Coláiste na hOllscoile Corcaigh, Éire
University College Cork, Ireland

Department of Process & Chemical Engineering
School of Engineering
National University of Ireland
University College Cork
Head of Department: Dr. Jorge Oliveira

Comparison of Deterministic, Stochastic and Fuzzy Logic Uncertainty Modelling for Capacity Extension Projects of DI/WFI Pharmaceutical Plant Utilities with Variable/Dynamic Demand

Dipl.-Ing. Frank Riedewald

Thesis presented to the National University of Ireland in fulfilment of the requirements for the Degree of Doctor of Philosophy (Ph.D.)

Supervisor: Dr. Edmond Byrne

June 2011

Table of Contents

List of Figures

List of Figs. ctn.

List of Figs. ctn.

List of Figs. ctn.

List of Figs. ctn.

List of Tables

List of Tables ctn.

Declaration

This PhD thesis is my own work and has not been submitted for another degree, either at University College Cork or elsewhere.

Frank Riedewald
29th June 2011

Acknowledgements

I would like to extend my thanks to a number of people. I am very grateful to my supervisor Dr. Edmond Byrne for his constant advice, support and discussions throughout the course of this work. I would like to thank Dr. Kevin Cronin, also from the Department of Process and Chemical Engineering UCC, for his helpful suggestions on the approach and structure of this research, and for his comments on various drafts of the manuscript.

I wish to thank Prof. Catherine Azzaro-Pantel, from the from the Laboratoire de Génie Chimique University of Toulouse, and Dr. Ken Brown from the Department of Computer Science UCC for taking the time to act as external and internal examiners.

I wish to thank Ms. Claire Healy, senior process engineer at WSP-CEL Cork, for her helpful comments on numerous versions of the manuscript.

I would also like to thank Mr. Kieran Goode, Managing Director of WSP-CEL Cork, and Mr. Pat Moriarty, Managing Director of CDGA Cork, who both generously allowed me the use of their office facilities to complete this work.

A very special thanks to my family, my brother and my sisters, to my parents Friedrich and Gisela, and to my wife Bereniece and our children Irina and Karl, for their ongoing encouragement, support and love.

Abstract

The last 30 years have seen Fuzzy Logic (FL) emerging as a method either complementing or challenging stochastic methods as the traditional method of modelling uncertainty. But the circumstances under which FL or stochastic methods should be used are shrouded in disagreement, because the areas of application of statistical and FL methods are overlapping with differences in opinion as to when which method should be used. Lacking are practically relevant case studies comparing these two methods. This work compares stochastic and FL methods for the assessment of spare capacity on the example of pharmaceutical high purity water (HPW) utility systems. The goal of this study was to find the most appropriate method modelling uncertainty in industrial scale HPW systems.

The results provide evidence which suggests that stochastic methods are superior to the methods of FL in simulating uncertainty in chemical plant utilities including HPW systems in typical cases whereby extreme events, for example peaks in demand, or day-to-day variation rather than average values are of interest. The average production output or other statistical measures may, for instance, be of interest in the assessment of workshops. Furthermore the results indicate that the stochastic model should be used only if found necessary by a deterministic simulation. Consequently, this thesis concludes that either deterministic or stochastic methods should be used to simulate uncertainty in chemical plant utility systems and by extension some process system because extreme events or the modelling of day-to-day variation are important in capacity extension projects. Other reasons supporting the suggestion that stochastic HPW models are preferred to FL HPW models include:

1. The computer code for stochastic models is typically less complex than FL models, thus reducing code maintenance and validation issues.
2. In many respects FL models are similar to deterministic models. Thus the need for a FL model over a deterministic model is questionable in the case of industrial scale HPW systems as presented here (as well as other similar systems) since the latter requires simpler models.

3. A FL model may be difficult to "sell" to an end-user as its results represent "approximate reasoning" a definition of which is, however, lacking.
4. Stochastic models may be applied with some relatively minor modifications on other systems, whereas FL models may not. For instance, the stochastic HPW system could be used to model municipal drinking water systems, whereas the FL HPW model should or could not be used on such systems. This is because the FL and stochastic model philosophies of a HPW system are fundamentally different. The stochastic model sees schedule and volume uncertainties as random phenomena described by statistical distributions based on either estimated or historical data. The FL model, on the other hand, simulates schedule uncertainties based on estimated operator behaviour e.g. tiredness of the operators and their working schedule. But in a FL model of a municipal drinking water distribution system the notion of "operator" breaks down.
5. Stochastic methods can account for uncertainties that are difficult to model with FL. The FL HPW system model does not account for dispensed volume uncertainty, as there appears to be no reasonable method to account for it with FL whereas the stochastic model includes volume uncertainty.

Glossary

This thesis uses terms specific to discrete-event simulation (DES) and uncertainty theory. This glossary briefly explains these terms within the context of a High Purity Water (HPW) system model. More information on DES can be found in textbooks such as Law & Kelton [1], Banks et al. [2] or Fishman [3].

Term	Meaning
Activity	A pair of linked events; here opening and closing of a valve on the HPW distribution system.
Crisp	Variable not including uncertainty or perfect knowledge
Entity	Any component modelled. Here referring to valves, pumps, tanks etc.
Event	A change in the system; here opening / closing of HPW offtake valves.
Event Table	A record of all events listed in chronological order per entity scheduled to occur; here schedule of opening/closing of HPW valves.
System	The HPW generating plant and distribution loop

Acronyms and Abbreviations

Acronyms /Abbreviations	Explanation
30 MC	30 simulated days (Monte Carlo Simulation)
CDF	Cumulative Density Function
CFU	Colony Forming Units
CIP	Cleaning in Place
CUP	Central Processing Unit
DES	Discrete Event Simulation
DI	Deionised Water as defined in the EU
DI/WFI	DI or WFI water generation and distribution system or both
EMEA	European Medicines Evaluation Agency
EP	European Pharmacopeia
EPA	Environmental Protection Agency
EPANET	Name of software developed by the US EPA's Water Supply and Water
EU	Endotoxin Units (EU) (measure of toxin which may be produced upon cell destruction)
EU	European Union
FDA	Food and Drug Administration (US Pharmaceutical Regulatory Authority)
FL	Fuzzy Logic
h	Hour
IMB	Irish Medicines Board (Irish Pharmaceutical Regulatory Authority)
IU	International Unit (amount of a substance, based on biological activity)

Acronyms Abbreviations	Explanation
LIFO	First In, First Out
LSS	Low Carbon Stainless Steel
m	Minute
Max	Maximum
MC	Monte Carlo
Min	Minimum
NPSH	Net Positive Suction Head
PC	Personal computer
PDF	Probability Distribution Function
PID	Proportional–Integral–Derivative controller
PW	Purified Water (generic name for DI & WFI)
R_a	Average Roughness
RNG	Random Number Generator
RS	Rough Sets
s	Second
USP	US-Pharmacopeia
VBA	Visual Basic for Applications
WFI	Water for Injection as defined in the EU or USP
WH	Wichmann-Hill
WHO	World Health Organisation

Nomenclature

Variable	Explanation	Units
A	General set or event	N/A
a	Constant or variable	N/A
$act_{i,k}$	Opening/closing event of valve i, k	N/A
$act_{i,k}^{Wait}$	Opening/closing event of valve i, k waiting to be served	N/A
B	General set or event	N/A
b	Constant or variable	N/A
C	Constant	N/A
c	Constant	N/A
d	Constant	N/A
E	General set	N/A
f_{Div}	Diversity factor	N/A
h	Constant	N/A
i	Integer parameter	N/A
j	Integer parameter	N/A
$\mathfrak{J}$	Field of sets	N/A
k	Integer parameter	N/A
m	Integer parameter	N/A
m_1, m_2, ..., m_7	Integer parameter	N/A
n	Integer parameter	N/A

Variable	Explanation	Units
N	Integer parameter	N/A
n_{op}	Number of operators	N/A
$n_{op,\,min}$	Minimum number of operators	N/A
P(A)	Probability of event A	N/A
P_i	Uncertainty in percent of full volume	%
POS	Possibility	N/A
$Pr_{i,k}$	Priority rules for each $act_{i,k}$	N/A
t	Time	h:m:s
tap_i	Valve (tap) i along the distribution system	N/A
$t^{B}_{2,j}$	Beginning of core break	h:m:s
$t^{B}_{3,j}$	End of core break	h:m:s
$t^{close}_{i,k}$	Scheduled closing time for each $act_{i,k}$	h:m:s
$t^{close}_{i,k,new}$	New closing time for each $act_{i,k}$	h:m:s
$t^{D}_{i,k}$	Time delay for each $act_{i,k}$	h:m:s
$t^{D,De}_{i,k}$	Defuzzified time delay for each $act_{i,k}$	%
$t^{Delay,1}_{i,k}$	Time delay caused by operator for each $act_{i,k}$	h:m:s
$t^{Delay,2}_{i,k}$	Time delay caused by operator for each $act_{i,k}$	h:m:s
$t^{Delay,\,No\,Op.}_{i,k}$	Delay caused by no operator being available for $act_{i,k}$	h:m:s
$t^{min,\,D}_{i,k}$	Minimum duration of each $act_{i,k}$	h:m:s
$t^{open}_{i,k}$	Scheduled opening time for each $act_{i,k}$	h:m:s

Variable	Explanation	Units
$t^{open}_{i,k,new}$	New closing time for each $act_{i,k}$	h:m:s
$t^{open,R1}_{i,k,new}$	New opening time due to influence of Rule 1	h:m:s
$t^{open,R4}_{i,k,new}$	New opening time due to influence of Rule 4	h:m:s
$t^{open,R5}_{i,k,new}$	New opening time due to influence of Rule 5	h:m:s
t^{sim}	Simulated time	s
t_E	Time at end of simulated time	h:m:s
t_{max}	Maximum allowable time	h:m:s
t_{Rule1}	Sum defined as: $t^{Sim} + t_{max}$	h:m:s
t_0	Time at start of simulated time	h:m:s
$\Delta t^{offtake}_{i,k}$	Opening/closing interval for each $act_{i,k}$	h:m:s
$\Delta t^{uncertain}_{i,k}$	Uncertain opening time	h:m:s
u	Element of set Ω	N/A
$U_{i,k}$	Binary variable either (-1) or (1)	N/A
U(X)	Uniform distribution	N/A
v	Element of set Ω	N/A
$\dot{V}_{DI,1}$ to $\dot{V}_{DI,2}$	Individual offtake from the DI loop	m^3/h
$V_{DI,L}$	Water volume in the DI storage tank	m^3
$V_{DI,L,St}$	Water volume in DI storage tank at t_0	m^3
$\dot{V}_{Design}$	Design flowrate	m^3/h
$\dot{V}_{DI,Gen}$	Volumetric flowrate DI from DI water generator	m^3/h

Variable	Explanation	Units
$\dot{V}_{DI,off}$	Volumetric off take from the DI system	m^3/h
$\dot{V}_{DI,Pump}$	DI Distribution pump delivery volume	m^3/h
$\dot{V}_{DI,WFI}$	Volumetric DI flowrate to WFI generation plant	m^3/h
$\dot{V}_{Max}$	Maximum flowrate	m^3/h
$\dot{V}_{RW}$	Raw water flowrate, typical drinking water inlet to DI-water generation plant.	m^3/h
$\dot{V}_{WFI,Blow}$	WFI generation blowdown	%
$\dot{V}_{WFI,Gen}$	Volumetric flowrate WFI from WFI generation plant	m^3/h
$\dot{V}_{WFI,Gen,max}$	Maximum volumetric flowrate from WFI generating plant	m^3/h
$V_{WFI,L}$	WFI volume in the WFI storage tank	m^3
$V_{WFI,L,C}$	Temperature compensated water volume in WFI storage tank	m^3
$V_{WFI,L,max}$	Maximum allowable water volume in WFI storage tank	m^3
$V_{WFI,L,min}$	Minimum allowable water volume in the WFI storage tank	m^3
$V_{WFI,L,St}$	Start volume in WFI storage tank at t_0	m^3
$\dot{V}_{WFI,1}$ to $\dot{V}_{WFI,2}$	Individual offtake from the WFI loop	m^3/h
$\dot{V}_{WFI,off}$	Volumetric off take from the WFI system	m^3/h
$\dot{V}_{WFI,off,i}$	Volumetric offtake for each act_i	m^3/h

Variable	Explanation	Units
$V_{WFI,off,i,k}$	Dispensed water volume for each act_i	m^3
$\dot{V}_{WFI,off\ i,k}$	Volumetric offtake for each $act_{i,k}$	m^3/h
$\dot{V}^{actual}_{WFI,off,i,k}$	Volumetric flowrate for each $act_{i,k}$	m^3/h
$\Delta\dot{V}^{uncertain}_{WFI,off,i,k}$	Stochastic volumetric flowrate uncertainty for each $act_{i,k}$	m^3/h
$\dot{V}_{WFI,Pump}$	WFI Distribution pump delivery volume	m^3/h
x	Element of set and general variable	N/A
X	Universe of discourse	N/A
X	Random variable, uniformly distributed $\{0,...,1\}$	N/A
X_D	Defuzzified value	N/A
Y_i	Generic stochastic variable	N/A
$Z_{i,k}$	Generic stochastic variable	N/A

Fuzzy Sets		
Variable	**Explanation**	**Units**
$\underset{\sim}{A}$	Generic fuzzy set	N/A
$\underset{\sim}{A}_{\alpha}$	α -cut of fuzzy set $\underset{\sim}{A}$	N/A
$\underset{\sim}{B}$	Generic fuzzy set	N/A
$\underset{\sim}{C}$	Generic fuzzy set	N/A
$\mu_{\underset{\sim}{A}}(x)$	General membership function	N/A
$\mu_{\underset{\sim}{B}}(x)$	General membership function	N/A
$\mu_{\underset{\sim}{B}}(t)$	Membership Function “Operator Break Pattern”	h:m:s
$\mu_{\underset{\sim}{D}}^{i,k}(t)$	Membership Function “Duration of Tasks” for each $act_{i,k}$	h:m:s
$\mu_{\underset{\sim}{S}}(t)$	Membership Function Operator “Shift Pattern”	h:m:s
$\mu_{\underset{\sim}{T}}(t)$	Membership Function “Operator Tiredness	N/A
$\mu_{\underset{\sim}{Out}}^{0\%}(r)$,..., $\mu_{\underset{\sim}{Out}}^{100\%}(r)$	Membership Function “Output”	N/A

Greek Variables		
Variable	**Explanation**	**Units**
α	Shape parameter α of the Beta distribution and α-cut of a fuzzy number	N/A
β	Shape parameter β of the Beta distribution	N/A
ζ	Elementary events	N/A
η	Elementary events	N/A
μ	Mean of the Normal distribution	N/A
ξ	Elementary events	N/A
ρ_{20}	Water density at 20 °C: ρ_{20} = 998.21 kg/m^3	kg/m^3
ρ_{80}	Water density at 80 °C: ρ_{80} = 971.79 kg/m^3	kg/m^3
σ	Standard deviation σ of the Normal distribution	N/A
$\Gamma(\alpha)$	Gamma function	N/A
χ_A	Interval A	N/A
Ω	General set	N/A

1 Introduction

The pharmaceutical industry is an important industrial sector, both in terms of providing vital medicines for the well-being of people as well as being an employer. In the 27 member states of the European Union the pharmaceutical industry directly employs over 500,000 people in about 4,400 companies manufacturing various kinds of pharmaceutical preparations including vaccines, homeopathic preparations, dental fillings and bandages [5, 6]. Among these pharmaceutical preparations, parenterals (i.e. injectable drugs) form a large and important group. Many vaccines against ailments such as tetanus, rubella, measles, flu, mumps and hepatitis must be delivered to the body as injectables. The importance of parenterals is further illustrated by a list of 306 essential drugs compiled by World Health Organisation (WHO) [7], of which about 44% are injectables [8] requiring Water for Injection (WFI) as a delivery vehicle.

In the future injectable drugs may become even more prevalent as the vast majority of biopharmaceuticals in development or in production must be injected into the bloodstream [9, 10]. This is because the molecules produced by biotechnology i.e. proteins, hormones and antibodies are generally speaking large molecules (typically 100 to 1,000 times larger than drugs based on chemical synthesis), which often cannot be delivered to the intended target via the oral route. In contrast the small molecules based on chemical synthesis typically allow oral route delivery [9, 10].

1.1 High Purity Water in the Pharmaceutical Industry

For the production of injectables WFI is required not only in the formulation of the medication, but also for cleaning and rinsing of production equipment as is for instance stipulated by the European Medicines Evaluation Agency (EMEA) [11]. Hence WFI systems may form an important part of a (bio)pharmaceutical production facility, while hardly deserving the label "utility" since the WFI forms an integral part of the formulation of the medicine.

Purified pharmaceutical grade water such as Deionised Water (DI) and WFI in particular is generally the most expensive utility employed in a pharmaceutical plant. The production of WFI

is a complicated process involving a number of steps [12]. Multi-media filters, ion exchangers, cartridge filters, reverse osmosis units, continuous electro-deionisation units and distillation are generally employed to achieve the required purity requirements of WFI. An indication of the purity levels required is given by Table 1.1, which compares the purity of naturally occurring waters to purified water. WFI is typically produced by distillation, the most expensive and energy intensive technology used in its production. This is because distillation must be used should the medication be destined for the European market as stipulated by the European Pharmacopeia (EP) [12, 13] (see Table 1.2). Furthermore, it is also energy demanding to distribute WFI within a production facility as the temperature of the WFI in the distribution system is typically maintained at temperatures exceeding 80 °C and controlled within a temperature band of ±5.0 ºC in order to maintain a low bacterial count [12, 14]. The costs of routine quality control [13] and the cost of regulatory scrutiny [12, 13] further explain the high value of pharmaceutical waters.

Types of Water	Impurities [ppm]	Purity [%]
Sea Water	30,000	97
Potable Water	500	99.95
Mountain Water	50	99.995
Purified Water	1	99.9999

Table 1.1 Water impurities and water purity for various waters *(from Dabbah [15])*

Many different DI/WFI distribution configurations are used in industry. The literature describes distribution systems without any buffer tank [16], with numerous tanks along the distribution system [13], batch systems [13] and network distribution systems (see Figure 1.2). The vast majority of all DI/WFI systems in industry are, however, of the form of the looped system with storage tank as schematically shown in Figure 1.1 [13]. The reason for the success of this DI/WFI configuration is that the hydraulic conditions and temperature profiles along the entire distribution system can readily be maintained.

	US Pharmacopeia (USP)	European Pharmacopeia (EP)
Source water	Drinking water, EPA, EU	Drinking water or purified water
Preparation	Distillation or other suitable process	Distillation
Total Organic Carbon (TOC)	0.5 ppm	0.5 ppm
Conductivity	1.1 µS 20 °C	1.1 µS 20 °C
Appearance	N/A	Clear and odourless
Bacterial endotoxins	Not more than 0.25 USP EU/mL	Less than 0.25 IU/mL
Bacterial count	10 CFU/100ml	10 CFU/100 ml

Table 1.2 Comparison of the stipulations of the US and European Pharmacopeia on WFI [12, 17]

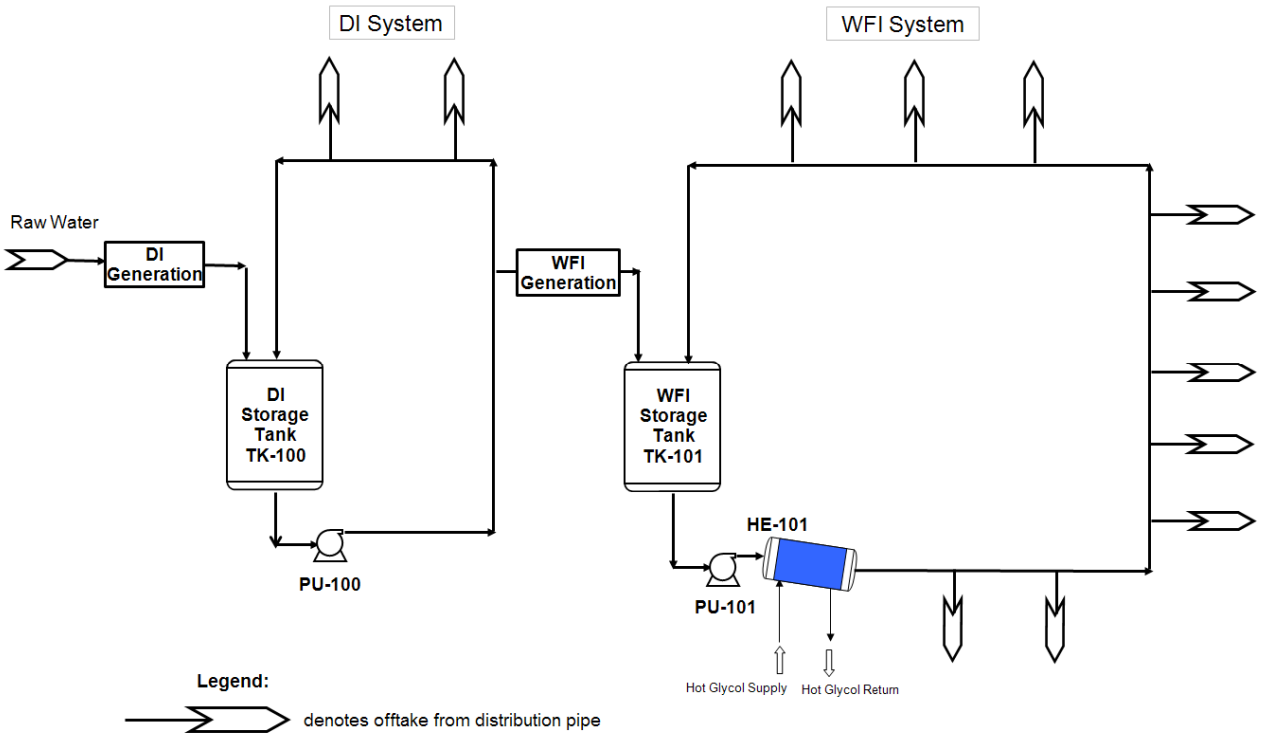


Figure 1.1 Process flowsheet of a typical DI/WFI distribution system

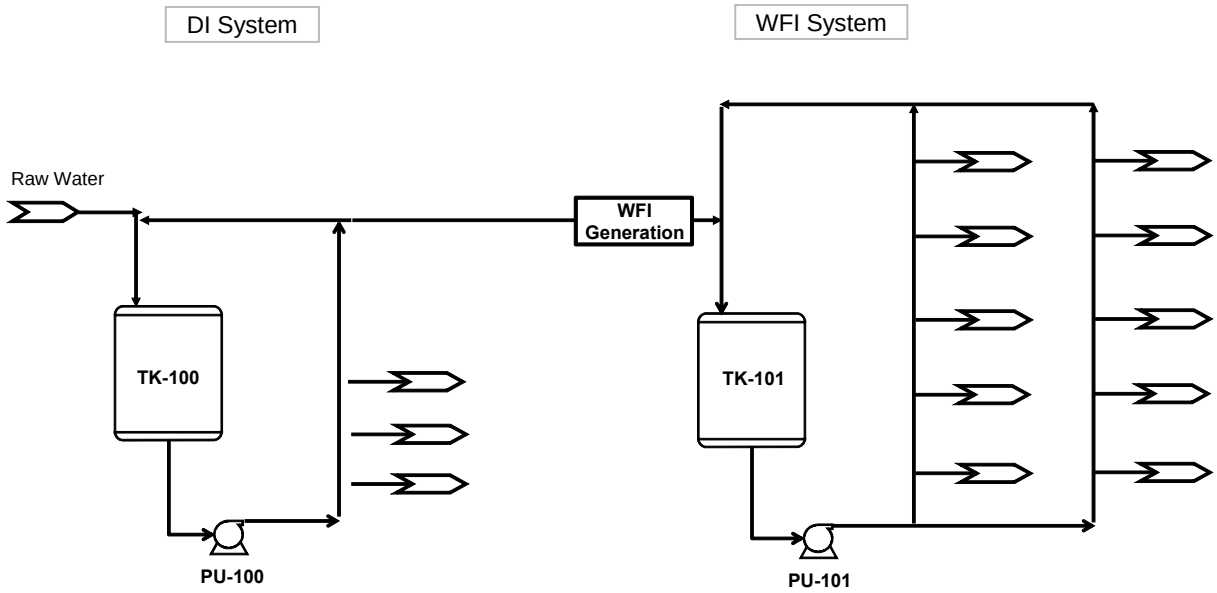


Figure 1.2 Process flowsheet of a looped DI/WFI network-looped distribution system

There is no general agreement on some design details of pharmaceutical DI/WFI systems, such as the requirements for minimum water velocities [12, 13] or the value of reducing internal pipe roughness [13]. As a result end-user preferences are common [13]. Typical design parameters of DI/WFI systems as applied in practice are summarised in Table 1.3. One important design parameter for DI/WFI systems is reliability, as physical breakdowns of DI/WFI systems may result in prolonged and costly production shutdowns. Nevertheless nowadays DI/WFI systems can be built which may operate continuously for many years [16]. This work is not concerned with any of these design parameters however, and these are discussed in detail by, for instance, Meltzer [12] or the ISPE Baseline Guide on Water Systems [13]. Instead it is concerned with the effective sizing of a DI/WFI system i.e. the size of the DI/WFI generation and distribution system including the storage tank volume as a function of expected demand.

Parameter	DI System	WFI System
Min. fluid velocity in the distribution pipe	not specified to min. 1 m/s	min. 1 m/s
Material of construction	316 LSS	316 LSS
Surface finish	0.8 – 0.5 μm Ra	0.5 – 0.25 μm Ra or better and electropolished
Max length of Dead legs	< 2 pipe diameters	< 2 pipe diameters to zero dead leg valves
Operating temperature	Ambient	~ 70 to 80 °C
Operating pressure (pipe)	Above ambient at all times	Above ambient

Table 1.3 Typical design parameters of industrial size DI/WFI systems [13]

1.2 Thesis Motivation and Objectives

In a study investigating the production capacity of General Motors, Eppen et al. [18] express their suspicion that most American companies have accumulated excessive manufacturing capacities. The pharmaceutical industry may also have accumulated excessive capacities for certain systems such as DI/WFI systems. This is likely because of the usual utility sizing procedures, which involves incorporating maximum loads as described by Winkel et al. [19]. As this combined maximum load is often too large to be economical, Ming et al. [20] cite the practice of applying a diversity factor to the maximum flow thereby reducing the maximum flow to a more economical figure. The problem with this design approach is that it cannot realistically assess the spare capacity of a given WFI system [20-22], because it does not determine the DI/WFI demand profile as it evolves over time. With simulation, on the other hand, the DI/WFI demand profile can be obtained and it allows experimentation with the WFI system in a virtual world.

Simulation can provide important insights. Saraph [21] describes a deterministic simulation of a purified water system. Computer experiments allowed Saraph [21] to show that a recurrent WFI shortage was a result of poor scheduling of water consumption from different WFI consumers, rather than a water-generating problem, resulting in savings of over $1,100,000. Alexander [23] also used a deterministic model of an existing WFI system to find if the system is capable of delivering a possible higher WFI demand. He then experimented with the model to find the most cost effective strategy for the system to deliver the increased WFI demand.

The question to be answered by a simulation project of an existing WFI system may hence be twofold. First and foremost: Can the additional WFI process demand be supported by the existing WFI system capacity? Second, failing the first approach: What is the minimum level of investment, for instance, in additional tank capacity or increasing the water generation capacity that needs to be made to support the additional DI/WFI demand?

In order to answer these questions a model should be "as simple as possible, but no simpler" to cite part of a title of a paper by Sanchez [24], who in turn quoted Albert Einstein. Crucially the modelling of uncertainty may play an important role; for instance Pinedo [25] stresses that a good model should include uncertainty. The main reason for including uncertainty is that according to Klir and Wierman [26] actual production plants are liable to be influenced by uncertainty, which may stem from various sources:

1. Delay in the arrival or shortage of raw materials (e.g. Vieira et al. [27]).
2. Breakdowns of machinery (e.g. Vieira et al. [27]).
3. Human (operator) variation and operator absenteeism (e.g. Vieira et al. [27]).
4. Uncertainty in measuring devices (e.g. Drosg [28]).
5. Other unknown uncertainties/factors.

The purpose of including uncertainty in a model is to increase acceptance of the model by the end-user as noted by Klir and Wierman [26], as a purely deterministic model cannot capture the inherent uncertainties of real production systems. Therefore modelling uncertainty should increase the usefulness and acceptance of a DI/WFI model, as the results of the simulation may "feel more real". As a consequence the deterministic WFI simulations of Saraph [21] and Alexander [23] have both a potential weak spot in that the deterministic approach taken is not able to represent the uncertainty inherent in DI/WFI systems.

Many different methods of modelling uncertainty are proposed in the literature, stochastic and Fuzzy Logic being two of the most popular ones. A long standing debate on which of these two methods should be used to describe uncertainty can be found in the literature. In a monograph entitled "Fuzzy Logic and Probability Applications" [29] a summary of these debates is provided. Unfortunately no guidance under which circumstances either method should be used is given. One of the reasons for this lack of guidance may be found in the apparent lack of direct comparisons of these two approaches to uncertainty. This leads on to the following research questions to be investigated in this work:

- What is the most appropriate mathematical method to model uncertainty in DI/WFI systems?
- To what extend can these results be extended to other systems?

1.3 Thesis Contributions

The contributions of this work to the body of scientific knowledge can be summarized as follows:

- This work is the first study known to the author which compares stochastic and fuzzy logic (FL) simulation models.
- This work highlights some limitations of FL models such as the inability to show day-to-day variations that are typical of chemical plants, an inability to model all uncertainties (in this case volume uncertainty) and increased program complexity compared with stochastic modelling. This work therefore makes a contribution towards qualifying some of the more extreme claims surrounding the capabilities of FL.
- The work demonstrates how stochastic methods can be superior to the FL methods where extreme events rather than average figures are deemed important.
- This thesis demonstrates that the importance of including uncertainty in modelling should not be overstated. The case study shows that the deterministic model should be used first and the stochastic approach used only if found necessary by the results of a deterministic simulation. In other words including uncertainty to a model may not yield any useful increase in information from the model other than adding time and effort.
- Three different models for schedule uncertainty are proposed: deterministic, stochastic and FL. The FL model of a DI/WFI system is the first of its kind known to the author.
- Two different models for dispensed volume uncertainty are proposed: deterministic and stochastic.
- A literature review on uncertainty modelling is provided.

1.4 Thesis Structure

This introductory chapter provides an overall framework for this thesis, explains the reasoning behind including uncertainty in DI/WFI systems, states the research questions and outlines the contribution to scientific knowledge provided by this work.

A literature review is presented in Chapter 2 and is primarily devoted to a review of uncertainty theories and their methods. The difficulty of choosing an appropriate uncertainty theory is highlighted from a theoretical and practical point of view. The literature review also includes a brief discussion on how a DI/WFI system may fit into the framework of Complexity Theory.

Three different DI/WFI models, each modelling uncertainty in a different way are developed in Chapter 3. These three models are :

i) A purely deterministic model, which does not include uncertainty.
ii) A stochastic model, which deterministically models uncertainty via statistical distributions, and
iii) A fuzzy logic model, which deterministically models uncertainty via fuzzy logic.

In Chapter 4 the materials and methods are outlined and discussed, whereas in Chapter 5 the results of the deterministic, stochastic and fuzzy logic simulations are presented and discussed.

Finally, Chapter 6 contains a discussion on the main finding of this work, that is, that the stochastic and deterministic methods are the preferred methods to describe uncertainties in DI/WFI systems and that fuzzy logic is not a suitable method for uncertainty modelling in DI/WFI systems. The thesis ends with hints on possible future work.

The appendices contain an introduction into Fuzzy Set Theory operations, the Visual Basic (VBA) listings of the deterministic/stochastic and Fuzzy Logic programs developed and the input data sets used for the simulations. Excel VBA was used as the programming language causing its own problems. These problems and their workarounds are also discussed in the appendices.

1.5 Publications

Work associated with this thesis has been published at, accepted by and submitted to the following conferences and peer reviewed scientific journals to date:

1. Riedewald, F. and Byrne, E. (2010), Simulation of DI/WFI Distribution Systems: exploring deterministic, stochastic and fuzzy logic as methods of uncertainty description with Excel as the modelling platform, Annual IChemE Computer Aided Process Engineering Poster Day, University of Leeds, England, 12 May 2010.
2. Frank Riedewald, Edmond Byrne, Kevin Cronin, Comparison of Deterministic and Stochastic Simulation for Capacity Extension of High Purity Water Delivery Systems, PDA Journal of Science and Technology, accepted for publication, 19th May 2011.
3. Frank Riedewald, Edmond Byrne, Kevin Cronin, A Fuzzy Logic Model of Deionised and Water for Injection Systems for Sizing and Capacity Assessment under Uncertainty, Journal of Pharmaceutical Innovation, revised manuscript submitted.
4. Frank Riedewald, Edmond Byrne, Kevin Cronin, A Stochastic Model for Performance Analysis of Pharmaceutical High Purity Water Systems, Simulation Modelling Practice and Theory, revised manuscript submitted.

2 Review of Relevant Literature

2.1 Introduction

Section 2.3. of this chapter reviews relevant uncertainty theories and methods as may be used to describe volume and schedule uncertainties of DI/WFI systems. The emphasis is placed on probability and possibility theory and their methods, statistical and fuzzy logic respectively, as these two theories are the most popular ones and these two theories of uncertainty were hence used in the models developed for this thesis. The question of under which circumstances the modeller should select probability or possibility theory is also considered. The hypothesis of this thesis is that the two theories are *competitive rather than complementary*, because time and budget constraints will generally not allow for the modelling of a DI/WFI system using both approaches. Sanchez [24], for instance, contemplates that too many simulation projects have run out of money or time or both before a functioning model had been developed. Ultimately, as will be outlined, no generally accepted "hard" criteria could be found in the literature, leaving the choice between the different theories open to context and personal conceptions, values and preferences.

The hypothesis of this work that the theories of probability and possibility are *competitive rather than complementary* is made from an industrial point of view of limited resources. Furthermore it requires that both possibility and probability theory may be used to solve the problem. Of course probability and possibility theory are based on different axioms as is outlined in this chapter. Therefore probability and possibility theory may indeed be *complementary* in the sense that they may offer different bases for their application. Moreover in some cases possibility theory (fuzzy logic) may be more appropriate than probability theory. This is because the application of probability theory requires at least some historical data, which is also assumed to exist here. Should historical data be lacking or the uncertain parameters be given by natural language, possibility theory may be considered a better choice in modelling the system in question. Consequently the hypothesis of this work is only claimed within the framework of

capacity extension projects of DI/WFI systems or other similar chemical plant utilities and processes from a pragmatic (time and money) resources perspective.

The next section briefly considers the DI/WFI system in the framework of complexity theory and suggests that while a DI/WFI system operating with human intervention can be considered a complex system, the system itself can generally be regarded as being merely complicated and thus for most operational purposes deterministic modelling techniques can be considered adequate and appropriate.

2.2 DI/WFI Systems and Complex System Theory

Large municipal electrical systems have been classified as complex systems [30]. Since electrical systems are utility systems, one may ask if DI/WFI systems are also complex systems. Unfortunately the notion of a complex system is difficult to define as there is no generally accepted definition [30, 31]. In the absence of such a definition it may be best to define a complex system by some of their common properties. Among the properties of complex systems are the following:

- The system consists of a large number of diverse though interconnected components, which produce interactive responses. Feedback (positive or negative) may also be included to, for instance, improve the survival of a system i.e. adaption via evolution [30].
- A small disturbance to the system may cause a large unpredictable output i.e. butterfly effect [31].
- Emergence and self-organisation may result as the system evolves with its environment [31] in ways that are inherently unpredictable.
- Uncertainty is thus inherent in a complex systems. A complex system cannot be modelled through reducing it to discrete elements which can be modelled and aggregated to predict overall system performance. Instead a complex system must be modelled as a whole.

- Deterministic models of complex systems completely break down when such systems, operating far from equilibrium, flip into an entirely new and unpredictable regime following the passing of a bifurcation or tipping point, often brought on by non-linear positive feedback loops.

Complex systems may thus give rise to chaotic behaviour [30, 31]. An example of such instability on a grand scale is the 2003 blackout on the East Coast of the United States [32]. Such instabilities may be explained by the May-Wigner Theorem [31], which states that for some systems interconnectivity should not increase beyond a certain threshold.

DI/WFI systems may also be viewed as complex systems if human behaviour (such as valve operation), economic impacts, etc. are included. But the DI/WFI models developed as part of this work are all inherently deterministic and thus do not recognise complexity. Figure 2.1 shows how the DI/WFI models may fit within the framework of complex system theory. The models describe a complicated system, though not a complex one, while human interaction through operator intervention is modelled deterministically. Within the context of the overall (DI/WFI) system investigated here, this is acceptable since uncertainties related to human intervention described by the DI/WFI model are relatively small as indicated in Figure 2.1. The model itself is merely a complicated one, and while the additional human input may add a degree of complexity to the system, for the purposes of this work this can be neglected and the frame narrowed in order to obtain reasonable models which apply in the vast majority of cases. Thus the possibility of, for example, no operators at all being available to activate valves or inadequate supply of raw water available to the system is not considered, nor is it judged to be required to be modelled for the purposes of this work. Clearly the DI/WFI models can be used to solve specific problems only. As a result, unforeseen behaviour i.e. emergent properties or chaos is not displayed by the DI/WFI models. For this system and work therefore, this approach is deemed sufficient.

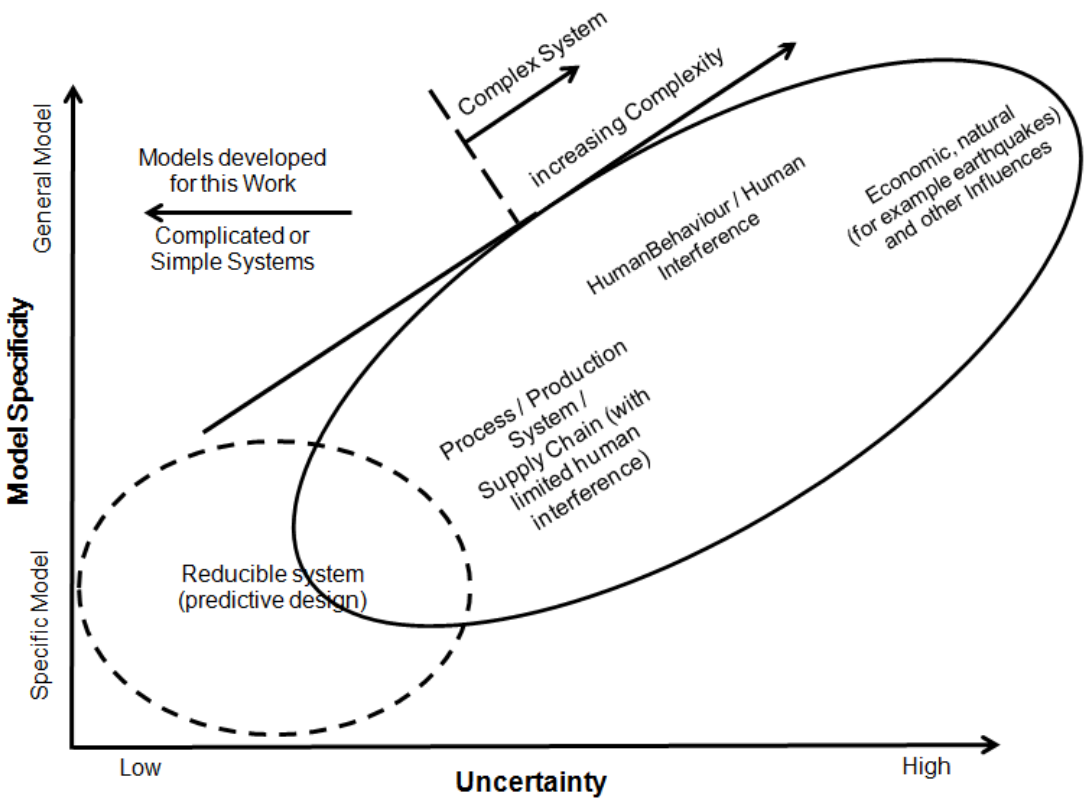


Figure 2.1 How the developed DI/WFI models of this thesis may fit within the framework of complex system theory.

2.3 Review of the Relevant Literature on Uncertainty

This section reviews the relevant literature on uncertainty with the following question in mind: Under what circumstances should which mathematical method be used to model uncertainty in DI/WFI systems?

The literature on uncertainty is vast and an entire account of the field is well beyond the scope of this dissertation, as it would, according to Ross and Kreinovich [33] require volumes. Therefore this chapter has been restricted to a review of five popular methods handling uncertainty: uncertainty factor, interval, rough sets, fuzzy, and statistical methods. Out of these different methods, fuzzy logic and stochastic methods are the most popular ones and considered in detail, as the others are rejected for various reasons as methods to model uncertainty in DI/WFI systems.

2.3.1 Theoretical Foundations of Uncertainty Theories

Before the 1940's the situation regarding uncertainty theories was quite straightforward, in the sense that only probability theory was available. Since then the situation has changed and possibility theory, evidence theory and general theory of uncertainty have been added to the fold as shown schematically in Figure 2.2.

This section first introduces probability theory, because it is the oldest, best known and developed uncertainty theory, followed by possibility theory and the other theories of uncertainty.

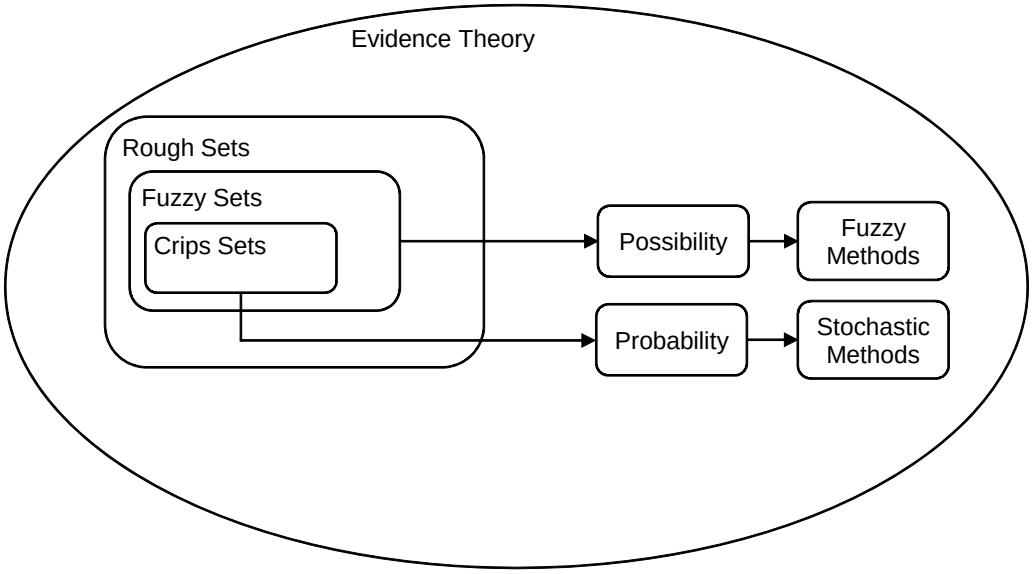


Figure 2.2 Overview of uncertainty theories and methods

2.3.1.1 Probability Theory

Probability theory is the foundation of statistics and was originally founded to study stochastic i.e. random processes such as gambling, meaning that probability theory describes the likelihood of an event taking place. Since its inauguration about 400 years ago probability theory has grown into a comprehensive and vast body of literature.

The modern five axioms of probability theory were proposed by Kolmogorov [34] in 1933 and these are:

Let E be a collection of elements $\xi, \eta, \zeta, \ldots$, which we shall call *elementary events*, and $\mathfrak{I}$ a set of subsets of E; the elements of the set E will be called *random events.*

1. $\mathfrak{I}$ is a field of sets.
2. $\mathfrak{I}$ contains the set E.
3. To each set **A** in $\mathfrak{I}$ is assigned a non-negative real number $P(A)$. This number $P(A)$ is called the probability of the event **A**.
4. $P(\mathfrak{I})$ equals 1.
5. If **A** and **B** have no element in common, then

$$P(A + B) = P(A) + P(B)$$

(2.1)

Within probability theory, uncertainty has generally been segregated into two groups [35-37]: *objective* and *subjective* as shown in Figure 2.3.

Objective, *frequentist*, or *epistemic* uncertainty is the uncertainty of repeatable events such as games of chance, or random events i.e. measurement errors. Therefore in principle epistemic uncertainty can be reduced by repeating the events or in other words by collecting more information. Objective uncertainties can hence be measured and its outcome should not depend on the experimenter.

Subjective or *aleatory* uncertainty (sometimes also referred to Bayesian [38]) is any uncertainty not being labelled objective. Subjective uncertainty is often seen as the more fundamental of the interpretations of uncertainty [35, 36, 38, 39]. This is, because even epistemic uncertainty i.e. measurements are somewhat subjective, as they are relative to a finite body of evidence, always leaving some room for interpretation [39]. The interpretation also recognises that different people may assign different probabilities to the same event based on their experiences. Hence

subjective probability allows assigning probabilities to unique events for which the concept of frequency breaks down. Such events may be the chance a particular horse will win a race [36] or that a defendant is guilty [36]. Finetti [40] uses the unique result of a football game as an example.

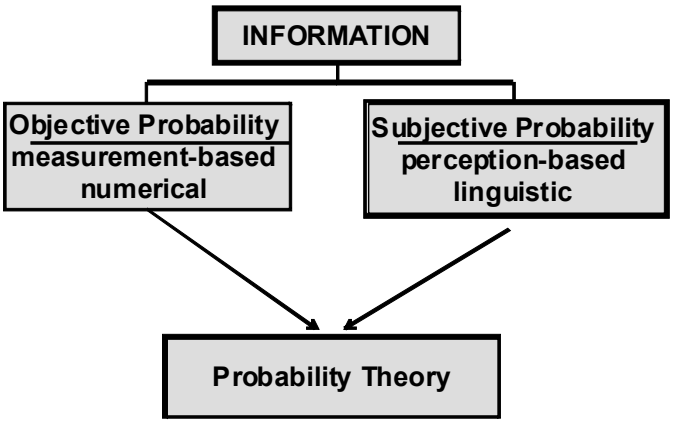


Figure 2.3 Statistician's view of uncertainty as revealed from the literature survey

2.3.1.1.1 Application of Probability Theory - Statistical Methods

Modelling uncertainty with statistical methods is subject to many textbooks on discrete-event simulation such as Banks et al. [2, 41], Fishman [3], Law and Kelton [1], and is the method of choice for most modellers concerned with schedule uncertainty [25, 42].

While several textbooks propose to estimate probability functions from available data [1-3, 41, 43, 44], the issue of choosing appropriate statistical distributions is not addressed in this work. Moreover, many statistical distributions are provided in Bury [45] and Evans et al. [46].

2.3.1.2 Possibility Theory

Compared to probability theory, possibility theory is a more recent development. Furthermore possibility theory is not a generally accepted theory and some of its methods are still under development. But it could be applied as an alternative to probability theory as a method to describe uncertainty in the simulation of DI/WFI systems.

A possibility distribution, denoted $POS(x)$, maps to each element u in a set Ω a degree of possibility in the interval [0, 1] and has the following three axioms [47]:

1. $POS(0) = 0$
2. $POS(\Omega) = 1$
3. $POS(u \cup v) = \max\ (POS(u),\ POS(v))$ for any disjoint subsets of u and v in Ω.

(2.2)

Axioms 1 and 2 are similar to probability theory [47] (see equation 2.1). Axiom 1 indicates that Ω describes all possibilities and that outside of Ω the possibility is zero. The second axiom states that at least one element of Ω must have the possibility 1 or that at least one event must be possible [47]. Axiom 3 is the so called "maxitivity axiom" [47], which differentiates possibility from probability theory. It results in subadditivity of the elements of the disjoint subsets u and v. Subadditivity means that the sum of two elements, here u and v, is always less or equal to the sum of the function's values of each element i.e. $f(u + v) \leq f(u) + f(v)$ [48] (see also equation 2.7). Note that the concept of *necessity* [47, 49] in possibility theory is ignored here.

According to Dubois [47] possibility can be interpreted in four different ways: (1st) *feasibility*, for example "it is possible to solve this problem" (2nd) *plausibility*, for example "it is possible that the train arrives on time", (3rd) *logic*, referring to consistency with available information; namely, stating that a proposition is possible or that it does not contradict this information. The (4th) interpretation of possibility means *allowed or permitted*. A more detailed discussion on the

philosophical background of possibility theory is, for instance, provided by Dubois and Prade [49], or Klir [26, 50].

At least three different practical interpretations of possibility have been presented. This work adopts the most common interpretation of possibility; that is the interpretation of possibility proposed by Zadeh [51] in 1978. In this paper Zadeh suggested to use fuzzy sets as the basis for possibility; an interpretation generally referred to as Fuzzy Logic (FL). The Zadeh interpretation of possibility is generally adapted in the engineering literature [52-54] and has been extensively used in expert [54-57] and control systems engineering [58, 59].

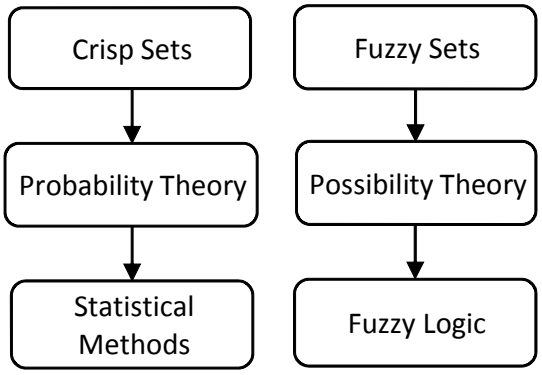


Figure 2.4 Relationship between crisp sets, probability theory and statistical methods (left) and fuzzy sets, possibility theory and fuzzy logic (right)

At first glance, the Zadeh interpretation of possibility appears to be a simple extension of probability theory as shown in Figure 2.4, with the fuzzy set generalising the crisp set. Yet, it is a bit more complicated than that. Firstly, from an axiomatic point of view possibility theory is different to probability theory (see equations 2.1 and 2.2) and secondly, the procedure of FL is different to the procedure of statistics as can be seen comparing the basic procedures of both methods (see sections 2.3.1.1.1 and 2.3.1.2.1).

In classical set theory and as applied in probability theory, the membership of elements of a set is assessed in binary terms (member/non-member) resulting in crisp sets, denoted A, usually in the

interval [0,1]. Zadeh [60] extended the classical crisp set theory to include fuzzy sets, denoted $\underset{\sim}{A}$ in his seminal 1964 paper, allowing gradual membership values as shown in Figure 2.5.

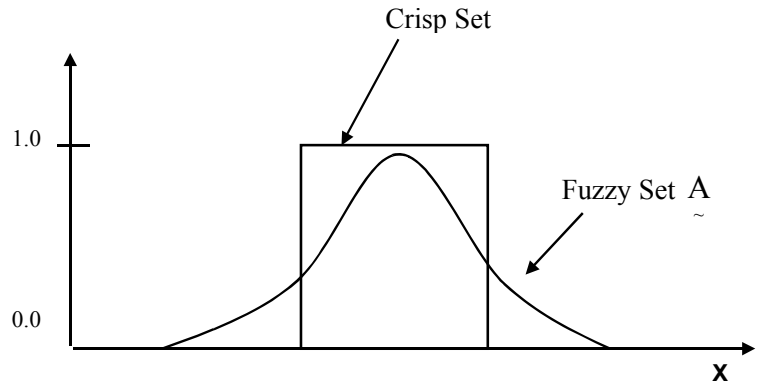


Figure 2.5 Crisp set A and fuzzy set $\underset{\sim}{A}$ over the universe of discourse X (*after Ross [53]*)

Similar to classical set theory, the members of a fuzzy set express membership of a set. In a fuzzy set the membership can be gradual allowing degrees of membership or to cite Zadeh [61]: "in fuzzy logic everything is allowed to be a matter of degree". This gradual degree of membership can be used to describe uncertainty.

The other two interpretations of possibility are first that possibility is the limit of the plausibility of nested sets of evidence and second that possibility is the upper limit of probability. More information on these two interpretations of possibility can be found in Aughenbaugh [62] and Nikolaidis et al. [63], while Klir [50] provides an extensive overview on the current state of uncertainty theories.

Whereas probability theory classifies uncertainty into *subjective* and *objective*, in the framework of possibility theory the classification of uncertainty is rather more involved and no agreement among the proponents of possibility theory on how to classify uncertainty has yet been reached. An impression of the various classifications of uncertainty proposed is provided graphically by Nikolaidis [64], which is reprinted in Figure 2.6.

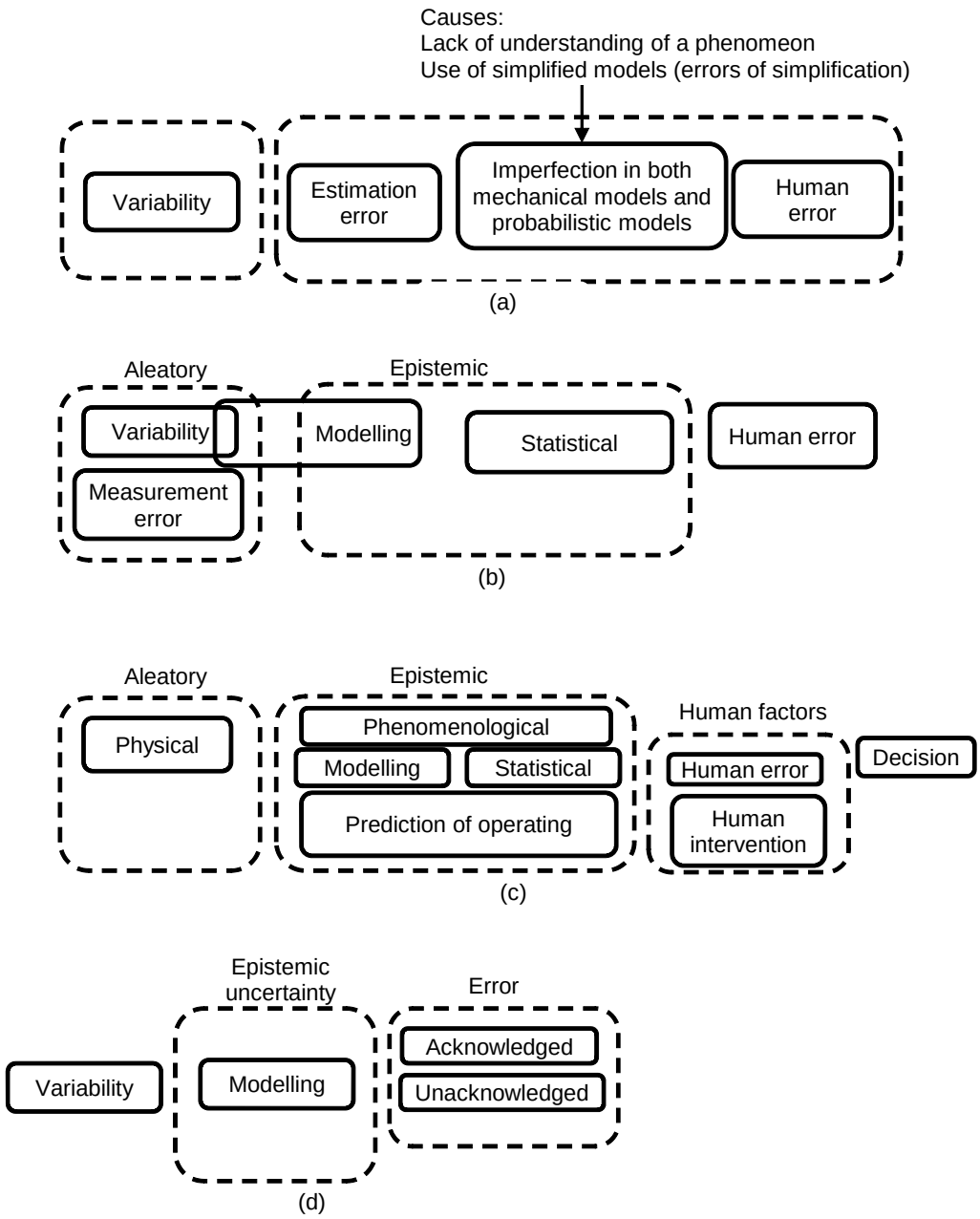


Figure 2.6 Taxonomies of uncertainty as proposed by various proponents of possibility theory (image reproduced from Nikolaidis [64] with kind permission from the author and the publisher). (a) refers to the proposed uncertainty taxonomy by Der Kiureghian, (b) to the proposal by Haukass, (c) to the proposal by Melchers and (d) to the proposal by Oberkampf. More information on these proposals can be obtained from Nikolaidis [64].

2.3.1.2.1 Application of Possibility Theory - Fuzzy Logic

This section presents the basic concepts of fuzzy logic, or the application of possibility theory by fuzzy sets as proposed by Zadeh [51]. More detailed information about Fuzzy Logic (FL), which is sometimes called Fuzzy Set Theory, can be found in textbooks such as Klir and Wierman [26], Klir [50], Ross [53] or Zimmermann [52].

FL is split into three main steps [52, 53, 65] as depicted in Figure 2.7:

1. **Fuzzification**
   The first step is to convert or map the crisp input values to fuzzy membership functions.
2. **Inference Engine (Rule Evaluation, Aggregation & Ranking)**
   In the second step, predetermined rules are fired triggered by the fuzzy input values obtaining intermediate fuzzy results. These intermediate fuzzy results are aggregated into a single fuzzy set and if necessary ranked.
3. **Defuzzification**
   The final step is to convert the resulting fuzzy set from the inference engine to a crisp output value.

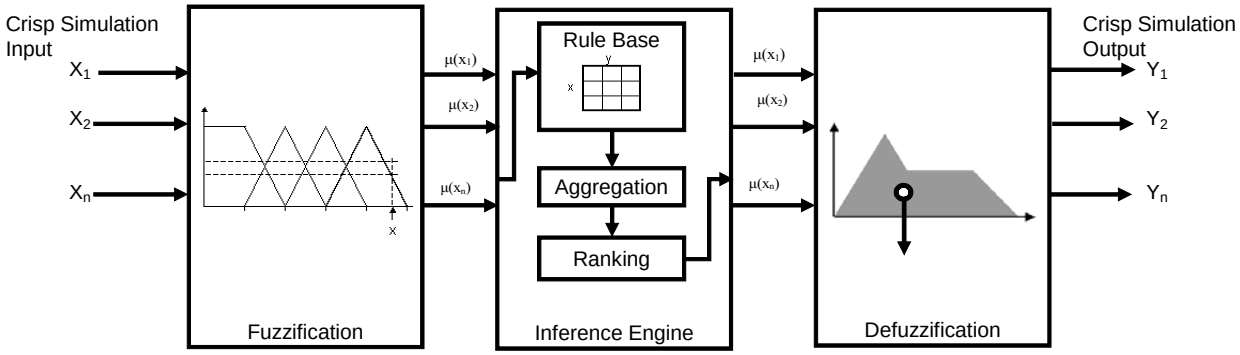


Figure 2.7 Block flow diagram of a fuzzy-logic simulation *(after Mendel [65])*

None of the three steps in a fuzzy simulation is trivial. Each step requires the modeller to make *value* decisions based in parts on *subjective* factors without prior knowledge of which one may be best suited to the problem or system at hand [52, 53].

1. Fuzzification

Fuzzification, the first step in a fuzzy simulation, transforms the initially crisp input variable into a fuzzy variable.

The question of how to obtain an appropriate membership function is not addressed here. Two introductory texts addressing this question are Ross [53] or Zimmermann [52]. As an example, consider the fuzzy sets for the universe of temperature X in Figure 2.8. Three triangular membership functions are shown: "low", "medium" and "high". The example crisp input variable x = 8 would fuzzify into 0.8 membership in the set "medium" and 0.4 membership in the set "high".

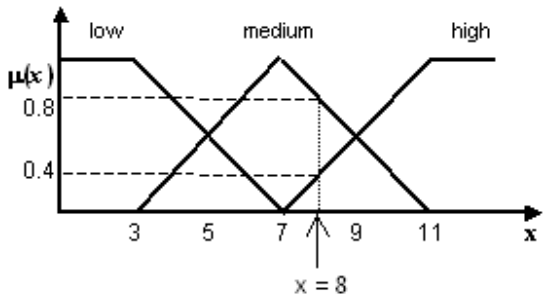


Figure 2.8 Fuzzification of a crisp input in domain X *(after Ross [53])*

2. Inference Engine (Rule-Base, Aggregation, Ranking)

The second step, the so called inference engine, expresses the relationships between the input and output sets. The inference engine itself is composed of the rule-base, aggregation and ranking section.

The first step of the inference engine is the *Rule-Base Evaluation* [66]. The rules of the system, which assume the relationship between the input and the outputs, are both deterministic and known and are encoded in the form:

IF *condition* THEN *result*

The *condition* is called the antecedent and the *result* is called the consequent. In most applications the antecedent will contain conditions of more than one domain as depicted in Figure 2.9:

IF *Antecedent 1* is *slow* AND *Antecedent 2* is *bad*,
THEN *result* is Fuzzy MAX Operation.

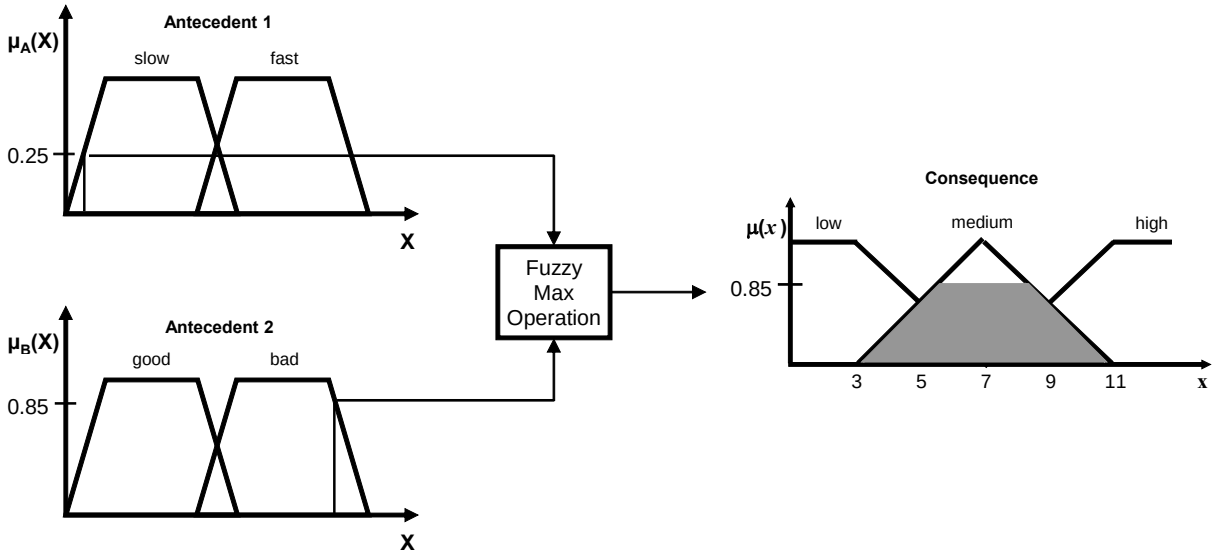


Figure 2.9 Example rule-base evaluation *(after Ross [53])*

The rule base can become very large depending on the number of input and output domains and the number of individual membership functions in these domains [53]. For example a system with a single output and two input domains, each of which is partitioned into six fuzzy sets, requires thirty-six rules to describe the relationship between the inputs and the outputs.

Since fuzzy logic is dealing with fuzzy numbers and not with crisp numbers, a fuzzy number may result in more than one rule being evaluated ("fired"). Therefore after all the rules have been evaluated, the results may have to be *aggregated* into a single fuzzy result. As an example Figure 2.10 shows the fuzzy max operation of two consequence rule evaluations into one aggregated output set. The main problem of aggregation is to choose an appropriate aggregation

procedure out of the many available [52, 53]. Appendix 1 gives some examples of other aggregation procedures such as union and intersection.

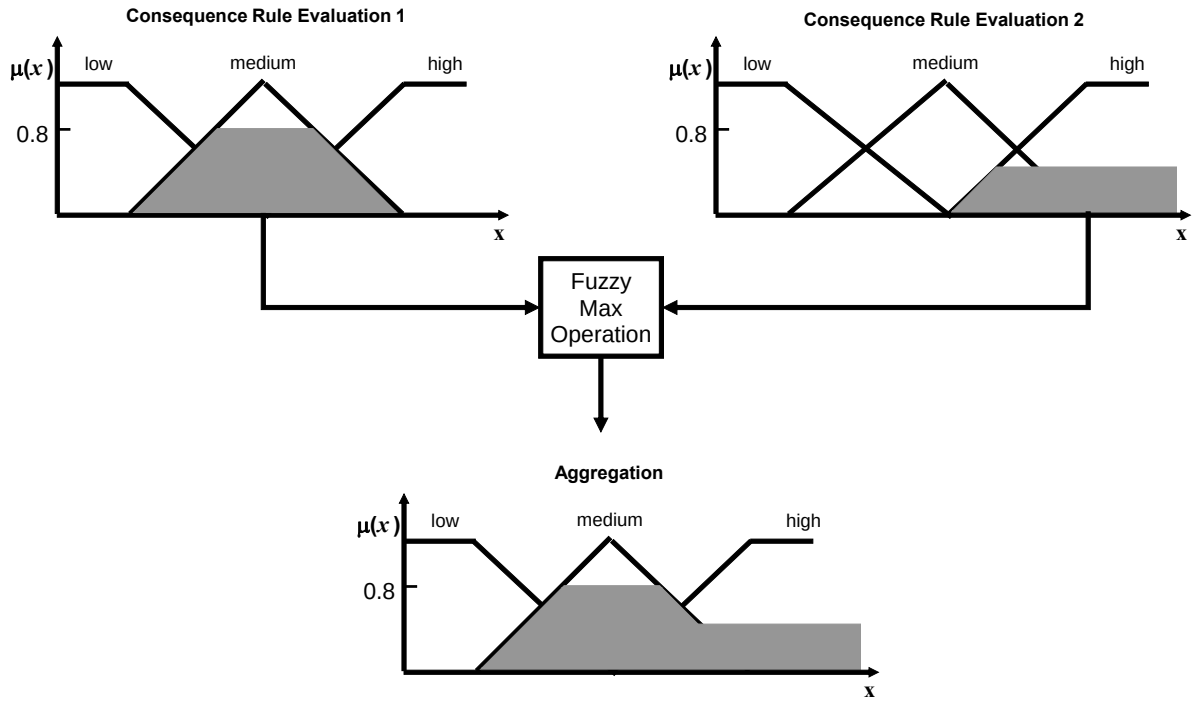


Figure 2.10 Example aggregation *(after Ross [53])*

Ranking is the sorting of fuzzy sets according to certain mathematical rules. Ranking of fuzzy numbers can prove difficult for a number of reasons. For one the ranking procedure outcome may depend on the ranking procedure itself. For another numerous ranking procedures have been proposed, but none of the ranking procedures is generally accepted and a number of ranking procedures may work in a given circumstance, leaving the choice of the ranking procedure open to personal preference [67]. More information on ranking is provided by Ross [53] and Zimmermann [52], whereas a comprehensive survey on ranking methods is offered by Wang and Kerre [68, 69].

3. Defuzzification

The third and last step in a fuzzy simulation is to subject the result of the inference engine to a defuzzifier to obtain a single real number as the crisp output. Defuzzification is therefore the opposite step to fuzzification returning a fuzzy output set back to a single crisp value [53].

The choice of the defuzzification method also depends on the problem, in other words heuristics [70], because of two difficulties. First, numerous defuzzification methods exist. Roychowdhury and Pedrycz [71] describe 12 defuzzification methods with the *standard ad hoc methods* of the center-of-gravity, mean-of-maxima, centre-of–mean and midpoint-of-area being the most popular ones. Second, neither any proposed underlying mathematical meaning of the defuzzification step nor its relationship with the actual operation of the system is/can be made clear [70, 71], leaving, once again a part of the FL procedure open to personal preference and criticism. In practice, two defuzzification methods find common use however [52, 53, 71]. These are the methods of center-of-gravity and mean-of-maxima.

The center-of-gravity defuzzification method is used by this work (see Section 3.3.1.10). Figure 2.11 provides a depiction of this method, which is defined by [53]:

$$X_D = \frac{\int \mu(x) \cdot x dx}{\int \mu(x) dx} \tag{2.3}$$

The mean-of-maxima defuzzification method (see Figure 2.11) is defined by:

$$X_D = \frac{a + b}{2} \tag{2.4}$$

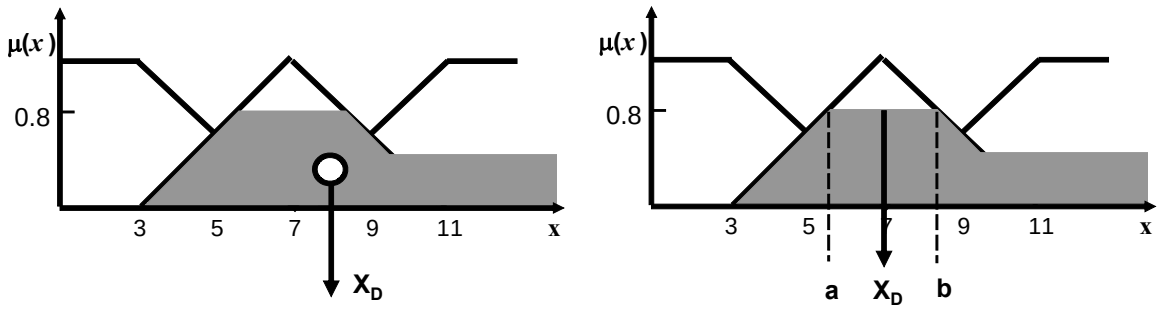


Figure 2.11 Center-of-gravity (left) and mean-of-maxima defuzzification (right) *(after Ross [53])*

2.3.1.3 Evidence Theory and Generalised Information Theory

Two theories *evidence theory* and the *generalized information theory* (GIT) which both include probability and possibility theory as a special case [50] are mentioned here for reasons of completeness, but are not applied in this work to describe uncertainty in DI/WFI systems.

Evidence theory deals primarily with so called *belief functions*. That is to say, the information available is very weak and this situation is characterized as having incomplete knowledge. No publication applying evidence theory to model uncertainty in dynamic systems could be found in the literature. But literature on how evidence theory can be applied in engineering design, specifically reliability engineering has been published. A recent review on these developments and how evidence theory is applied in engineering design is detailed by Oberkampf and Helton [72].

Some have attempted to develop an all encompassing theory of uncertainty, the so-called "general theory of uncertainty". A recent textbook by Klir [50] provides a current overview detailing these efforts.

2.3.2 Mathematical Methods Describing Uncertainty

Figure 2.12 shows schematically five different mathematical methods describing uncertainty and how much uncertainty is inherent in each. The figure is based on an image developed by Zadeh [73], who for good reasons did not include the method *Service Factor* in his image, as it is an

"*Engineering Approach*" rather than a *Scientific Approach* as classified in Figure 2.12. The following sections will describe these five methods of uncertainty modelling in more detail and detail why only the fuzzy logic and statistical methods were selected for this work.

Other less popular methods describing uncertainty than the five listed above such as interval-valued probabilities [50], fuzzy probability [50], imprecise coherent probabilities [74], coherent lower previsions [74] and info-gap models [74] are not reviewed here.

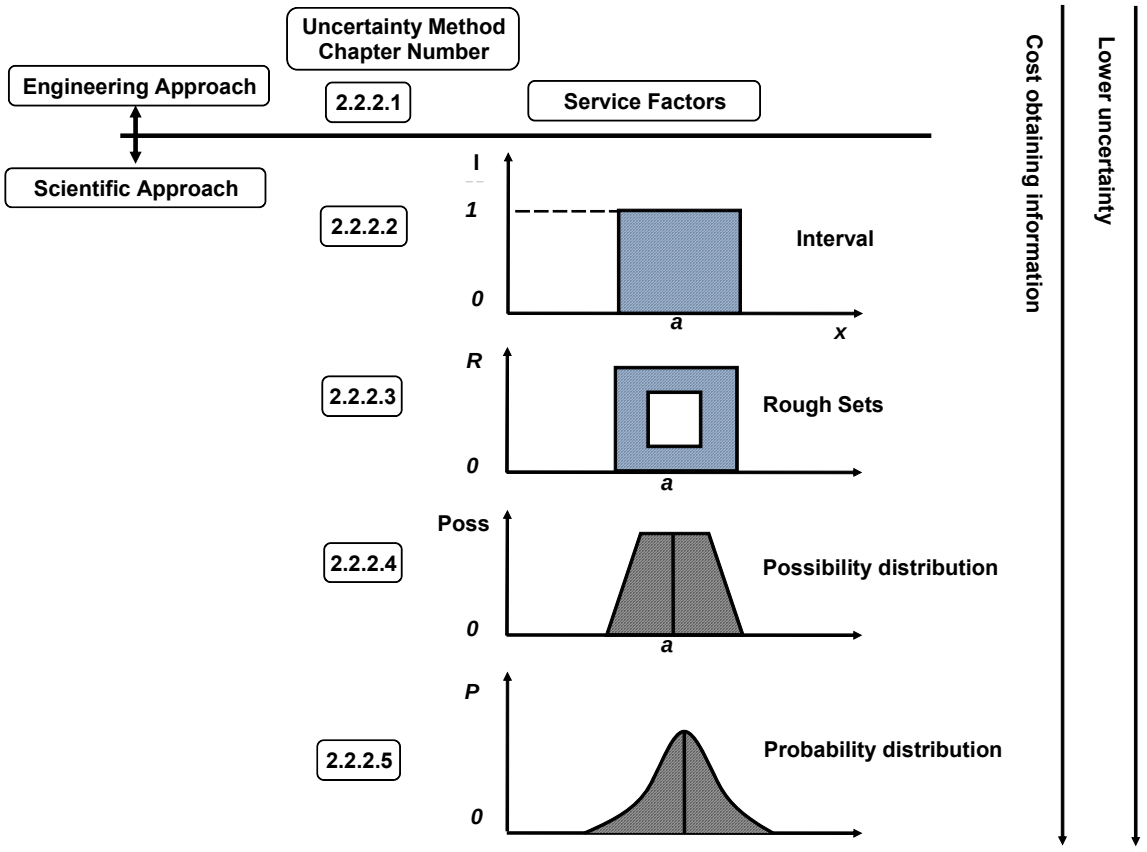


Figure 2.12 Relative order of the five uncertainty methods investigated (*after Zadeh [73]*)

2.3.2.1 Service Factors / Diversity Factors

In the industrial practice a diversity factor is often used to handle uncertainty in many dynamic systems including electrical distribution [75] and WFI distribution systems [13, 20]. The ISPE Baseline Guide [13], a pharmaceutical industry guide regarding the design and operation of High Purity Water systems, suggests using a diversity factor to "level out anticipated usage" without giving any hints about how to obtain a reasonable factor. In practice, the diversity factor is usually set somewhere between 0.7 and 0.8 [20].

The diversity factor denoted f_{Div} is defined as [20]:

$$f_{Div} = \frac{\dot{V}_{Design}}{\dot{V}_{Max}} \tag{2.5}$$

McQueen et al. [22] explain how the diversity factor method is woefully inadequate in terms of obtaining the design load of an electrical distribution system, as it cannot model the dynamics of the system. Extending the argumentation of McQueen et al. [22] to DI/WFI systems makes it clear that a diversity factor should not be used to calculate the spare capacity of a DI/WFI system should any degree of precision be required. Furthermore, the notion of a diversity factor has no foundation in any (rationally based) theory of uncertainty. The diversity factor method can thus be discounted as a viable method to treat uncertainty in DI/WFI systems where a degree of precision is required e.g. around making investment decisions.

2.3.2.2 Interval Methods – Bounded Uncertainty

Interval methods or bounded uncertainty [76] are often applied if only the upper and lower value are known but no information whatsoever on the behaviour of the data between those two boundary values is available. Therefore interval methods require little information.

Bounded uncertainty can be described [53] as a crisp set in which an element *x* belonging to a set **A** is defined as $x \in \mathbf{A}$, whereas an element that is not a member of **A** is noted as $x \notin \mathbf{A}$:

$$\chi_A(x) = \begin{cases} 1 & \text{for } x \in A, \\ 0 & \text{for } x \notin A. \end{cases} \tag{2.6}$$

As bounded uncertainty requires only knowledge of the boundaries and no information on the behaviour between these boundaries, in other words very little information, it was discounted as a description of uncertainty for DI/WFI systems.

2.3.2.3 Rough Sets

Rough sets were proposed by Pawlak [77] in 1982 and have enjoyed applications ranging from machine learning to decision finding in expert systems [77]. Rough sets are similar to intervals in that there is an upper and lower boundary, but the approximation to the uncertain value **A** is by lower and upper crisp sets. Rough sets may be seen as a formalism of uncertainty with a possible accuracy located somewhere between the interval and fuzzy set methods [78]. So far, rough sets have not, to the knowledge of the author, been used in simulations of schedule uncertainties. This combined with rough sets being only slightly more accurate than interval methods, has led to the decision not to utilise rough sets for this work.

2.3.2.4 Possibility Theory – Fuzzy Logic

As a motivation for the decision to simulate uncertainty in DI/WFI systems with FL, the next subsection reviews some of the successes FL has enjoyed over the last couple of years, whereas the subsection following that reviews FL in the simulation of water and other related systems.

2.3.2.4.1 Fuzzy Logic Successes

Fuzzy Logic (FL) has enjoyed substantial successes in control system design including commercial applications. McNeill [58] cites fuzzy control of vacuum cleaners, industrial furnaces, washing machines, elevators, TV sets, and the control of the subway in the Japanese city of Sendai. A recent review on the current state of the art of FL control system engineering is provided by Feng [59].

A high level of theoretical and practical sophistication has been reached, particularly in the control of non-linear systems. The great advantage of fuzzy logic in control systems engineering is that the behaviour of the system can be modelled without requiring a detailed mathematical model. Instead, FL simulates the behaviour of the human operator controlling it. Consequently systems can be controlled, which may be difficult or impossible to control using a mathematical model or which may be slow to respond using the classical PID controller approach. Examples of such FL control systems described in the literature include highly non-linear systems such as pH-control [79], a fluidized bed reactor [80] or dissolved oxygen in a pilot plant [56].

FL is also applied in expert systems [81]. Examples of such expert systems as they may apply in the chemical industry are provided by Azadeh et al. [55] and Zio et al. [57]. Azadeh et al. [55] published a paper describing a pump diagnostic expert system utilising FL. Zio et al. [57] presented a FL expert system modelling dependence of successive operations involving human interactions. The stated advantage of the FL model is improved traceability and repeatability rather than dependent on the opinion of an expert [57].

2.3.2.4.2 Fuzzy Logic Methods for System Analysis

According to Ross [53] and Zimmermann [52] FL gives the modeller the means to deal with ill defined, vague, uncertain data, for which probability theory does not apply as, for instance, the data are not random. Moreover, FL allows the mathematical treatment of linguistic information [82], as FL is put forward as a method to handle vague, linguistic statements often only based on the subjective opinion of 'experts'. In the context of this work, however, the ability of FL to simulate schedule uncertainty is of most interest.

A number of authors among them Azzaro-Pantel et al. [83], Perrone et al. [67], Zhang et al. [84] and Nucci and Grieco [85] have combined discrete event simulation with FL to simulate workshop problems. Workshops are optimization problems in manufacturing, employing sequential, parallel or a mixture of both operations as exemplary shown in Figure 2.13. Consider n jobs J_1, J_2, ..., J_n , which may be different to each other, to be scheduled on m machines m_1, m_2, ... m_m. The objective is to minimize the total length of the production schedule for the n jobs. This work, however, is not concerned with the details of workshop simulation and therefore refers the reader to the relevant literature, for instance Pinedo [25].

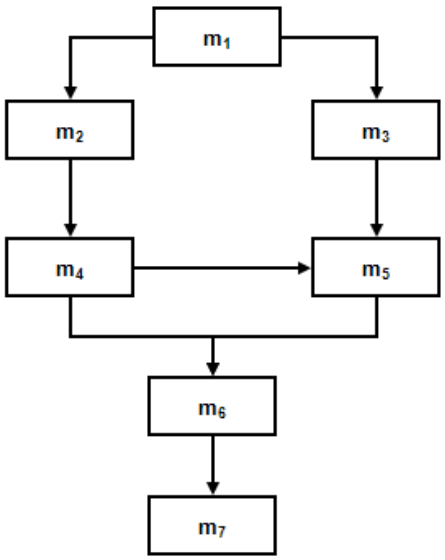


Figure 2.13 Example workshop flow

The main challenge in FL simulations of workshops or other time sequencing operations is keeping the fuzzy clock up to date and especially avoiding or reducing the number of time paradoxes, because ranking of overlapping fuzzy numbers in time sequencing operations such as workshops is an unsolved problem [67, 83]. All too often *ad hoc* procedures have to be implemented [85], as with an increasing simulation time the number of time paradoxes typically increase [85].

Researchers have proposed new ranking methods hoping to overcome the time paradox problem. Perrone et al. [67] compared a ranking method they proposed with some of the existing ones showing that the new one performs better. Unfortunately as with all ranking methods proposed so far, it may only work in a particular circumstance, which in the case of Perrone et al. is a simple one-machine production system. Therefore it is not clear how this ranking method performs in a multi-machine workshop as per Figure 2.13.

An important limitation of some ranking methods is noted by Zhang et al. [84]. Some ranking methods cannot accept fuzzy numbers other than triangular or trapezoidal. This limitation, of course, becomes difficult to overcome should the result of an aggregation procedure be a fuzzy distribution other than triangular or trapezoidal. In 2008 Nucci and Grieco [85] proposed a new fuzzy ranking method called TRM (Temporal Ranking Measure) and benchmarked it to a workshop test case as provided by [83]. While this new ranking method would appear to be working well [85], it does not represent the final verdict on ranking problems in fuzzy discrete event simulations. In summary, it would appear the research on FL is not advanced enough to be applied on industrial workshop problems.

For this work, however, the time paradox problem is of no consequence, as the FL model of the DI/WFI distribution systems is not modelled as a workshop. A workshop is an optimisation problem comprising of a number of sequential machines as shown in Figure 2.13. Should such a problem be simulated by FL the ranking procedure may be critical to avoid time paradoxes. In the FL DI/WFO model, sequential operations are not modelled as interdependencies of the valve actuations are excluded from the model (see Section 3.3.1). Instead every valve operation is modelled on its own independent from the opening times of other valves. Not every calculation

involving fuzzy sets is time dependent. Revelli and Ridolfi [86] applied fuzzy sets on the uncertainties in water distribution networks. They tested fuzzy sets on two small water distribution networks with the uncertain pipe roughness given as a fuzzy number. The calculation was run a number of times with different α-cuts (see Appendix 1 for more information on α-cuts) of the fuzzy numbers reducing the problem to interval mathematics. Branisavljevic and Ivetic [87] expanded the α-cuts method proposed by Revelli and Ridolfi [86] to a real water distribution system. They used the public domain EPANET [4] software for the modelling of the municipal water distribution network of Becej, Serbia (453 nodes and 616 pipes). A genetic algorithm was applied on the α-cut input variables in order to find a relationship between input variation and output sensitivity. This information may then be used to improve the model [87].

Perrone et al. [67] conclude that the results of a FL simulation always contain the results of a stochastic simulation, although FL requires only one iteration as opposed to a stochastic Monte Carlo simulation which requires multiple runs. However, their conclusion is based on a number of assumptions such as using triangular-shaped FL and statistical distributions, a one-machine production system and a LIFO (Last-In-First-Out) strategy for the queuing system. Therefore it may be wrong to generalise their conclusion.

Zhang et al. [84] found that a FL simulation is fast to execute, as only one run is required. In addition FL models are easy to built because FL membership functions are often only triangular or trapezoidal. Furthermore Zhang et al. [84] corroborates an earlier paper of Perrone et al. [67] in that the statistical results of a FL simulation always contain the results of a stochastic simulation.

Azzaro-Pantel et al. [83] included the opinion of experts in their FL model of a semiconductor workshop. The experts gave approximate operator intervention times in the following linguistic terms: “It is impossible that this task will take less than 2 min and more than 10 min (values of and the possibility for this task to be carried out within approximately 4 or 6 min is very high)“ which Azzaro-Pantel et al. [83] translated into trapezoidal fuzzy numbers. A novel approach to defuzzification is presented by Azzaro-Pantel et al. [83], who did not defuzzify the results of

their simulations. Whereas this circumvents the difficulties associated with defuzzification (see Section 2.3.1.2.1), it leaves the end-user with the task of interpreting the fuzzy results of a simulation. A potential difficulty is that it requires an understanding of FL by the end-user, which is, taken the complexity of a FL procedure into account, somewhat doubtful. However, a FL workshop analysis as performed by Azzaro-Pantel et al. may not require a defuzzification step, because the authors may be available to interpret the results. But the FL model of a DI/WFI system proposed in Chapter 3.3 requires it, as this step is required to finally compute the incurred time delay.

In summary, one can say that a FL uncertainty simulation of a system may work, but success with this method appears not to be a given and dependent on the application as well as the appropriateness and quality of the selected inputs. Moreover, according to Chen and Lin [88], who published extensively on the possible application of FL simulation of semiconductor workshops [88-90], the statistical approach is still the most accurate method to simulate workshops and is often used as a benchmark for FL models, which leads on to the next section on statistical methods in simulation.

2.3.2.5 Probability Theory – Statistical Methods

Statistical methods are the method of choice to model schedule and other uncertainties encountered in industrial practice [25, 33, 42]. Therefore statistical methods were chosen to simulate uncertainty in DI/WFI systems. In the following two subsections, some successes of statistical methods are mentioned, and then statistical methods used in the simulation of water distribution systems are briefly reviewed.

2.3.2.5.1 Statistical Methods Successes

The successes of statistical methods to solve problems involving uncertainty in many fields such as finance, physics, biology, chemistry and engineering [91, 92] are undisputed, and therefore statistical methods hardly need to be promoted. A small sample of recent papers from various engineering disciplines may serve to illustrate this point:

Cronin et al. [93] applied a stochastic model to improve the prediction that hoses used to convey chemicals may rupture because of thermal expansion of the trapped solvents if atmospheric conditions changed. Page et al. [94] present a stochastic simulation of occupant presence, the results of which may provide the inputs for, for instance, building heating models. Semaan and Zayed [95] provide a description of a stochastic diagnostic tool modelling the performance of subway stations, which is shown to model reality better than purely deterministic models as uncertainty is taken into account. The optimal design of electrostatic precipitators used to abate particle emissions from flue gases in for example coal-fired power plants and cement plants is improved by a stochastic method proposed by Zhao and Zheng [96]. Stochastic modelling was also used by Nakamura et al. [97] to improve the particle mixing prediction on grates feeding waste-to-energy incinerators. The goal was to improve the design of the grates and to gain a better understanding of the parameters controlling the combustion process.

Statistical methods have also been successfully applied in control system engineering and expert systems. The cutting operations of a lathe, for instance, are difficult to control, but this control problem may be solved with a stochastic predictive control system as suggested by Goggos and King [98]. An early example of the stochastic control of an industrial type continuous chemical reactor is provided by MacGregor and Wong [99]. The main difficulty in applying stochastic control is modeling the reactor to the accuracy required. MacGregor and Wong [99] overcame some of these difficulties by utilizing linear empirical relationships they found between the input and output parameters.

It is possible to integrate uncertainty in expert systems using stochastic methods, as described by Zvarova [100] and Spiegelhalter et al. [101] on the example of medical expert systems. An example of an stochastic expert system in engineering is presented by Matijevics and Józsa. [102], who built a system for the assessment of the reliability of electric power networks in which uncertainty is modelled via stochastic distributions.

2.3.2.5.2 Statistical Methods for Water System Analysis

Stochastic methods have been used to address various questions about different municipal water distribution systems such as pipe failure rates and maximum possible water demand.

Watson et al. [103] developed a discrete-event Bayesian pipe failure model of a municipal water distribution system to test a number of pipe maintenance policies in order to identify optimal pipe repair strategies. They found that the Bayesian approach is particularly useful in reducing data collecting problems, that is, missing or incomplete data. This paper shows the ability of the Bayesian approach to learn, should more information become available.

Pipe failure rates or, generally speaking, the survival data of many systems, are usually assessed with statistical analysis [104]. Boxall et al. [105] developed statistical expressions to predict annual pipe failure rates of water distribution mains from two sample data sets as a function of diameter, length and age of the pipe. To complicate matters further Dehghan et al. [104] recently noted that the failure rate prediction of pipes may be complicated by seasonal influences. They statistically analyzed the pipe failures of a municipal water distribution pipes in Melbourne, Australia, and found that pipe failure rates are also influenced by seasonal rainfall and amount of rainfall.

The designer and operator of a municipal water distribution is usually only interested in the maximum possible flow through the system in order to size the pumps and pipes, whereas the water demand as it evolves over time is of lesser or no interest. Wong at al. [106] used the classical approach comprising of data collection, finding a stochastic distribution and Monte Carlo simulation to calculate the maximum simultaneous water demand from domestic washrooms in tower blocks in Hong Kong. They found that the likely maximum simultaneous water demand is only 50-60% of the original design assumptions made, opening the possibility of reducing the installation costs of pipe installations in tower blocks of similar size. The peak water demand from a small town municipal water distribution network system was simulated with stochastic distributions by Tricarico et al. [107], who developed a new formula for the maximum demand from a municipal water systems as a function of the number of connected inhabitants.

One rare example where the water demand profile as it evolves over time is of interest is provided by Janković-Nišić et al. [108], who tried to find optimum locations for flow meters to detect leaks in a municipal water systems, while uncertainty in demand was modelled with stochastic distributions. They combined a stochastic model with the EPANET [4] software. A municipal water distribution system comprising 402 nodes and 439 pipes was simulated with the duration and flow of every appliance in a household as the random variable.

2.3.3 Criteria for Selecting Probability or Possibility Theory

This section concentrates on probability and possibility theory and their methods because these two methods were chosen to simulate uncertainty in DI/WFI systems. Although the suggestion has been made that probability and possibility theory are "complementary rather than competitive" [109], from a programming and project execution point of view this may not be entirely true, as generally there will not be enough time available to model a DI/WFI system utilising both approaches should one approach fail. Therefore, this work sees probability and possibility theory as competing. But, if these two approaches to uncertainty modelling are competing, the question arises which criteria, if any, the literature offers to select one theory over the other. Unfortunately as will be shown, the literature cannot answer this question. On the contrary the literature review reveals two camps; in one the proponents of probability theory, in the other the proponents of possibility theory. Both camps appear to be unable to come to an agreement under which circumstances which theory of uncertainty should be used [74]. A consequence of this general disagreement is that the modeller may have base the decision between these two theories of uncertainty on personal preference only, because a rationally-based decision criteria is lacking. But, because possibility theory is the younger method of uncertainty modelling, it should offer clear advantages over probability theory in order to be applied as lacking clear advantages there would hardly be a point doing so. Unfortunately the advantages possibility theory may have over probability theory are not clear from the literature review.

The remainder of this section is devoted to a comparison of the differences between probability and possibility theory from the following points of view:

1. Axiomatic Differences Between Probability and Possibility Theory
2. Theoretical Differences between Probability and Possibility Theory
3. Practical Differences between Probability and Possibility Theory

2.3.3.1 Axiomatic Differences between Probability and Possibility Theory

Probability and possibility theory have been compared before, not in simulation, but in the field of engineering reliability or risk analysis. Nikolaidis et al. [63] compared possibility and probability theory and stated that the main difference between the axioms of possibility and probability theory is that probability is additive whereas possibility is subadditive, as is shown in the following caparison:

Probability Theory	Possibility Theory	
$\sum_{i=1}^{n} P(U_i) = 1$	$\sum_{i=1}^{n} POS(U_i) \geq 1$	(2.7)

The work of Nikolaidis et al. [63] is concerned with catastrophic failures of components or a whole system under uncertainty. In other words it is reliability engineering where either probability or possibility distributions are combined and the resulting joint probability or possibility distribution is calculated as per equations 2.7 above. The outcome of their work is that sometimes possibility and other times probability theory offers the best results and that **both** should be used if little information is available i.e. during design development as information may be lacking. The reason why both approaches to uncertainty should be used if only little information is available is because it is unclear which approach gives the more conservative result. However, if enough data are available probability theory should be used [63].

Aughenbaugh [62] rejects possibility theory as an approach to solve engineering problems on theoretical reasons, for possibility theory having no operational definition. He argues that it is simply not clear what possibility means, hence drawing conclusions from it may be misleading.

There may also be a rather simple reason for rejecting possibility theory and FL in particular, in that it is superfluous. Many researchers, among them Lindley [35, 36, 110, 111], Almond [39] and others [74, 112] reject possibility theory on such grounds. They assert that probability theory can do all that possibility can do. Lindley [110] proclaims the "inevitability of

probability" as the only acceptable means of describing uncertainty. Furthermore Lindley [111] states that no one has given an example that has a more satisfactory solution with possibility rather than with probability theory. This statement made by Lindley in 1995 may carry some weight: the literature review carried out as part of this work could not find any papers giving such an example. That said, many engineers and scientists appear to be more comfortable with FL than with statistical methods as Almond [39] and Laviolette et al. [112] speculate.

2.3.3.2 Theoretical Differences between Probability and Possibility Theory

Inherent fuzziness rather than randomness, the lack of a probability distribution or the difficulty of obtaining one, or because only the opinion of an expert expressed in vague linguistic terms may be available, are often cited as reasons to select FL over statistical methods.

Lack of a Probability Distribution

The question of how to obtain a suitable distribution, be it a possibility or probability distribution, is not addressed in this work because Banks et al. [41] and Law and Kelton [1] already provide basic guidance on how to find suitable probability distributions whereas Ross [53] provides the same for possibility distributions. Indeed finding a suitable probability distribution can be an elaborate, time consuming and difficult undertaking [1]. This difficulty has led some researchers for instance Ross [53], Zhang et al. [84], Nucci and Grieco [85] and Azzaro-Pantel et al. [83] to conclude that possibility might then be better than probability theory, as FL membership functions can be drawn from very little information. This is a curious reason for choosing a possibility (membership function) over a probability (statistical) distribution, because no reason is given why or how a possibility distribution could be estimated more easily than a probability distribution.

Expert Opinion

Despite repeated claims of superiority of FL in modelling expert opinion as expressed by Ross [53], Perrone et al. [67], Siler and Buckley [81] or Hong et al. [84], it is not the exclusive realm of modelling expert opinion as was already shown above. Lindley [110] for instance is very

clear on this having published a paper stating that very point in its title: "The Probability Approach to the Treatment of Uncertainty in Artificial Intelligence and Expert Systems".

Linguistic Expressions

Often FL is used in computing linguistic expressions (Zimmermann [52], Ross [53], Zadeh [82], Perrone et al. [113]). Language is inherently vague for example in expressions such as "John is tall", "Jack is young", because different people in different situations may provide different answers as Zimmermann [114] points out. However, the situation regarding linguistic information is not as clear-cut as Figure 2.14 may imply. Walley and de Cooman [115] investigated the claim the FL is better suited than statistical methods to model linguistic uncertainty. They dispute the claim that fuzzy has any special properties over statistical methods in modelling linguistic uncertainty. In an earlier, but not as detailed paper, Lindley [110] comes to the same conclusion as Walley and de Cooman [115].

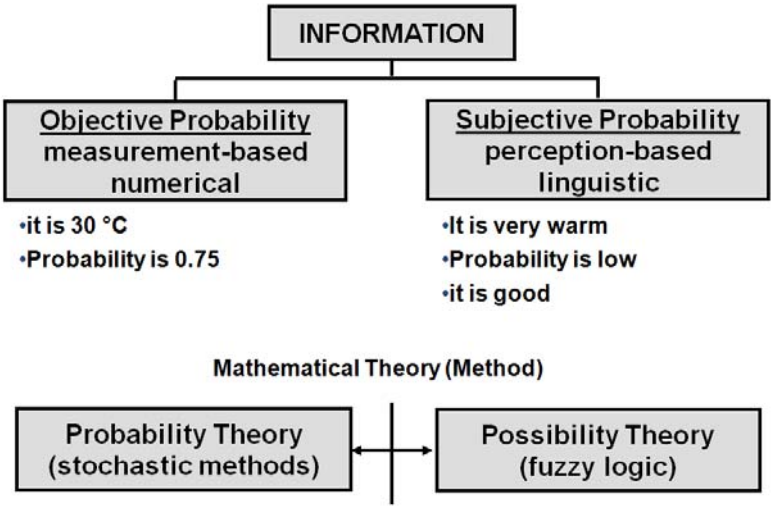


Figure 2.14 Fuzzy logic proponents view of the difference between perception and measurement based information *(after Zadeh [73])*

Inherent Fuzziness

Zadeh [82] expressed the "difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are fuzzy." Many papers on fuzzy logic, among them Ross [53], Zimmermann [52] and Dassisti and Galantucci [116] motivate their use of fuzzy logic by stating that probability theory cannot be used as the uncertainty is fuzzy rather than random.

General agreement has been reached that random processes or what statisticians call *objective* uncertainty should be treated with stochastic methods (Dassisti and Galantucci [116]). But this general agreement falls apart on the issue what enough data constitutes, as in some cases proponents of the FL method such as Dassisti and Galantucci [116] or Zeng et al. [117] state that possibility theory was applied as not enough data was available to formulate a probabilistic distribution.

The above authors, all proponents of FL, seem to ignore the *subjective* interpretation of probability, which allows description of events with stochastic methods, which are not related to frequency and are non-repeatable. The statisticians view of Figure 2.14 may hence be rather different not including any references to fuzzy logic as can be seen in Figure 2.15.

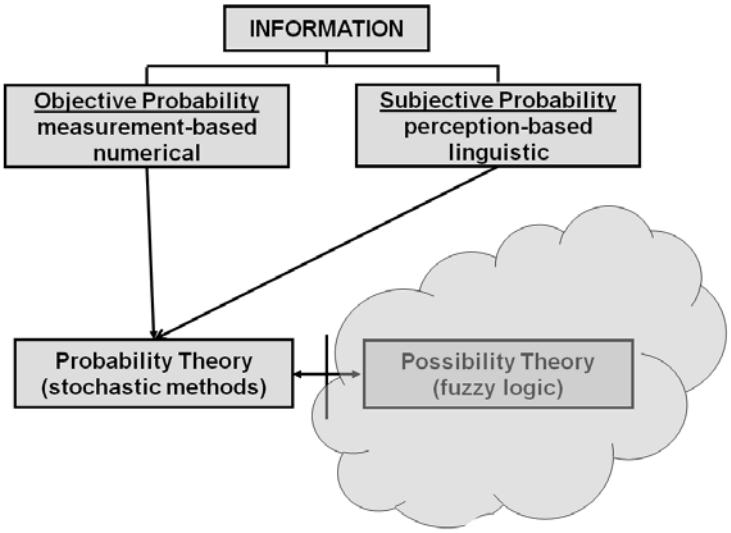


Figure 2.15 Statisticians' view of possibility theory as revealed from the literature survey
The cloud indicates the few – if any – applications statisticians would envisage for possibility theory.

In summary both probability and possibility theories are capable of attempting to handle subjective data including fuzziness, expert opinions and linguistic expressions.

2.3.3.3 Practical Differences between Probability and Possibility Theory

FL is riddled with practical difficulties. Many subjective choices may have to be made for the development of the membership function [39, 52, 53], the rule base [53], ranking [53, 68, 69], the aggregation procedures [52, 53] and the defuzzification procedure [53, 70, 71, 118] as was shown above, throwing into question the claim of some for example Ross [53] that FL is "natural". The claim of FL being "natural" may be related to Zadeh, who makes the following statement in his 1965 seminal paper [60]: "The notion of a fuzzy set provides a convenient point of departure for the construction of a conceptual framework which parallels in many respects the framework used in the case of ordinary sets, but is more general than the latter and, potentially, may prove to have a much wider scope of applicability, particularly in the fields of pattern classification and information processing. Essentially, such a framework provides a **natural** [note: not bold in original text] way of dealing with problems in which the source of imprecision is the absence of sharply defined criteria of class membership rather than the presence of random variables." Whereas the classification of uncertain information by fuzzy sets may indeed feel "natural", in the sense that it presents a model which acknowledges the role of 'possibilities' in reality, extending the claim of being "natural" to the entire fuzzy logic procedure is clearly not. Various papers e.g. [37, 39, 62, 110-112, 119, 120] criticise FL for lacking clear semantics that is applying ad-hoc rules to satisfy a situation. But this lack of clear semantics may be a serious deficiency to some, it is not for others: Dubois and Prade [54] and Ross [53] for instance declare this inherent lack of clear semantics of FL is part of the richness and ultimate strength of FL.

Dubois and Prade [54], who are two of the main investigators of possibility theory, admit that possibility theory has not been developed into the comprehensive and coherent framework as probability theory when they write: "For each semantic approach to probability, there exists a coherent and extensive explanation that justifies why the laws of probability theory should be adhered to, and thought experiments that explain how degrees of probability can be obtained. This is perhaps still the weak point of fuzzy set theory at the present time despite the existing

literature on the elicitation of membership grades.“ Of course it may be unfair to compare two theories of uncertainty when one (probability) is nearly 400 years older than the other (possibility). This thesis, therefore, takes a leap of faith and assumes that the proponents of possibility will eventually achieve a coherent and comprehensive theory. This approach to FL is taken as FL evidently works for many applications and engineers trying to find a solution to a problem may not care about gaps in the underlying theory if the method works.

2.4 Conclusions and Summary of Chapter

This chapter has tried to answer which theory of uncertainty should be used to describe uncertainty in a DI/WFI system. The question posed is difficult to answer as the literature does not appear to give any indication as to the circumstances under which possibility theory should be used over probability theory or vice versa. If anything, the literature review reveals a transition area between the two theories in which it is not clear which theory should be used. Should one be prepared to reject FL and all other theories of uncertainty as Lindley [35, 36, 110, 111] strongly recommends, only probability theory remains to model uncertainty.

This work does not adopt Lindley’s stand on FL outright. Instead, FL is not rejected for the simulation of uncertainty of DI/WFI systems because the literature reveals many successful, even commercial applications of FL. Furthermore a number of authors such as Ross [53], Ross and Kreinovich [33] and Moeller and Beer [121] believe that FL could be an alternative to statistical methods or at a minimum provide a different view or to cite Zadeh [109] once more “Probability Theory and Fuzzy Logic Are Complementary Rather Than Competitive”. Therefore, this work sets out to investigate if FL has indeed something to offer on the description of uncertainty in DI/WFI systems.

3 Mathematical Formulation of the DI/WFI Models

The previous chapter was concerned with the theoretical differences between the theories of probability and possibility. This chapter shows the practical differences between the two, highlighting again the fundamental differences between probability and possibility theory. Three discrete-event DI/WFI models are proposed in this chapter, which treat uncertainty in different ways, namely a purely deterministic (hereafter simply called "deterministic"), a stochastic and a model based on fuzzy logic (FL). As the deterministic model forms the basis for the stochastic and FL models it is presented first. Following this, the stochastic and FL models are developed.

Four limitations of the models are noted. First, the hydraulics of the DI/WFI system is not modelled. Therefore, the physical configuration of the DI/WFI distribution system, be it the typical loop, a deadleg or another system [13] is of no concern here. Figure 3.2 provides a graphical depiction of the simulated DI/WFI system showing that the layout of the distribution pipe is irrelevant for the models, because it is not needed to calculate the demand profile.

Some pharmaceutical production may require a minimum fluid velocity (i.e. 1m/s) in the return leg from the distribution loop to the storage tank in the belief that this is helpful in maintaining a low bacterial count (Meltzer [12]), demanding that the hydraulics be checked. Though this was originally thought to minimise bacterial attachment to surfaces and hence retard biofilm formation and subsequent microbial contamination, it is now recognised that the water velocity in the distribution loop is largely irrelevant (see for instance, Stoodley et al.[122] or Klauer [123]).

A second limitation of the models is that breakdowns of the DI/WFI system are excluded from the model, for DI/WFI systems are generally highly reliable as reported by Junker et al. [16]. A third limitation is that interdependencies between the various WFI opening operations are not modelled. In reality this is an incorrect assumption as for instance, the operation of filling a tank should precede the operation of transferring its contents. Lastly, all the DI/WFI models simulate

a 24-hour period. The models cannot simulate operations which exceed this timeframe without re-programming.

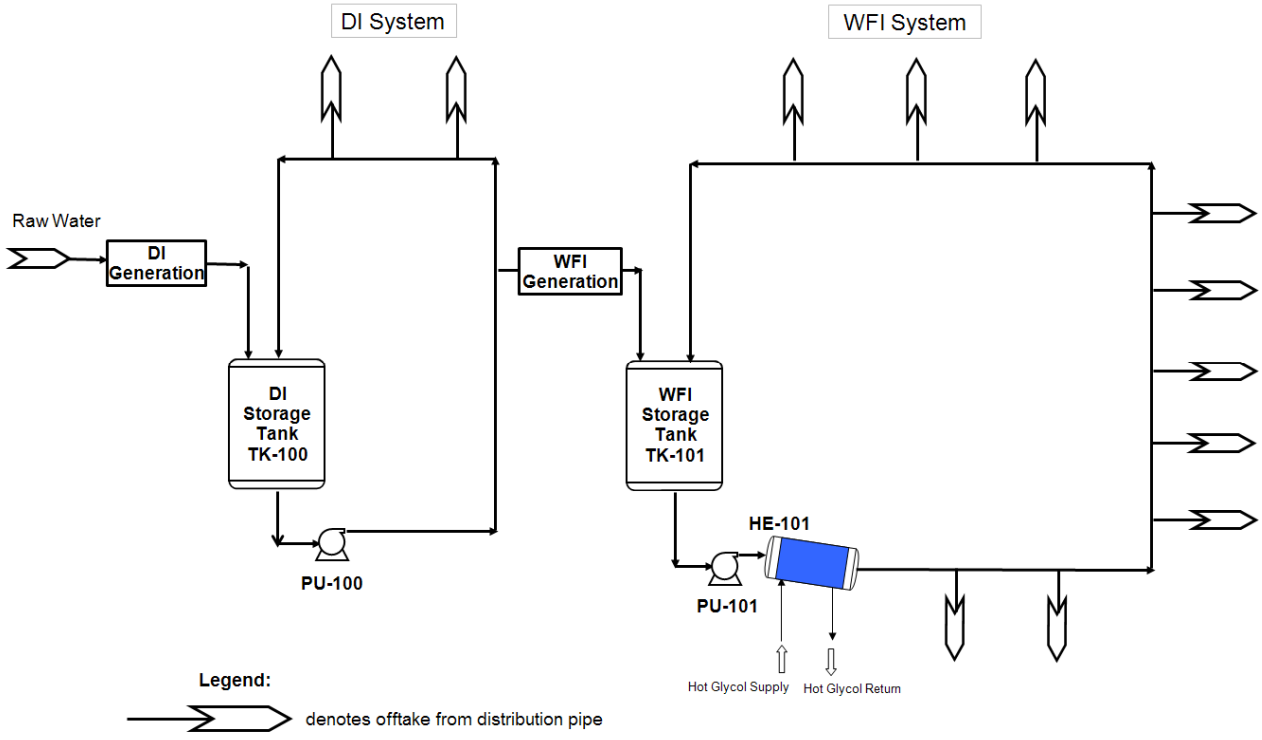


Figure 3.1 Process flowsheet of a typical industrial DI/WFI system

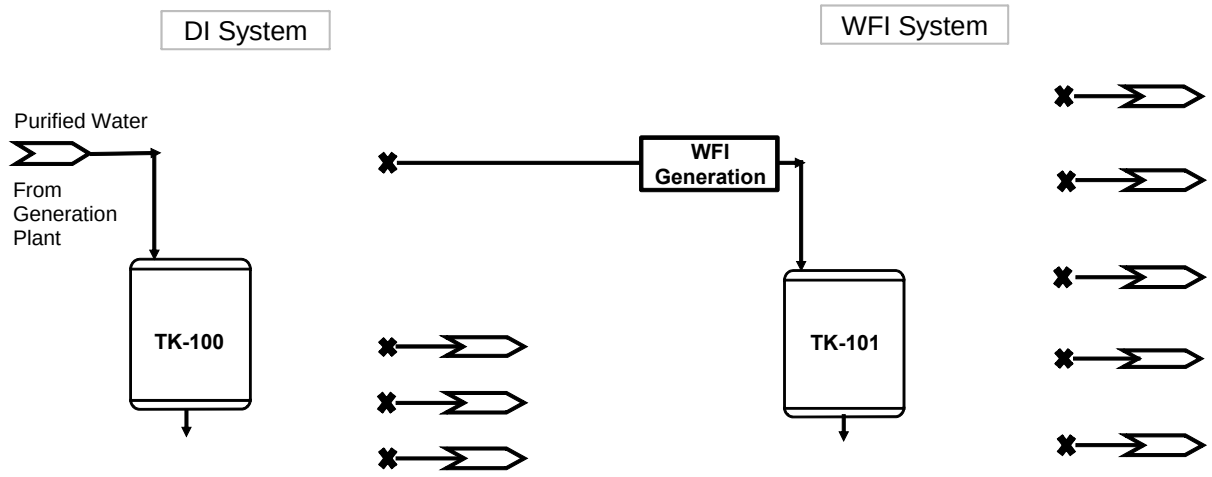


Figure 3.2 Process flowsheet of the DI/WFI system as modelled
This image indicates that the models of the DI/WFI are independent of the piping layout of the DI/WFI system. Therefore the layout of the distribution pipe i.e. loop, deadleg system etc. is irrelevant for the models.

The computer code listing of all the programs developed for this thesis are included in the appendices. Appendix 2 contains the Visual Basic for Applications (VBA) code of the stochastic and deterministic models, while Appendix 3 contains the code for the FL model.

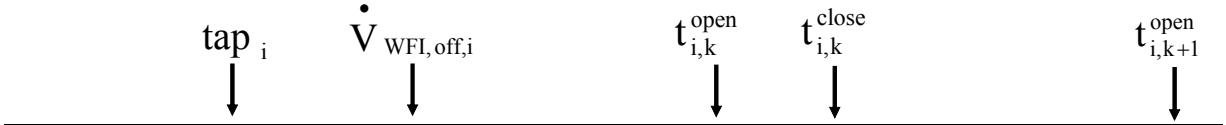
A general simplification can be made in the presentation of the models. The mathematical models of the DI and WFI generation and distribution systems are geometrically similar, as is evident from the process flowsheet given in Figure 3.1. For this reason, the model of the WFI system is presented first, followed by a section presenting the differences the model of the DI system has to the model of the WFI system.

3.1 Deterministic Model of WFI Demand

The deterministic model of a WFI system does not, by definition, include uncertainty. Therefore a deterministic simulation may not describe reality very well, as uncertainty is an integral part of any production environment [26, 124], but such models may still be useful. For example, the WFI models of Alexander [23] and Saraph [21] are both deterministic models, yet both models have been used in practice to make investment decisions.

The mathematical model of the deterministic case is formulated as a discrete event simulation (see Section 4.2.1) to compute the demand profile from the WFI distribution systems and a continuous simulation to calculate the water level variation in the WFI storage tank and is presented in the following text.

The WFI demand from the WFI loop is computed via discrete-event simulation. The event table shown in Figure 3.3 lists all the required input variables for the deterministic simulation such as flowrate and opening / closing times of all WFI valves. Some of the columns in the event table shown in Figure 3.3 are empty. These empty columns are used for the input of data of the stochastic simulation and are of no interest for a deterministic simulation and are therefore shown blank.



WFI Water Offtake Input Data

						1				
No.	Tag No.	Description of Offtake	Water Offtake [m^3/h]	Max. Uncertainty [%]	Description of Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]
1		ORIGINAL PROCESS DATA								
2		1. BULK PROCESSING SUITE								
3	T-100	Pre-rinse	1.5			2:10	2:15			10:00
4	T-100	Fill	1			2:00	2:06			10:30
5	T-101	Pre-rinse	1.5			3:10	3:17			11:25
6	T-101	Fill	1			3:15	3:25			11:15
7	T-102	Pre-rinse	1.5			1:20	1:30			9:30
8	T-102	Fill	1			3:15	3:25			10:15
9	T-103	Pre-rinse	1.5			1:00	1:10			9:00

Figure 3.3 Event table or user input interface for the deterministic simulation of a WFI System as presented within Excel (excerpt)
The variables shown above the event table indicate the user input columns of the variables required for a deterministic simulation.

A general assumption for the all models presented in this thesis is that all valve opening and closing times are independent from each other with the exception of the precedence constraints

(see below). With this assumption, the mathematical formulation of the deterministic discrete-event WFI distribution loop can begin.

The WFI loop contains *n* offtake valves denoted tap_i with $i = 1, 2, \ldots, n$ as listed on the event table (see Figure 3.3 & 3.4 for nomenclature). Each tap_i is associated with $k = 1, 2, \ldots, m$ activities denoted $\mathrm{act}_{i,k}$ comprised of two events: opening and closing of the valve in question.

The opening/closing interval denoted $\Delta t_{i,k}^{\mathrm{offtake}}$ associated with each $\mathrm{act}_{i,k}$ is composed of an opening time denoted $t_{i,k}^{\mathrm{open}}$ and a closing time denoted $t_{i,k}^{\mathrm{close}}$ as depicted graphically in Figure 3.4.

For the deterministic simulation the user has to provide for each $\mathrm{act}_{i,k}$ the opening times $t_{i,k}^{\mathrm{open}}$, the closing times $t_{i,k}^{\mathrm{close}}$ and the volumetric flowrate denoted $\dot{V}_{\mathrm{WFI,off,i}}$ as indicated in Figure 3.3.

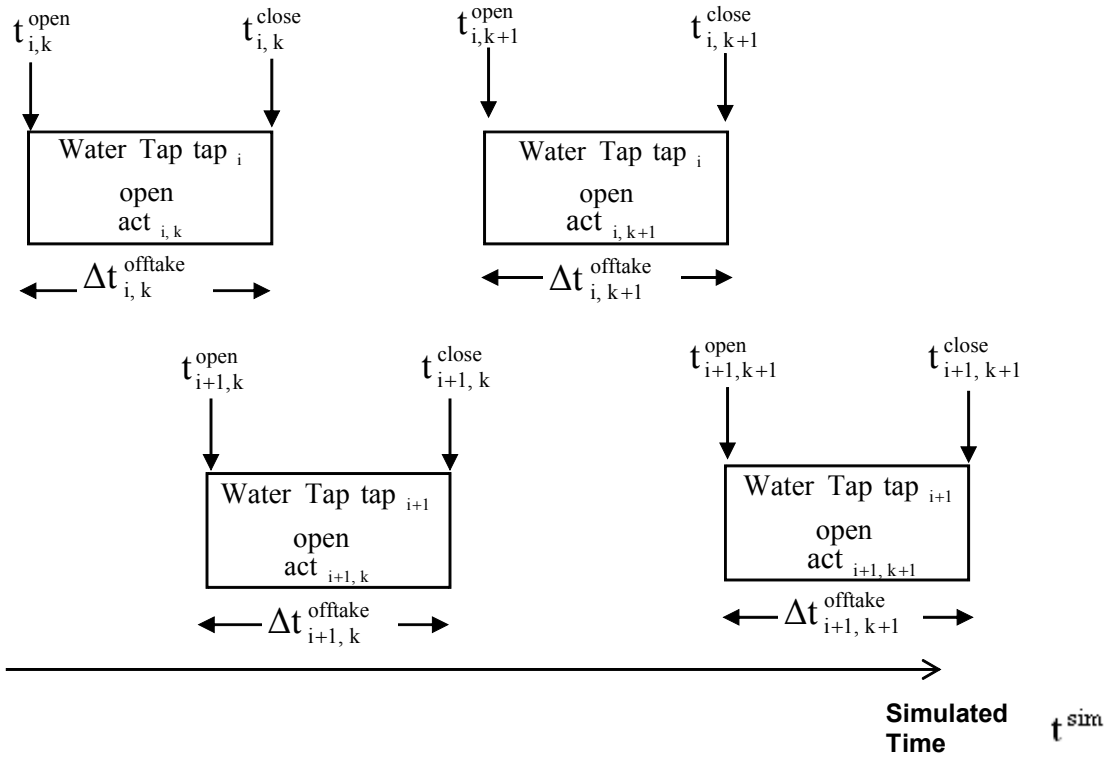


Figure 3.4 Nomenclature for the deterministic DI/WFI discrete-event simulation

Note: $act_{i,k}$ could finish before $act_{i+1,k}$ commences (see also event table Fig. 3.3).

The mathematical model of the tap opening is subject to a number of constraints:

1. Allocation constraints: A tap can only serve one process at any one time; it is not possible to open a tap while it is still serving another task.
2. Precedence constraints: This constraint ensures the correct sequence of operations of $act_{i,k}, act_{i,k+1}, \ldots, act_{i,k+m}$. The tap close time $t^{close}_{i,k}$ of $act_{i,k}$ must precede the opening time $t^{open}_{i,k+1}$ of $act_{i,k+1}$ defined by:

$$t^{close}_{i,k} \prec t^{open}_{i,k+1} \qquad \forall i = 1,2,\ldots,n; \quad \forall k = 1,2,\ldots,m \tag{3.1}$$

Furthermore the open time must precede the close time for each $act_{i,k}$:

$$t_{i,k}^{open} \prec t_{i,k}^{close} \qquad \forall i = 1,2,...,n; \quad \forall k = 1,2,...,m \tag{3.2}$$

3. Volume constraints: The volumetric offtake $\dot{V}_{WFI,off\,i,k}$ from the distribution loop for each $act_{i,k}$ is assumed to be constant for a given *i* and all *k* reducing $\dot{V}_{WFI,off,i,k}$ to $\dot{V}_{WFI,off,i}$ as shown in Figure 3.3. This simplification is justified as a WFI distribution system is a pumped, re-circulated system operating under a relatively constant pressure.
4. Duration constraints: For each $act_{i,k}$ the time interval $\Delta t_{i,k}^{offtake} = t_{i,k}^{close} - t_{i,k}^{open}$ is assumed to be constant, not being subject to uncertainty. This is a reasonable assumption, as only validated volumes $V_{WFI,off,i,k}$ are dispensed from the WFI distribution system and volume uncertainties are small.
5. Non-negativity constraints: None of the input data or the calculated values must assume negative values.

The model uses a step width of one second moving the simulated time denoted t^{sim} forward. Once the simulated time has moved forward by one second, the water offtake denoted $\dot{V}_{WFI,\,off}$ from the WFI distribution system for every time frame t^{sim} can finally be calculated:

$$\dot{V}_{WFI,off} = \sum_{i=1}^{n} \dot{V}_{WFI,\,off,i} \qquad \forall t^{sim} = 1,2,...,86400 \tag{3.3}$$

with t^{sim} = 86,400 representing a 24-hour period expressed in seconds.

The VBA program of the deterministic model of the DI/WFI system is a subset of the stochastic model and its VBA program listing is included in Appendix 2. The program performs a deterministic simulation if the Check Box named "Check to exclude time & offtake uncertainty" on the simulation model spreadsheet is ticked (see Figure 3.5).

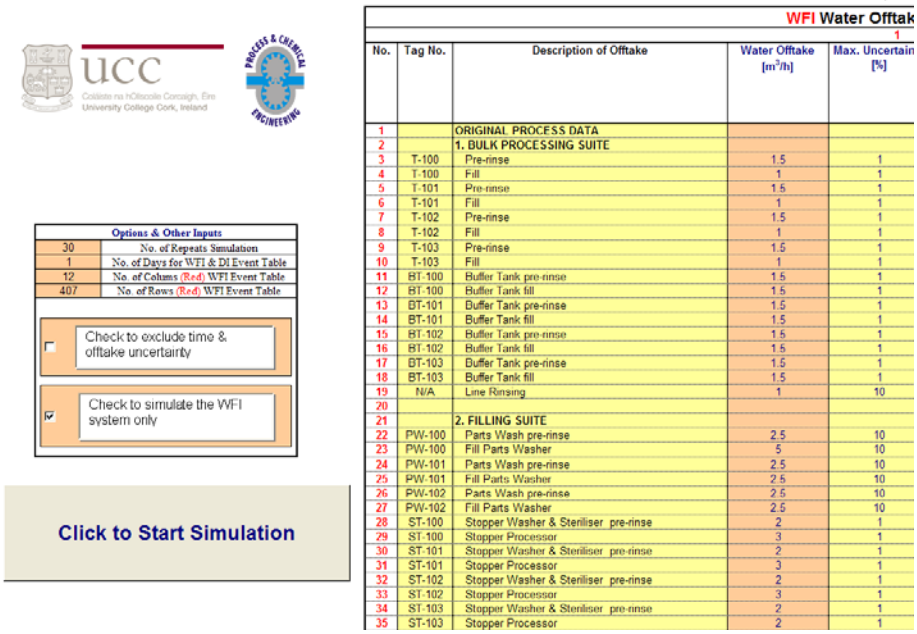


Figure 3.5 Screenshot of the user interface for the deterministic and stochastic simulation as presented within Excel (excerpt)

3.1.1 Summary of Required Input Variables for the Deterministic Model

The Table 3.1 below summarises the required input variables for the WFI distribution model simulation.

Parameter	Description
$\dot{V}_{WFI,off,i}$	Volumetric flowrate for each tap_i
$t_{i,k}^{open}$	Opening time of each $act_{i,k}$
$t_{i,k}^{close}$	Closing time of each $act_{i,k}$

Table 3.1 Input variables required for the deterministic WFI distribution model

3.1.2 WFI Storage Tank and Generating System

Once the water demand $\dot{V}_{WFI,off}$ from the WFI distribution system for all timeframes t^{sim} has been computed, the variation of the water level in the storage tank denoted $V_{WFI,L}$ over a 24-hour period can be computed from a mass-balance (see Figure 3.6) of the WFI storage tank as follows:

$$\frac{dV_{WFI,L}}{dt} = V_{WFI,L} + (\dot{V}_{WFI,Gen} - \dot{V}_{WFI,Off}) \cdot dt \tag{3.4}$$

with the initial condition of the WFI storage tank volume at $t = t_0$ as:

$$V_{WFI,L} = V_{WFI,L,St} \tag{3.5}$$

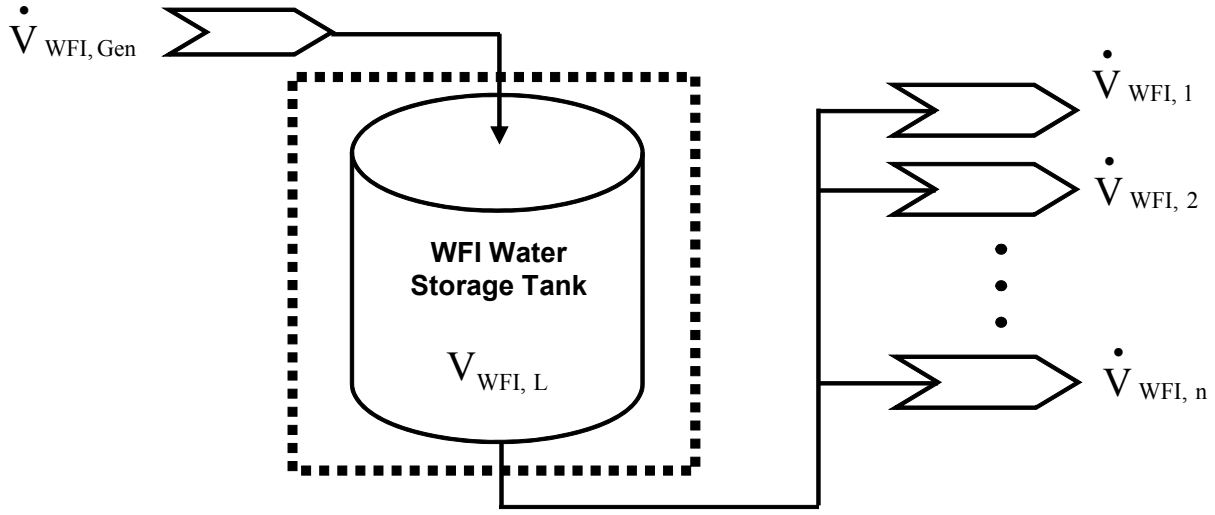


Figure 3.6 Mass balance of the WFI-storage tank

The following assumptions are made to arrive at equation (3.4):

1. Water is incompressible; therefore the water removal from the distribution system equals the water removal from the storage tank.
2. The water temperature of the WFI system is set at 80 ºC and controlled within a temperature band of ±5.0 ºC in order to maintain a low bacterial count [12, 14]. Consequently, the water density along the WFI distribution system is assumed to be constant.

The differential equation (3.4) is integrated by the Euler method, which according to Luyben et al. [125] is a suitable method for this problem. The integration step is, in line with the step width of the discrete event simulation, one second.

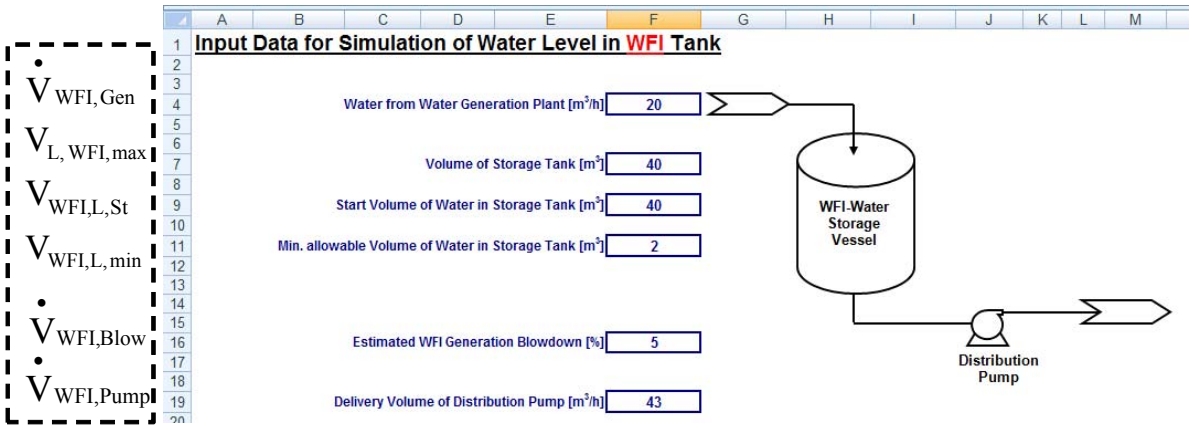


Figure 3.7 User interface WFI showing the storage tank and generation plant data (Excel screenshot) The input variables as denoted in the text are depicted in the dotted box.

Furthermore the water level in the WFI storage tank is bounded between a minimum and maximum volume:

$$V_{WFI,L,min} \leq V_{WFI,L} \leq V_{WFI,L,max} \tag{3.6}$$

With equation 3.6 above, the mathematical description of the WFI storage and distribution system is complete. Missing is the mathematical formulation of the WFI generation plant, which is developed in the following text.

The raw water for the WFI generation plant is provided by the DI loop. A mass-balance on the WFI generation plant (see Figure 3.8) provides the amount of DI water required as:

$$\dot{V}_{DI,WFI} = \dot{V}_{WFI,Gen} + \dot{V}_{WFI,Blow} \tag{3.7}$$

$\dot{V}_{WFI,Blow}$ in the above equation denotes the amount of water from the WFI generation plant going to blowdown; here assumed to go to drain. The blowdown of a WFI plant is usually between 5-8% of the raw (DI) water [12]. The modeller can set the blowdown to an appropriate level via the "WFI Storage Tank and Generation Plant" user interface as depicted in Figure 3.7. Furthermore this user interface accepts the storage tank inputs.

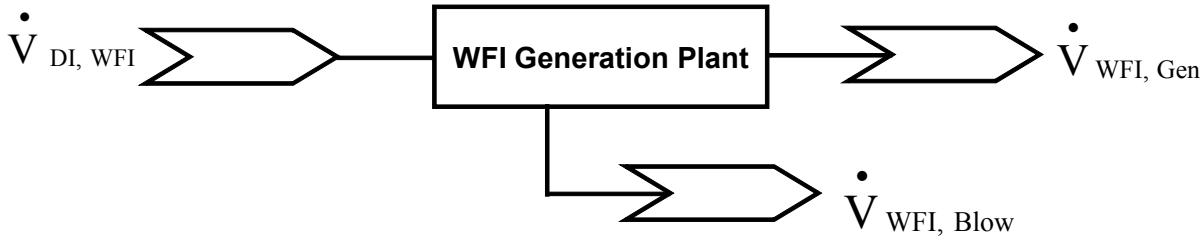


Figure 3.8 Mass balance of the WFI-generation

The flow of water $\dot{V}_{WFI,Gen}$ from the WFI generation plant to the WFI storage tank is controlled to maintain the level in the WFI storage tank at its maximum allowable level $V_{WFI,L,max}$. The control is assumed to be perfect; any delay, overshooting or dampening effects are ignored. The water flow from the WFI generation plant replenishes the level in the WFI storage tank in any timeframe t^{sim} depending on the offtake from the WFI loop according to four rules:

1. Should the WFI offtake from the distribution loop $\dot{V}_{WFI,off}$ be less than the capacity of the WFI generation plant $\dot{V}_{WFI,Gen}$ and the water level $V_{WFI,L}$ in the WFI storage tank is as its maximum, then the level in storage tank will be maintained by the water flow from the WFI generation station:

$$\dot{V}_{WFI,Gen} = \dot{V}_{WFI,off} \quad \text{if} \quad V_{WFI,L} = V_{WFI,L,max} \quad \forall t^{sim} = 1, 2, ..., 86400 \qquad (3.8)$$

2. Should the water level $V_{WFI,L}$ in the WFI storage tank drop below the maximum allowable value, then the flowrate from the WFI generation plant shall increase to its maximum value to replenish the water in the storage tank to its maximum level as soon as possible:

$$\dot{V}_{WFI,Gen} = \dot{V}_{WFI,Gen,max.} \quad \text{if } V_{WFI,L} < V_{WFI,L,max} \quad \forall t^{sim} = 1, 2, ..., 86400 \qquad (3.9)$$

3. Should the water level $V_{WFI,L}$ in the WFI storage tank exceed the maximum allowable value, then the flowrate from the WFI generation system shall be zero and the level in the storage tank shall be the maximum allowable level:

$$\dot{V}_{WFI,Gen} = 0 \quad \text{if} \quad V_{WFI,L} = V_{WFI,L,max} \quad \forall t^{sim} = 1,2,...,86400 \qquad (3.10)$$

4. Should the water level $V_{WFI,L}$ in the WFI storage tank drop below the minimum allowable level $V_{WFI,L,min}$ the simulation terminates as an impossible situation arose.

It is assumed that the production process requires WFI at 20°C and that the volumetric measurement of the amount of WFI dispensed is also made at 20°C. The WFI is, however, stored and distributed at 80 ºC to maintain a low bacterial count [12], resulting in a temperature difference between the storage and distribution system and the raw and dispensed water as shown in Figure 3.9. This temperature difference requires a density compensation of the amount of water in the WFI water storage tank in order to obtain a temperature compensated WFI water volume denoted $V_{WFI,L,C}$ as follows:

$$V_{WFI,L,C} = V_{WFI,L} \frac{\rho_{20}}{\rho_{80}} \tag{3.11}$$

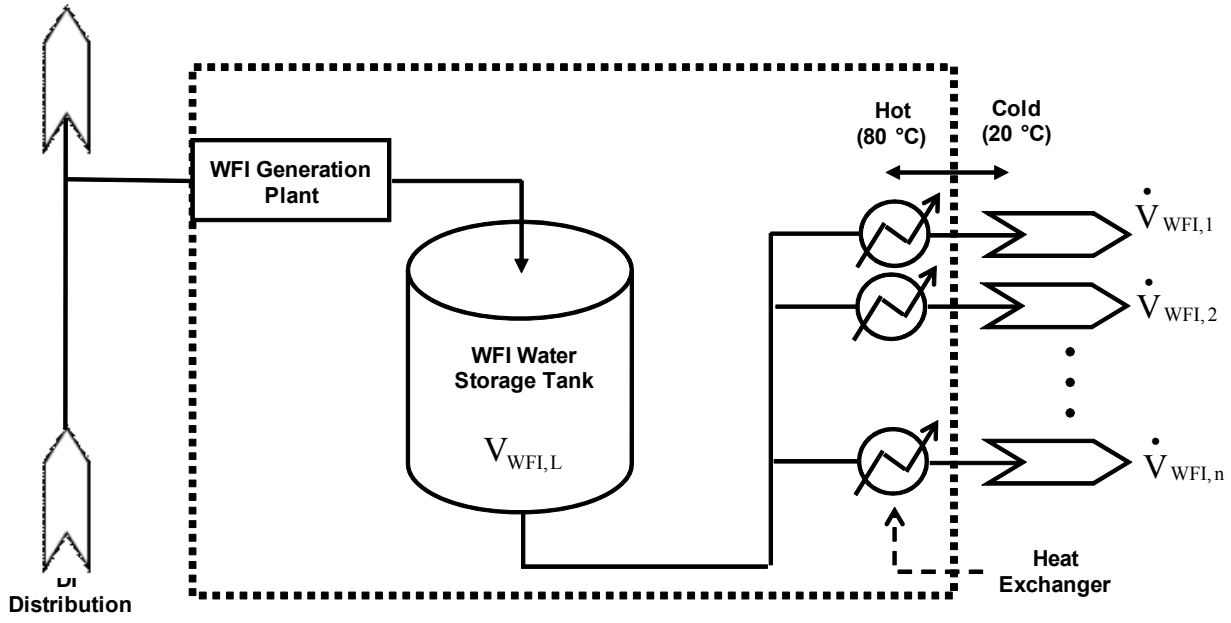


Figure 3.9 Temperature profile in the WFI system as modelled

3.1.3 DI Storage Tank and Generating System

The mathematical model of the DI water system is similar to the WFI system. Therefore this section only describes the parts of the DI model, which are different to the WFI model. The mass-balance of the DI tank (see Figure 3.10), which has to include the feed water flow $\dot{V}_{DI,WFI}$ (see equation 3.7) to the WFI generation plant as follows:

$$\frac{dV_{DI,L}}{dt} = V_{DI,L} + (\dot{V}_{DI,Gen} - \dot{V}_{DI,off} - \dot{V}_{DI,WFI}) \cdot dt \tag{3.12}$$

with the initial condition:

$$V_{DI,L} = V_{DI,L,St} \tag{3.13}$$

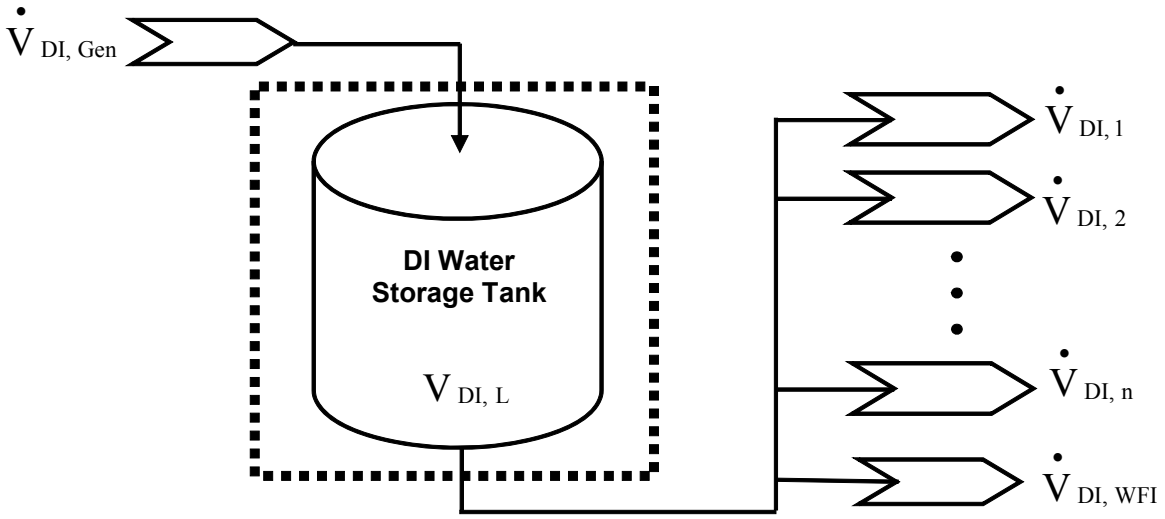


Figure 3.10 Mass balance of the DI-storage tank

The DI water is assumed to be generated, stored and distributed at a constant 20 ºC. Therefore temperature compensation as per equation 3.11 is not required. The DI water generation system is also different to the WFI generation plant, having negligible blowdown as shown in Figure 3.11; hence:

$$\dot{V}_{DI,Gen} = \dot{V}_{RW} \tag{3.14}$$

Figure 3.11 Mass balance of the DI-generation plant

3.1.4 Summary of Required Input Variables for Storage Tank Model

The Table 3.2 below summarises the required input variables for the WFI storage tank simulation. The input variables for the DI simulation are similar.

Parameter	Description
$\dot{V}_{WFI,Gen}$	WFI from generation plant
$V_{WFI,L,max}$	Maximum volume of WFI in storage tank
$V_{WFI,L,St}$	Start volume in WFI storage tank at t_0
$V_{WFI,L,min}$	Minimum allowable volume in WFI storage tank
$\dot{V}_{WFI,Blow}$	WFI generation blowdown (not required for DI storage tank)
$\dot{V}_{WFI,Pump}$	WFI water distribution pump
ρ_{20}	Water density at 20 °C (not required for DI storage tank)
ρ_{80}	Water density at 80 °C (not required for DI storage tank)

Table 3.2 Input variables required for the WFI storage tank model

3.2 Stochastic Model of WFI Demand

The previous section described the deterministic model of a WFI system. This section extends the deterministic simulation to include dispensed volume and schedule uncertainty modelled with stochastic distributions. Stochastic methods are the classical approach to uncertainty simulation [25] and are subject to a number of textbooks on discrete-event simulation such as Banks [2, 41], Law and Kelton [1], Fishman [3] to list a few. As the deterministic model forms the basis for the stochastic simulation, this section only describes the additional assumptions and mathematical expressions required to extend the deterministic model to a stochastic model of a WFI system.

All data associated with a WFI system are subject various levels of uncertainty [26, 124]. Two different uncertainties were chosen for this dissertation to be modelled; (a) dispensed volume and (b) schedule uncertainty:

a. Dispensed volume uncertainty is the uncertainty associated with the amount of water dispensed from the distribution system for each $\mathrm{act}_{i,k}$. Volume uncertainty is a function of the measurement system and the type of operation. For automatic operation, i.e. filling of a vessel, an accuracy of ± 1% of full volume is typically achieved. Dispensed volume uncertainty of manual cleaning operations is, on the other hand, larger [21]. This work assumes the accuracy of the WFI consumption of the manual operations i.e. manual cleaning is assumed to be ± 10% of full volume.
b. In a real production environment the opening and closing times of each $\mathrm{act}_{i,k}$ are uncertain [26, 124]. Schedule uncertainty is a function of the production schedule, which may experience uncertainties caused by for instance variations in operator performance or unavailability of operators for manual interventions.

In the following subsections, the mathematical formulation of flowrate uncertainty is developed first, followed by the development of the stochastic schedule uncertainty.

3.2.1 Stochastic Model of Dispensed WFI Volume

The amount of WFI dispensed is subject to uncertainty caused by for instance uncertainty in the flowrate measurement [28] or by manual filling or cleaning regimes [21]. This uncertainty varies the volume of the actually dispensed WFI for each $act_{i,k}$ denoted $\dot{V}^{actual}_{WFI,off,i,k}$.

P_i Y_i $\Delta t^{uncertain}_{i,k}$ $Z_{i,k}$

WFI Water Offtake Input Data

1

No.	Tag No.	Description of Offtake	Water Offtake [m³/h]	Max. Uncertainty [%]	Description of Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty
1		ORIGINAL PROCESS DATA							
2		1. BULK PROCESSING SUITE							
3	T-100	Pre-rinse	1.5	1	u	2:10	2:15	0:30	u
4	T-100	Fill	1	1	u	2:00	2:06	0:20	u
5	T-101	Pre-rinse	1.5	1	u	3:10	3:17	0:30	u
6	T-101	Fill	1	1	u	3:15	3:25	0:30	u
7	T-102	Pre-rinse	1.5	1	u	1:20	1:30	0:20	u
8	T-102	Fill	1	1	u	3:15	3:25	0:20	u
9	T-103	Pre-rinse	1.5	1	u	1:00	1:10	0:20	u
10	T-103	Fill	1	1	u	6:00	6:30	0:20	u
11	BT-100	Buffer Tank pre-rinse	1.5	1	u	1:15	1:20	0:20	u

Figure 3.12 Event table WFI for the stochastic simulation of a WFI system as presented within Excel (excerpt)
The variables shown above the event table indicate the user input columns of the additional variables required for a stochastic simulation.

Let Y_i (see Figure 3.12 and 3.13 for nomenclature) denote a stochastic variable describing the stochastic distribution of the dispensed uncertain WFI volume denoted $\Delta\dot{V}^{uncertain}_{WFI,off,i,k}$ for each $act_{i,k}$. Let P_i denote the maximum uncertainty in percent of full volume. Furthermore let $U_{i,k}$ be a stochastic variable representing the chances of $\Delta\dot{V}^{uncertain}_{WFI,off,i,k}$ being negative or non-negative, as it is possible that the flowrate $\dot{V}_{WFI,off,i}$ of each $act_{i,k}$ is more or less than nominal. Then, the flowrate uncertainty $\Delta\dot{V}^{uncertain}_{WFI,off,i,k}$ for each $act_{i,k}$ is calculated as:

$$\Delta \dot{V}^{uncertain}_{WFI,off,i,k} = \dot{V}_{WFI,off,i} \cdot Y_i \cdot U_{i,k} \cdot P_i / 100 \quad \forall i = 1,2,..., n \quad \forall k = 1,2,.., m \qquad (3.15)$$

Note that $\dot{V}_{WFI,off,i}$, Y_i and P_i are independent of *k* (see volume constraint, Section 3.1).

The actual flowrate $\dot{V}^{actual}_{WFI,off,i,k}$, that is the flowrate including flowrate uncertainty, for each $act_{i,k}$ is computed as:

$$\dot{V}^{actual}_{WFI,off,i,k} = \dot{V}_{WFI,off,i} + \Delta \dot{V}^{uncertain}_{WFI,off,i,k} \qquad \forall i = 1,2,..., n \quad \forall k = 1,2,.., m \qquad (3.16)$$

A number of simplifications are made to reduce the dimensions of Y_i and $U_{i,k}$:

1. Each Y_i could be described by many statistical distributions [45, 46]. The model as included in Appendix 2 uses three statistical distributions namely the Normal, the Beta and the uniform distribution (see Appendix 2). In addition, the parameters of the three distributions are, if applicable, constant throughout the simulation, further reducing the computational burden.

   It is important to allow the user to choose between different stochastic distributions, as for instance measurement uncertainty may not always be best described by the Normal distribution [28]. The Normal distribution is, however, the most widely used distribution in statistics [126] and is therefore included in the model. The uniform distribution is often used if no information on the shape of the distribution is available [126]. The beta distribution is included in the model as it can easily adopt different shapes. It is hoped that these three distributions will provide enough flexibility for the industrial practice. Of course, with additional programming effort, the stochastic flowrate model can be changed to accept other stochastic distributions or distributions with varying parameters.

2. Each $U_{i,k}$ is reduced to a binary variable, which can be either negative or non-negative depending on the result of the uniform probability distribution denoted $U(X)$ as follows:

$$U_{i,k} = \begin{cases} -1, & \text{if} \quad U(X) \le 0.5 \\ 1, & \text{if} \quad U(X) > 0.5 \end{cases} \tag{3.17}$$

with X denoting a random variable.

Therefore for the stochastic flowrate simulation the user has to provide P_i, Y_i and its parameters for each $act_{i,k}$ as shown in Figure 3.13.

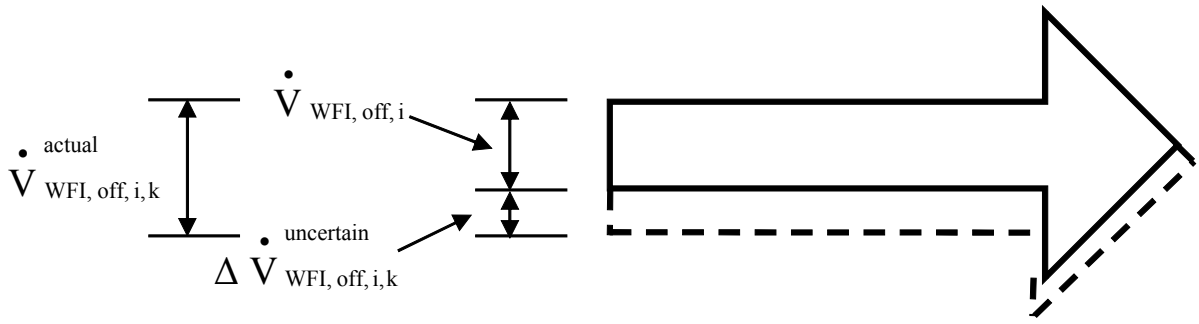


Figure 3.13 Nomenclature for the stochastic flowrate uncertainty of $act_{i,k}$
The solid lines indicate the crisp flowrate as given in the event table, whereas the dotted lines indicate the variation of the crisp flowrate by stochastic uncertainties.

3.2.2 Stochastic Model of Schedule or Tap Opening/Closing Times

Further to volume offtake uncertainty, the uncertainty of the opening and closing times of each $\mathrm{act}_{i,k}$ is modelled via stochastic distributions, which is the subject of this section. This uncertainty varies the *opening* $t_{i,k}^{open}$ and *closing times* $t_{i,k}^{close}$ for each $\mathrm{act}_{i,k}$ of the crisp schedule.

The precedence constraints presented above must also be observed by the stochastic model if the schedule is subject to stochastic uncertainties. These uncertainties in time will shift the *opening* $t_{i,k}^{open}$ and *closing times* $t_{i,k}^{close}$ backwards or forwards in time, creating new *opening* $t_{i,k,new}^{open}$ and *closing times* $t_{i,k,new}^{close}$, which must not clash with events in an earlier or later timeframe or mathematically:

$$t_{i,k,new}^{open} \prec t_{i,k,new}^{close} \qquad \forall i = 1,2,...,n; \quad \forall k = 1,2,...,m \tag{3.18}$$

To avoid a possible violation of the precedence constrain, the VBA program of the DI/WFI system checks the event table for any possible violations of the precedence constraints taken the maximum possible time variation into account before proceeding with the simulation. Should a possible violation be found, the user is notified of the location of the possible violation on the event table and the program terminates.

Let $Z_{i,k}$ (see Figure 3.14 for nomenclature) denote a stochastic variable expressing the stochastic distribution of the uncertain opening time denoted $\Delta t_{i,k}^{uncertain}$ for each $\mathrm{act}_{i,k}$. Furthermore let $U_{i,k}$ be a stochastic variable representing the chances of $\Delta t_{i,k}^{uncertain}$ be negative or non-negative, as it is possible for the new opening times denoted $t_{i,k,new}^{open}$ of each $\mathrm{act}_{i,k}$ to

be earlier or later than originally scheduled. Hence the new opening time $t^{open}_{i,k,new}$ is calculated as:

$$t^{open}_{i,k,new} = t^{open}_{i,k} + \Delta t^{uncertain}_{i,k} \cdot Z_{i,k} \cdot U_{i,k} \qquad \forall i = 1,2,...,n \quad \forall k = 1,2,..,m \qquad (3.19)$$

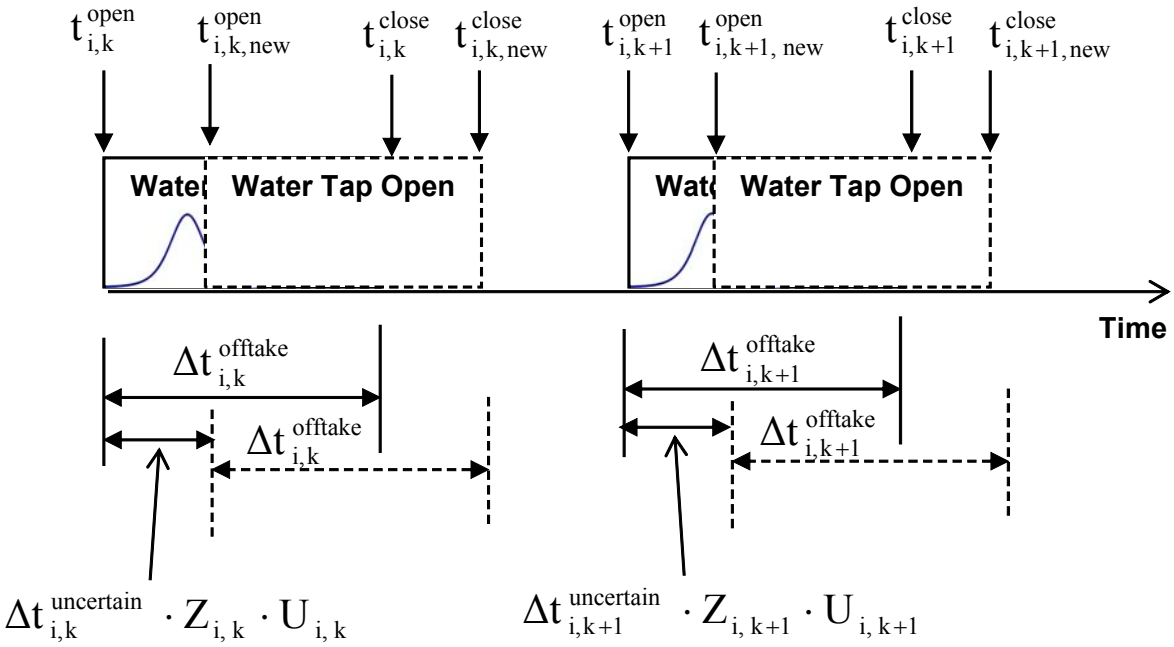


Figure 3.14 Nomenclature for the stochastic schedule uncertainty
The solid lines indicate the crisp schedule as given in the event table, whereas the dotted lines indicate the variation of the crisp schedule by stochastic distribution.

Similar to the stochastic volume flowrate model described above, a number of simplifications are made to reduce the dimensions of $Z_{i,k}$ and $U_{i,k}$:

1. Each $Z_{i,k}$ could be described by many statistical distributions [45, 46]. The same comments made for Y_i apply to $Z_{i,k}$.

2. Each $U_{i,k}$ is reduced to a binary variable, which can be either negative or non-negative depending on the result of the uniform probability distribution $U(X)$ as follows:

$$U_{i,k} = \begin{cases} -1, & \text{if} \quad U(X) \leq 0.5 \\ 1, & \text{if} \quad U(X) > 0.5 \end{cases} \tag{3.20}$$

with X denoting a random variable.

Further, it is assumed (see Section 3.1), that the tap opening intervals $\Delta t_{i,k}^{offtake}$ are constant and that only the opening $t_{i,k}^{open}$ and, with it, the closing times $t_{i,k}^{close}$ for each $act_{i,k}$ will shift forwards and backwards in time. Hence, the new closing time $t_{i,k,new}^{close}$ for each $act_{i,k}$ can be determined by:

$$t_{i,k,new}^{close} = t_{i,k,new}^{open} + \Delta t_{i,k}^{offtake} \qquad \forall i = 1,2,...,n \qquad \forall k = 1,2,..,m \tag{3.21}$$

3.2.3 Summary of Required Input Variables for the Stochastic Model

The Table 3.3 below summarises the required input variables for the stochastic simulation excluding the variables needed for the storage tank simulation.

Parameter	Description
Deterministic Model	
$\dot{V}_{WFI,off,i}$	Volumetric offtake for each act_i
$t_{i,k}^{open}$	Scheduled opening time for each $act_{i,k}$
$t_{i,k}^{close}$	Scheduled closing time for each $act_{i,k}$
Model including Volume Uncertainty	
P_i	Uncertainty in percent of full volume
Y_i	Stochastic description of uncertainty. Here uniform, Normal or Beta distribution (see below).
Model including Schedule Uncertainty	
$\Delta t_{i,k}^{uncertain}$	
$Z_{i,k}$	Stochastic description of uncertainty. Here uniform, Normal or Beta distribution (see below).
Statistical Distributions	
Normal Distribution	Mean μ and Standard deviation σ
Beta Distribution	Shape parameters α, β and lower bound a and upper bound b

Table 3.3 Input variables required for the stochastic WFI distribution model

3.2.4 Water Storage Tanks & Generating Systems – Mathematical Model

The mathematical model of the DI/WFI storage tanks and generation plants is the same as the deterministic model described in Section 3.1.2.

3.3 Fuzzy Logic Model of WFI Demand

The stochastic model simulates schedule uncertainty based on observed or estimated time variations of the opening times of the WFI valves. The Fuzzy Logic (FL) model, on the other hand, simulates schedule uncertainty by modelling operator behaviour. In practice, despite the advances of automation, operators interaction with the process is common in modern pharmaceutical plants. Therefore the schedule may be affected by human fatigue, motivation, availability and other influences, which are simulated by the FL model proposed here. As a result, the FL model describes schedule uncertainty fundamentally different to the stochastic model, as will be further evident from the development of the mathematical formulation of the FL model in this section.

For two reasons the FL model does not model volume uncertainty. First, schedule uncertainty plays a more important role than volume uncertainty, as the uncertainty of the latter is by comparison small. Secondly it is difficult to construct a reasonable model to model volume uncertainty with FL. Therefore dispensed DI or WFI volume uncertainties are not included in the FL model.

Similar to the previous section on the stochastic formulation, this section only describes the additional assumptions and mathematical expressions required to formulate the FL model of a DI/WFI system; again taking the deterministic model as its basis. Once more the models for the WFI and DI systems are similar (see Figure 3.1), requiring that only the formulation of the WFI system needs to be presented.

3.3.1 Fuzzy Logic Model of Tap Opening/Closing Times

FL is a logic for *approximate reasoning* [52, 61, 127, 128]; in other words it is a logic capable to deal with relatively large uncertainties. Specifically, estimates of the uncertainties are often given verbally by experts [26, 50, 53, 82, 128]. Consequently, this work uses "expert" knowledge to model schedule uncertainty of the offtake from a DI/WFI system. The overall objective of the fuzzy simulation model is to describe schedule uncertainty caused by operator behaviour based on a combination of personnel experience, expert advice and literature sources. The influence of the operators on WFI demand is modelled over a 24-hour period, three shift operation, which leads to a fundamentally different model for the demand uncertainty formulation from a WFI system compared to the stochastic model.

There are three principle causes of a particular valve opening event $\mathrm{act}_{i,k}$ to be delayed and not to occur at the scheduled time as depicted in Figure 3.15:

1. The delay denoted $t_{i,k}^{\mathrm{Delay,\,No\,Op.}}$ caused by "No Operator Available" as the number of operators denoted n_{op} is limited.
2. The delay denoted $t_{i,k}^{\mathrm{Delay,1}} + t_{i,k}^{\mathrm{Delay,2}}$ as a result of "Operator Delay" due to, for instance, tiredness of the operator or the operator being on a break when a task is about to start. This delay is a function of the fuzzy number "Operator Setup Time" (see event table Figure 3.17), the various Membership Functions and the Rule Base.
3. The delay denoted $t_{i,k}^{\mathrm{min,\,D}}$ due to "Minimum Duration of Task", as operators will require a minimum amount of time for a task, for example, connecting a hose to a vessel. This delay is a function of the fuzzy number "Operator Setup Time" (see event table Figure 3.17), the various Membership Functions and the Rule Base.

The details of the various delays are described in this section. In summary the FL model describes a plant in which the operator will have to stay with the WFI operation, specifically till the WFI valve is closed again. Only when a task is finished, the operator becomes available for other tasks.

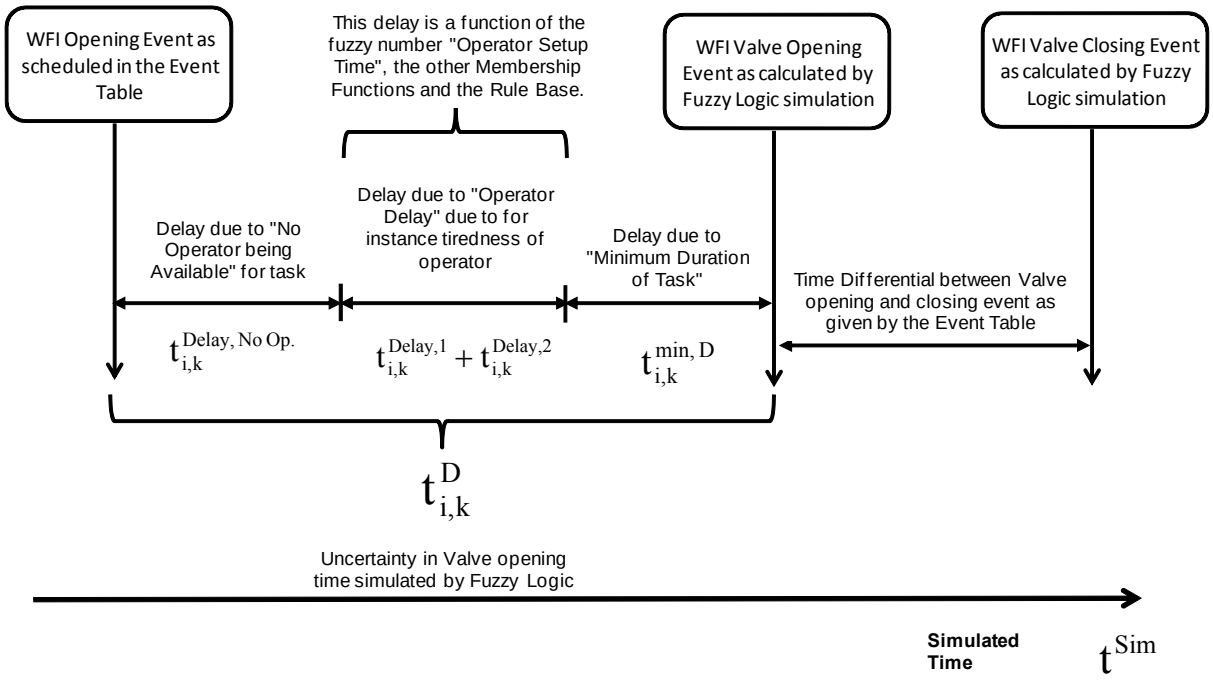


Figure 3.15 Principle of the fuzzy logic calculation of valve opening time

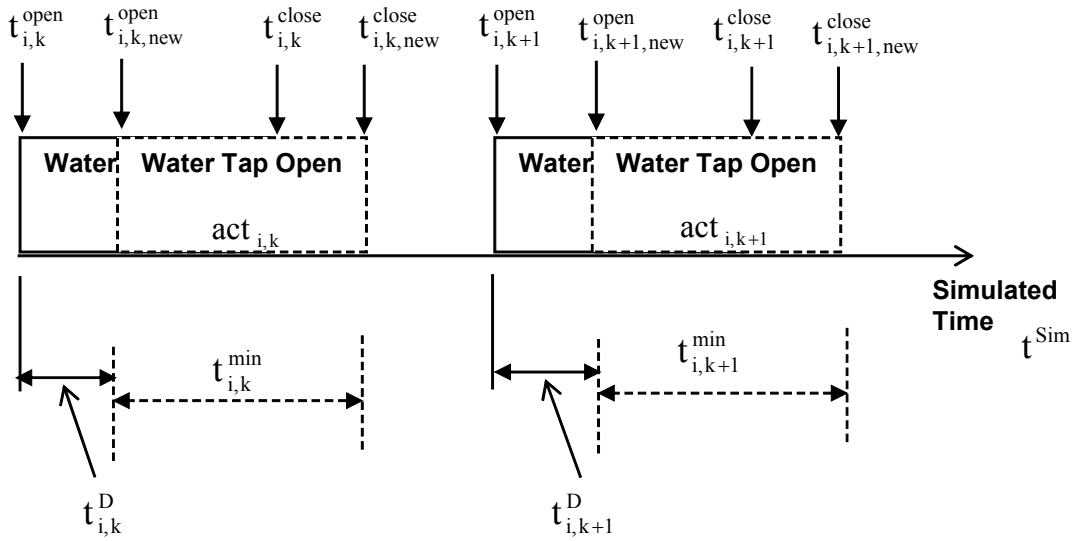
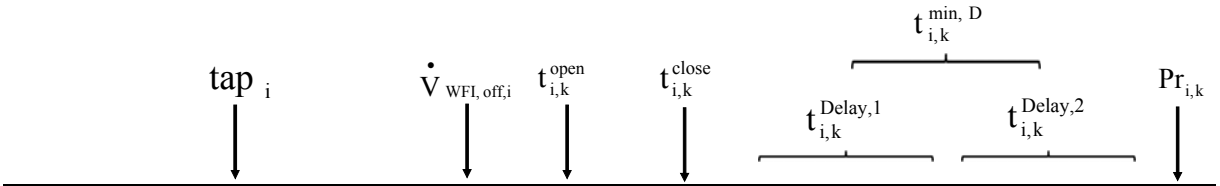


Figure 3.16 Nomenclature for the fuzzy logic model of a WFI system



WFI Water Offtake Input Data

No.	Tag No.	Description of Offtake	Water Offtake [m³/hr]	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority
1		ORIGINAL PROCESS DATA								
2		1. BULK PROCESSING SUITE								
3	T-100	Pre-rinse	1.5	2:00	2:05	0:00	0:00	0:00	0:00	2
4	T-100	Fill	1	2:00	2:06	0:00	0:02	0:08	0:10	3
5	T-101	Pre-rinse	1.5	3:10	3:17	0:00	0:00	0:00	0:00	2
6	T-101	Fill	1	3:15	3:25	0:00	0:02	0:08	0:10	3
7	T-102	Pre-rinse	1.5	1:20	1:30	0:00	0:00	0:00	0:00	2
8	T-102	Fill	1	3:15	3:25	0:00	0:02	0:08	0:10	3
9	T-103	Pre-rinse	1.5	1:00	1:10	0:00	0:00	0:00	0:00	2
10	T-103	Fill	1	6:00	6:30	0:00	0:02	0:08	0:10	3
11	BT-100	Buffer Tank pre-rinse	1.5	1:15	1:20	0:00	0:00	0:00	0:00	2
12	BT-100	Buffer Tank fill	1.5	2:00	2:10	0:00	0:02	0:08	0:10	2

Figure 3.17 Event table (Excel screenshot) of a fuzzy simulation of a WFI system (excerpt) The variables shown above in the event table indicate the user input columns of these variables.

For each $act_{i,k}$ the new opening time denoted $t^{open}_{i,k,new}$ is calculated by adding a time delay denoted $t^{D}_{i,k}$ (see Figure 3.16 for a graphical depiction of the nomenclature and Figure 3.17 for the required model input variables) to the originally scheduled opening time $t^{open}_{i,k}$:

$$t^{open}_{i,k,new} = t^{open}_{i,k} + t^{D}_{i,k} \qquad \forall i = 1,2,...,n \qquad \forall k = 1,2,..,m \tag{3.22}$$

Note: In the following text the term "$\forall i = 1,2,...,n \qquad \forall k = 1,2,..,m$" is generally omitted.

The new closing time denoted $t^{close}_{i,k,new}$ of $act_{i,k}$ is computed as:

$$t^{close}_{i,k,new} = t^{open}_{i,k,new} + (t^{close}_{i,k} - t^{open}_{i,k}) \tag{3.23}$$

The opening interval $(t_{i,k}^{close} - t_{i,k}^{open})$ for each $act_{i,k}$ is constant because volume uncertainty is not modelled. The time delay $t_{i,k}^{D}$ (see Figure 3.15 & 3.16) on the other hand, is not constant and is modelled as a function of the interaction of ten different influences:

1. Number of Operators
2. Work Instructions – Sequence, Priority Operations
3. Membership Function "Duration of Tasks"
4. Membership Function "Operator Shift Pattern"
5. Membership Function "Operator Tiredness"
6. Membership Function "Operator Break Pattern"
7. Membership Function "Output"
8. Rule Base
9. Aggregation & Ranking
10. Defuzzification

In the first ten of the next eleven subsections each of the ten different influences listed above are explained. Finally, in the eleventh subsection the overall time delay $t_{i,k}^{D}$ (see Figure 3.15 & 3.16) is determined.

3.3.1.1 Number of Operators

The FL model includes the number of operators denoted n_{op} as an input variable (see Figure 3.18). Clearly, should the number of operators be too low, an $act_{i,k}$ may have to be delayed, as was shown by Azzaro-Pantel et al. [83] with their FL model of a workshop of a semiconductor production facility.

The model presented here assumes that, the number of operators on site over all shifts is constant.

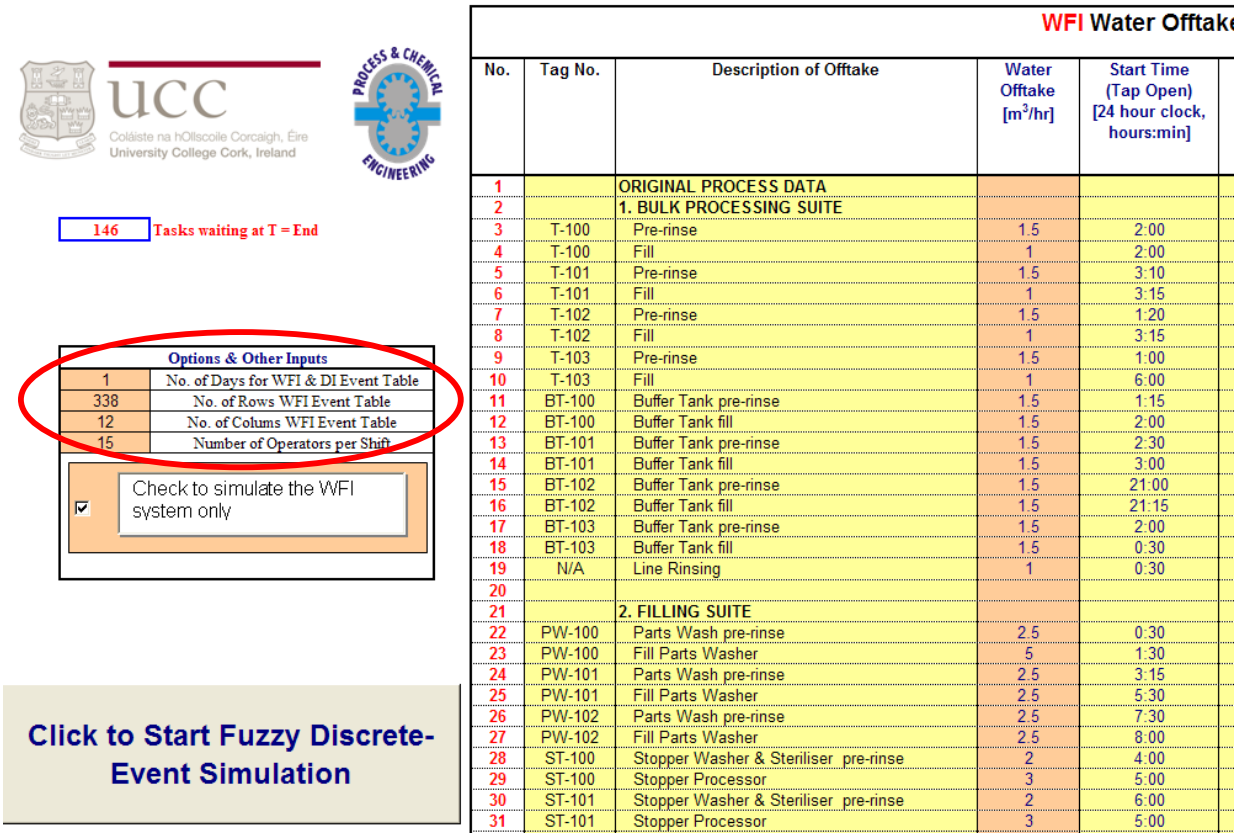


Figure 3.18 Input field for the number of operators n_{op} within Excel

3.3.1.2 Work Instructions – Sequence, Priority Operations

A production plant can only operate in a safe and efficient manner, if work instructions and in particular scheduling instructions are followed [25, 129]. The FL model of the DI/WFI system assumes that tasks are served on a first-come-first-serve or First-In First-Out (FIFO) basis, which was also adapted by Azzaro-Pantel et al. [83] in their fuzzy workshop model. Furthermore, the model assumes the operator can perform one task at a time only and that the operator is instructed to finish a job once begun. For instance, if an operator is occupied with a task, this operator will not be available to perform other tasks, even if those tasks have a higher priority.

Should more than one task be waiting, the one with the highest priority will be served first. If more than one task with the same priority is waiting, the one with the longest waiting time will be served first.

The production manager has assigned priorities denoted $\mathrm{Pr}_{i,k}$ to the individual tasks as follows:

$\mathrm{Pr}_{i,k} = 1$; Delay to next shift allowed.

$\mathrm{Pr}_{i,k} = 2$; Delay to after break allowed.

$\mathrm{Pr}_{i,k} = 3$; No delay allowed – proceed as soon as possible.

The assigned priorities rules $\mathrm{Pr}_{i,k}$ are inputs to the FL model as shown in Figure 3.17. These priority rule work instructions result in the operators not being allowed to look into the future, for example, delaying a low priority task for a higher priority one starting just seconds after the lower priority job.

The upshot of the work instructions, combined with the limiting number n_{op} of operators, is that a low priority $\mathrm{act}_{i,k}$ may have to be delayed for a long time. Should any $\mathrm{act}_{i,k}$ be delayed due to, for instance, unavailability of operators, the new starting time of this operation will be:

$$t_{i,k}^{\text{Delay, No Op.}} = \begin{cases} t^{\text{Sim}}, & \text{if an operator was previously not available,} \\ & \text{but is available now.} \\ 0, & \text{otherwise} \end{cases} \tag{3.24}$$

with $t_{i,k}^{Delay,\,No\,Op.}$ denoting the delay of an operation, because of the limiting number of operators and priority rules and t^{Sim} denoting the current simulated time.

3.3.1.3 Membership Function "Duration of Tasks"

Every $act_{i,k}$ is associated with a trapezoidal membership function $\underset{\sim}{\mu}_{D}^{i,k}(t)$ "Duration of Tasks" which describes how long an operator might be occupied with a task. Trapezoidal or triangular functions are popular in fuzzy logic, because of their simplicity and ease of computation [52, 53]. Moreover, Dubois and Prade [49] state that trapezoidal and triangular membership functions are generally sufficient for fuzzy simulations.

For the purposes of this thesis it is assumed that the membership function "Duration of Tasks" is given by expert advice following the approach of Azzaro-Pantel et al. [83]. In their publication, an expert gave the following information on a task: "It is impossible that this task will take less than 2 min and more than 10 min (values of and the possibility for this task to be carried out within approximately 4 or 6 min is very high)". Information such as this can be mathematically approximated by trapezoidal membership functions of the form:

$$\underset{\sim}{\mu}_{D}^{i,k}(t) = \begin{cases} 0, x \le a \\ (x-a)/(b-a), x \in (a,b) \\ 1, x \in (b,c) \\ (d-x)/(d-c), x \in (c,d) \end{cases} \tag{3.25}$$

A plot of the trapezoidal membership function $\underset{\sim}{\mu}_{D}^{i,k}(t)$ with the values (1, 2, 3, 5) is given in Figure 3.19. The limiting case $t_{i,k}^{min,\,D} = 0$, meaning that $act_{i,k}$ is performed without a minimum duration can be modelled by reducing the trapezoidal function to a triangular one.

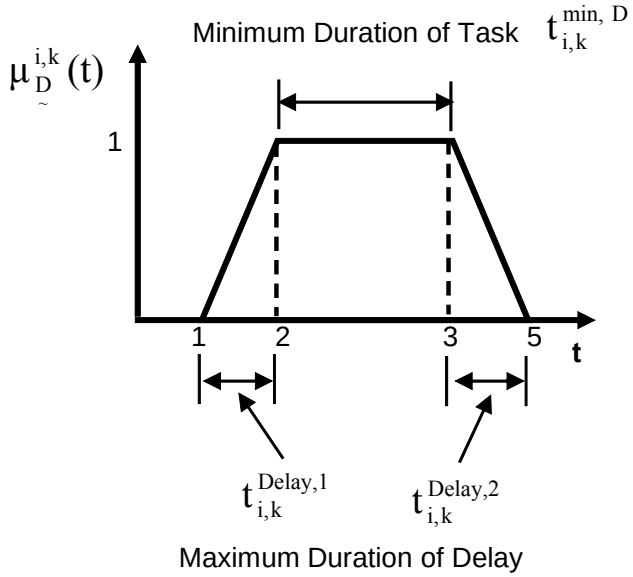


Figure 3.19 Membership function "Duration of Tasks" $\mu_{\tilde{D}}^{i,k}(t)$

The four parameters of the trapezoidal membership function $\mu_{\tilde{D}}^{i,k}(t)$ are input variables of the FL model as shown in Figure 3.17.

3.3.1.4 Membership Function Operator "Shift Pattern"

The membership function $\mu_{\tilde{S}}(t)$ "Shift Pattern" as shown in Figure 3.20 is another trapezoidal function. This membership function approximates the efficiency of operators during the start and end of their respective shifts. The efficiency of operators is assumed to be low at the beginning of a shift, recovering quickly, but tailing off towards the end of a shift as suggested by Jay et al. [130] and Bloom [131].

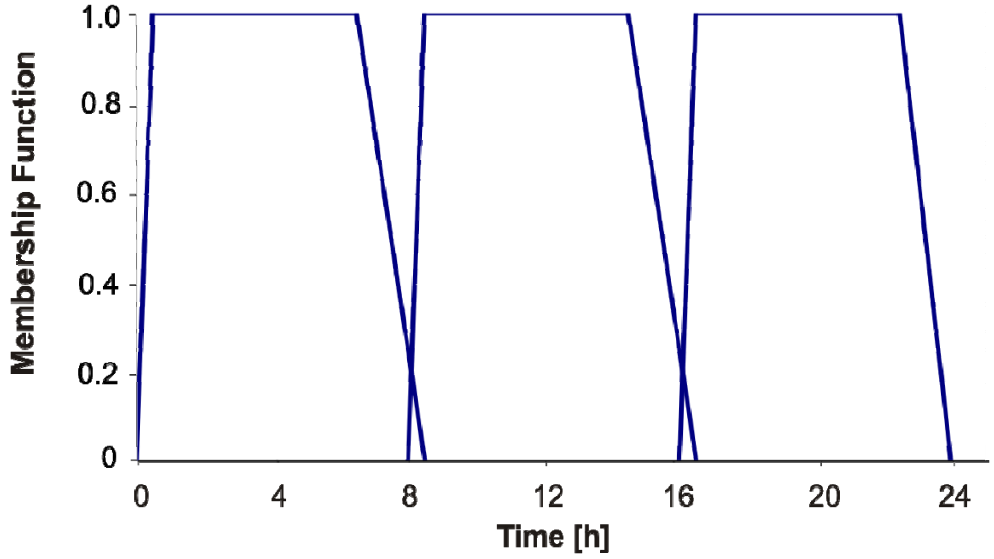


Figure 3.20 Membership function "Shift Pattern" $\mu_{\underset{\sim}{S}}(t)$

3.3.1.5 Membership Function "Operator Tiredness"

An operator works less efficient should tiredness set in. The membership function $\mu_{\underset{\sim}{T}}(t)$ "Operator Tiredness" as shown in Figure 3.21 is based on Johnsson and Fröberg. [132]. These authors found an increase in errors made by operators at around 3:00 and 15:00 hours, which they explained by an increased level of tiredness. At 3:00 in the morning the tiredness level of the operators is modelled to be reaching its peak value, as found by Dahlgren [133], who suggested that operators are more affected by tiredness during night shifts. This increased tiredness during night shifts is modelled by a higher membership function grade of tiredness during the timeframe from 21:00 to 5:00 as depicted in Figure 3.21. The level of tiredness as shown in the figure below is estimated and may not reflect reality very well.

Excluded from the FL model are influences such as sudden illnesses, forgetfulness or oversight of the operator.

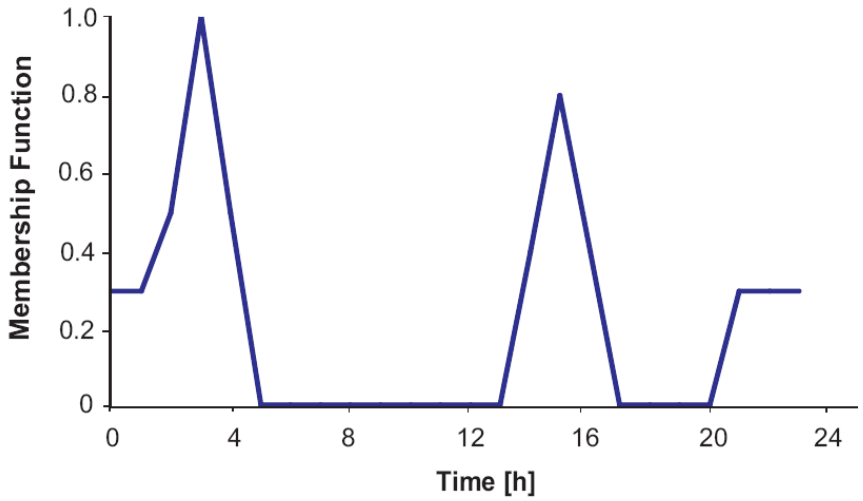


Figure 3.21 Membership function "Operator Tiredness" $\underset{\sim}{\mu}_{T}(t)$

3.3.1.6 Membership Function "Operator Break Pattern"

During an 8 hour shift operation employees are, by European Union law [134], entitled to two breaks. The FL model assumes the first break to be a short break of 10 minutes, while the second one is a longer break of 30 minutes. Furthermore, the model assumes that the operators are allowed some variation in the starting time of their break, as indicated by the trapezoidal break pattern membership function "Operator Break Pattern" denoted $\underset{\sim}{\mu}_B(t)$ displayed in Figure 3.22.

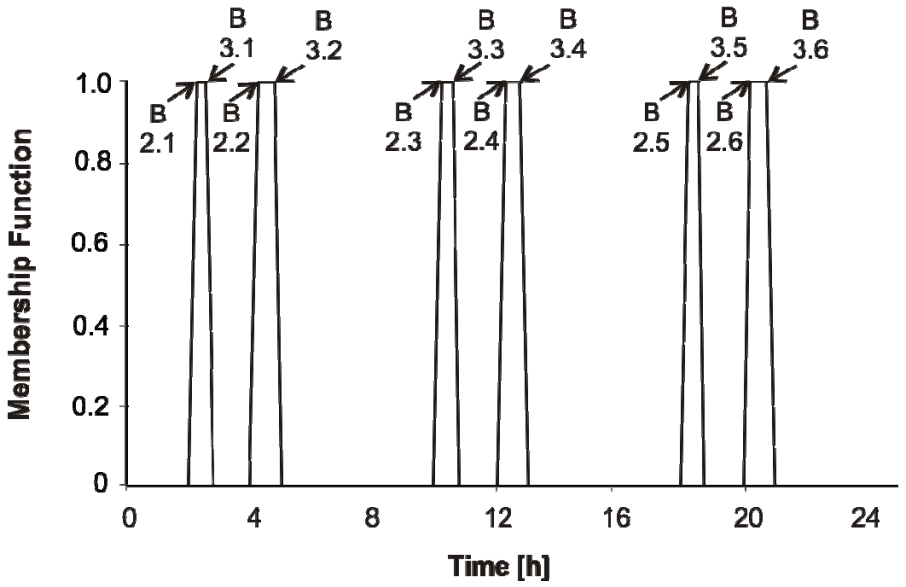


Figure 3.22 Membership function "Operator Break Pattern" $\underset{\sim}{\mu}_B(t)$

The arrows indicate the time of the start $t^B_{2,j}$ and end $t^B_{3,j}$ of the core breaks $\underset{\sim}{\mu}_B(t) = 1$; with j = 1,2,..,6.

During core break times i.e. $\underset{\sim}{\mu}_B(t) = 1$, the model assumes operators will not be available to start any process operations regardless of the priority assigned to the task. Instead, a task scheduled to start during a break will be delayed to a time denoted $t^B_{3,j}$ with j = 1,2, ..., 6, a time just after the core break.

3.3.1.7 Membership Function "Output"

The last membership function required for the FL simulation of a DI/WFI system is the "Output" membership function as depicted in Figure 3.23. This output function delivers the crisp output value of the FL simulation and is composed of eleven triangular membership functions denoted $\underset{\sim}{\mu}_{Out}^{0\%}(r)$, $\underset{\sim}{\mu}_{Out}^{10\%}(r)$ to $\underset{\sim}{\mu}_{Out}^{100\%}(r)$ with *r* denoting the output from the rule base. These *Rule Base Output values r* originate from the rule base, which is described in the following subsection entitled "Rule Base".

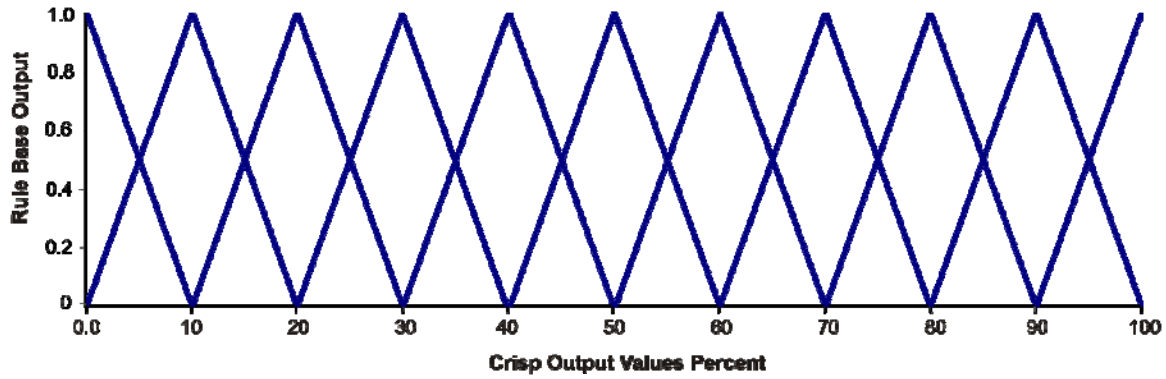


Figure 3.23 Output membership functions $\underset{\sim}{\mu}_{Out}^{0\%}(r)$, $\underset{\sim}{\mu}_{Out}^{10\%}(r)$, …, $\underset{\sim}{\mu}_{Out}^{100\%}(r)$

Note: The modeller can adjust the parameters of the four fuzzy membership functions "Shift Pattern" $\underset{\sim}{\mu}_S(t)$, "Operator Tiredness" $\underset{\sim}{\mu}_T(t)$, "Operator Break Pattern" $\underset{\sim}{\mu}_B(t)$ and "Output" $\underset{\sim}{\mu}^{0\%}_{Out}(r)$, $\underset{\sim}{\mu}^{10\%}_{Out}(r)$ to $\underset{\sim}{\mu}^{100\%}_{Out}(r)$ by accessing the appropriate Excel tab as shown in Figure 3.24.

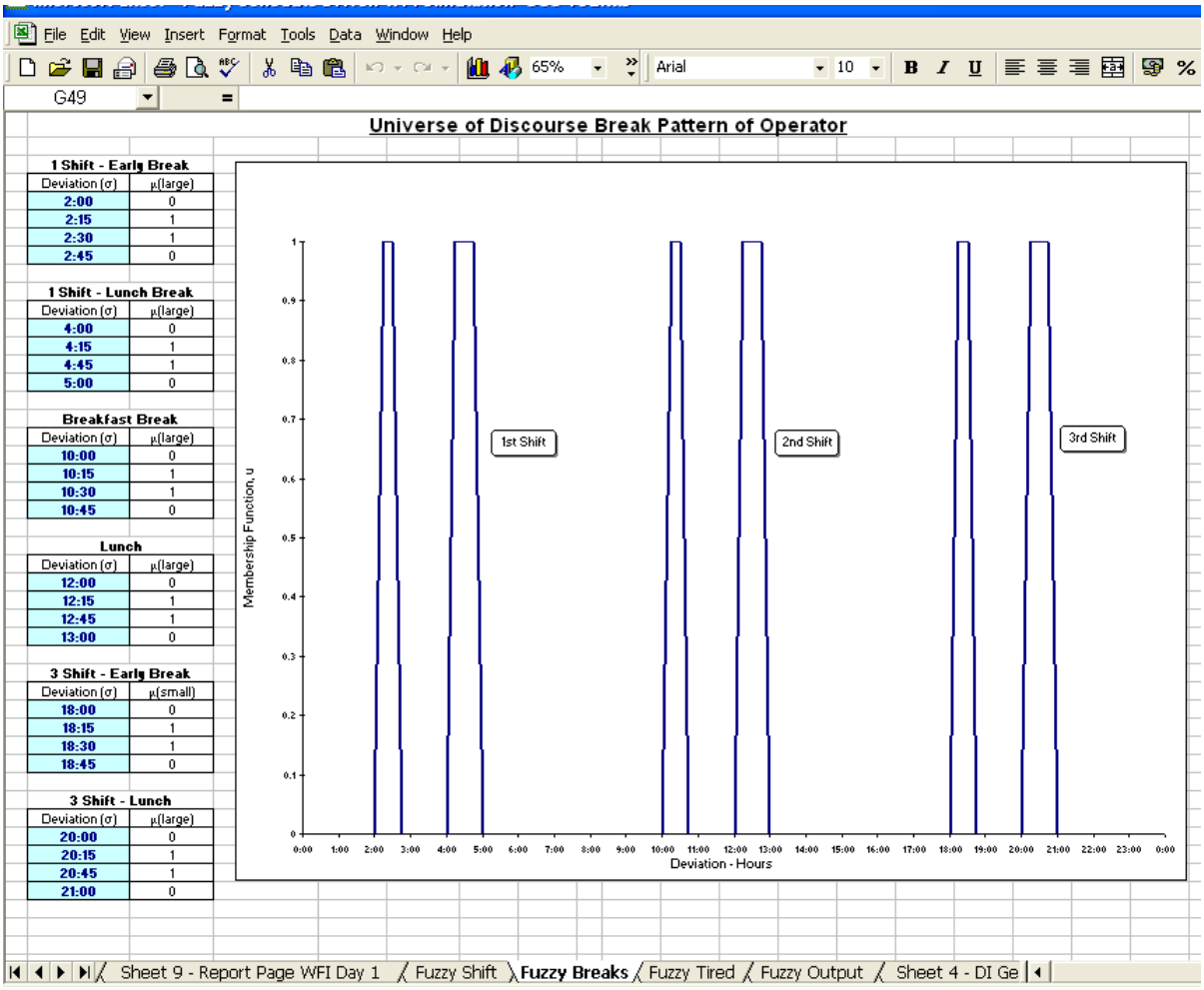


Figure 3.24 Partial screenshot showing the input interface for the membership function "Operator Break Pattern" $\underset{\sim}{\mu}_B(t)$ with Excel tabs "Fuzzy Breaks"

Other Excel tabs named "Fuzzy Shift", Fuzzy Tired" and "Fuzzy Output" allow access of the other membership functions of similar name.

3.3.1.8 Rule Base

The rule-base establishes the relationship between the various input membership functions and the output membership function. Specifically it establishes by how much each $act_{i,k}$ may be delayed by $t_{i,k}^{Delay,1} + t_{i,k}^{Delay,2}$ (see Figure 3.15) caused by the influence of the various membership functions and rules. To this end, six rules describing the behaviour of operators on plant operations are formulated as follows:

Rule 1: The model assumes that the operators place a high value on having breaks with their peers. Therefore, operators will not start low priority tasks i.e. $Pr = 1$ or $Pr = 2$, should the outcome be that they might miss the core break $\mu_{\tilde{B}}(t) = 1$.

The operator assesses the possibility of having to work on a low priority task during a break by adding the maximum duration of a task to the current simulated time t^{Sim}. Should the result be that there is an overlap with a core break, the operator will delay this task to a time after the break or formally:

$$t_{i,k,new}^{open,R1} = \begin{cases} t_{3,j}^{B} & \text{, if } \textbf{Condition A} \text{ is true.} \\ 0 & \text{, otherwise} \end{cases} \tag{3.26}$$

With j = 1, 2, ... , 6 (see Figure 3.22 for explanation of $t_{3,j}^{B}$).

Condition A is calculated as follows:

The maximum duration of $act_{i,k}$ denoted t_{max} is calculated as:

$$t_{max} = (t_{i,k}^{Delay,1} + t_{i,k}^{Delay,2}) + t_{i,k}^{min,D} \quad (3.27)$$

Should the time denoted t_{Rule1}, calculated as the sum of the current simulated time t^{Sim} and t_{max}

$$t_{Rule1} = t^{Sim} + t_{max} \quad (3.28)$$

of $act_{i,k}$ either fall within a core break time $\mu_{\tilde{B}}(t) = 1$ or be beyond a core break time the operator will not start $act_{i,k}$. Instead the operator will delay the start of $act_{i,k}$ to a time after the break.

In addition to the delay of $act_{i,k}$ to $t_{i,k,new}^{open,R1}$ expressed in crisp time, there is another delay expressed as a fuzzy number: the efficiency of the operator for the first task after the break is assumed to be slightly reduced, causing a delay to the duration of $\mu_{\tilde{Out}}^{10\%}(r) = 1$.

Rule 2: This rule evaluates how the shift transition from one operator to another influences the start $t_{i,k}^{open}$ of each $act_{i,k}$. During a shift transition period two operators may be available, which, however, does not translate into two operators being available for a given task. On the contrary, the model assumes that the two operators are

occupied with other tasks during the shift changeover such as briefly communicating the status of the current process operation. Therefore, the shift transition period may result in a brief delay in operations.

During the time of a shift transition, the mean of the membership values $\mu_S(t)$ calculated from the shift membership values denoted $\mu_{S1}(t)$ and $\mu_{S2}(t)$ from each operator and given as:

$$\mu_S(t) = \frac{1}{2}(\mu_{S1}(t) + \mu_{S2}(t)) \tag{3.29}$$

Five different intervals were chosen to calculate a possible delay to the operations caused by a shift handover. The delays caused by shift handover are, however, assumed to be smaller than for instance possible delays by tiredness. The following equation is used to decide on the grade of the "Output" function $\underset{\sim}{\mu}^{0\%}_{Out}(r)$ … $\underset{\sim}{\mu}^{100\%}_{Out}(r)$ as follows:

$$\begin{aligned}
&\text{If } 0.9 < \mu_S(t) < 1 \quad \text{then} \quad \underset{\sim}{\mu}^{10\%}_{Out}(r) = 1 \\
&\text{If } 0.8 < \mu_S(t) < 0.9 \quad \text{then} \quad \underset{\sim}{\mu}^{20\%}_{Out}(r) = 1 \\
&\text{If } 0.6 < \mu_S(t) < 0.8 \quad \text{then} \quad \underset{\sim}{\mu}^{50\%}_{Out}(r) = 1 \\
&\text{If } 0.4 < \mu_S(t) < 0.6 \quad \text{then} \quad \underset{\sim}{\mu}^{70\%}_{Out}(r) = 1 \\
&\text{If } 0 < \mu_S(t) < 0.4 \quad \text{then} \quad \underset{\sim}{\mu}^{100\%}_{Out}(r) = 1
\end{aligned} \tag{3.30}$$

Note: Because the shift membership function from both operators is taken into account (equation 3.29), $\mu_S(t)$ is generally small resulting in $\underset{\sim}{\mu}^{100\%}_{Out}(r)$ in the simulation experiments of Chapter 5. This appears to be in line with operational experience.

Rule 3: Rule 3 describes how the operator's tiredness as given by the membership function "Operator Tiredness" $\mu_T(t)$ is influencing the "Output" function $\underset{\sim}{\mu}^{0\%}_{Out}(r)$, $\underset{\sim}{\mu}^{10\%}_{Out}(r)$, … $\underset{\sim}{\mu}^{100\%}_{Out}(r)$ in the following manner:

$$
\begin{aligned}
&1)\ \text{If } 0 < \mu_T(t) < 0.15 \quad \text{then} \quad \underset{\sim}{\mu}^{0\%}_{Out}(r) = 1 \\
&2)\ \text{If } 0.15 < \mu_T(t) < 0.3 \quad \text{then} \quad \underset{\sim}{\mu}^{0\%}_{Out}(r) = 1 \text{ and } \underset{\sim}{\mu}^{10\%}_{Out}(r) = 1 \\
&3)\ \text{If } 0.3 < \mu_T(t) < 0.35 \quad \text{then} \quad \underset{\sim}{\mu}^{10\%}_{Out}(r) = 1 \\
&4)\ \text{If } 0.35 < \mu_T(t) < 0.4 \quad \text{then} \quad \underset{\sim}{\mu}^{10\%}_{Out}(r) = 1 \text{ and } \underset{\sim}{\mu}^{20\%}_{Out}(r) = 1 \\
&5)\ \text{If } 0.4 < \mu_T(t) < 0.5 \quad \text{then} \quad \underset{\sim}{\mu}^{20\%}_{Out}(r) = 1 \\
&6)\ \text{If } 0.5 < \mu_T(t) < 0.6 \quad \text{then} \quad \underset{\sim}{\mu}^{20\%}_{Out}(r) = 1 \text{ and } \underset{\sim}{\mu}^{30\%}_{Out}(r) = 1 \\
&7)\ \text{If } 0.6 < \mu_T(t) < 0.7 \quad \text{then} \quad \underset{\sim}{\mu}^{30\%}_{Out}(r) = 1 \\
&8)\ \text{If } 0.7 < \mu_T(t) < 1 \quad \text{then} \quad \underset{\sim}{\mu}^{30\%}_{Out}(r) = 1,\ \underset{\sim}{\mu}^{40\%}_{Out}(r) = 1
\end{aligned}
\qquad (3.31)
$$

Notes:

1. Should the operator be coming off a break, the tiredness rule (Rule 3) is not invoked.
2. The rule base (equation 3.31) is an estimate based on operational experience.

Rule 4: This rule establishes the delays to operations, which were scheduled to start while the operator is on a break i.e. $\underset{\sim}{\mu}_B(t) = 1$.

In this case, the start of operation $act_{i,k}$ is delayed to:

$$t_{i,k,new}^{open,R4} = \begin{cases} t_{3,j}^{B} & ,\text{if } \underset{\sim}{\mu}_{B}(t) = 1 \\ 0 & ,\text{otherwise} \end{cases} \tag{3.32}$$

With j = 1, 2, ... , 6 (see Figure 3.22 for explanation of $t_{3,j}^{B}$).

Further to the delay of $act_{i,k}$ to $t_{i,k,new}^{open,R4}$ expressed in crisp time, there is another delay expressed as a fuzzy number. The model assumes that the efficiency of the operator for the first task after returning from a break will not be at maximum efficiency, but for the first ask after the break it is: $\underset{\sim}{\mu}_{Out}^{20\%}(r) = 1$.

Rule 5: Despite the work instruction that priority 3 tasks should be dealt with immediately, operational experience indicates that operators do sometimes delay batches deliberately. Therefore, it is assumed that the operators will delay priority 3 tasks, if the core break time i.e. $\underset{\sim}{\mu}_{B}(t) = 1$ is getting close causing a delay of $act_{i,k}$ as follows:

$$t_{i,k,new}^{open,R5} = \begin{cases} t_{3,j}^{B} & ,\text{if } \underset{\sim}{\mu}_{B}(t) > 0.6 \text{ and } Pr = 3 \\ 0 & ,\text{otherwise} \end{cases} \tag{3.33}$$

With j = 1, 2, ... , 6 (see Figure 3.22 for explanation of $t_{3,j}^{B}$).

Further to the delay of $act_{i,k}$ to $t^{open,R5}_{i,k,new}$ expressed in crisp time, another delay expressed as a fuzzy number is added. For this thesis it is assumed that the operator's efficiency after the break will be very good i.e. $\underset{\sim}{\mu}^{0\%}_{Out}(r) = 1$, as the operator feels "guilty" having deliberately delayed a process operation in order not to miss the break.

3.3.1.9 Aggregation & Ranking

In the course of a FL simulation run, some of the above rules of the rule base may be triggered at the same time, which may result in two or more fuzzy outputs of the membership function "Output", for example $\underset{\sim}{\mu}^{0\%}_{Out}(r) = 1$ and $\underset{\sim}{\mu}^{30\%}_{Out}(r) = 1$, demanding ranking and aggregation of the different fuzzy outputs into one overall output.

Ranking is the sorting of fuzzy numbers in order of, for example, which is largest. Ranking of fuzzy numbers is a challenge [53] and is still subject to research and debate [68]. The difficulty with fuzzy ranking is that so many ranking methods exist, therefore requiring intuition of the modeller. Moreover none of ranking methods has emerged as a generally accepted method as was identified by Wang and Kerre [68] in a survey of fuzzy ranking methods, where they identified 35 different ranking methods.

Aggregation is a fuzzy set procedure by which two or more fuzzy sets are combined into one fuzzy set. This thesis does not apply the standard aggregation operations of union or intersection as described in Appendix 1. It was felt that it is unrealistic to aggregate very different fuzzy output numbers into one overall fuzzy number.

Instead the following ranking and aggregation procedure is used:

1. Ranking: Should the fuzzy output membership functions be apart or not overlap, only the larger of the two will be forwarded to the next step i.e. defuzzified, maximising the output in the process. For example in the case of the two membership functions $\underset{\sim}{\mu}_{Out}^{0\%}(r)=1$ and $\underset{\sim}{\mu}_{Out}^{60\%}(r)=1$, the lower fuzzy membership function $\underset{\sim}{\mu}_{Out}^{0\%}(r)=1$ is discarded, leaving only $\underset{\sim}{\mu}_{Out}^{60\%}(r)=1$ for defuzzification.
2. Aggregation: Should the fuzzy output membership functions be adjacent to each other, both will be forwarded to defuzzification, without any aggregation procedure being performed. For example, the two membership functions $\underset{\sim}{\mu}_{Out}^{50\%}(r)=1$ and $\underset{\sim}{\mu}_{Out}^{60\%}(r)=1$, both fuzzy membership functions will be forwarded to be defuzzified.

This aggregation procedure is, of course, *ad hoc* but it appears to suit this FL simulation. A description of a similar ranking/aggregation procedure could not be found in the literature.

3.3.1.10 Defuzzification

The defuzzification procedure used to obtain a crisp output is the center-of-gravity method, itself being a popular method [52, 53], but not one of the easiest to implement in computer code because of the need to solve integrals [135]. But two properties ensure a straightforward center-of-gravity defuzzification procedure. Firstly, all the output membership functions are triangular, symmetrical functions and secondly, due to the rule base and the aggregation/ranking procedure, all outputs have the value of unity.

The defuzzification procedure is twofold. Firstly, should the aggregation procedure return one fuzzy output number; the defuzzification scalar is located below the "peak" of the triangle. For example, the defuzzification of $\underset{\sim}{\mu}_{Out}^{30\%}(r)=1$ equals a defuzzified value $t_{i,k}^{D,De}=30\%$, as graphically displayed in Figure 3.25.

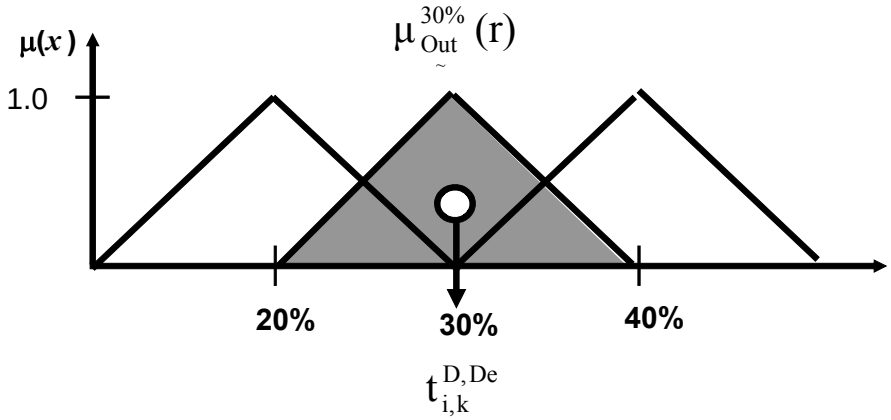


Figure 3.25 Example defuzzification to $t_{i,k}^{D,De} = 30\%$ of "Output" $\mu_{\tilde{Out}}^{30\%}(r) = 1$

Secondly, should the defuzzification procedure return two fuzzy numbers, the defuzzified scalar is the mean of the defuzzified values of the individual fuzzy numbers. For example, the defuzzification of $\mu_{\tilde{Out}}^{30\%}(r) = 1$ and $\mu_{\tilde{Out}}^{40\%}(r) = 1$ equals a defuzzified value of $t_{i,k}^{D,De} = 35\%$.

Note: The rule base, aggregation and defuzzification step can only be adjusted by changing the VBA program (see Appendix 3 for a program listing of the FL model).

3.3.1.11 Schedule Uncertainty

Finally, the time delay denoted $t_{i,k}^{D}$ of $act_{i,k}$ due to the influence of the limited number of operators, various membership functions and the rule base can be calculated as the sum of all delays:

$$t_{i,k}^{D} = t^{sim} + t_{i,k}^{Delay,\,No\,Op.} + t_{i,k,new}^{open,R1} + t_{i,k,new}^{open,R4} + t_{i,k,new}^{open,R5} + t_{i,k}^{min,D} + t_{i,k}^{D,De} \cdot \frac{t_{i,k}^{Delay,1} + t_{i,k}^{Delay,2}}{100} \qquad (3.34)$$

The new opening event $t_{i,k,new}^{open}$ for each $act_{i,k}$ is:

$$t_{i,k,new}^{open} = t_{i,k}^{D} + t_{i,k}^{open} \qquad (3.35)$$

while the new closing event $t_{i,k,new}^{close}$ for each $act_{i,k}$ is:

$$t_{i,k,new}^{close} = t_{i,k,new}^{open} + (t_{i,k}^{close} - t_{i,k}^{open}) \qquad (3.36)$$

3.3.2 Summary of Required Input Variables for the Fuzzy Logic Model

To recap the FL model requires the input parameters as shown in Table 4.4 below.

Parameter	Description
$\dot{V}_{WFI,off,i}$	Volumetric offtake for each act_i
$t_{i,k}^{open}$	Scheduled opening time for each $act_{i,k}$
$t_{i,k}^{close}$	Scheduled closing time for each $act_{i,k}$
n_{op}	Number of operators, see Section 3.3.1.1
$Pr_{i,k}$	Priority rules for each $act_{i,k}$, see Section 3.3.1.2
Membership Function "Duration of Tasks" $\tilde{\mu}_D^{i,k}(t)$	"Duration of Task" for each $act_{i,k}$, see Section 3.3.1.3
Membership Function Operator "Shift Pattern" $\tilde{\mu}_S(t)$	Various inputs, see Section 3.3.1.4
Membership Function "Operator Tiredness" $\tilde{\mu}_T(t)$	Various inputs, see Section 3.3.1.5
Membership Function "Operator Break Pattern" $\tilde{\mu}_B(t)$	Various inputs, see Section 3.3.1.6
Membership Function "Output" $\tilde{\mu}_{Out}^{0\%}(r)$, $\tilde{\mu}_{Out}^{10\%}(r)$ to $\tilde{\mu}_{Out}^{100\%}(r)$	Various inputs, see Section 3.3.1.7
Rule Base	Various inputs, see Section 3.3.1.8
Aggregation & Ranking	Various inputs, see Section 3.3.1.9
Defuzzification	Various inputs, see Section 3.3.1.10

Table 3.4 Input variables required for the fuzzy logic WFI distribution model

3.3.3 Water Storage Tanks & Generating Systems – Mathematical Model

The mathematical model of the DI/WFI storage tanks and generating plants of the FL model is the same as the deterministic model described in Section 3.1.2.

3.4 Summary – Mathematical Formulation of the DI/WFI Models

This chapter developed the mathematical formulation of the deterministic, stochastic and FL models of a DI/WFI system. In the previous chapter - Chapter 2 - the theoretical differences between probability and possibility theory were reviewed and it was shown that a criterion enabling modellers to choose between any of the different methods describing uncertainty as a function of the available information or any other "hard" criteria does not exist. Instead, modellers are left to apply their own experience and preferences when choosing a mathematical method of describing uncertainty. This chapter shows that the selection the modeller makes is not a trivial one because it has large practical consequences for model construction and subsequent programming, testing, validation and calibration. And, as will be shown in Chapter 5, the choice between probabilistic and FL methods also influences the results of the simulation, its general usability and interpretation.

4 Materials and Methods

The materials and methods used in the execution of this research are described in this chapter. First, the available materials are outlined and then the methods for calculating the capacity of a DI/WFI system and for comparing the results are described. The final section of this chapter outlines the reasons for selecting Excel as the simulation platform.

4.1 Materials

A WFI system, as depicted in Figure 4.1, was selected as an illustrative case study. All data used was obtained from the design data of a highly automated secondary pharmaceutical bulking, freeze drying and vial filling facility known to the author through his involvement in its original design. Due to commercial concerns the name and location of the facility are confidential.

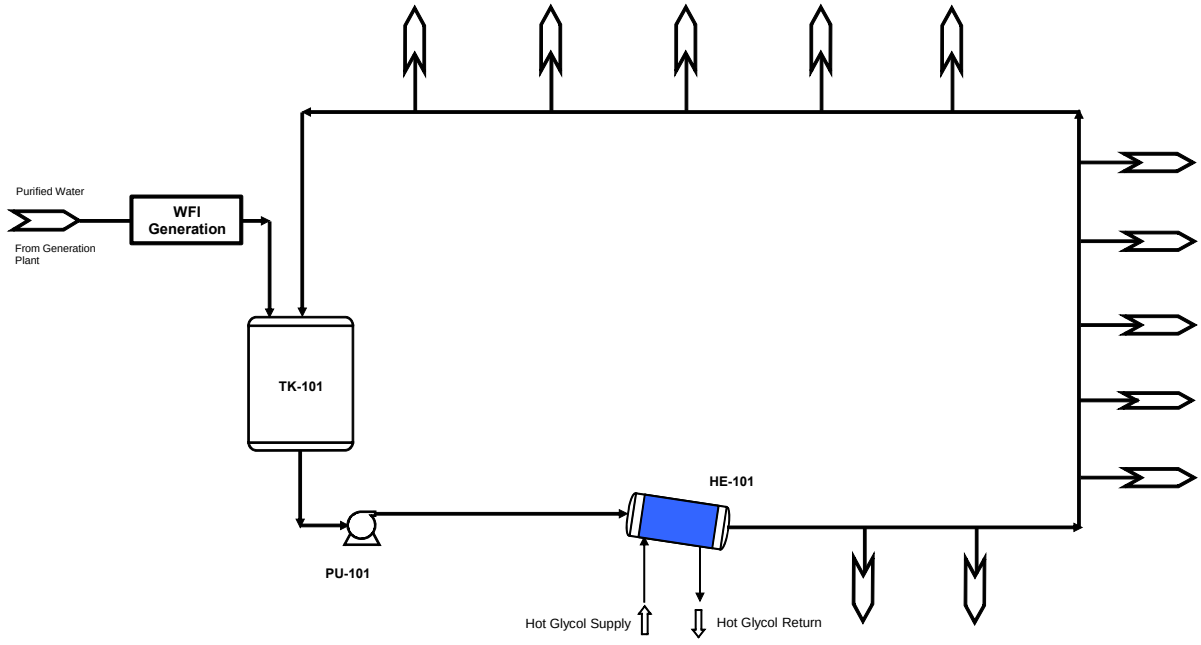


Figure 4.1 Process flowsheet showing a WFI generation and distribution system

4.1.1 General Data - Process

The process conditions of the WFI distribution system under investigation (see Table 4.1 for selected actual mechanical and hygienic design data for the facility being investigated) are as follows:

1. The WFI generating capacity is $\dot{V}_{WFI,Gen} = 8.8 \ \ m^3/h$. This capacity may increase in steps as shown in Table 4.3.
2. The WFI storage tank volume is $V_{WFI} = 40.0 \ \ m^3$. Additional storage capacity of up to 40 m^3 can be provided.
3. The WFI storage tank is always assumed to be full at time index $t_0 = 0$.
4. The WFI system is operating around the clock on a 7 day, 24-hour and 3 shift operations basis.

Parameter	Description
WFI Storage Tank	Storage tank volume = 40 m^3; sanitary design; nitrogen padded; internal surfaces polished to SF3 as per ASME BPE [136]; rated to full vacuum, 3.45 bar(g) and 149 °C, sanitisable with steam.
WFI Distribution Pump	Pump flowrate = 43 m^3/h, head = 45 m; sanitary service; internal surfaces polished to SF3 as per ASME BPE; sanitisable with steam.
WFI Distribution Loop Data	ASTM A-270 Type 316L stainless steel tubing; hot recirculated system with a min. allowable temperature of 65 °C; internal surfaces mechanically polished plus electropolished to SF4 ASME BPE (0.38 µm); standard outer finish; pharmaceutical grade; sanitisable with steam; fluid velocity ~ 1.5 m/s; DN 100; fittings: generally butt-welded, zero-static valves (minimum deadlegs).

Table 4.1 Design data of the WFI generation and distribution system investigated in this study.

The limitations of the WFI system are:

1. The minimum allowable volume of water in the WFI storage tank is $V_{WFI,min} = 2.0 \ \ m^3$ due to NPSH requirements of the WFI distribution pump PU-101.
2. The WFI generation blowdown to drain is $\dot{V}_{WFI,Blow} = 5\%$ and is assumed to be the same for additional WFI generation plants.
3. The WFI distribution pump has a maximum delivery capacity of 43 m^3/h (DN100 pipe with fluid velocity of 1.5 m/s).
4. The WFI storage tank must be full again towards the end of the simulation.

WFI is used in the production process, which takes place in four different processing suites:

1. Bulk Processing Suite (comprising of tanks, pre-rinse, etc. operations)
2. Filling Suite (comprising of vial filling, stopper washer and other operations)
3. Washing Suite (such as parts washer)
4. CIP Suite (for example vial filler and lyophiliser CIP operations)

These different process suites are of modular construction and can be connected to the WFI distribution system via existing spare connections with minimal disruption. Increases in output of product from the production facility i.e. increase in filled product can be accomplished in two ways; firstly by maximizing the use of the equipment and secondly by increasing the number of filling suites. Should the number of filling suites be increased, the number of the other suites may have to be increased also, taking the relative process capacities between the different suites into account. For example, one Process/Mixing suite can, as is noted in Table 4.2, cater for two Filling Line suites, though one CIP suite can accommodate two Filling Line and four Process/Mixing suites.

Process Suite	Relative Capacity of Suite
Bulk Processing / Mixing	2 x Filling Line
Bulk Processing / Mixing	2 x Washing
CIP	2 x Filling Line
CIP	4 x Bulk Processing / Mixing

Table 4.2 Stepwise increase in the number of process suites

Similar to the stepwise increase in processing suites, the WFI generation plant output can only increase in discrete steps as shown in Table 4.3, as the preferred vendor only offers these WFI generation plants.

Model Series	WFI Generation Plant Output [litre/h]
WFI500	500
WFI1200	1,200
WFI2200	2,200
WFI3200	3,200
WFI4200	4,200
WFI5200	5,200
WFI6200	6,200
WFI8800	8,800
WFI11000	11,000
WFI14000	14,000

Table 4.3 WFI generating plant capacities

4.1.2 WFI Event Data Set – Deterministic and Stochastic Simulation

The uncertainty data, that is the stochastic distribution and the uncertainty interval, were estimated by production personnel. About 13% of the tasks of are manual filling operations i.e. parts washing and line rinsing, which are assumed having a volumetric accuracy of ±10% of full volume. All other dispensing operations are automated having an accuracy of ±1% of full volume. The event data set for the deterministic and stochastic simulations are included in Appendix 6 and 7 respectively.

4.1.3 WFI Event Data Set – Fuzzy Logic Simulation

The input data set for the Fuzzy Logic (FL) simulation uses the same crisp opening and closing times for the off take valves as the deterministic and stochastic simulations. The FL model requires the number of operators, priority information on the process, shift pattern, shift breaks and operator tiredness.

Two different fuzzy set-up times are used to describe the timeframe for manual operations. The first set assumes that the operators may require up to 5 minutes for a task, whereas the second one assumes the operators may require up to 10 minutes for a given task. A task was allocated either the 5 or 10 minutes fuzzy time on advice from production personnel. For simplicity, the number of manual operations is equally split between the two fuzzy set-up times. More information on the fuzzy set-up time is provided in Chapter 3. The event data set for the FL simulations is included in Appendix 8.

4.2 Methods

4.2.1 Discrete Event Simulation

Discrete event simulation (DES), one of the most popular simulation techniques [137], is used to find the WFI demand from the water distribution loop as it evolves over time, while a continuous simulation computes the variation of the level in the WFI storage tank. DES is often used to gain insights into complex systems, that elude analytical treatment. It is most applicable for systems in which the state of the system changes at discrete times, such as patient arrivals at hospitals [41], car arrivals at traffic intersections [41], many manufacturing systems [41] including pharmaceutical production and packaging [138, 139] or indeed WFI systems [21, 23]. In WFI

systems the state of the system changes should a valve on the distribution loop open or close, changing the water demand from the loop as is graphically shown in Figure 4.2.

Traditionally DES was not a tool for optimisation [2]. Efforts are now under way to overcome this limitation by linking DES with a genetic algorithm as was proposed by Dietz et al. [140]. In the context of this work, however, finding an optimal solution is not necessarily of concern, as one should keep in mind that optimisation problems have many variations. Saraph [21], for instance, used simulation to show that the WFI shortage at the Bayer facility in California was caused by poor water management rather than undersized equipment avoiding capital investment into the WFI system.

More detailed information on DES can be found in Banks [2, 41], Law and Kelton [1], Carson [141], or Fishman [3].

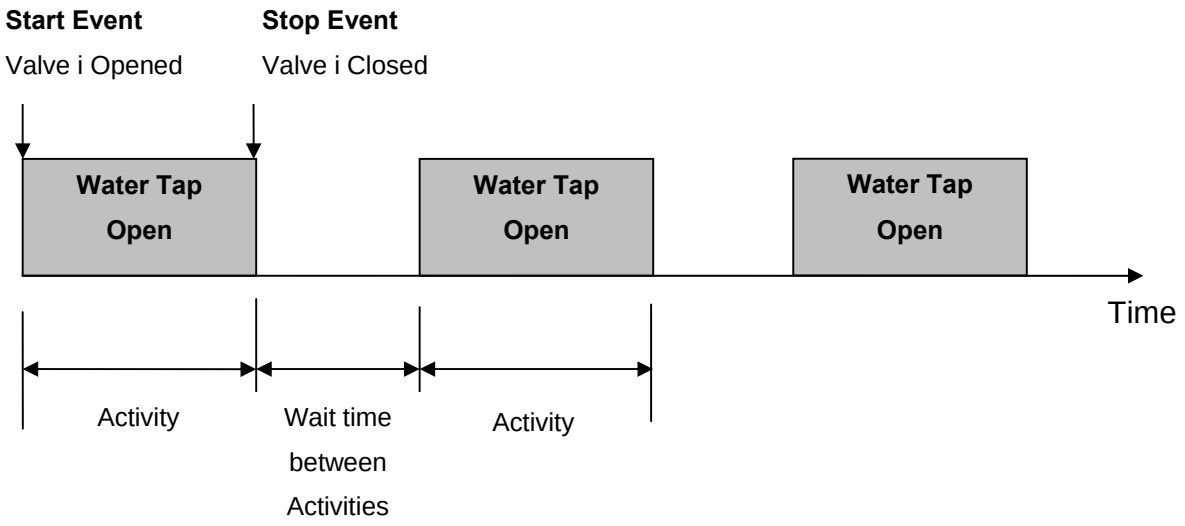


Figure 4.2 On the workings of a discrete-event simulation

4.2.2 Objective Function and other Requirements

The simulation models developed for this thesis shall be used to identify the hidden capacity of an existing WFI system. In other words the objective function is to avoid capital expenditure

into the WFI system, should an increased production demand require more WFI. Therefore, a definition relating *capacity* to *capital expenditure* in the context of a WFI system is required.

Leeuw [142] proposed an economic definition of capacity linking *capacity* and *capital expenditure*. This author views capacity as the maximum output a system can achieve and a higher output can only be achieved with more investment. Extending this definition to the problem posed here, the capacity of a WFI is defined as the capacity that can be achieved with (1) re-scheduling or (2) with relatively minor investments such as additional WFI storage or generation capacity or both, while avoiding major investments i.e. a new WFI system. In this context re-scheduling is the preferred option, as it is cost neutral as far as the existing WFI system is concerned, as shown in Figure 4.3 the decision tree for a WFI upgrade. But it must be stated that this work sees the WFI system on its own and ignores the process system. Whether for instance re-scheduling of the process is actually possible is not investigated here.

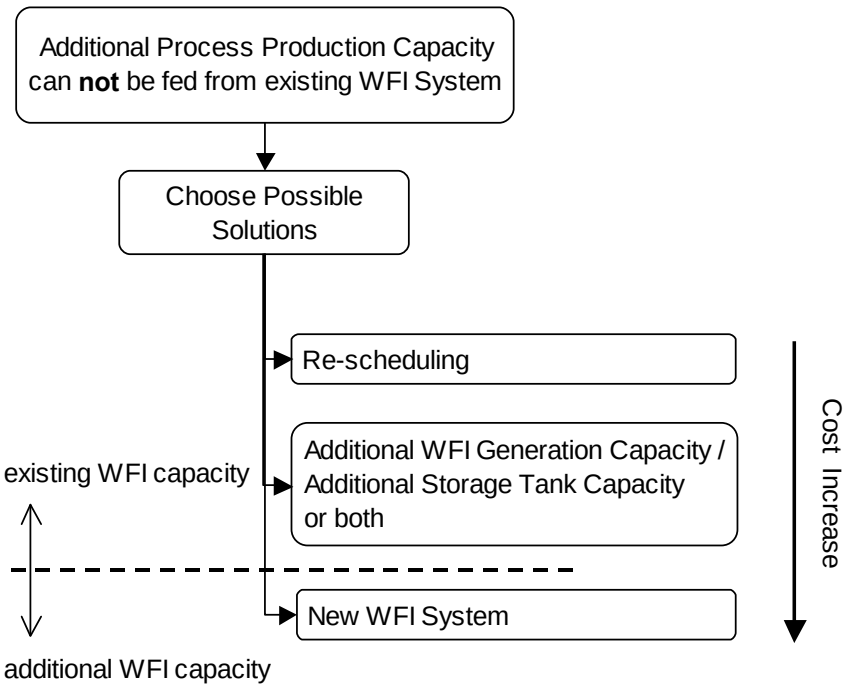


Figure 4.3 Decision tree for the WFI system upgrade

4.2.3 Procedure for Assessing Available WFI System Spare Capacity

The procedure for assessing if an existing WFI system can accommodate an increase in WFI demand is graphically given by Figure 4.4. The number of process suites and with it the WFI demand is increased, till the capacity of the WFI distribution system is exhausted. The capacity of the WFI system may be limited by the delivery capacity of the main distribution pump, the WFI generation plant, the WFI storage tank or a combination of the latter two. Should the increase in WFI demand not be accommodated by the "existing" WFI system, first re-scheduling, which in this case is staggered demand and then additional WFI storage capacity, additional WFI generation capacity or both are applied to accommodate the increased demand.

The number of processing suites is increased again until it becomes clear that only a new, additional, WFI system is capable of supporting the increased production demand. Other restrictions such as space limitations for the additional process suites or operational and process reliability issues are ignored.

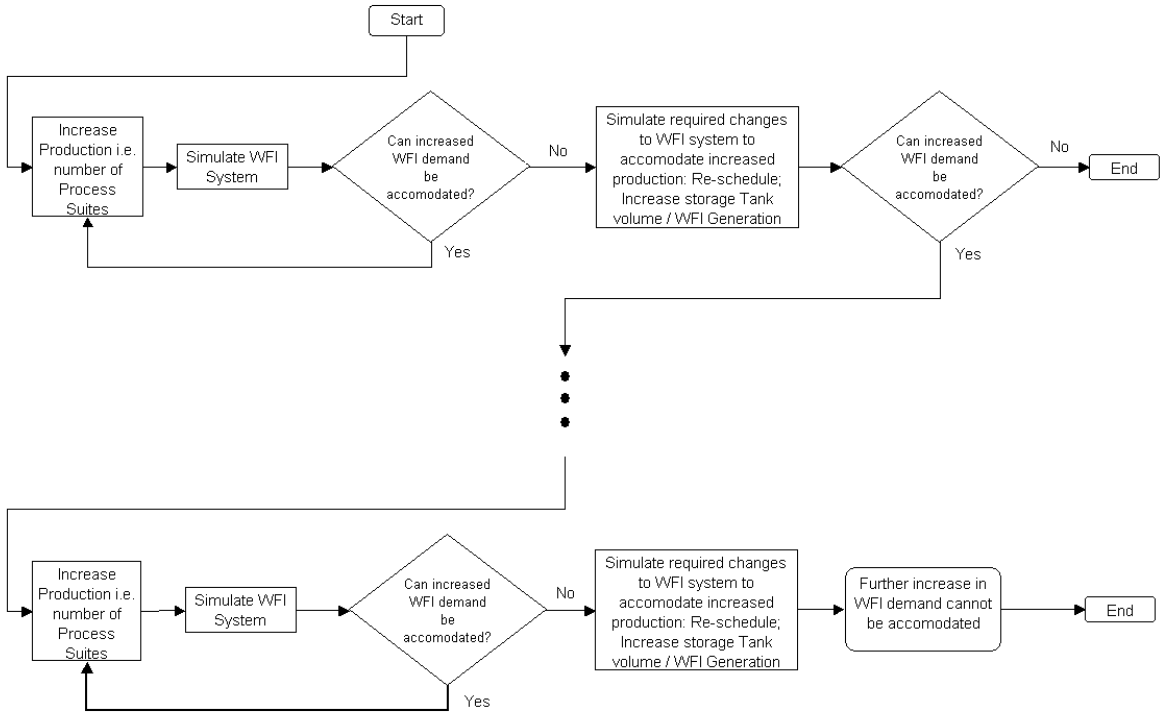


Figure 4.4 Methodology flowchart to assess if further increases in DI/WFI demand are possible

4.2.4 Qualitative Comparison to Reveal the Most Appropriate Method

The method used in finding the most appropriate mathematical method to describe uncertainty in existing DI/WFI systems is by comparing the results of the deterministic, stochastic and FL simulations. The comparisons are made on a qualitative basis, because the conclusions drawn should be applicable for all DI/WFI systems and not just for the specific system investigated.

In order to make the comparisons between the different models as fair as possible, the inputs to the various models must be as similar as possible. The deterministic model was used to determine the capacities of the purified water storage tank and purified water generation plant capacity for all increases. Furthermore the deterministic simulations set down the staggered starting times of any increases. These capacities and starting times were subsequently used as inputs for the stochastic and FL simulations. More information on this method is provided in Chapter 5.

4.2.5 The Monte Carlo Method for Solving the Stochastic Model

The Monte Carlo (MC) method [1, 2, 126, 143] is used to solve the stochastic model. A difficulty with any MC simulation is to decide how often the simulation has to be repeated to be confident about the results, because a general criteria does not exist [143]. For the purposes of this thesis 30 repeats are deemed sufficient as is further outlined in Section 5.3.2.

The MC method is not the only method to solve stochastic models [144]. But it is a method that is often applied and compared to the other methods conceptually easy to understand requiring no advanced mathematics. For these reasons MC was chosen.

4.2.6 Model Validation

Model validation is a critical part of the model building process, proving the suitability of the model for the problem at hand. For this work the end-user demanded a very high level of confidence into the model and provided the final approval on the validity of the model. Sargent [145] describes fifteen different techniques to model validation. Out of these fifteen techniques seven are applicable to validate the DI/WFI model. All of these seven techniques were used, hence maximising the confidence into the model. Firstly the model was validated against the

real system denoted "Existing Process" in Table 5.1 For the actual (start-up design) project that is represented by the data in this thesis the end-user demanded a very high level of confidence from the model and provided the final approval on the validity of the model for the *real* system. All capacity increases simulated in this work are future possible processes (see Table 5.1) and therefore historical data does not exist for those simulations. Secondly the *Internal Validity* test was applied to the model to ensure consistency of the results, which the results showed. Thirdly the relationship between input and outputs was checked by the end-user and the model developer if reasonable (*Face Validity* test). Fourthly the graphical outputs were checked if reasonable (*Operational Graphics* test). Fifthly the *Extreme Condition* test was performed. For instance, an offtake of 10,000 m^3/h should be readily visible in the results if the background offtake is less than 50 m^3/h. Or for the FL model an input of one instead of forty operators should display reasonable results. Sixthly the results of the *Sensitivity Analysis* should be reasonable, which they are as shown in Chapter 5. Seventhly specific inputs were followed through the simulation flow to check if the logic of the model is correct (*Traces* test). Moreover the FL model also passed the *Degenerate* test. For example, the number of operations waiting at the end of a simulated day should increase with decreasing number of operators (see Section 5.4.1). The computer models passed all of these validation tests, and are therefore considered by the author, who has 20 years industrial experience in the design of pharmaceutical plants including WFI plants and the end-user, to be accurate for the purposes of this work. An example of a non-applicable validation test is the *Animation* test, since movements of parts through for instance a factory do not occur here.

Data validity issues are also highlighted by Sargent [145]. Allocation and precedence constraints (see Section 3.1) must be adhered to by the input data. As large amounts of data are difficult to check manually, the stochastic model includes a subroutine, which checks the input data for suitability for the simulation. The program of the fuzzy simulation does not include this subroutine as the input data was first used on the stochastic simulation and then a similar only slightly modified data set was used for the fuzzy simulation.

4.2.7 Analysis of Simulation Results

Most discrete event simulations require output analysis; typically a statistical analysis as noted by Law and Kelton [1], Banks [2], Sanchez [146], Nakayama [147] and Alexopoulos [148]. Here, however, as is explained in Chapter 5 graphical outputs are used instead of statistical analysis.

The most relevant results are the graphical displays of the level variation in the storage tank and the WFI demand profile over time as shown in a sample output interface in Figure 4.5. These graphs give a measure of the availability of WFI, as a lack of WFI may put the product at risk or at best may cause a production delay. Another graphical output is provided by the probability density function (PDF) of the chances of how often the WFI demand outstrips the flowrate of the WFI distribution pump leading to starvation and delays of these operations. Details on these PDF's are provided in Chapter 5.

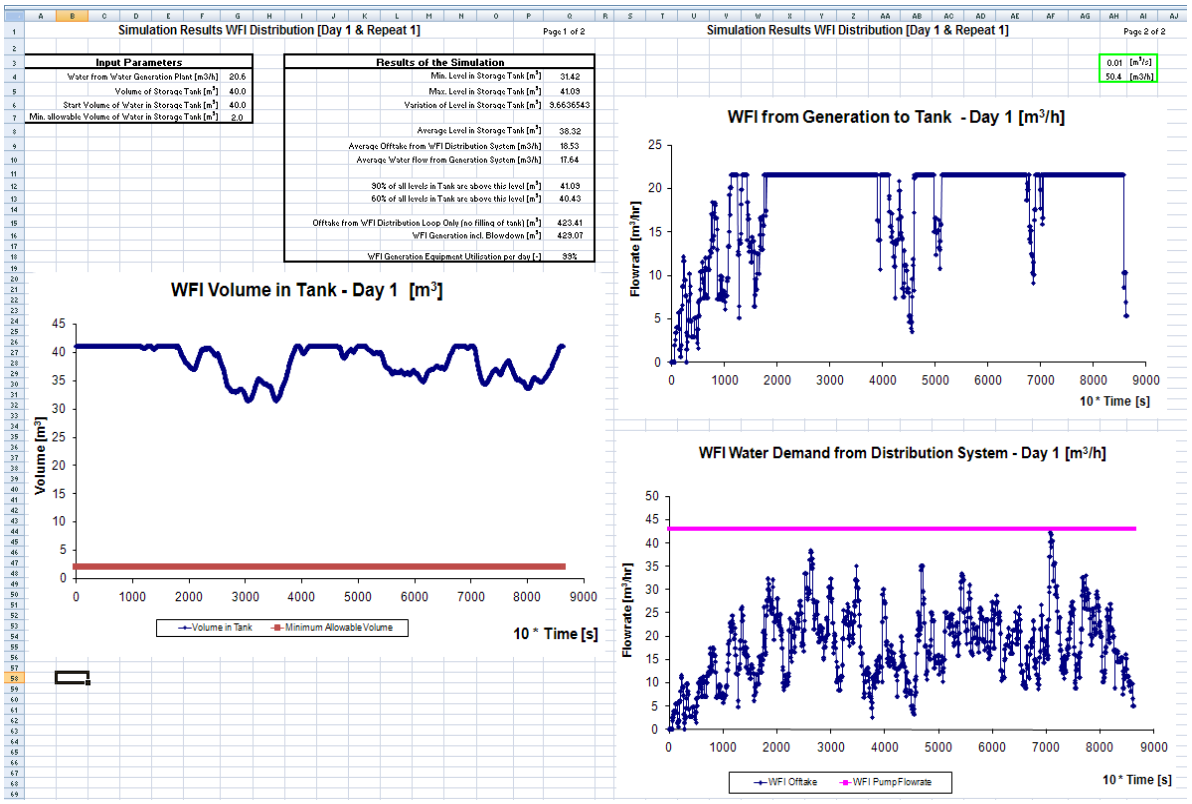


Figure 4.5 Sample output interface of the stochastic WFI simulation *(Excel screenshot)*

4.2.8 Software Selection

This study utilizes the standard version of Excel 2003 as the platform to execute the simulation of an industrial scale pharmaceutical water system. Excel was chosen over commercially available discrete-event simulation packages to provide a cost effective, portable, and open model.

The cost of simulation packages appears to be a hurdle in limiting the widespread application of simulation in practice. Greasley [149] and Hlupic [150] note that cost is a major issue for industry, making the standard version of Excel attractive, because it is a de-facto industry standard [151, 152] and hence available on most PCs in industry.

Excel has been used as a discrete event simulation package before. Greasley [153] used Excel and published the VBA (Visual Basic for Applications) source-code to simulate supermarket queues in an educational setting with a view that students would learn the basics of discrete event simulation with Excel before moving on to more sophisticated simulation packages. Some authors, for instance Thiriez [154], utilise Excel add-ons such as Crystal Ball to build professional models. However, utilising the standard version of Excel as the platform for a professional model without the use of any add-ons keeps the cost of the model to its minimum and enables sharing of the model between many users.

The Excel spreadsheet offers three different approaches to program: (1) the spreadsheet grid environment, (2) VBA, which is a object-orientated general purpose programming language [155], or (3) a combination of the two. Early on during the course of this work, it was decided that all the programming would be done in VBA and not on the spreadsheet grid. Instead the spreadsheet grid serves as the input/output interface only.

The reason for rejecting programming on the spreadsheet is that it is prone to errors, which are difficult to find and debug [156, 157]. It is well established that spreadsheet modelling is error prone [156, 158]. Deficits in design and testing of the program [156, 157] and the unstructured nature of the spreadsheet [156, 157] often result in mistakes in spreadsheets. Programming in VBA on the other hand should reduce programming errors, provided the principles of good

programming [159, 160] are followed. As a consequence of writing the VBA with code maintenance in mind, no emphasis was put on efficiency of code execution. Programming in VBA also offers a versatility and availability for code maintenance and debugging that the spreadsheet environment cannot offer. A disadvantage of using VBA is that, apart from the Random Number Generator (see Appendix 4) the entire source code had to be written.

The Excel workbooks of all the models developed are available for download as open public domain software from the internet site of the Department of Process and Chemical Engineering UCC [161, 162]. No guarantee whatsoever is given that the code is correct or suitable for any particular purpose as per the GNU General Public Licence agreement [163].

5 Simulation of an Industrial Scale Water for Injection System – Numerical Experiments

5.1 Introduction

This chapter exhibits the experimental results of the three different WFI models. The results of the deterministic simulations are followed by the results of the stochastic simulations and finally the results of the fuzzy logic (FL) simulations.

The relevant input data sets of the simulation experiments are included in the Appendices. Appendix 6 includes the input data for the deterministic, Appendix 7 the input data for the stochastic simulations and Appendix 8 the input data for the FL simulations.

5.2 Results of the Deterministic Simulations

The results of the deterministic simulations are presented first as they are used as a baseline for the stochastic and FL simulations as outlined in the previous chapter. The deterministic simulations are used to calculate the WFI tank storage volume, the WFI generation capacity and the starting times arrangements for the additional processes. All of these three parameters are retained as inputs for the stochastic and FL simulations. The additional incremental in water demand from the systems over and above the original baseline demand due to the imposition of various additional process features are named "1st Increase", "2nd Increase", "3rd Increase" and so forth (see Table 5.1). The required WFI storage tank volumes and WFI generation capacities for each additional process are given in Table 5.1 which also lists the overall WFI demand and the cumulative number of all $act_{i,k}$ (opening / closing events of a WFI valve) for each increase with the "7th Increase" having 829 $act_{i,k}$.

The deterministic simulation predicts that the capacity of the WFI storage tank at 40 m^3 can remain constant throughout all the increases. Figure 5.2 (left hand panels) display the water level in the storage tank over the simulated day for the "1st Increase" to the "7th Increase",

demonstrating that the existing WFI storage tank capacity of 40 m^3 is sufficient throughout all increases, with the dotted line indicating the minimum allowable volume in the storage tank of 2 m^3, which is not predicted to be reached. The right hand panels display the WFI demand from the distribution system over time, with the dotted line indicating the flowrate of the WFI distribution pump PU-101 of 43 m^3/hr. But the deterministic simulation also predicts that the WFI system can only deliver this increased WFI demand, if the production starting times of the additional processes are staggered by at least 2 hours as illustrated in Figure 5.1 avoiding simultaneous maximum demand from the various WFI users. The 2 hour staggering time was found by trial and error. No attempt was made to optimize i.e. reducing this staggering time of 2 hours. However, a 1.5 hour staggering time results in the model predicting the WFI tank being empty at some point during the day and was therefore discounted as a solution.

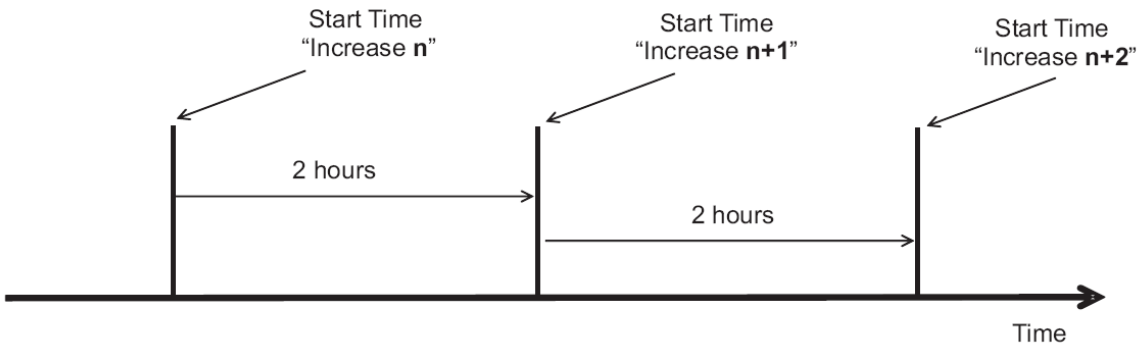


Figure 5.1 Staggered arrangement of process start times of "Increase n" and "Increase n+1"

Name of Increase	**Description of Process Capacity Increase**	**Cumulative Number of** $act_{i,k}$ (opening and closing events of a valve)	**WFI Offtake from Distribution System over 24-hours [m^3]**	**Results of the Deterministic Simulations**	
				WFI Generating Capacity [m^3/h]	**WFI Storage Tank Volume [m^3]**
Existing Process	-	188	129	8.8	40
Future Possible Processes (all simulated)					
1st Increase	Maximise use of equipment	266	160	8.8	40
2nd Increase	Add second filling suite, this second filling suite requires additional CIP and Washing Suites	429	278	14	40
3rd Increase	Maximise use of equipment added in second increase	496	305	14	40
4th Increase	Add third filling suite	655	433	20.6	40
5th Increase	Maximise use of equipment added in 4th increase	713	456	22.8	40
6th Increase	Add fourth filling suite	778	521	28.5	40
7th Increase	Maximise use of equipment added in 6th increase	829	546	29	40

Table 5.1 Required WFI generation capacity and WFI storage tank volume for the "Existing Process" and the "1st Increase" to "7th Increase" as established by the deterministic simulation.

A discrete event simulation generates large amount of data, which needs to be analysed. Usually a statistical analysis is performed. Two papers by Nakayama [147] and Sanchez [146] give hints on how to perform statistical analysis suitable for most discrete event simulations. Savage [164], however, argues that statistical point measures provide little useful information for dynamical systems, because events of special interest such as minimum water level in the storage tank may be hidden in the average figures. This very point applies to the statistical point measures of the deterministic WFI simulations given in Table 5.2. Therefore statistical point measures are not a useful criterion to assess if an existing WFI system can support an increased WFI demand. Instead a different criterion must be found.

Statistical Measure	Existing Process	Increase						
		1st	2nd	3rd	4th	5th	6th	7th
Min. Level in Storage Tank [m^3]	37.09	32.45	25.06	17.81	25.11	29.92	31.41	29.31
Average Level in Storage Tank [m^3]	40.64	39.59	37.81	35.21	35.55	37.55	39.32	38.50
Average Offtake from WFI Distribution System [m^3/h]	5.61	6.87	12.11	13.21	18.73	19.96	22.72	23.71
Average Water flow from Generation System [m^3/h]	5.39	6.65	11.58	12.72	18.04	19.01	21.72	22.74
90% of all levels in Tank are above this level [m^3]	41.09	41.09	41.09	41.09	41.09	41.09	41.09	41.09
60% of all levels in Tank are above this level [m^3]	41.09	41.04	40.67	38.20	36.84	39.64	41.09	41.09
WFI Generation Equipment Utilisation per day [-]	84%	89%	97%	97%	98%	98%	98%	98%

Table 5.2 Statistical point measures of the deterministic simulation

This different criterion can either be the graphical output of the simulations or an appropriate numerical measure. The graphical output may provide information on the predicted behaviour of the WFI level in the storage tank or the WFI demand profile from the distribution system over time as is shown in Figure 5.2 for the "Existing Process" and the "1st Increase" to the "7th Increase". Furthermore a numerical measure can be constructed. From a production point of view, unavailability of WFI, which may result in production delays, is of most interest. Therefore the numerical measure assessing the suitability of an WFI system to cope with increases in WFI demand is:

- Minutes per day the WFI demand exceeds the WFI distribution pump flowrate of 43 m^3/h

Applying this measure to the results of the deterministic simulation reveals that the deterministic simulation predicts WFI starvation of some process operations from the "5th Increase" onwards as can be seen from Table 5.3 and graphically from Figure 5.2 right hand panels showing WFI demand peaks exceeding the WFI distribution pump flowrate of 43 m^3/h as indicated by the dotted line. During these peak demand times process operations fed from the WFI valves located towards the end of the distribution loop will be starved of WFI and these process operations will be delayed till the overall WFI demand has dropped below 43 m^3/h.

Name of Increase	**Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate [Min]**
Existing Process	0.0
1st Increase	0.0
2nd Increase	0.0
3rd Increase	0.0
4th Increase	0.0
5th Increase	25.0
6th Increase	70.0
7th Increase	110.0

Table 5.3 Results of the numerical measure of the deterministic simulation

The starvation times are also used as a criterion to terminate further demand increases. The deterministic simulation predicts a cumulative WFI starvation time of 110 minutes for the "7th Increase", which is assumed to be too long of a cumulative production delay per day. Therefore no further increases are simulated.

A welcome side effect of increasing the WFI demand is an increase in utilisation of the WFI generation equipment. As shown in Table 5.2 the utilisation of the WFI generation system rises from 84% for the "Existing Process" to 98% for the "4th Increase" and later increases in WFI demand. The results of the stochastic and FL simulations display a similar increase in utilisation (see Tables 5.4 and 5.12).

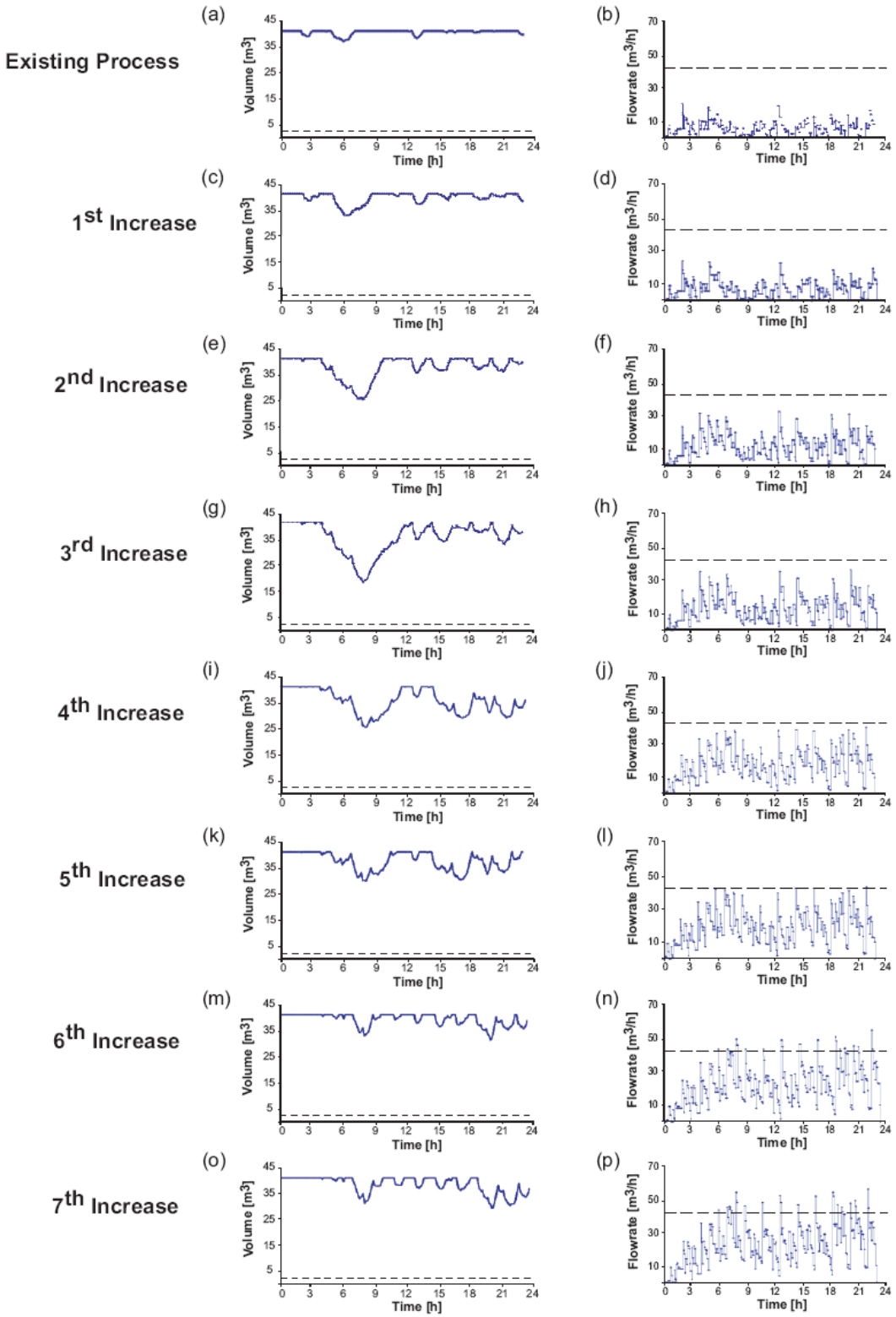


Figure 5.2 Graphical results of the deterministic simulation for the ”Existing Process” and the “1st Increase” to “7th Increase”

At first glance the behaviour of the water level in the storage tank is somewhat intriguing (see Figure 5.2 left hand panels). For the "3rd Increase" the level of water in the storage tank is predicted to be quite low during the morning, but for subsequent increases this low water level does not occur anymore. This is for two reasons. Firstly the additional demand on the WFI system is staggered by 2 hours resulting in a more equal water demand profile throughout the day and secondly the "4th Increase" requires a large incremental increase in WFI generating capacity to ensure WFI is available at all times (see Table 5.1). Consequently from the "4th Increase" onwards the level in the storage tank is more equal throughout the day. At the same time it is apparent from the water demand profile (see right hand side panels of Figure 5.2), that the water demand is increasing with the number of increases and is also more equally spread throughout the day.

5.2.1 Sensitivity Experiment

An experiment showing the influence of setting the WFI generation plant capacity lower than the capacities given in Table 5.1 was performed on the deterministic model. Figure 5.3 shows how the water level in the storage tank behaves over time for the "7th Increase" should the water generation capacity be lower than 29 m^3/h. These graphs results demonstrate that the WFI generation capacity should not be set lower than the capacities given in Table 5.1, otherwise the WFI storage tank will not be full again towards the end of the 24 hour simulation time frame.

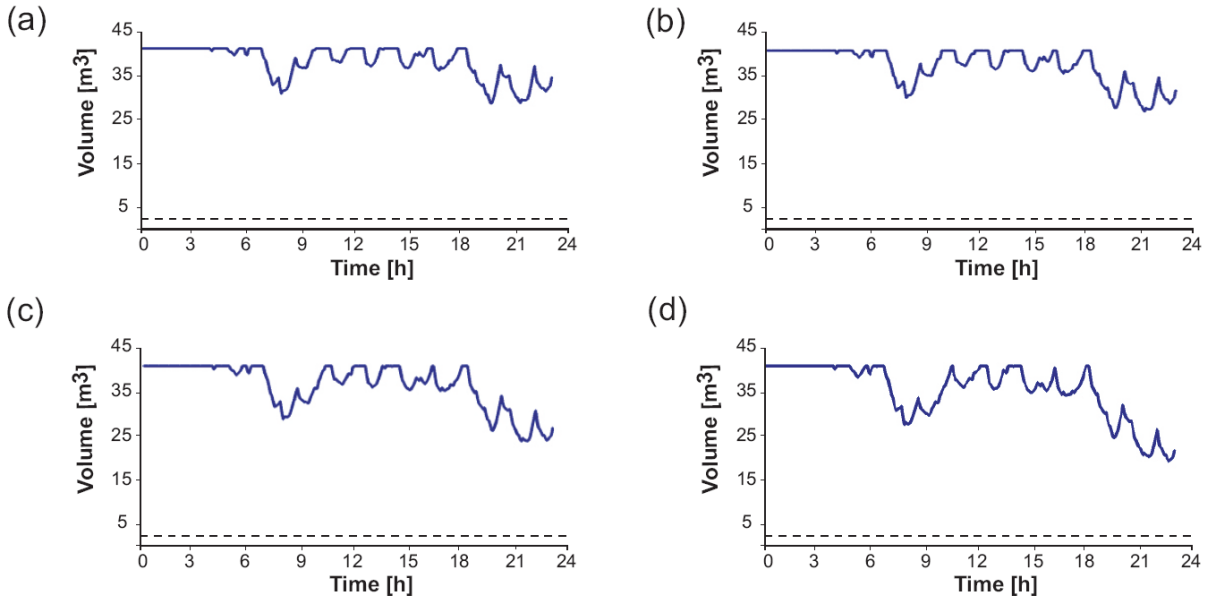


Figure 5.3 Graphical results showing the predicted level in the storage tank should the WFI generation plant capacity of the "7th Increase" be set to (a) 28.5 m^3/h, (b) 28 m^3/h, (c) 27 m^3/h and (d) 26 m^3/h instead of 29 m^3/h

5.3 Results of the Stochastic Model Simulations

A Monte Carlo (MC) simulation generated the results of the stochastic simulations. Thirty simulated days (30 MC) of data were generated for each "1st Increase" to "7th Increase", giving slightly different results for each of the thirty simulated days. This is similar to a real WFI system, which would also show day-to-day variations. This day-to-day variation may ensure that the results of the stochastic simulations appear more "real" compared to the results of the deterministic simulation, which are static not showing any day-to-day variation. As an example, Figures 5.4 and 5.5 display how the water level in the WFI storage tank and the WFI demand profile for the "5th Increase" and "7th Increase" are predicted to behave for six simulated days.

Statistical Measure	Increase				
	3rd	4th	5th	6th	7th
Min. Level in Storage Tank [m^3]	23.40	28.98	30.84	27.84	32.30
Average Level in Storage Tank [m^3]	36.97	37.88	38.17	39.21	39.72
Average Offtake from WFI Distribution System [m^3/h]	13.13	18.85	19.83	22.79	23.68
Average Water flow from Generation System [m^3/h]	12.50	17.95	18.89	21.75	22.54
90% of all levels in Tank are above this level [m^3]	41.09	41.09	41.09	41.09	41.09
60% of all levels in Tank are above this level [m^3]	39.40	40.12	40.95	41.09	41.09
WFI Generation Equipment Utilisation per day [-]	99%	99%	99%	99%	99%

Table 5.4 Statistical point measures of the stochastic simulation of the "3rd Increase" to the "7th Increase"

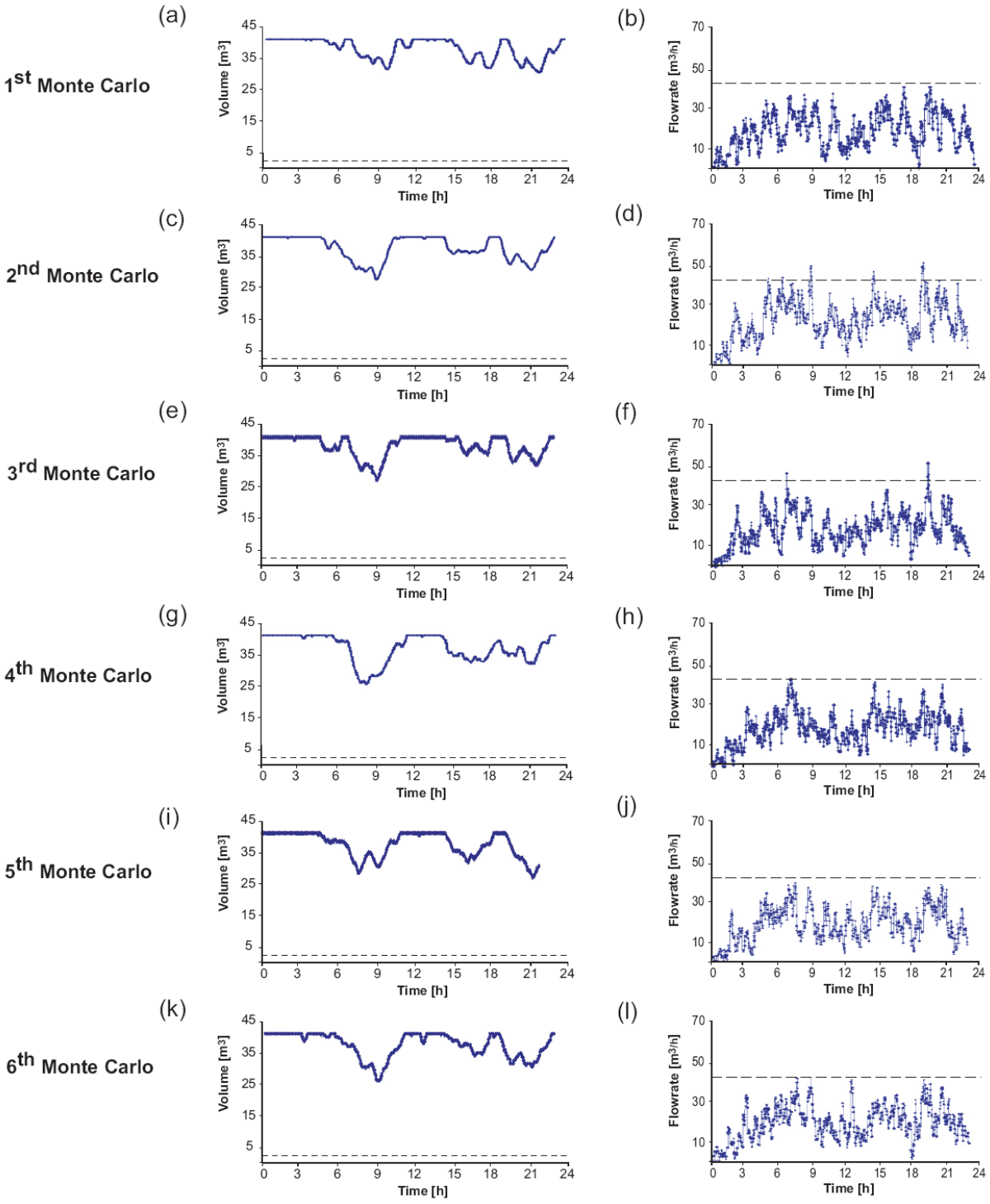


Figure 5.4 Graphical results of the stochastic simulation: "5th Increase", 1st to 6th MC run

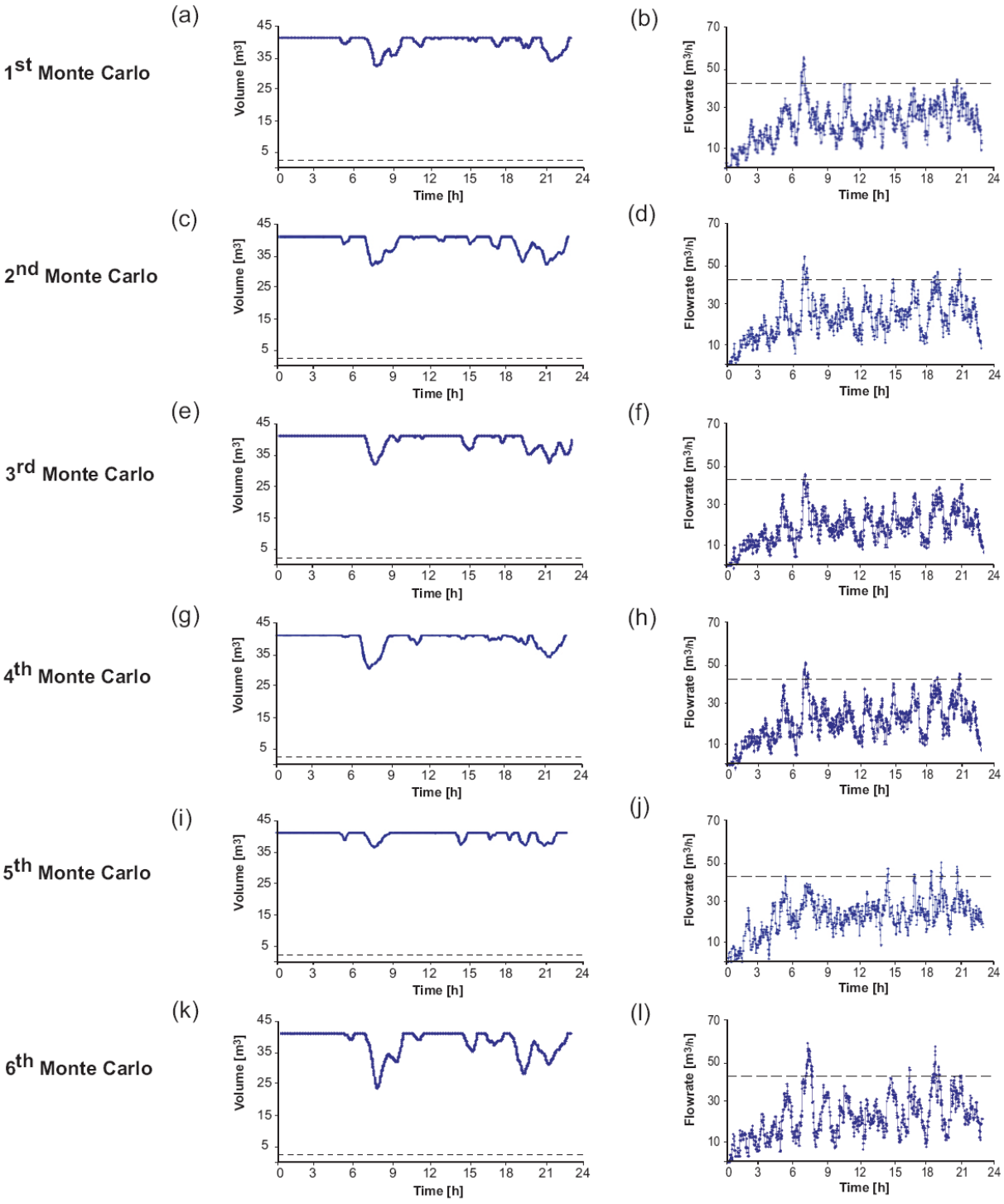


Figure 5.5 Graphical results of the stochastic simulation: "7th Increase", 1st to 6th MC run

Description of Input Data / Statistical Measure	5th Increase		6th Increase		7th Increase	
	Deterministic	Stochastic	Deterministic	Stochastic	Deterministic	Stochastic
Min. Level in Storage Tank [m^3]	29.92	30.84	31.41	31.76	29.31	32.33
Average Level in Storage Tank [m^3]	37.55	38.17	39.32	39.99	38.50	39.74
Average Offtake from WFI Distribution System [m^3/h]	19.96	19.83	22.72	22.68	23.71	23.63
Average Water flow from Generation System [m^3/h]	19.01	18.89	21.72	21.60	22.74	22.52
90% of all levels in Tank are above this level [m^3]	41.09	41.09	41.09	41.09	41.09	41.09
60% of all levels in Tank are above this level [m^3]	39.64	40.95	41.09	41.09	41.09	41.09
WFI Generation Equipment Utilisation per day [-]	98%	99%	98%	99%	98%	99%

Table 5.5 Comparison of the statistical point measures of the deterministic and the stochastic simulations of the "5th Increase" to the "7th Increase"

Similar to the statistical point measures of the deterministic simulations, the statistical point measures of the stochastic simulation given in Table 5.4 provide little useful information in assessing if a WFI distribution system is capable of delivering additional WFI should the demand from the production process increase. And it is worthwhile to point out that the statistical point measures of the deterministic and stochastic simulations are similar as is evident from Table 5.5.

For the same reasons as the deterministic simulations, the results of the stochastic simulations are assessed by the numerical measure of how long the WFI demand may exceed the WFI distribution pump flowrate. However, the results of the stochastic simulations must be presented differently to the results of the deterministic simulations as 30 MC runs are performed. Whereas the deterministic simulation gives one result per Increase, the results of the stochastic simulation as presented in Table 5.6 are given as minimum, maximum and average figures derived from the 30 MC runs. From this table it would appear that the stochastic simulation give more conservative results than the deterministic simulation as it predicts WFI starvation from the "4th Increase" onwards, which the deterministic simulation does not predict. This is, of course, an

inherent feature of a Monte Carlo simulation repeating the simulation a number of times each with different values for schedule and volume.

Name of Increase	Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate [Min]			
	Deterministic Model	Stochastic Model		
		Max.	Min.	Ave.
Existing Process	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0
4th Increase	0.0	31.0	0.0	3.3
5th Increase	25.0	11.0	0.0	3.2
6th Increase	70.0	54.0	0.0	15.4
7th Increase	110.0	68.0	3.0	23.1

Table 5.6 Comparison of the results of the numerical measure of the stochastic and the deterministic simulation

The interpretation of Max., Min. and Ave. in Table 5.6 can be difficult as these figures do not give any probability or distribution of the data. An improved presentation of the results may be provided by utilising a probability density function (PDF) of the chances of how often the WFI demand outstrips the flowrate of the WFI distribution pump, as shown in Figure 5.6 for the "4th Increase" and the "7th Increase". The PDF of the "4th Increase", for example, shows that the majority of WFI starvation time intervals are of short duration (< 5 min) and that the WFI starvation times exceeding 15 minutes are of low probability.

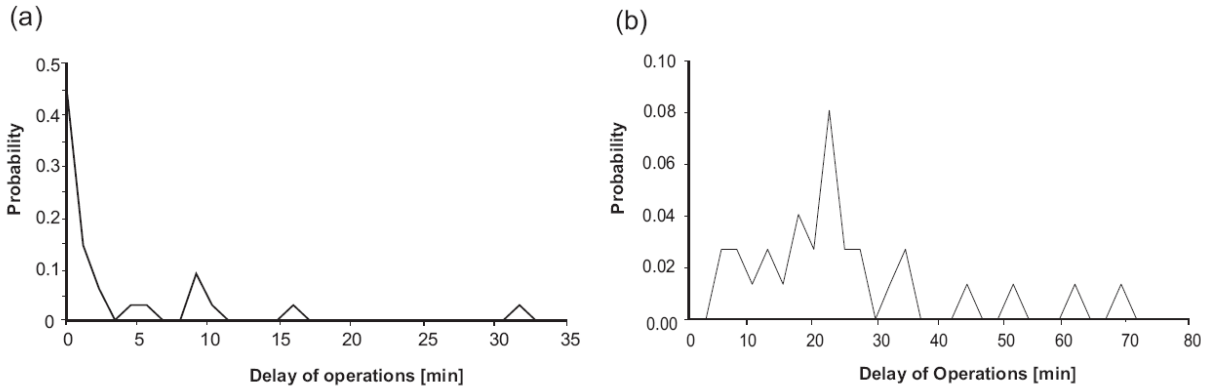


Figure 5.6 PDFs of WFI starvation of the stochastic simulations of the (a) 4th Increase" and (b) "7th Increase"

5.3.1 Sensitivity Experiments

Three sensitivity experiments are performed on the stochastic model to find parameters having a significant influence on the results. The first experiment tests the sensitivity of dispensed volume uncertainty on the results of the simulation. The second experiment tests the sensitivity the type and shape of the statistical distribution have on the results of the DI/WFI simulation. In a third experiment the influence of varying the maximum schedule uncertainty is investigated.

The **first** sensitivity experiment shows that the influence of dispensed volume uncertainty is negligible. In the "Original Data Set" the accuracy of dispensed WFI of automated tasks is, in line with situations in practice, set to be ±1% of full volume, while for manual tasks a maximum variation of ±10% of full volume was assumed. The sensitivity of variations in volume uncertainty on the results can best be assessed by Table 5.7 showing a comparison between the "Original Data Set" and two repeat experiments with all volume uncertainty inputs set to either ±0% or ±10 %. From this table it would appear that the influence of volume uncertainty is small. Furthermore Figure 5.7 visually shows that there is little difference between the results of the simulation with ±0% or ±10 % uncertainties in dispensed WFI volume.

Of course setting the volume uncertainty higher, say ±30% of full volume, may yield results, whose variation may not be deemed negligible as shown in Table 5.8. In this table one large

variation in the order of 100% between the "Original Data Set" and the "30% Volume Uncertainty" is highlighted. The graphical outputs given in Figure 5.8 also display clear differences indicating that volume uncertainties should at least for the present system not be larger than ±30% in order to be negligible. But, generally speaking, a volumetric uncertainty of ±30% would violate the accuracy requirements of pharmaceutical formulations and is hence not realistic.

Name of Increase	**Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate – Stochastic Model – 30 MC Runs [Min]**								
	Original Data Set			**Volume Uncertainty set to 0%**			**Volume Uncertainty set to 10%**		
	Max.	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**
Existing Process	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4th Increase	31.0	0.0	3.3	31.0	0.0	3.3	27.0	0.0	3.3
5th Increase	11.0	0.0	3.2	10.0	0.0	2.9	12.0	0.0	3.6
6th Increase	54.0	0.0	15.4	54.0	0.0	14.2	54.0	0.0	15.8
7th Increase	68.0	3.0	23.1	58.0	2.0	22.1	67.0	2.0	24.6

Table 5.7 Comparison of the results of the numerical measure of the stochastic simulations of the "Original Data Set", "0% Volume Uncertainty" and "10% Volume Uncertainty"

Name of Increase	Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate – Stochastic Model – 30 MC Runs [Min]					
	Original Data Set			Volume Uncertainty set to 30%		
	Max.	Min.	Ave.	Max.	Min.	Ave.
Existing Process	0.0	0.0	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0	0.0	0.0
4th Increase	31.0	0.0	3.3	29.0	0.0	3.9
5th Increase	11.0	0.0	3.2	21.0	0.0	5.0
6th Increase	54.0	0.0	15.4	42.0	0.0	17.2
7th Increase	68.0	3.0	23.1	82.0	2.0	27.3

Table 5.8 Comparison of the results of the numerical measure of the stochastic simulations of the "Original Data Set" and "30% Volume Uncertainty"

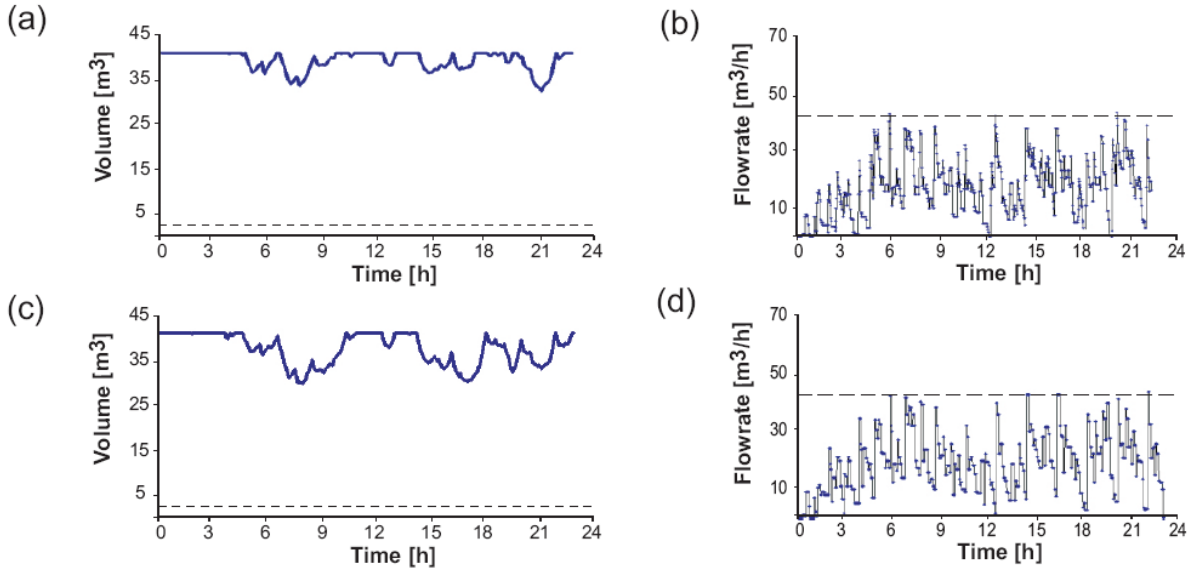


Figure 5.7 Comparison of the graphical results of the stochastic simulation: "5th Increase", 1st MC Run, (a & b) "0% Dispensed Volume Uncertainty" and (c & d) "10% Dispensed Volume Uncertainty"

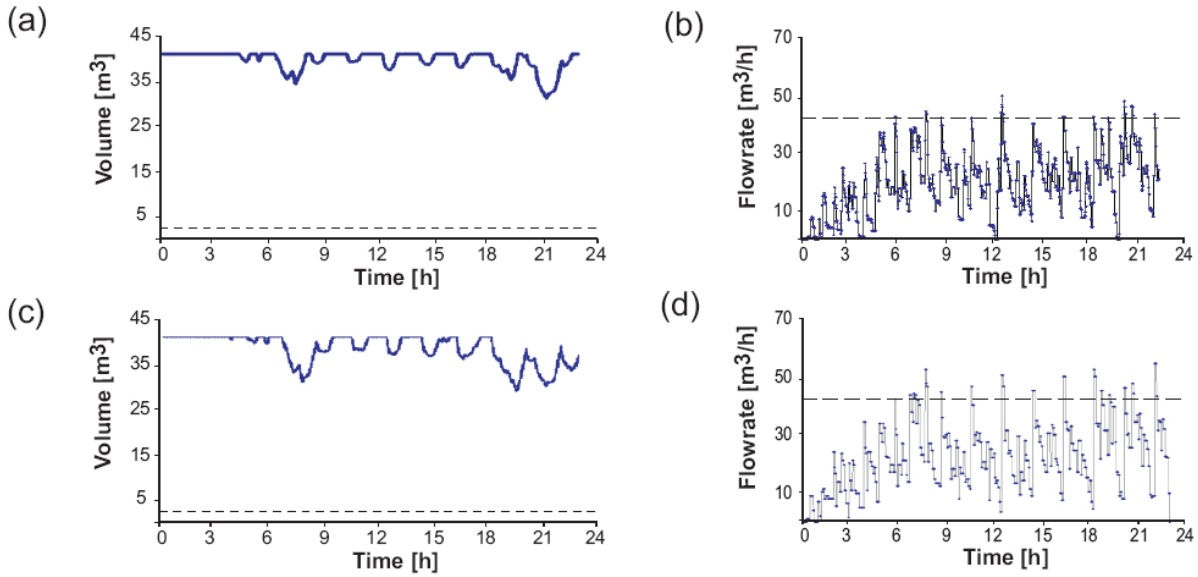


Figure 5.8 Comparison of the graphical results of the stochastic simulation: "7th Increase", 1st MC Run, (a & b) "0% Dispensed Volume Uncertainty" and (c & d) "30% Volume Uncertainty"

The **second** sensitivity experiment tests the influence of varying the type and shape of the stochastic distribution used on the results. Replacing all the stochastic distributions in the "Original Data Set" with the uniform distribution and running the simulation gives the results as shown in Table 5.9 and Figure 5.9. The results show a relatively small influence on the results of the simulation should all distributions of the "Original Data Set" be replaced by the uniform distribution.

Name of Increase	**Deterministic Model**	**Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate – Stochastic Model – 30 MC Runs [Min]**					
		Original Data Set			**Uniform Distribution Only**		
		Max.	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**
Existing Process	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4th Increase	0.0	31.0	0.0	3.3	27.0	0.0	2.7
5th Increase	25.0	11.0	0.0	3.2	14.0	0.0	3.7
6th Increase	70.0	54.0	0.0	15.4	52.0	0.0	15.3
7th Increase	110.0	68.0	3.0	23.1	77.0	6.0	24.6

Table 5.9 Comparison of the results of the numerical measure of the stochastic simulations "Original Data Set" and "Uniform Distribution Only" and the results of the deterministic simulation

The results of the second sensitivity experiment, experimenting with the type of distribution, appear to show that the influence of the type of distribution on the results of a simulation of a WFI distribution system may be small. Consequently, for the WFI system investigated, estimating the statistical distributions from very little information may be acceptable.

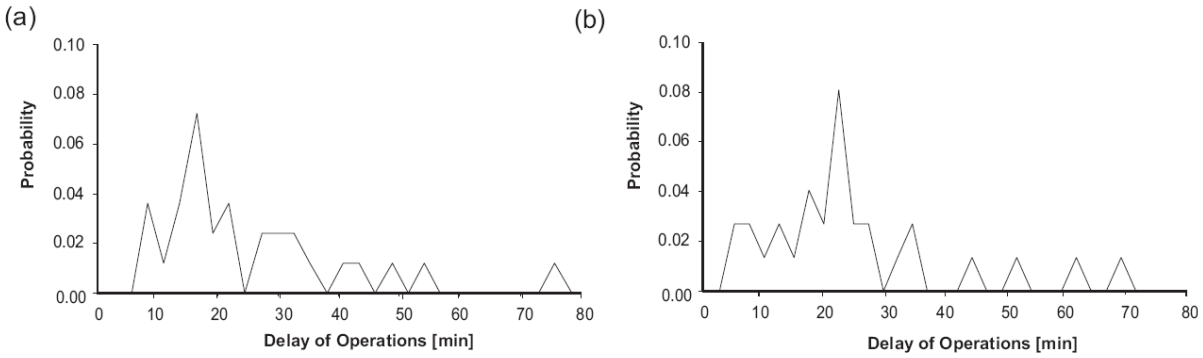


Figure 5.9 Comparison of the PDFs of WFI starvation of the stochastic simulations (a) "Uniform Distribution" and (b) "Original Data Set" of the "7th Increase"

The shape of the distribution may also have an influence on the outcome of a simulation experiment. This is shown in the following experiment utilising two different Normal distributions one with a standard variation ($\sigma = 0.05$) and one with a larger standard deviation of ($\sigma = 0.2$) as graphically shown in Figure 5.10. As may be expected the more central the distribution, the more pronounced its influence on the results (here: minutes per day the WFI demand exceeds the WFI pump flowrate) will be, as the amount of variation on the opening times of the valves allowed is reduced (see Table 5.10 and 5.11). Generally the results display a rise in the values of the maxima and average figures from the "Original Data Set" to the "Normal Distribution Only ($\mu = 0.5$ and $\sigma = 0.2$)" to the values of the "Normal Distribution Only ($\mu = 0.5$ and $\sigma = 0.05$)". Similar results, although not as obvious as the numerical measure, may be observed from the graphical outputs given in Figures 5.10 and 5.11, which demonstrate that the results of the simulations may be difficult to interpret using the graphical outputs only.

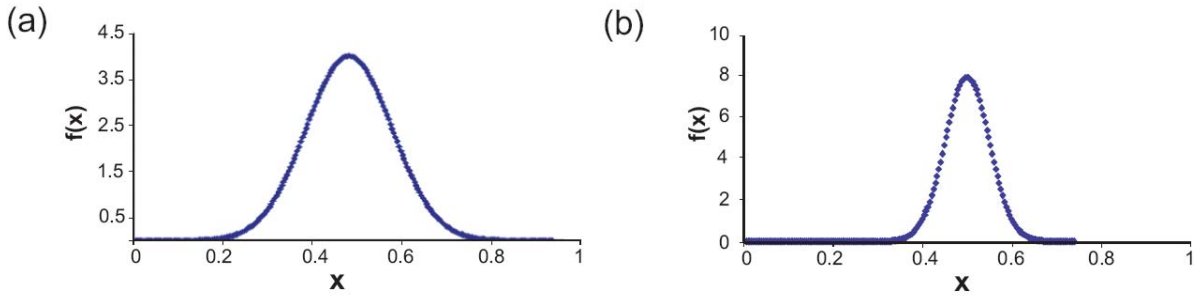


Figure 5.10 Normal distribution with parameters (a) $\mu = 0.5$ and $\sigma = 0.2$ and (b) $\mu = 0.5$ and $\sigma = 0.05$

Name of Increase	Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate – Stochastic Model – 30 MC Runs [Min]								
	Original Data Set			Normal Distribution Only ($\mu = 0.5$; $\sigma = 0.2$)			Normal Distribution Only ($\mu = 0.5$; $\sigma = 0.05$)		
	Max.	Min.	Ave.	Max.	Min.	Ave.	Max.	Min.	Ave.
Existing Process	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4th Increase	31.0	0.0	3.3	21.0	0.0	3.1	18.0	0.0	4.3
5th Increase	11.0	0.0	3.2	18.0	0.0	4.5	36.0	0.0	8.7
6th Increase	54.0	0.0	15.4	57.0	0.0	18.1	59.0	11.0	30.8
7th Increase	68.0	3.0	23.1	62.0	0.0	29.6	92.0	6.0	43.0

Table 5.10 Comparison of the results of the numerical measure of three stochastic simulations: "Original Data Set" , "Normal Distribution Only $(\mu = 0.5 \text{ and } \sigma = 0.2)$" and "Normal Distribution Only $(\mu = 0.5 \text{ and } \sigma = 0.05)$"

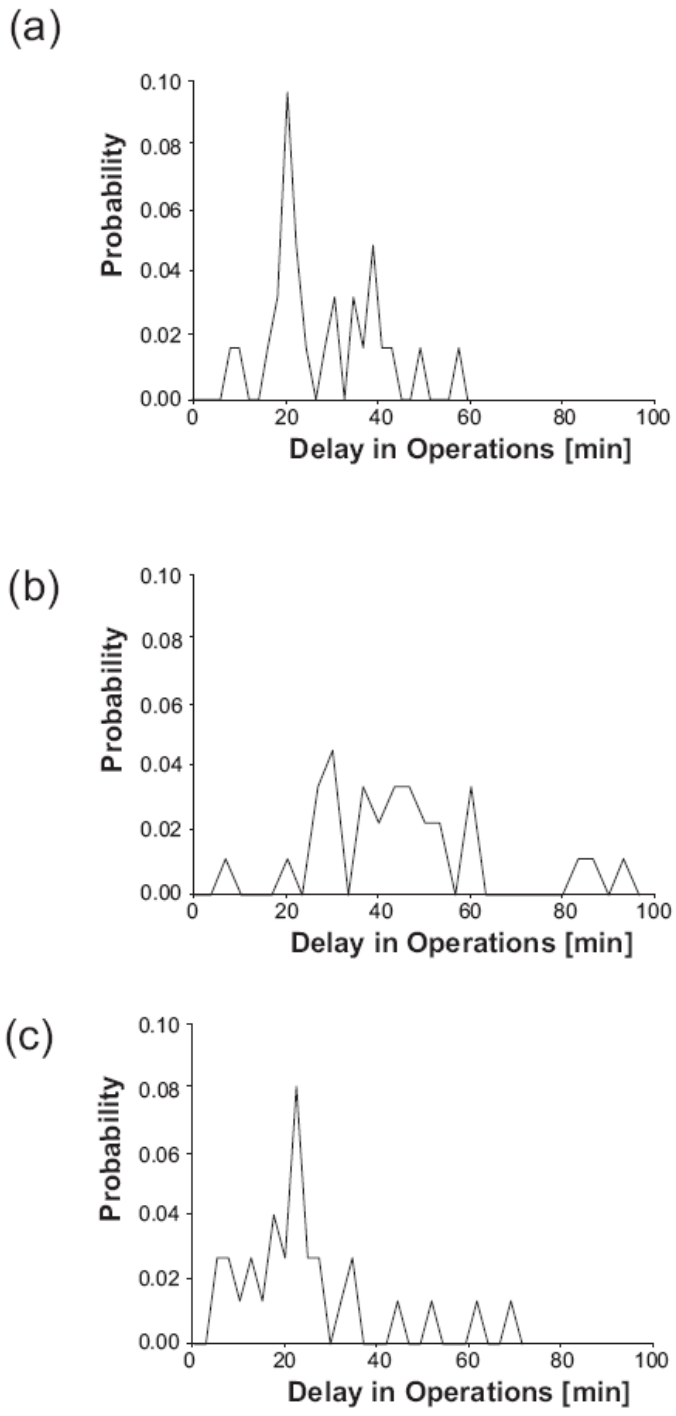


Figure 5.11 Comparison of the PDFs of three different stochastic simulations of the "7th Increase": (a) "Normal Distribution Only $(\mu = 0.5 \text{ and } \sigma = 0.2)$", (b) "Normal Distribution Only $(\mu = 0.5 \text{ and } \sigma = 0.05)$" and (c) "Original Data Set"

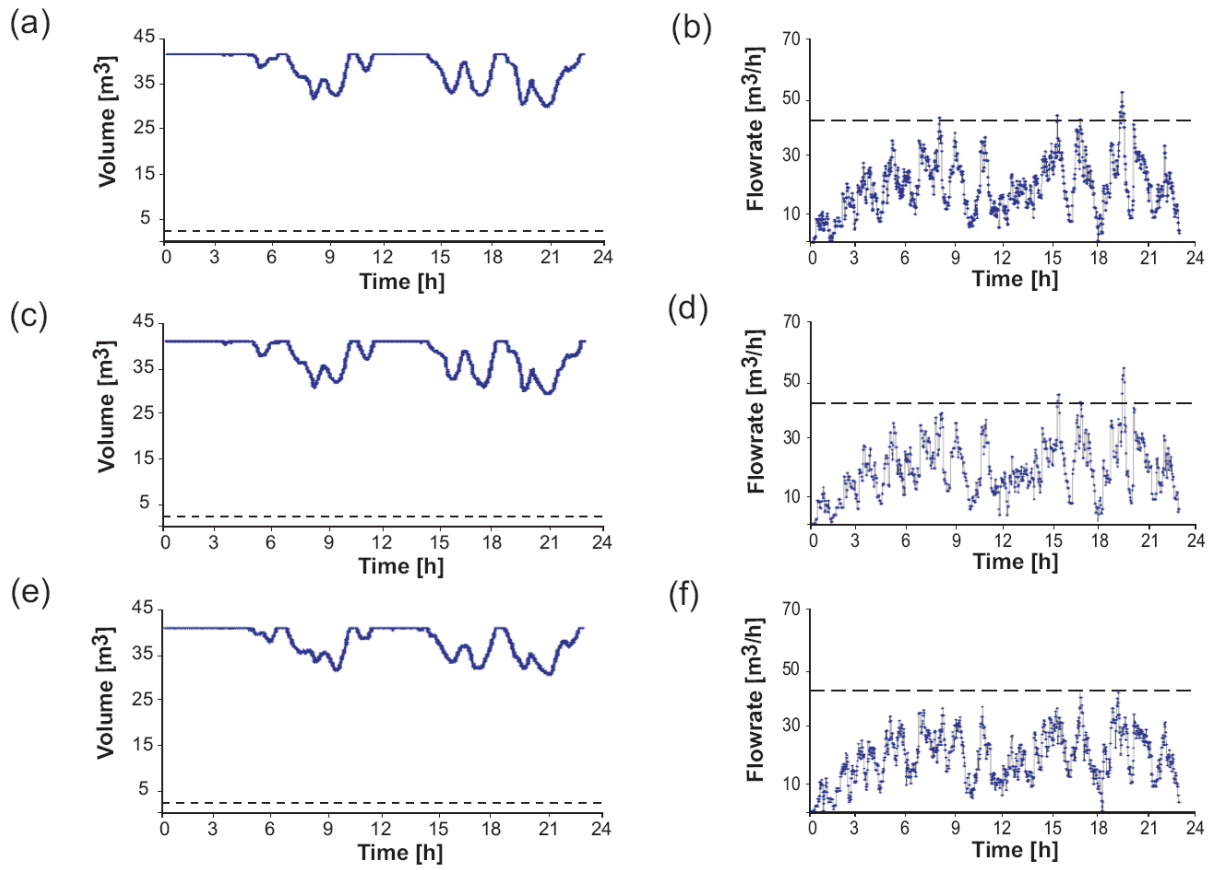


Figure 5.12 Graphical results of the stochastic simulation "5th Increase", 1st MC Run: (a & b) "Normal Distribution Only $(\mu = 0.5 \text{ and } \sigma = 0.2)$ ", (c & d) "Normal Distribution Only $(\mu = 0.5 \text{ and } \sigma = 0.05)$ " and (e & f) "Original Data Set"

In the **third** sensitivity experiment, the maximum uncertainty for each $act_{i,k}$ was first set to 10 minutes and then repeated with nearly all $act_{i,k}$ set to 30 minutes. Not every $act_{i,k}$ could be set to 30 minutes, because of precedence conditions (see Section 3.1.), requiring a small number of $act_{i,k}$ to remain at 10 minutes. This sensitivity experiment shows an expected large influence of schedule uncertainty on the simulation results as is evident from Figure 5.13.

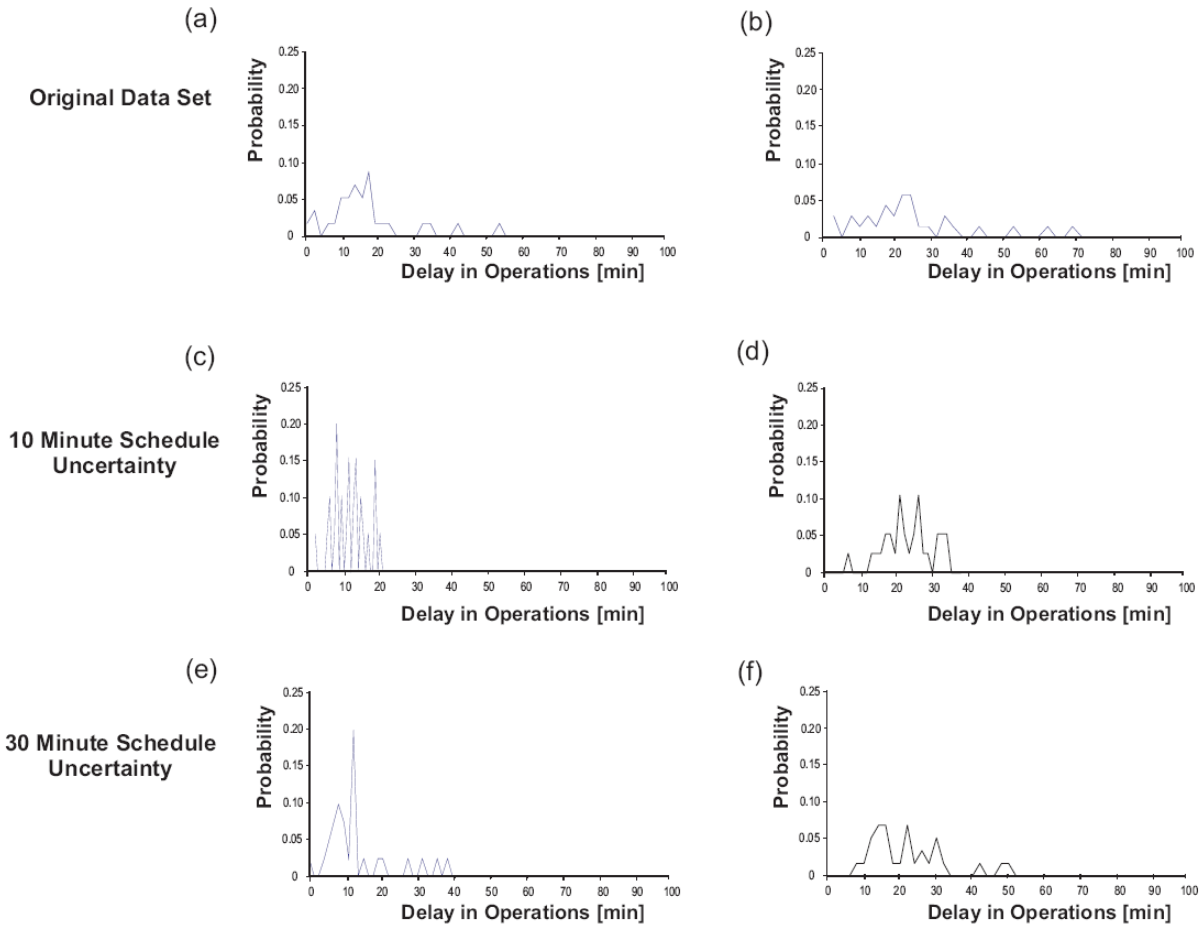


Figure 5.13 Comparison of the PDFs of WFI starvation of three different stochastic simulations: (a & b) "Original Data Set", (c & d) "10 Minutes Schedule Uncertainty" and (e & f) "30 Minutes Schedule Uncertainty" for the "6th Increase" (left) and the "7th Increase" (right)

5.3.2 A Note on the Number of Monte Carlo Runs Executed

This subsection explains the reasoning for deeming a seemingly low number of 30 MC runs sufficient for the stochastic simulations. As can be seen from Table 5.11 top table, should only 10 MC runs be executed, a number of inconsistencies in the results arise. For instance, stepping from the "4th Increase" to the "5th Increase" ("Original Data Set", maximum and average) should result in an increase and not in a decrease in the time the WFI distribution pump flowrate is exceeded. Another inconsistency is evident in the results of the "6th Increase". The maximum result for the "Original Data Set" and "Uniform Distribution Only" is 22, whereas for the "Normal Distribution Only" the result is 49. Such a variation or lack thereof in the results is deemed to be unacceptable. But if 30 MC runs are executed, the results are 54, 52 and 59 (see Table 5.11 bottom table), which is deemed to be within the required accuracy. It must, however, be emphasised that increasing the number of MC runs to 30 does not remove all data inconsistencies, but it reduces the impact of them. For example, the results of the "Original Data Set" do still display an inconsistency between the "4th Increase" and the "5th Increase" as can be seen from Table 5.11 bottom table. But for quantitative comparisons as applied in this thesis the data appears to be sufficiently consistent. It is possible to increase the number of MC runs to 50, maybe to 100. However there are three potential limitations to doing this. Firstly, the simulation time increases with the number of simulations. Secondly, the random number generator used for this work may not be able to support the increased number of simulations (see Appendix 4) requiring its substitution with a more modern one [165]. And finally the Excel file size may get excessively large as all data of the simulations are stored on the spreadsheet.

From Figure 5.14 comparing the plots of the PDF's for the "5th Increase" and "7th Increase" for both 10 and 30 MC runs, it is further evident that 10 MC runs lack resolution compared to 30 MC runs.

Name of Increase	Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate – Stochastic Model – **10 MC Runs [Min]**									
	Deterministic Model	**Original Data Set**			**Normal Distribution Only (μ = 0.5; σ = 0.2)**			**Normal Distribution Only (μ = 0.5; σ = 0.05)**		
		Max.	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**
Existing Process	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4th Increase	0.0	15.0	0.0	2.4	10.0	0.0	3.3	15.0	0.0	4.4
5th Increase	25.0	7.0	0.0	1.3	14.0	0.0	4.7	25.0	0.0	8.4
6th Increase	70.0	22.0	1.0	11.1	22.0	0.0	9.8	49.0	9.0	23.6
7th Increase	110.0	53.0	3.0	21.4	77.0	12.0	32.0	80.0	16.0	39.2

Name of Increase	Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate – Stochastic Model – **30 MC Runs [Min]**									
	Deterministic Model	**Original Data Set**			**Normal Distribution Only (μ = 0.5; σ = 0.2)**			**Normal Distribution Only (μ = 0.5; σ = 0.05)**		
		Max.	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**	**Max.**	**Min.**	**Ave.**
Existing Process	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4th Increase	0.0	31.0	0.0	3.9	27.0	0.0	2.7	15.0	0.0	2.9
5th Increase	25.0	11.0	0.0	3.2	14.0	0.0	3.7	25.0	0.0	7.3
6th Increase	70.0	54.0	0.0	10.2	52.0	0.0	15.3	58.0	5.0	23.6
7th Increase	110.0	68.0	3.0	23.1	77.0	6.0	24.1	80.0	10.0	36.2

Table 5.11 On the stability of the results of the stochastic simulation; 10 MC runs (top table) and 30 MC runs (bottom table)

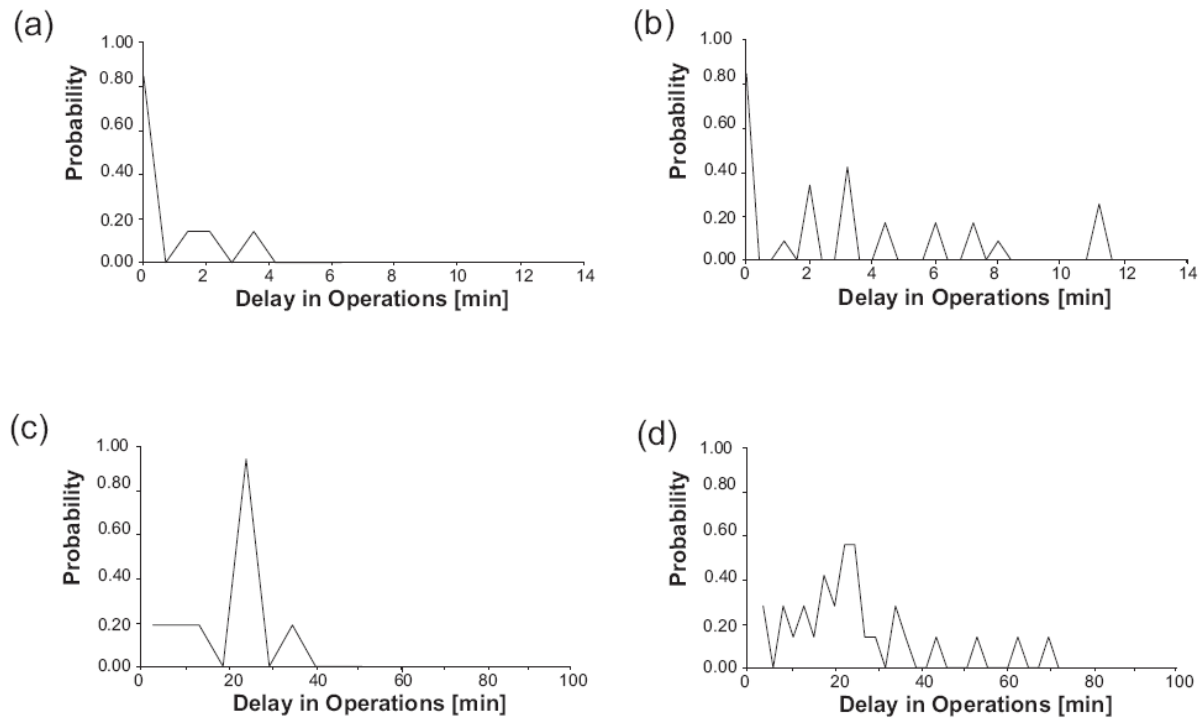


Figure 5.14 Comparison of the PDFs of WFI starvation of the stochastic results of the (a & b) "5th Increase" and (c & d) the "7th Increase" for (a & c) 10 MC runs and (b & d) 30 MC runs

In Table 5.11 an anomaly with the variation of the distributions is apparent, which is related to the low number of MC runs executed. In the table the variance of the distributions is first increased from left to right and then for the final column on the right decreased again. Therefore the Max values for the "Original Data Set" and the "Normal Distribution Only ($\mu = 0.5$; $\sigma = 0.2$)" should be higher than compared to the deterministic case. Moreover the Max values for the "Normal Distribution Only ($\mu = 0.5$; $\sigma = 0.2$)" should be higher than for "Normal Distribution Only ($\mu = 0.5$; $\sigma = 0.05$)", This is not generally the case as the number of MC runs are too low. But as can be observed by comparing the Max Values of the 10 MC run with the 30 MC run, it diminishes with an increasing number of executed MC runs. Hence increasing the number of MC runs should eliminate this anomaly. However for this work no attempt was made to find how many MC runs are required to eliminate this anomaly.

5.4 Results of the Fuzzy-Logic Simulations

According to Ross [53] and Zadeh [128] FL is a logic for "approximate reasoning". In other words, the uncertainties involved in a FL simulation may be large. Because of that volume uncertainty is not included in the FL model (see Section 3.3). This decision was subsequently shown to be correct by the sensitivity analysis of the stochastic model, which illustrate that volume uncertainty is negligible for the process investigated.

Statistical Measure	Existing Process	Increase						
		1st	2nd	3rd	4th	5th	6th	7th
Min. Level in Storage Tank [m^3]	38.94	34.19	33.56	27.04	30.54	32.54	34.14	31.29
Max. Level in Storage Tank [m^3]	41.09	41.09	41.09	41.09	41.09	41.09	41.09	41.09
Variation of Level in Storage Tank [m^3]	5.00	8.78	15.61	22.31	17.69	13.33	6.65	8.87
Average Level in Storage Tank [m^3]	40.95	40.29	39.88	38.61	38.48	39.42	40.19	39.72
Average Offtake from WFI Distribution System [m^3/h]	4.23	5.40	9.46	10.60	15.91	16.87	19.39	20.31
Average Water flow from Generation System [m^3/h]	4.23	5.41	9.46	10.60	15.91	16.88	19.40	20.42
90% of all levels in Tank are above this level [m^3]	41.09	41.09	41.09	41.09	41.09	41.09	41.09	41.09
60% of all levels in Tank are above this level [m^3]	41.09	41.09	41.09	41.06	40.93	41.09	41.09	41.09
WFI Generation Equipment Utilisation per day [-]	84%	86%	95%	96%	97%	97%	97%	97%

Table 5.12 Statistical point measures of the fuzzy simulation for the "Existing Process" and the "1st Increase" to the "7th Increase" ($act_{i,k}^{Wait} = 0$)

An advantage of FL over stochastic methods noted by Zhang et al. [84] and Perrone at al. [67], is that one run of a fuzzy simulation contains all the statistical information of multiple runs of a statistical model (see Table 5.12). And indeed the statistical point measures of the deterministic, stochastic and FL simulation are quite similar as can be seen from Table 5.13. But statistical point measures may not provide useful information in assessing the suitability of an existing WFI system to cope with demand increases. Therefore no conclusion as to the most appropriate method of simulating a DI/WFI systems can be derived from the comparison of the statistical point measures of the various methods.

Statistical Measure \ Description of Input Data Set	5th Increase			6th Increase			7th Increase		
	Deterministic	Stochastic	FL	Deterministic	Stochastic	FL	Deterministic	Stochastic	FL
Min. Level in Storage Tank [m^3]	29.92	30.84	32.54	31.41	27.84	34.14	29.31	32.30	31.29
Average Level in Storage Tank [m^3]	37.55	38.17	39.42	39.32	39.21	40.19	38.50	39.72	39.72
Average Offtake from WFI Distribution System [m^3/h]	19.96	19.83	16.87	22.72	22.79	19.39	23.71	23.68	20.31
Average Water flow from Generation System [m^3/h]	19.01	18.89	16.88	21.72	21.75	19.40	22.74	22.54	20.42
90% of all levels in Tank are above this level [m^3]	41.09	41.09	41.09	41.09	41.09	41.09	41.09	41.09	41.09
60% of all levels in Tank are above this level [m^3]	39.64	40.95	41.09	41.09	41.09	41.09	41.09	41.09	41.09
WFI Generation Equipment Utilisation per day [-]	0.98	0.99	0.97	0.98	0.99	0.97	0.98	0.99	0.97

Table 5.13 Comparison of the statistical point measures of the deterministic, stochastic and FL $(act_{i,k}^{Wait} = 0)$ simulations for the "5th Increase", "6th Increase" and the "7th Increase"

The FL model includes the number of operators as an input parameter. For the FL results given in Table 5.12 and 5.13 the number of operators are set at the required minimum number of operators denoted $n_{op,\,min}$ so that no operations are left waiting (denoted $act_{i,k}^{Wait} = 0$) at the end of the simulation timeframe.

Utilising the numerical measure as already applied presenting the results of the deterministic and stochastic simulations may provide a better assessment of the suitability of FL for modelling uncertainty in DI/WFI. In Table 5.14 a comparison of the results of the deterministic, stochastic and FL simulations is provided. Because the FL simulation does not use random numbers, its results are deterministic. The FL simulation predicts WFI starvation from the "6th Increase" onwards. Therefore the results of a FL may or may not be as conservative as the results of a deterministic or stochastic simulation.

Name of Increase	Cumulative Times the WFI Demand exceeds the WFI Pump Flowrate Capacity [Min]				
	Deterministic Model	Stochastic Model 30 MC Runs			FL Model
		Max.	Min.	Ave.	$(act_{i,k}^{Wait} = 0)$
Existing Process	0.0	0.0	0.0	0.0	0.0
1st Increase	0.0	0.0	0.0	0.0	0.0
2nd Increase	0.0	0.0	0.0	0.0	0.0
3rd Increase	0.0	0.0	0.0	0.0	0.0
4th Increase	0.0	31.0	0.0	3.3	0.0
5th Increase	25.0	11.0	0.0	3.2	0.0
6th Increase	70.0	54.0	0.0	15.4	10.8
7th Increase	110.0	68.0	3.0	23.1	40.5

Table 5.14 Comparison of the results of the numerical measure of the FL with the deterministic and stochastic simulations

The results of the FL simulations do not show day-to-day variations, which is a result of FL being deterministic. But as can be seen from Figures 5.15 to 5.17 the graphical results of the FL simulations are somewhat different to the results of the deterministic simulations and may give additional insights into the behaviour of the system.

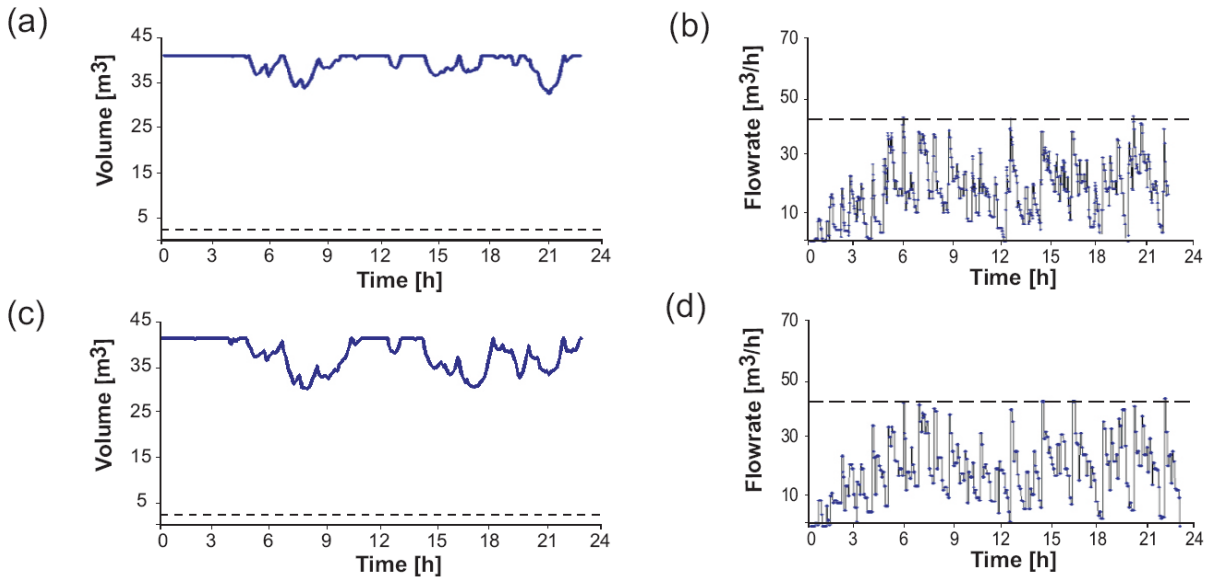


Figure 5.15 Comparison of the graphical results of the (a & b) FL simulation "5th Increase" with $n_{op,\,min} = 33$ with the results (c & d) of the deterministic simulation

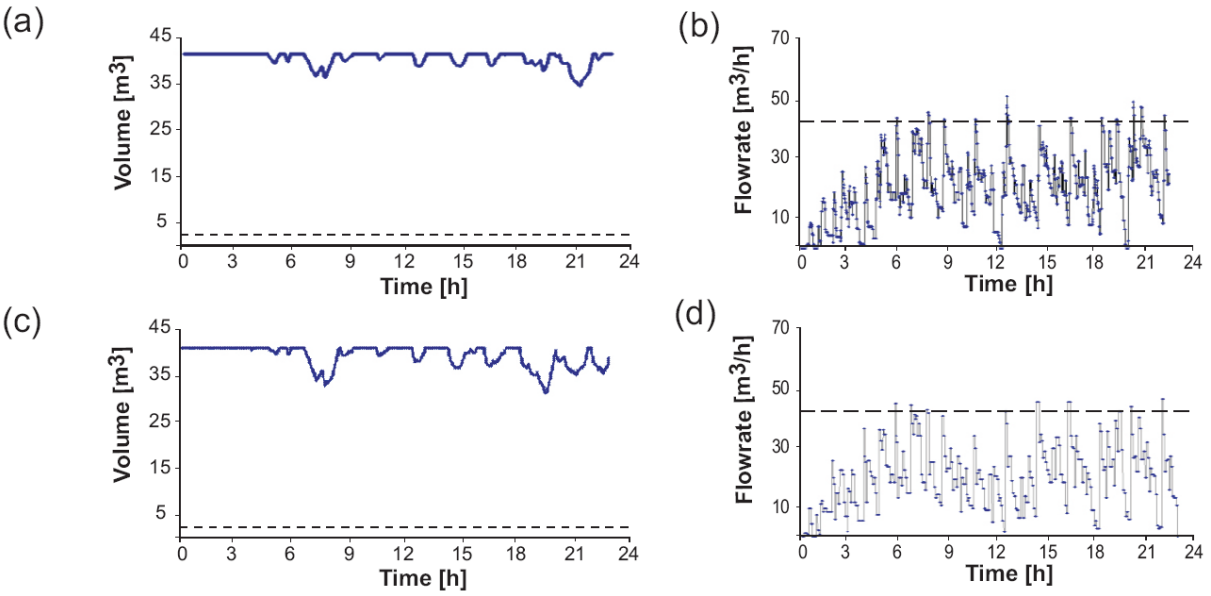


Figure 5.16 Comparison of the graphical results (a & b) of the FL simulation "6th Increase" with $n_{op,min} = 35$ with the results (c & d) of the deterministic simulation

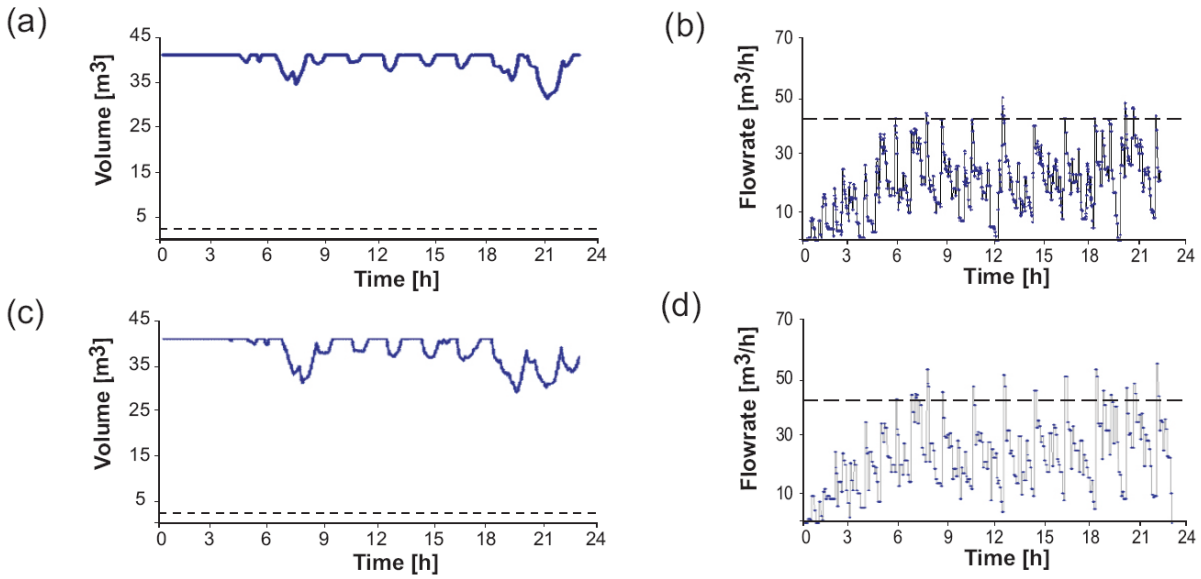


Figure 5.17 Comparison of the graphical results (a & b) of the FL simulation "7th Increase" with $n_{op,min} = 38$ with the results (c & d) of the deterministic simulation

5.4.1 Sensitivity Experiments

Three different sensitivity experiments were performed on the FL model to test the influence variations of the input variables have on the outcome of the FL simulation:

1. In the first sensitivity experiment all membership functions "Duration of Tasks" $\tilde{\mu}_D^{i,k}(t)$ are first set to 10 minutes ($t_{i,k}^{Delay,1} = 2$ min, $t_{i,k}^{min,D} = 6$ min, $t_{i,k}^{Delay,2} = 2$ min) and then repeated with all of them set to 30 minutes ($t_{i,k}^{Delay,1} = 5$ min, $t_{i,k}^{min,D} = 20$ min, $t_{i,k}^{Delay,2} = 5$ min).
2. The influence of the number of operators n_{op} on the FL model results is investigated in the second set of experiments.
3. In the third experiment the time delays $t_{i,k}^{Delay,1}$ and $t_{i,k}^{Delay,2}$ of all membership functions "Duration of Tasks" $\tilde{\mu}_D^{i,k}(t)$ are set to zero.

The **first** set of FL sensitivity experiments demonstrates an expected large difference between the results of the 10 min and 30 min schedule uncertainty (see Figure 5.18). Moreover as expected the required minimum number of operators is larger for the "30 Minutes Schedule Uncertainty" than the "10 Minutes Schedule Uncertainty" experiment.

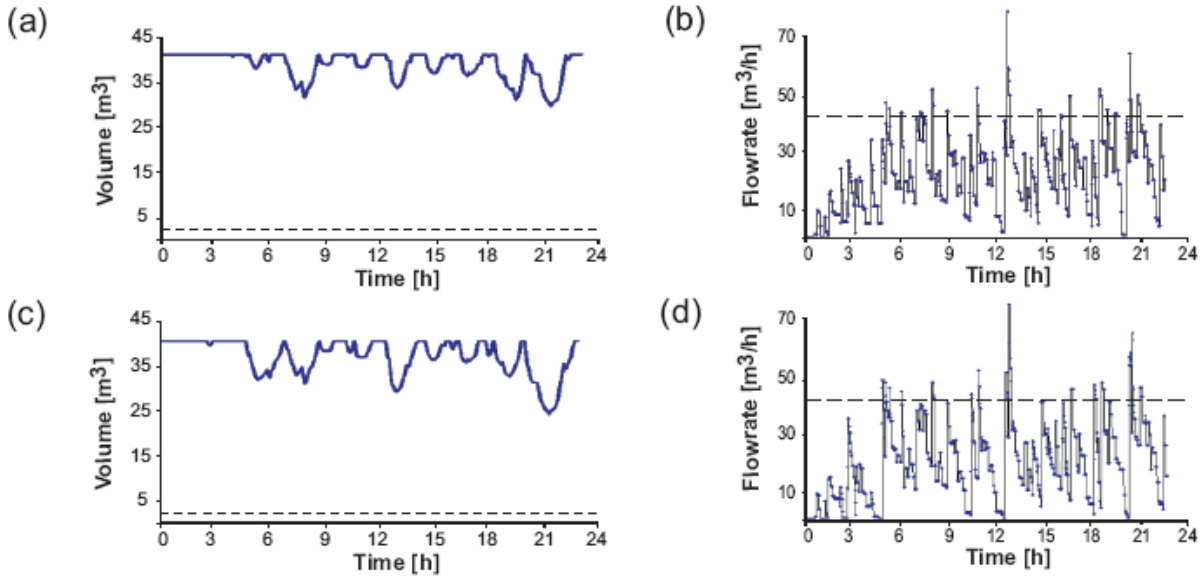


Figure 5.18 Graphical results of the fuzzy simulation "7th Increase" (a & b) "10 Minutes Schedule Uncertainty" for each $act_{i,k}$ and $n_{op,min} = 42$; (c & d) "30 Minutes Schedule Uncertainty" for each $act_{i,k}$ and $n_{op,min} = 59$

The **second** sensitivity experiments tests the influence the number of operators n_{op} has on the simulation results of the FL model. Unsurprisingly, the required minimum number of operators, $n_{op,min}$ rises in line with the increases as shown in Figure 5.19.

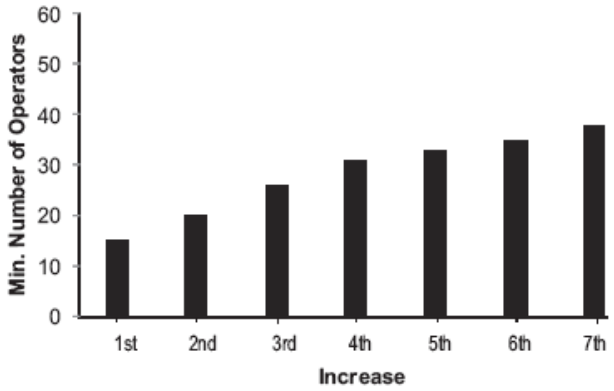


Figure 5.19 Minimum number of operators $n_{op,min}$ over the Increases

Should the number of operators be below the minimum required number of operators $n_{op,min}$, then the WFI level in the storage tank also depends on the numbers of operators. This is shown in Figure 5.20 and 5.21 for the "5th Increase" with the numbers of operators set to $n_{op} = 30$ and $n_{op} = 27$ respectively.

Including the number of operators in a model of a utility system violates the principle that the process operations rather than the utility operations should determine the man-power requirements of a facility. After all a utility should support a process system and not vice versa. Consequently including the number of operators as is done in the FL model of a WFI system is questionable. But at a minimum this sensitivity experiment does validate the FL model on a qualitative basis.

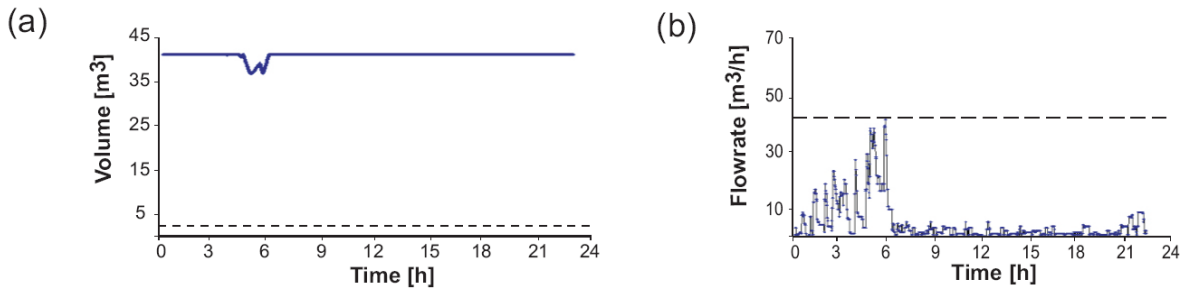


Figure 5.20 Graphical result of the fuzzy simulation "5th Increase" with $n_{op} = 30$ resulting in $act_{i,k}^{Wait} = 445$

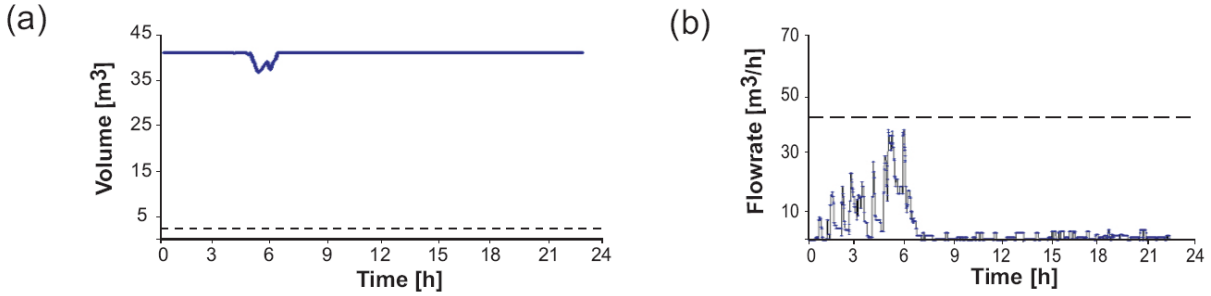


Figure 5.21 Graphical result of the fuzzy simulation "5th Increase" with $n_{op} = 27$ resulting in $act_{i,k}^{Wait} = 475$

In the **third** experiment all time delays $t_{i,k}^{Delay,1}$ and $t_{i,k}^{Delay,2}$ of the membership functions "Duration of Tasks" $\mu_{\tilde{D}}^{i,k}(t)$ are set to zero. This removes all operator task uncertainties from the FL simulation and may lead to a deterministic simulation. However, the rule base, for example that during break times operators do not start any processes (see Section 3.3.1.8), still applies. And indeed comparing the predictions of the behaviour of the WFI level in the storage tank of the deterministic and the FL simulations shows differences between the two simulations as can be seen from Figure 5.22. Therefore setting all time delays $t_{i,k}^{Delay,1}$ and $t_{i,k}^{Delay,2}$ of the FL model to zero does not yield the same result as the deterministic simulation. But, of course, deterministic input conditions should be simulated with the deterministic model and not the FL model. The net effect is similar to the second FL sensitivity experiment of having an experiment with no real practical value apart from validating the FL model on a qualitative basis.

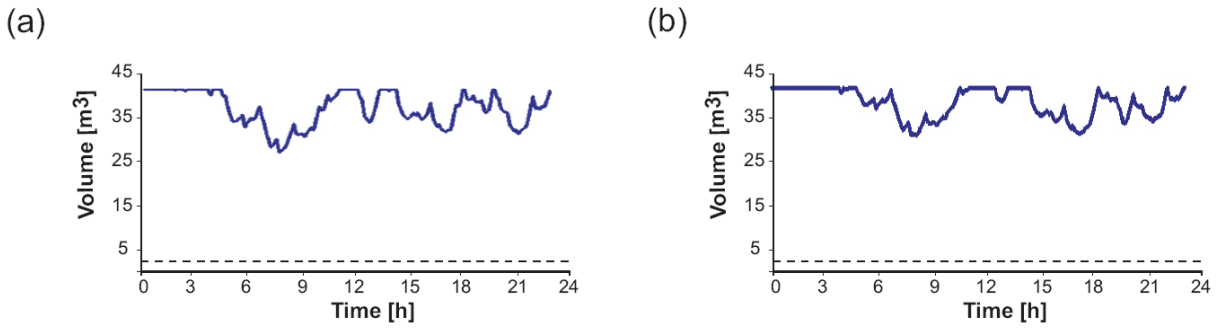


Figure 5.22 Comparison of the graphical results water level in the storage tank of the (a) fuzzy logic and the (b) deterministic simulation of the "5th Increase"
All time delays $t_{i,k}^{Delay,1}$ and $t_{i,k}^{Delay,2}$ of the membership functions "Duration of Tasks" are set to zero with $n_{op,min} = 31$.

5.5 Computation Times

Perrone et al. [67] and Zhang et al. [84] suggest that an advantage of FL over stochastic methods is their reduced computational time, because a FL simulation requires only one run whereas a stochastic simulation requires multiple runs. Clearly the stochastic simulation requires a significantly longer computation time than the FL simulation as is evident from Table 5.15, which provides a comparison of the deterministic, stochastic and FL simulation for the "7^{th} Increase". But the computation times have to be seen in context. A WFI capacity expansion project will not be executed very often. Therefore simulation times of ten hours or even longer would not pose a particular problem and hence computation times do not provide a basis for selecting FL over stochastic or deterministic methods.

Description of Simulation Run	**Computation Time**
Deterministic Model, 7^{th} Increase	5 m, 31 s
FL Model, 7^{th} Increase	9 m, 19 s
Stochastic Model, 30 MC Simulations, 7^{th} Increase	3 h, 10 m

Table 5.15. Comparison of the computation times of the deterministic, stochastic and FL simulations These computation times were obtained by executing the simulations on a Lenovo laptop computer running under Microsoft Windows XP equipped with an Intel x86, Family 6 Model 15, Stepping 6 processor and a memory capacity of 1,038,700 kilobyte.

5.6 Chapter Summary

This chapter presents the results of the various simulations. As only one WFI system is investigated, generalisations of the results should be avoided. However, one result can be generalised: Uncertainty modelling in DI/WFI systems should be executed with deterministic or stochastic methods rather than FL methods where there is a cost/time imperative. This will be further discussed in the next chapter.

6 Discussion and Future Work

This work investigates which method is most appropriate to model uncertainty in industrial scale DI/WFI systems. Three popular methods: deterministic, stochastic and FL are considered in detail. Of these three only the deterministic method has been applied in practice [21, 23], but the deterministic method does not directly deal with uncertainty, it ignores it. The stochastic method, which is widely accepted method of describing uncertainty and has been used in many fields [25, 33, 42]. FL simulations are often benchmarked against stochastic methods, as is pointed out by Chen and Lin [88]. This leaves FL as a relatively untested method, requiring a justification if used. Because this justification is difficult to make, this thesis concludes that either deterministic or stochastic methods should be used to model uncertainty in the context of DI/WFI systems expansion projects. In the following pages the reasons for this conclusion will be further discussed under five headings:

1. FL and Stochastic Methods are Fundamentally Different
2. FL may be more Complex Compared to a Stochastic Model
3. FL is a Deterministic Method
4. FL may be Difficult to “Sell” to an End-User
5. Stochastic Methods may be Superior to the Methods of FL

1. FL and Stochastic Methods are Fundamentally Different

The stochastic model simulates uncertainty of the DI/WFI offtake valves opening and closing times based on estimated (subjective) or historical (objective) information utilising statistical distributions. In contrast, the FL model simulates the uncertainty of the DI/WFI offtake valves opening and closing times based on estimated operator behaviour, number of operators and other influences. Therefore the modelling philosophies of the stochastic and FL models are fundamentally different. As already stated in the previous chapter, including the number of operators and their behaviour in a model of a utility system is questionable, as the process and not its utility system should determine the number of operators in a production facility. As a result, this part of the FL model of a DI/WFI system is problematic from a standpoint of modelling philosophy.

2. FL may be more Complex Compared to a Stochastic Model

Applying the easiest measure of code complexity, that is, lines of computer code, as suggested by Brooks and Tobias [166], the FL model needs about 1,200 lines of code whereas the stochastic model requires about 400 lines of code or three times less. Moreover, the FL model requires more Do-Loops, If-End blocks and calls to subroutines than the stochastic model further adding to code complexity. This reveals a difference between the stochastic and FL models from a programming point of view: Code writing, debugging, code validation and code maintenance of a FL model are more complex compared to a stochastic model. And as Musselman [167] has pointed out, increased complexity makes failure of a simulation project more likely. But these differences in programming length and complexity may simply be related to VBA. Code writing issues would be different and easier should a discrete-event simulation package such as ARENA be used for the FL model.

The payback from a more complex model should be a gain of insight into the behaviour of a system. Unfortunately extreme events as the most valuable result of a DI/WFI simulation are understated by the FL model (see Chapter 5.4, specifically Table 5.14). But it must be said that this may be a result of the inputs to the FL model not being suitable or accurate enough. Therefore, a FL simulation may not give an increased insight into the behaviour of the system compared to the results of a less complex stochastic model.

3. FL is a Deterministic Method

Neither the FL nor the deterministic simulations are able to display day-to-day variations as typical for real production facilities. Furthermore the results of a FL simulation may be similar to the results of a deterministic simulation as is evident from the case study (see Table 5.14 and Figures 5.15 to 5.17). This similarity in the results between the FL and deterministic model is not entirely surprising as in these two models random numbers are not used. But if FL is a deterministic method, the deterministic approach should be used as it has been successfully applied in practice [21, 23].

4. FL Models are not Flexible

The FL model of the DI/WFI system does not include dispensed volume uncertainty, because there appears to be no reasonable method to account for it. The FL model treats schedule uncertainty by attempting to model the behaviour of the operators through their tiredness level and other influences. It appears unreasonable to model volume uncertainty in a similar fashion to schedule uncertainty that is by again modelling the behaviour of the operators as volume uncertainty is primarily a function of the measurement system, which is conveniently and normally treated as a random i.e. stochastic phenomenon [28]. However this may not be a negative score for FL as first FL is "approximate reasoning" only and can therefore, by definition, not model all uncertainties and second volume uncertainty may often be negligible as is, for instance, shown by the case study (see Section 5.3.1).

5. Stochastic Methods may be Superior to the Methods of FL

FL has been put forward as a tool to handle lack of or incomplete information. But from the sensitivity experiments of the stochastic model given in Chapter 5.3.1 it appears that estimating the statistical distributions from limited information is also acceptable as the influence of the distribution on the simulation results may be small, throwing into question the claim of some authors such as Ross [53], Zhang et al. [84] or Nucci and Grieco [85] that FL is better suited than statistical methods in simulating systems if only limited information is available. Indeed statistical methods can and have been used if only limited information is available. As pointed out by Lindley [36], Walley [37] and Laviolette et al. [112] the inputs to stochastic models may

be entirely subjective. Consequently stochastic models can also achieve results, which may have to be labelled "approximate reasoning".

But stochastic methods can do more. Stochastic methods also have the ability to improve their accuracy [38], they can learn so to speak, should more information become available. Therefore a stochastic model can be used should initially only little information be available and re-used should more information become available. FL models, on the other hand, have this ability only up to a point as FL, by definition, is a logic for "approximate reasoning" only.

The proponents of FL [52, 53, 109] agree with the proponents of statistical methods [36, 37, 112] that statistical methods should be used if *enough* information is available. But this consensus breaks down on the question what *enough* information constitutes. Since a definition of *enough* information is not given by either party, the selection between FL or stochastic modelling must be based on personal preference. But the modeller should be aware of a pitfall should the choice be FL: It may be difficult to upgrade a FL model to fit the improved state of information. And it may not be obvious if a FL model is still capable of supporting the improved state of information. Therefore stochastic methods may have an advantage over FL methods, as they may be able to avoid this situation.

In summary, it would appear that stochastic methods are best suited in modelling uncertainty in DI/WFI systems for capacity extension projects. But should a new DI/WFI system be investigated, the FL method may have advantages over stochastic methods, as the FL model gives the designer more parameters to investigate than the stochastic model, such as operator numbers, break times etc. Because FL simulations are faster to execute than stochastic simulations (see Table 5.15) various scenarios including production increases or design changes may be assessed in a short time. Another advantage of a FL model is that the membership functions due to their simplicity can easily be changed and the impact of such variations assessed [84].

The conclusion that stochastic methods should be used to model uncertainty in utility systems can be extended to process systems. Statistical methods should be used in the simulation of any

process or manufacturing system if extreme events or showing day-to-day variations as typical in real production plants are deemed important.

FL may, however, be useful in the simulation of manufacturing systems if the average production output or other statistical point measures rather than extreme events are of interest. For instance, in the investigation by Azzaro-Pantel et al. [83] or Chen and Lin [88], then FL may indeed be an alternative to stochastic methods. Nevertheless, two reasons to be cautious with FL remain. First Chen and Lin [88] state that the application of FL in the simulation of manufacturing systems is still subject to research. And second the results of this research indicate that the simpler deterministic model may practically yield the same statistical point measures as a FL simulation (see Table 5.13). But no definite answer regarding the value of FL in the simulation of manufacturing systems can be given by this research and is left to future research.

Despite the conclusion that stochastic methods may be better suited than FL methods in describing uncertainty in utility and process systems, stochastic methods are not necessarily straightforward. In particular, finding suitable statistical distributions describing the uncertainty may be labour intensive as Law and Kelton [1] caution. Two examples may illustrate this. Wong and Mui [106] interviewed over 500 households to find stochastic distributions which describe the water flushing requirements from residential estates. Jankovic-Nisic et al. [108] collected the water consumption from 33 households over a one week period in order to find appropriate statistical distributions for water demand duration and flowrates for a cluster of houses. Furthermore stochastic methods can be difficult to understand as is acknowledged by Laviolette et al. [112] and Almond [39]. But the author of this work believes FL to be equally, if not more, difficult to understand. For one in FL some theoretical and practical difficulties as outlined in Chapter 2 remain to be solved. For another there is a lack of textbooks on how to apply FL in discrete-event models. For statistical methods, on the other hand, the amount of literature available is substantial, as a visit to any university library may testify.

The entire discussion of modelling uncertainty with stochastic or FL methods is moot should there be no need to model uncertainty. Clearly including uncertainty in a model should result in

a gain of insight into the behaviour of a system. In practice, however, a deterministic model may be entirely suitable as was already shown by Alexander [23] and Saraph [21]. A similar result is obtained from the case study presented in the previous chapter, which utilises the deterministic method to size the capacities of the WFI generation plants and the volumes of the WFI storage tanks and establishes the staggered starting times for all process demand increases. In comparison the insight gained from the stochastic simulations is small only adding that WFI starvation defined as times when the WFI demand exceeds the WFI distribution pump flowrate (see Chapter 5.2) occurs at an earlier stage than predicted by the deterministic simulation. Hence the principle of "as simple as possible but no simpler" [24] must not be overwritten by a desire to include uncertainty or any other features if there is no need to do so.

6.1 Future Work

Other utility systems such as heat transfer media circuits or drinking water systems may also be simulated with the deterministic and stochastic models, once relevant modifications to the program were taken. However, the business incentives to justify a simulation project on these utilities may not exist. Furthermore the performance of ultrapure water systems as employed in the microelectronics industry may be assessed with the models. The models may also be extended to model more complicated DI/WFI distribution systems for example in which two or more storage tanks are placed in different locations along a DI/WFI distribution loop as shown in Figure 6.1.

This work performed no optimisation on the WFI system modelled, but it is possible to include an outer optimisation loop for optimisation purposes. For optimisation of new DI/WFI systems the FL model may turn out to be particularly useful due to the various number of inputs required hence increasing the options of parameters to optimise. Furthermore for optimisation purposes it may be useful to link the DI/WFI models to the process systems.

More comparative work assessing the suitability of stochastic and fuzzy logic methods for various systems are required. This is especially true for process systems where there appears to be no comparative study at all.

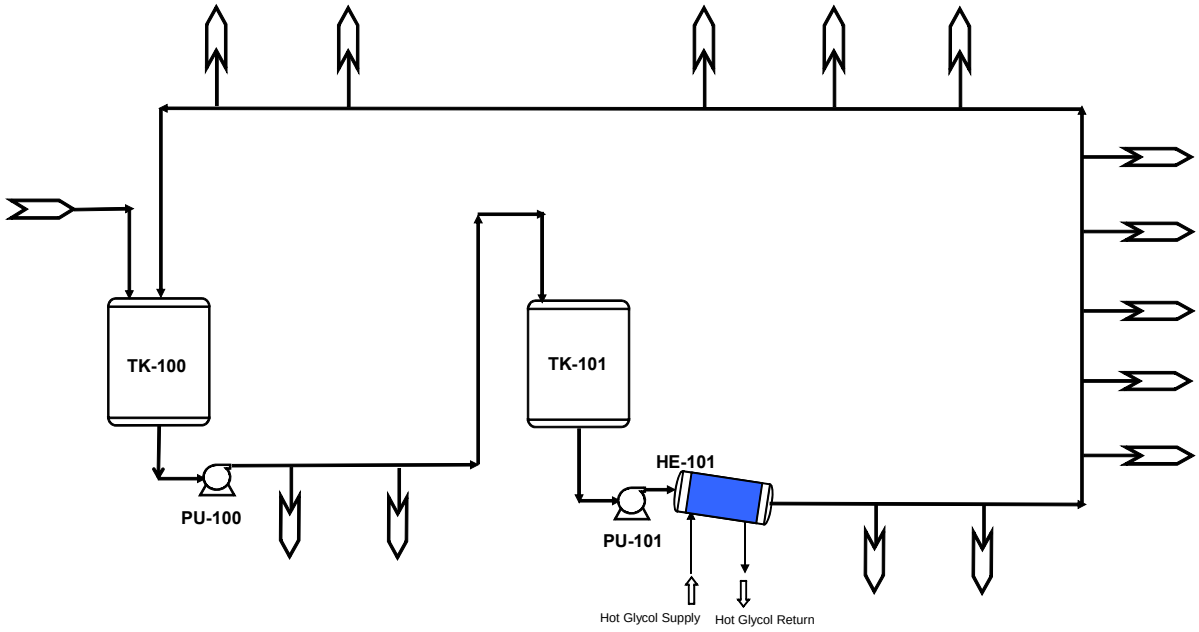


Figure 6.1 DI/WFI distribution system with two DI/WFI storage tanks arranged in series

This thesis finds FL problematic as a method to simulate schedule and volume uncertainty in chemical plant utilities. But there may be another way to utilise FL in modelling uncertainty in DI/WFI systems utilising α-cuts. Revelli and Ridolfi [86] or Branisavljevic and Ivetic [87] used α-cuts (see Appendix 1) to model uncertainty in municipal water systems. The applicability of α-cuts is not investigated here as some of the limitations of FL such as inability to display day-to-day variation, no guarantee of being conservative or inability to find extreme events remain. Hence a detailed investigation into the applicability of α-cuts in uncertainty modelling of DI/WFI systems is left to others.

7 Bibliography

1. Averill M. Law and David M. Kelton. 1991. *Simulation Modeling and Analysis*. 2nd ed. New York: McGraw-Hill.

2. J. Banks, J. Carson, B. L. Nelson and D. Nicol. 2005. *Discrete-Event System Simulation*. 4th ed. Upper Saddle River, New Jersey: Prentice Hall.

3. George S. Fishman. 1973. *Concepts and Methods in Discrete Event Digital Simulation*. New York: John Wiley & Sons.

4. Anonymous. EPANET: Software That Models the Hydraulic and Water Quality Behavior of Water Distribution Piping Systems. US EPA, http://www.epa.gov/nrmrl/wswrd/dw/epanet.html, last accessed 4th April 2011.

5. Anonymous. European business - Facts and Figures. Eurostat, 2009, European Commission, http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home, last accessed 4th April 2011.

6. Anonymous. The pharmaceutical industry in the European Union. Eurostat, 2005, European Commission, http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home, last accessed 19th February 2010.

7. Anonymous. WHO Model Lists of Essential Medicines. World Health Organisation, 2009, http://www.who.int/medicines/publications/essentialmedicines/en/, last accessed 4th April 2011.

8. Anonymous. Review of Injectable Drugs Included on the WHO Model List of Essential Drugs. World Health Organization, retrieved from: http://archives.who.int/eml/expcom/expcom12/review_of_injectable.pdf, last accessed 4th April 2011.

9. Ronald A. Rader. 2008. (Re)defining biopharmaceutical. *Nature Biotechnology* 26 (7): pp. 743-751.

10. Gary Walsh. 2006. Biopharmaceutical benchmarks 2006. *Nature Biotechnology* 24 (7): pp. 769-776.

11. Anonymous. Note for guidance on quality of water for pharmaceutical use, CPMP/QWP/158/01. European Medicines Evaluation Agency (EMEA), 2002, http://www.ema.europa.eu/pdfs/human/qwp/015801en.pdf, last accessed 4th April 2011.

12. Theodore H. Meltzer. 1997. *High-Purity Water Preparation: For the Semiconductor, Pharmaceutical, and Power Industries*. 2nd ed. Littleton, Colorado: Tall Oaks Publishing.

13. Anonymous. 2001. *Water and Steam Systems*. 1st ed. International Society for Pharmaceutical Engineering (ISPE).

14. Jose E. Martinez. 2004. Hyperthermophilic Microorganisms and USP Hot Water Systems. *Pharmaceutical Technology* February pp. 50-65.

15. Roger Dabbah. 2006. USP Pharmaceutical Waters, Part 1 Bulk Waters. *BioProcess International* March pp. 50-55.

16. B.H. Junker, M. Stanik, J. Adamca, K. LaRiviere, M. Abbatiello and P. Salmon. 1997. An ambient water loop system for USP purified water. *Bioprocess Engineering* 17 (5): pp. 277-286.

17. Roger Dabbah. 2006. USP Pharmaceutical Waters, Part 2 Packaged Waters and Harmonization Issues. *BioProcess International* April pp. 58-62.

18. Gary D. Eppen, Kipp R. Martin and Linus Schrage. 1989. A Scenario Approach to Capacity Planning. *Operations Research* 37 (4): pp. 517-527.

19. M. L. Winkel, L. C. Zullo, P. J. T. Verheijen and C. C. Pantelides. 1995. Modelling and simulation of the operation of an industrial batch plant using gPROMS. *Computers & Chemical Engineering* 19 (Supplement 1): pp. 571-576.

20. Ming Wu, Darren Sun and Joo Hwa Tay. 2004. Development of a practical model for capacity evaluation of ultrapure water systems. *Desalination* 161 (3): pp. 223-233.

21. Prasad V. Saraph. Biotech Industry: Simulation and Beyond, Proceedings of the 2001 Winter Simulation Conference, Arlington, Virginia, B. A. Peters, J. S. Smith, D. J. Medeiros and M. W. Rohrer, Eds. IEEE Computer Society: Arlington, Virginia, 2001; pp. 838-843.

22. Dougal H. O. McQueen, Patrick R. Hyland and Simon J. Watson. 2004. Monte Carlo Simulation of Residential Electricity Demand for Forecasting Maximum Demand on Distribution Networks. *IEEE Transaction on Power Systems* 19 (3): pp. 1685-1689.

23. Craig W. Alexander. Discrete Event Simulation for Batch Processing, Proceedings of the 2006 Winter Simulation Conference, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol and R. M. Fujimoto, Eds. IEEE: 2006; pp. 1929-1934.

24. Paul J. Sanchez. As simple as Possible, But no Simpler: A Gentle Introduction to Simulation Modeling, Proceedings of the 2006 Winter Simulation Conference, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol and R. M. Fujimoto, Eds. IEEE: 2006; pp. 2-10.

25. Michael L. Pinedo. 2008. *Scheduling Theory, Algorithms, and Systems*. 3rd ed. New York: Springer.

26. George J. Klir and Mark J. Wierman. 1999. *Uncertainty-Based Information: Elements of Generalized Information Theory*. 2nd ed. Heidelberg: Physica-Verlag.

27. Guilherme E. Vieira, Jeffrey W. Herrmann and Edward Lin. 2003. Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *Journal of Scheduling* 6 (1): pp. 39-62.

28. Manfred Drosg. 2009. *Dealing with uncertainties: a guide to error analysis*. 2nd ed. Dordrecht, Heidelberg, London, New York: Springer.

29. Timothy J. Ross, Jane M. Booker and W. Jerry Parkinson. The great debate. In *Fuzzy Logic and Probability Applications: Bridging the Gap, ASA-SIAM Series on Statistics and Applied Probability*, Timothy J. Ross, Jane M. Booker and W. Jerry Parkinson, Eds. SIAM, ASA: Philadelphia, Alexandria, VA, 2002.

30. Melanie Mitchell. 2009. *Complexity: a guided tour*. New York: Oxford University Press.

31. Péter Érdi. 2008. *Complexity Explained*. Berlin, Heidelberg: Springer.

32. Anonymous. Northeast Blackout of 2003, http://en.wikipedia.org/wiki/Northeast_Blackout_of_2003, last accessed 4th April 2011.

33. Timothy J. Ross and Vladik Kreinovich. 2006. Los Alamos National Laboratory Uncertainty Workshop: An Interval Perspective. *Reliable Computing* 12 pp. 65–71.

34. A.N. Kolmogorov. 1956. *Foundations of the Theory of Probability*. 2nd ed. New York: Chelsea Publishing Company, Download available University of London at: http://www.clrc.rhul.ac.uk/resources/fop/index.htm, last accessed 4th April 2011.

35. Dennis V. Lindley. 2000. The Philosophy of Statistics. *The Statistician* 49 (3): pp. 293-337.

36. Dennis V. Lindley. 2006. *Understanding Uncertainty*. 1st ed. New Jersey: John Wiley & Sons, Inc.

37. Peter Walley. 1990. *Statistical Reasoning with Imprecise Probabilities*. 1st ed. London: Chapman and Hall.

38. Jose M. Bernardo and Adrain F. M. Smith. 1994. *Bayesian Theory*. Chichester: John Wiley & Sons.

39. Russell G. Almond. 1995. Discussion: Fuzzy Logic: Better Science? Or Better Engineering? *Technometrics* 37 (3): pp. 267-270.

40. Bruno D. Finetti. 1972. *Probability, Induction and Statistics The art of guessing*. London: John Wiley & Sons.

41. 1998. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. New York: John Wiley & Sons, Inc.

42. J. Balasubramanian and I. E. Grossmann. 2003. Scheduling optimization under uncertainty - an alternative approach. *Computers & Chemical Engineering* 27 (4): pp. 469-490.

43. Kaddour Najim, Enso Ikonen and Ait-Kadi Daoud. 2004. *Stochastic Processes*. 1st ed. London: Kogan Page Science.

44. Cheryl Cihon and John K. Taylor. 2004. *Statistical techniques for data analysis*. 2nd ed. Boca Raton, Florida: CRC Press.

45. Karl Bury. 1999. *Statistical Distributions in Engineering*. Cambridge University Press.

46. Merran Evans, Nicholas Hastings and Brian Peacock. 1993. *Statistical Distributions*. 2nd ed. New York: John Wiley & Sons.

47. Didier Dubois. 2006. Possibility theory and statistical reasoning. *Computational Statistics & Data Analysis* 51 pp. 47–69.

48. Steven Schwartzman. 1996. *The Words of Mathematics: An Etymological Dictionary of Mathematical Terms Used in English*. Cambridge: The Mathematical Association of America.

49. Didier Dubois and Henri Prade. 1988. *Possibility theory: an approach to computerized processing of uncertainty*. New York: Plenum Press.

50. George J. Klir. 2006. *Uncertainty and information: foundations of generalized information theory*. New Jersey: John Wiley & Sons, Inc.

51. Lotfi A. Zadeh. 1978. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1 (1): pp. 3–28.

52. Hans-Jürgen Zimmermann. 1992. *Fuzzy Set Theory and its Applications*. 2nd ed. Dordrecht, The Netherlands: Kluwer Academic Publishers.

53. Timothy J. Ross. 2004. *Fuzzy Logic with Engineering Applications*. 2nd ed. West Sussex: John Wiley & Sons.

54. Didier Dubois and Henri Prade. 1997. The three semantics of fuzzy sets. *Fuzzy Sets and Systems* 90 (2): pp. 141-150.

55. A. Azadeh, V. Ebrahimipour and P. Bavar. 2009. A fuzzy inference system for pump failure diagnosis to improve maintenance process: The case of a petrochemical industry. *Expert Systems with Applications* 37 pp. 627-639.

56. A. Traoré, S. Grieu, S. Puig, L. Corominas, F. Thiery, M. Polit and J. Colprim. 2005. Fuzzy control of dissolved oxygen in a sequencing batch reactor pilot plant. *Chemical Engineering Journal* 111 (1): pp. 13-19.

57. E. Zio, P. Baraldi, M. Librizzi, L. Podofillini and V. N. Dang. 2009. A fuzzy set-based approach for modeling dependence among human errors. *Fuzzy Sets and Systems* 160 (13): pp. 1947-1964.

58. Martin F. McNeill and Ellen Thro. 1994. *Fuzzy Logic, A Practical Approach*. Boston: Academic Press Inc.

59. Gang Feng. 2006. A Survey on Analysis and Design of Model-Based Fuzzy Control Systems. *IEEE Transactions on Fuzzy Systems* 14 (5): pp. 676-697.

60. Lotfi A. Zadeh. 1965. Fuzzy Sets. *Information and Control* 8 pp. 338-353.

61. Lotfi A. Zadeh. 2006. Generalized theory of uncertainty (GTU) - principal concepts and ideas. *Computational Statistics & Data Analysis* 51 (1): pp. 15-46.

62. Jason Matthew Aughenbaugh. 2006. *Managing Uncertainty in Engineering Design using Imprecise Probabilities and Principles of Information Economics,* . http://westinghouse.marc.gatech.edu/Members/jaughenbaugh/papers_presentations/aughenbaugh_jason_m_200608_phd.pdf, last accessed 4th April 2011, PhD Thesis, Georgia Institute of Technology, George W. Woodruff School of Mechanical Engineering.

63. E. Nikolaidis, S. Chen, H. Cudney, R.T. Haftka and R. Rosca. 2004. Comparison of Probability and Possibility for Design Against Catastropic Failure Under Uncertainty. *ASME Journal of Mechanical Design* 126 pp. 386-394.

64. Efstratios Nikolaidis. 2005. *Types of Uncertainty in Design Decision Making*. Boca Raton, Florida: CRC Press LLC.

65. Jerry M. Mendel. March 1995. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE* 83 (3): pp. 345-377.

66. Didier Dubois and Henri Prade. 1996. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems* 84 (2): pp. 169-185.

67. G. Perrone, A. Zinno and S. Noto La Diega. 2001. Fuzzy discrete event simulation: A new tool for rapid analysis of production systems under vague information. *Journal of Intelligent Manufacturing* 12 pp. 309-326.

68. Xuzhu Wang and Etienne E. Kerre. 2001. Reasonable properties for the ordering of fuzzy quantities (I). *Fuzzy Sets and Systems* 118 (3): pp. 375-385.

69. Xuzhu Wang and Etienne E. Kerre. 2001. Reasonable properties for the ordering of fuzzy quantities (II). *Fuzzy Sets and Systems* 118 (3): pp. 387-405.

70. J. Recasens, D. Boixader and J. Jacas. 1999. Searching for Meaning on Defuzzification. *International Journal of Uncertainty* 7 (5): pp. 475-482.

71. Shounak Roychowdhury and Witold Pedrycz. 2001. A Survey of Defuzzification Strategies. *International Journal of Intelligent Systems* 16 pp. 679-695.

72. William L. Oberkampf and Jon C. Helton. 2005. *Evidence Theory for Engineering Applications*. Boca Raton, Florida: CRC Press LLC.

73. Lotfi A. Zadeh. 2005. Toward a generalized theory of uncertainty (GTU) - an outline. *Information Sciences* 172 (1-2): pp. 1-40.

74. Ferson Scott, Joslyn Cliff A., Helton Jon C., Oberkampf William L. and Sentz Kari. 2004. Summary from the epistemic uncertainty workshop: consensus amid diversity. *Reliability Engineering & System Safety* 85 (1-3): pp. 355-369.

75. Anthony J. Pansini. 2005. *Guide to electrical power distribution systems*. 6th ed. Boca Raton, Florida: Dekker/CRC Press.

76. R.L. Muhanna and R. L. Mullen. Interval Methods for Reliable Computing. In *Engineering Design Reliability Handbook*, Dan M. Ghiocel Efstratios Nikolaidis, Suren Singhal, Ed. CRC Press LLC: Boca Raton, Florida, 2005.

77. Zdzislaw Pawlak and Andrzej Skowron. 2007. Rudiments of rough sets. *Information Sciences* 177 (1): pp. 3-27.

78. Orlowska Ewa (Editor). 1998. *Incomplete Information: Rough Set Analysis*. Heidelberg: Physica-Verlag.

79. Katarina Kavsek-Biasizzo, Igor Skrjanc and Drago Matko. 1997. Fuzzy predictive control of highly nonlinear pH process. *Computers & Chemical Engineering* 21 (Supplement 1): pp. S613-S618.

80. N. M. Ghasem. 2006. Design of a Fuzzy Logic Controller for Regulating the Temperature in Industrial Polyethylene Fluidized Bed Reactor. *Chemical Engineering Research and Design* 84 (2): pp. 97-106.

81. William Siler and James J. Buckley. 2005. *Fuzzy expert systems and fuzzy reasoning*. New Jersey: John Wiley & Sons.

82. Lotfi A. Zadeh. 2002. From Computing with Numbers to Computing with Words − From Manipulation of Measurements to Manipulation of Perceptions. *International Journal of Applied Mathematics and Computer Science* 12 (3): pp. 307–324.

83. Catherine Azzaro-Pantel, Pascal Floquet, Luc Pibouleau and Serge Domenech. 1997. A Fuzzy Approach for Performance Modeling in a Batch Plant: Application to Semiconductor Manufacturing. *IEEE Transaction on Fuzzy Systems* 5 (3): pp. 338-357.

84. Hong Zhang, C. M. Tam and Heng Li. 2005. Modeling uncertain activity duration by fuzzy number and discrete-event simulation. *European Journal of Operational Research* 164 (3): pp. 715-729.

85. Francesco Nucci and Antonio Grieco. System analysis and assessment by fuzzy discrete event simulation, 2006 IEEE International Conference on Fuzzy Systems, Vancouver, BC, Canada, IEEE: Vancouver, BC, Canada, 2006; pp. 1591-1598.

86. Roberto Revelli and Luca Ridolfi. 2002. Fuzzy Approach for Analysis of Pipe Networks. *Journal of Hydraulic Engineering* 128 (1): pp. 93-101.

87. Nemanja Branisavljevic and Marko Ivetic. 2006. Fuzzy approach in the uncertainty analysis of the water distribution network of Becej. *Civil Engineering and Environmental Systems* 23 (3): pp. 221-236.

88. Toly Chen and Yu-Cheng Lin. 2009. A fuzzy back propagation network ensemble with example classification for lot output time prediction in a wafer fab. *Applied Soft Computing* 9 (2): pp. 658-666.

89. Toly Chen. 2003. A fuzzy back propagation network for output time prediction in a wafer fab. *Applied Soft Computing* 2 (3): pp. 211-222.

90. Toly Chen, Hsin-Chieh Wu and Yi-Chi Wang. 2009. Fuzzy-neural approaches with example post-classification for estimating job cycle time in a wafer fab. *Applied Soft Computing* 9 (4): pp. 1225-1231.

91. W. Mendenhall and T. Sincich. 2007. *Statistics for Engineering and the Sciences*. 5th ed. New Jersey: Prentice Hall.

92. Haym Benaroya and Seon Mi Han. 2005. *Probability models in engineering and science*. 1st ed. Boca Raton, Florida: Taylor & Francis.

93. Kevin Cronin, Edmond Byrne and Paul O'Leary. 2007. Prevention of thermo-hydraulic rupture of solvent transfer pipes in the pharmaceutical industry. *Journal of Loss Prevention in the Process Industries* 20 (1): pp. 7-14.

94. J. Page, D. Robinson, N. Morel and J. L. Scartezzini. 2008. A generalised stochastic model for the simulation of occupant presence. *Energy and Buildings* 40 (2): pp. 83-98.

95. Nabil Semaan and Tarek Zayed. 2010. A stochastic diagnostic model for subway stations. *Tunnelling and Underground Space Technology* 25 (1): pp. 32-41.

96. Haibo Zhao and Chuguang Zheng. 2008. A stochastic simulation for the collection process of fly ashes in single-stage electrostatic precipitators. *Fuel* 87 (10-11): pp. 2082-2089.

97. Masato R. Nakamura, Marco J. Castaldi and Nickolas J. Themelis. 2010. Stochastic and physical modeling of motion of municipal solid waste (MSW) particles on a waste-to-energy (WTE) moving grate. *International Journal of Thermal Sciences* 49 (6): pp. 984-992.

98. V. Goggos and R. E. King. 1997. Stochastic predictive control of mechatronic systems. *Mechatronics* 7 (2): pp. 129-140.

99. John F. MacGregor and A. K. L. Wong. 1980. Multivariate Model Identification and Stochastic Control of a Chemical Reactor *Technometrics* 22 (4): pp. 453-464

100. Jana Zvarova. 1999. Expert Systems and Relevant Information. *Environmetrics* 10 pp. 493-504.

101. David J. Spiegelhalter, A. Philip Dawid, Steffen L. Lauritzen and Robert G. Cowell. 1993. Bayesian Analysis in Expert Systems. *Statistical Science* 8 (3): pp. 219-247.

102. István Matijevics and Laws Józsa. 1995. An expert-system-assisted reliability analysis of electric power networks. *Engineering Applications of Artificial Intelligence* 8 (4): pp. 449-460.

103. T.G. Watson, C.D. Christian, A.J. Mason, M.H. Smith and R. Meyer. 2004. Bayesian-based pipe failure model. *Journal of Hydroinformatics* 6 pp. 259-264.

104. A. Dehghan, K. J. McManus and E. F. Gad. 2008. Statistical analysis of structural failures of water pipes. *Water Management* 161 (WM4): pp. 207-214.

105. J. B. Boxall, A. O'Hagan, S. Pooladsaz, A. J. Saul and D. M. Unwin. 2007. Estimation of burst rates in water distribution mains. *Water Management* 161 (WM2): pp. 73-82.

106. L.T. Wong and K.W. Mui. 2005. Determination of domestic flushing water consumption in Hong Kong. *Facilities* 23 (1/2): pp. 82-92.

107. C. Tricarico, G.de Marinis, R. Gargano and A. Leopardi. 2007. Peak residential water demand. *Water Management* 160 (WM2): pp. 115–121.

108. Bojana Jankovic-Nisic, Cedo Maksimovic, David Butler and Nigel J. D. Graham. 2004. Use of Flow Meters for Managing Water Supply Networks *Journal of Water Resource Planning and Management* March/April pp. 171-179.

109. Lotfi A. Zadeh. 1995. Discussion: Probability Theory and Fuzzy Logic Are Complementary Rather Than Competitive. *Technometrics* 37 (3): pp. 271-276.

110. Dennis V. Lindley. 1987. The Probability Approach to the Treatment of Uncertainty in Artificial Intelligence and Expert Systems. *Statistical Science* 2 (1): pp. 17-24.

111. Dennis. V. Lindley. 1995. Comments on "The Efficacy of Fuzzy Representations of Uncertainty". *IEEE Transactions on Fuzzy Systems* 2 (1): pp. 37.

112. Michael Laviolette, John W. Seaman, J. Douglas Barrett and William H. Woodall. 1995. A Probabilistic and Statistical View of Fuzzy Methods. *Technometrics* 37 (3): pp. 249-261.

113. G. Perrone, S. Noto La Diegaz and A . Zinno. 1998. A Fuzzy Discrete Event Simulator for Fuzzy Production Environment Analysis. *Annals of the CIRP* 47 (1): pp. 404-408.

114. H. J. Zimmermann. 2000. An application-oriented view of modeling uncertainty. *European Journal of Operational Research* 122 (2): pp. 190-198.

115. Peter Walley and Gert de Cooman. 2001. A behavioral model for linguistic uncertainty. *Information Sciences* 134 (1-4): pp. 1-37.

116. Michele Dassisti and Luigi M. Galantucci. 2005. Pseudo-fuzzy discrete-event simulation for on-line production control. *Computers & Industrial Engineering* 49 (2): pp. 266-286.

117. Jiahao Zeng, Min An and Nigel John Smith. 2007. Application of a fuzzy based decision making methodology to construction project risk assessment. *International Journal of Project Management* 25 (6): pp. 589-600.

118. Thomas A. Runkler. 1997. Selection of Appropriate Defuzzification Methods Using Application Specific Properties. *IEEE Transaction on Fuzzy Systems* 5 (1): pp. 72-79.

119. Michael Laviolette, John W. Seaman Jr., J. Douglas Barrett and William H. Woodall. 1995. A Probabilistic and Statistical View of Fuzzy Methods: Reply. *Technometrics* 37 (3): pp. 249-261.

120. Michael Laviolette and Jr. John W. Seaman. 1994. The Efficacy of Fuzzy Representations of Uncertainty. *IEEE Transactions on Fuzzy Systems* 2 (1): pp. 4-15.

121. Bernd Möller and Michael Beer. 2008. Engineering computation under uncertainty - Capabilities of non-traditional models. *Computers & Structures* 86 (10): pp. 1024-1041.

122. P. Stoodley, K. Sauer, D. G. Davies and J. W. Costerton. 2002. Biofilms as Complex Differentiated Communities. *Annual Review of Microbiology* 56 pp. 187-209.

123. Jörg Klauer. 2001. An Examination of Pipe Self Cleaning in High Purity Water. *Ultrapure Water* March pp. 56-60.

124. S. J. Honkomp, S. Lombardo, O. Rosen and J. F. Pekny. 2000. The curse of reality - why process scheduling optimization problems are difficult in practice. *Computers & Chemical Engineering* 24 (2-7): pp. 323-328.

125. Michael L. Luyben and William L. Luyben. 1993. *Essentials of process control*. New York: McGraw-Hill.

126. Kevin Cronin and James P. Gleeson. Monte Carlo Simulation. In *Handbook of Food and Bioprocess Modeling Techniques*, M. Shafiur Rahman Shyam S. Sablani, Ashim K. Datta, Arun S. Mujumdar, Ed. CRC Press: Boca Raton, 2007.

127. R. E. Bellman and L. A. Zadeh. 1970. Decision-Making in a Fuzzy Environment. *Management Science* 17 (4): pp. 141-164.

128. Lotfi A. Zadeh. 1975. The Concept of a Linguistic Variable and its Application to Approximate Reasoning I. *Information Sciences* 8 pp. 199-249.

129. S. S. Panwalkar and Wafik Iskander. 1977. A Survey of Scheduling Rules. *Operations Research* 25 (1): pp. 45-61.

130. Sarah M. Jay, Dawson Drew, Ferguson Sally and Lamond Nicole. 2008. Driver fatigue during extended rail operations. *Applied Ergonomics* 39 (5): pp. 623-629.

131. Wallace Bloom. 1967. *Shift Work and Human Efficiency, In: Studies in Personnel and Industrial Psychology*. Homewood, Illinois: The Dorsey Press.

132. Johnsson Anders and Jan E. Fröberg. 1975. Work Schedules and Biological Clocks. *Ambio* 4 (1): pp. 45-50.

133. Kerstin Dahlgren. Shiftwork, Work Scheduling and their Impact Upon Operators in Nuclear Power Plants, Human Factors and Power Plants, IEEE: 1988; pp. 517-521.

134. Anonymous. EC Council Directive 93/104/EC of 23rd November 1993 Concerning certain aspects of the organization of working time. European Commission, http://eur-lex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=en&numdoc=31993L0104&model=guichett, last accessed 4th April 2011.

135. Leonid Reznik. 1997. *Fuzzy Controllers*. Oxford: Butterworth-Heinemann.

136. Various, *ASME BPE-2009, The American Society of Mechanical Engineers*. ASME: New York, 2009.

137. S. Robinson. 2005. Discrete-event simulation: from the pioneers to the present, what next? *Journal of the Operational Research Society* 56 pp. 619–629.

138. Steve Chu, Prasad V. Saraph and Lee Schruben. Packaging Capacity Analysis of a Biopharmaceutical Production Operation, Proceedings of the 2005 Winter Simulation Conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong and J. A. Joines, Eds. IEEE: 2005; pp. 2449-2453.

139. Prasad V. Saraph. Simulating Biotech Manufacturing Operations: Issues and Complexities, Proceedings of the 2001 Winter Simulation Conference, B. A. Peters, J. S. Smith, D. J. Medeiros and M. W. Rohrer, Eds. IEEE: 2001; pp. 530-534.

140. A. Dietz, C. Azzaro-Pantel, L. Pibouleau and S. Domenech. 2007. Optimal design of batch plants under economic and ecological considerations: Application to a biochemical batch plant. *Mathematical and Computer Modelling* 46 (1-2): pp. 109-123.

141. John S. Carson. Introduction to Modelling and Simulation, Proceedings of the 2003 Winter Simulation Conference, S. Chick, P. J. Sánchez, D. Ferrin and D. J. Morrice, Eds. IEEE: 2003; pp. 7-13.

142. Frank De Leeuw. 1962. The Concept of Capacity. *Journal of the American Statistical Association* 57 (300): pp. 826-840.

143. George S. Fishman. 1999. *Monte Carlo: concepts, algorithms, and applications*. 3rd ed. New York: Springer.

144. Bart M. Nicolaï, Nico Scheerlinck and Maarten L.A.T.M. Hertog. Probabilistic Modeling. In *Handbook of Food and Bioprocess Modeling Techniques*, M. Shafiur Rahman Shyam S. Sablani, Ashim K. Datta, Arun S. Mujumdar, Ed. CRC Press: Boca Raton, 2007.

145. Robert G. Sargent. Verification and Validation of Simulation Models, Proceedings of the 2007 Winter Simulation Conference, S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew and R. R. Barton, Eds. IEEE: 2007; pp. 124-137.

146. Susan M. Sanchez. ABC's of Output Analysis, Proceedings of the 1999 Winter Simulation Conference, Phoenix, Arizona, United States, P. A. Farrington, H. B. Nembhard, D. T. Sturrock and G. W. Evans, Eds. ACM: Phoenix, Arizona, United States, 1999; pp. 24-32.

147. Marvin K. Nakayama. Output Analysis for Simulations, Proceedings of the 2006 Winter Simulation Conference, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol and R. M. Fujimoto, Eds. IEEE: 2006; pp. 36-46.

148. Christos Alexopoulos. A Comprehensive Review of Methods for Simulation Output Analysis, Proceedings of the 38th Conference on Winter simulation, Monterey, California, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol and R. M. Fujimoto, Eds. IEEE: Monterey, California, 2006; pp. 168-178.

149. Andrew Greasley. 2008. *Enabling a Simulation Capability in the Organisation*. London: Springer.

150. Vlatka Hlupic. 1999. Discrete-Event Simulation Software: What the Users Want. *Simulation Modelling Practice and Theory* 73 (362): pp. 363-370.

151. Andrew F. Seila. Spreadsheet Simulation, Proceedings of the 2006 Winter Simulation Conference, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol and R. M. Fujimoto, Eds. IEEE: 2006; pp. 11-18.

152. J. D. Pemberton and A. J. Robson. 2000. Spreadsheets in Business. *Industrial Management & Data Systems* 100 (8): pp. 379-388.

153. Andrew Greasley. 1998. An Example of a Discrete-Event Simulation on a Spreadsheet. *Simulation* 70 (3): pp. 148-162.

154. Hervé Thiriez. 2004. Spreadsheet-Based Professional Modelling. *INFORMS Transactions on Education* 4 (2): pp. 14-27.

155. Anonymous. Visual Basic User Guide, Microsoft® Excel, Version 5.0 and on-line help functions included with Excel. Microsoft Corporation.

156. Raymond R. Panko and Ralph H. Sprague. 1998. Hitting the wall: errors in developing and code inspecting a simple spreadsheet model. *Decision Support Systems* 22 (4): pp. 337-353.

157. Stephen G. Powell, Kenneth R. Baker and Barry Lawson. 2009. Impact of errors in operational spreadsheets. *Decision Support Systems* 47 (2): pp. 126-132.

158. Stephen G. Powell, Kenneth R. Baker and Barry Lawson. 2008. A critical review of the literature on spreadsheet errors. *Decision Support Systems* 46 (1): pp. 128-138.

159. J. Green, S. Bullen, R. Bovey and M. Alexander. 2007. *Excel 2007 VBA Programmers Reference*. Indianapolis: Wiley Publishing Inc.

160. Edsger W. Dijstra. Notes on Structured Programming. Technological University Eindhoven, http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD249.PDF, last accessed 4th April 2011.

161. Frank Riedewald. 2011. *Download DI/WFI Stochastic Excel Simulation Spreadsheet*. University College Cork, Department of Process & Chemical Engineering, http://www.ucc.ie/processeng/links/utility/stochastic

162. Frank Riedewald. 2011. *Download DI/WFI Fuzzy Logic Excel Simulation Spreadsheet*. University College Cork, Department of Process & Chemical Engineering, http://www.ucc.ie/processeng/links/utility/FuzzyModel

163. Anonymous. GNU Operating System. http://www.gnu.org/, last accessed 4th April 2011.

164. Sam Savage. 2002. The Flaw of Averages. *Harvard Business Review* November pp. 20-21.

165. Pierre L'Ecuyer, G. Henderson Shane and L. Nelson Barry. 2006. Chapter 3 Uniform Random Number Generation. In *Handbooks in Operations Research and Management Science,* Elsevier, and further information from L'Ecuyer website: http://www.iro.umontreal.ca/~lecuyer/myftp/papers last accessed 28 June 2011.

166. R. J. Brooks and A. M. Tobias. 1996. Choosing the Best Model: Level of Detail, Complexity, and Model Performance. *Mathematical and Computer Modelling* 24 (4): pp. 1-14.

167. Kenneth J. Musselman. Guidelines for Success. In *Handbook of Simulation*, Jerry Banks, Ed. John Wiley & Sons: 1998.

168. B. D. McCullough. 2008. Microsoft Excel's Not The Wichmann-Hill' random number generators. *Computational Statistics & Data Analysis* 52 (10): pp. 4587-4593.

169. A. Talha Yalta. 2008. The accuracy of statistical distributions in Microsoft® Excel 2007. *Computational Statistics & Data Analysis* 52 (10): pp. 4579-4586.

170. Hans Pottel. Statistical Flaws in Excel. University of Coventry, http://www.coventry.ac.uk/ec/~nhunt/pottel.pdf, last accessed 4th April 2011.

171. Su Yu-Sung. 2008. It's easy to produce chartjunk using Microsoft® Excel 2007 but hard to make good graphs. *Computational Statistics & Data Analysis* 52 (10): pp. 4594-4601.

172. B.A. Wichmann and I. D. Hill. 1982. Algorithm AS 183: An Efficient and Portable Pseudo-Random Number Generator. *Applied Statistics* 31 (1): pp. 188-190

173. A.G. Steele and R.J. Douglas. Monte Carlo. Institute for National Measurements Standards, National Research Council of Canada, http://inms-ienm.nrc-cnrc.gc.ca/qde/montecarlo/choosedownloads.html, last accessed 27th February 2010.

174. Leo Knüsel. 2005. On the accuracy of statistical distributions in Microsoft Excel 2003. *Computational Statistics & Data Analysis* 48 (3): pp. 445-449.

175. Jeffrey S. Simonoff. Statistical analysis using Microsoft Excel. New York University, Leonard. N. Stern School of Business, http://pages.stern.nyu.edu/~jsimonof/classes/1305/pdf/excelreg.pdf, last accessed 4th April 2011.

Appendices

Appendix 1 Fuzzy Set Theory

This appendix gives a brief overview on fuzzy set theory and fuzzy operations as it applies to this thesis. The information in this section is taken from textbooks on fuzzy set theory such as Klir and Wierman [26], Klir [50], and Ross [53], if not mentioned otherwise.

In classical set theory the membership of elements of a set is assessed in binary terms (member/non-member) resulting in so–called crisp sets, denoted A, usually in the interval *[0,1]*. Zadeh [60] extended the classical set theory to fuzzy sets, here denoted $\underset{\sim}{A}$, in 1964, allowing gradual membership values in what he dubbed Fuzzy Set Theory (FST) (see Figure App.1.1).

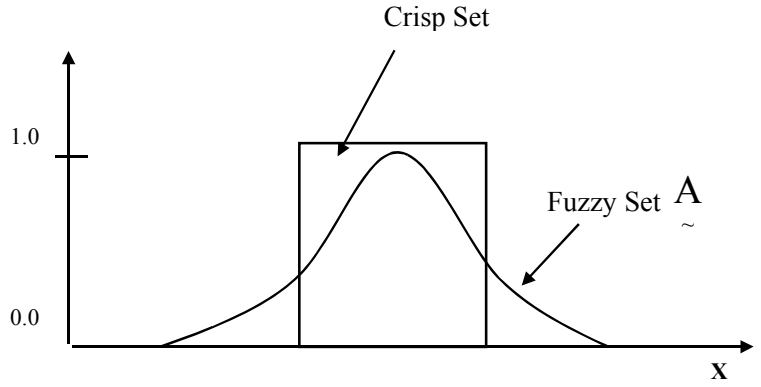


Figure App.1.1 Crisp set A and general fuzzy $\underset{\sim}{A}$ set on universe of discourse **X**

Definition 3.1: A fuzzy set $\underset{\sim}{A}$ in a universe of discourse ***X*** is characterized by a membership function $\mu_{\underset{\sim}{A}}(x)$ which maps each element x in ***X*** to a real number in the interval [0, 1]. For every $x \in \boldsymbol{X}$ the fuzzy set $\underset{\sim}{A}$ is defined as a set of ordered pairs x as:

$$\underset{\sim}{A} = \left\{ \left(x, \mu_{\underset{\sim}{A}}(x) \right) \mid x \in \underset{\sim}{A}, \mu_{\underset{\sim}{A}}(x) \in [0,1] \right\} \qquad \text{(App. 1.1)}$$

The membership function $\mu_{\underset{\sim}{A}}(x)$ is called the membership function of the fuzzy set and is continuous for this work. The function $\mu_{\underset{\sim}{A}}(x)$ quantifies the grade of membership of the elements *x* to the fundamental set ***X***. An element with the value 0 is not included in the given set while 1 describes a fully included member. Values between 0 and 1 characterize the fuzzy members and is referred to as the membership grade or degree of membership.

The following definitions are important in the theory of fuzzy sets:

Definition 3.2: Height

The height h of a fuzzy set $\underset{\sim}{A}$ is the maximum value of the membership function i.e.

$$h(A) = \max\left\{ \mu_{\underset{\sim}{A}}(x) \right\} \qquad \text{(App. 1.1)}$$

Definition 3.3: Core

The core value of a fuzzy set $\underset{\sim}{A}$ (see Figure App. 1.2) are all values of the set, which are characterized by full membership i.e.

$$\mu_{\underset{\sim}{A}}(x) = 1 \qquad \text{(App. 1.2)}$$

Definition 3.4: Support

The support of a fuzzy set $\underset{\sim}{A}$ (see Figure App. 1.2) are all values of the membership, which the set contains i.e.

$$\mu_{\underset{\sim}{A}}(x) > 0 \qquad \text{(App. 1.3)}$$

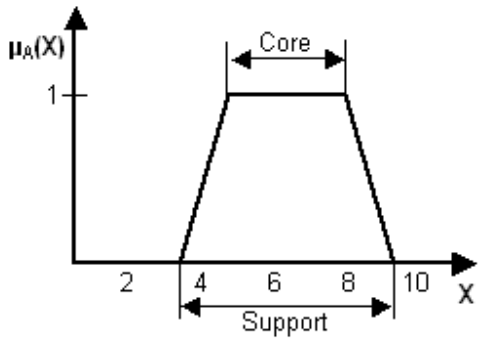


Figure App. 1.2 Core and support of a fuzzy membership function

Definition 3.5: Normal Fuzzy Set

A fuzzy set $\underset{\sim}{A}$ is called "normal" if at least one value of the set has full membership $\mu_{\underset{\sim}{A}}(x) = 1$. This work uses normal fuzzy sets only.

A fuzzy membership function $\mu_{\underset{\sim}{A}}(x)$ can have any appropriate form. In practice, however, the trapezoidal and triangular membership functions are most commonly used as they are easy to implement and computationally inexpensive [reference]. A trapezoidal membership function (see Figure App.1.3) is defined by four parameters:

$$\underset{\sim}{A} = \begin{cases} 0, x \le a \\ (x-a)/(b-a), x \in (a,b) \\ 1, x \in (b,c) \\ (d-x)/(d-c), x \in (c,d) \end{cases} \qquad \text{(App. 1.4)}$$

A trapezoidal membership function can be reduced to a triangular one by setting b = c.

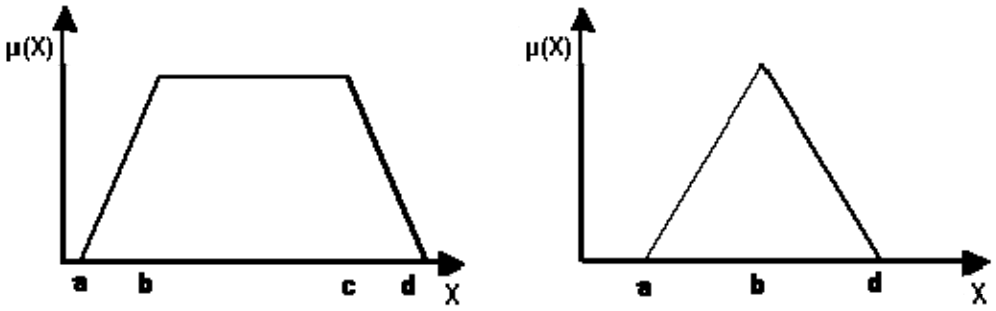


Figure App. 1.3 Trapezoidal (left) and triangular (right) membership function μ(x)

A Gaussian membership function (see Figure App.1.4) is defined as:

$$\underset{\sim}{A} = \exp\left(\frac{-(X-\mu)^2}{2\sigma^2}\right) \quad -\infty < X < \infty \qquad \text{(App. 1.5)}$$

with:

standard deviation $\sigma = \sqrt{\frac{\sum (X-\mu)^2}{N}}$

mean: $\mu = \frac{\sum X}{N}$

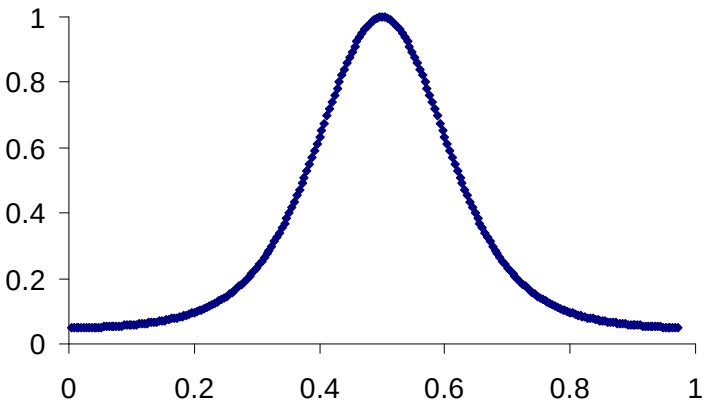


Figure App. 1.4 Gaussian membership function μ(x)

Definition 3.6: α-cut

The α-cut of a fuzzy set $\underset{\sim}{A}$ is defined as the set of the elements having at a minimum the membership value α as defined:

$$\underset{\sim}{A}_{\alpha} = \{x \in X | \mu_A(x) \geq \alpha\} \qquad \text{(App. 1.6)}$$

Is called "α-cut" as shown in Figure App.1.4.

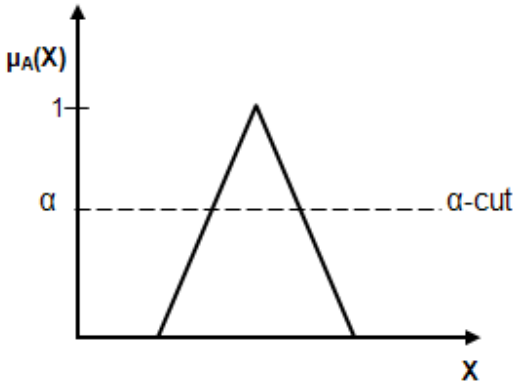


Figure App. 1.5 α-cut of a triangular fuzzy number μ(x)

Only symmetric membership functions are introduced here. However, membership functions do not need to be symmetric.

A large number of different fuzzy operations can be performed on membership functions. The most important operations are the so called standard fuzzy operations, which were originally proposed by Zadeh [60]:

Let $\underset{\sim}{A}$ and $\underset{\sim}{B}$ be two fuzzy sets with membership functions $\mu_{\underset{\sim}{A}}(x)$ and $\mu_{\underset{\sim}{B}}(x)$ respectively, then the following can be standards operations (see also Figure App. 1.5 & App 1.6.) can be performed:

a. **Union** $(\underset{\sim}{C} = \underset{\sim}{A} \cup \underset{\sim}{B})$

The membership function $\mu_{\underset{\sim}{A}\cup\underset{\sim}{B}}(x)$ of the union $\underset{\sim}{A}\cup\underset{\sim}{B}$ is defined for all $x \in X$ by:

$$\mu_{\underset{\sim}{A}\cup\underset{\sim}{B}}(x) = \max\left\{\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right\} \qquad \forall x \in X \qquad \text{(App. 1.7)}$$

The fuzzy union operator can be also be defined as the algebraic sum of two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ as follows:

$$\mu_{\underset{\sim}{A}\cup\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) + \mu_{\underset{\sim}{B}}(x) - \mu_{\underset{\sim}{A}}(x) \bullet \mu_{\underset{\sim}{B}}(x) \qquad \forall x \in X \qquad \text{(App. 1.8)}$$

The union operation in Fuzzy set theory is the equivalent of the **OR** operation in Boolean algebra.

b. **Intersection** $(\underset{\sim}{C} = \underset{\sim}{A} \cap \underset{\sim}{B})$

The membership function $\mu_{\underset{\sim}{A}\cap\underset{\sim}{B}}(x)$ of the intersection $\underset{\sim}{A}\cap\underset{\sim}{B}$ is defined for all $x \in X$ by:

$$\mu_{\underset{\sim}{A}\cap\underset{\sim}{B}}(x) = \min\left\{\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right\} \qquad \forall x \in X \qquad \text{(App. 1.9)}$$

The fuzzy intersection operator can also be defined as the algebraic product of two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ as follows:

$$\mu_{\underset{\sim}{A}\cap\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}} \bullet \mu_{\underset{\sim}{B}} \qquad \forall x \in X \qquad \text{(App. 1.10)}$$

The intersection operation in Fuzzy set theory is the equivalent of the **AND** operation in Boolean algebra.

c. **Complement** ($\underset{\sim}{C} = \underset{\sim}{A} + \underset{\sim}{B}$)

The membership function of the Complement of a Fuzzy set $\underset{\sim}{A}$ with is defined as the negation (see Figure App. 1.7) of the specified membership function:

$$\bar{\mu}_{\underset{\sim}{A}}(x) = 1 - \mu_{\underset{\sim}{A}}(x) \qquad \forall x \in X \qquad \text{(App. 1.11)}$$

The complement operation in Fuzzy set theory is the equivalent of the **NOT** operation in Boolean algebra.

Other mathematical fuzzy operations than the standard operations may be defined. These operations are, however, not needed here, but these operations will not be used for this thesis and are therefore not mentioned here.

Example:

Let $\underset{\sim}{A}$ and $\underset{\sim}{B}$ be two fuzzy sets with triangular membership functions $\mu_{\underset{\sim}{A}}(x)$ and $\mu_{\underset{\sim}{B}}(x)$ respectively, then the addition of the two fuzzy sets is computed as follows:

$$\mu_{\underset{\sim}{A}}(x) + \mu_{\underset{\sim}{B}}(x) = (A_1, A_2, A_3) + (B_1, B_2, B_3) = (A_1 + B_1, A_2 + B_2, A_3 + B_3) \qquad \text{(App. 1.12)}$$

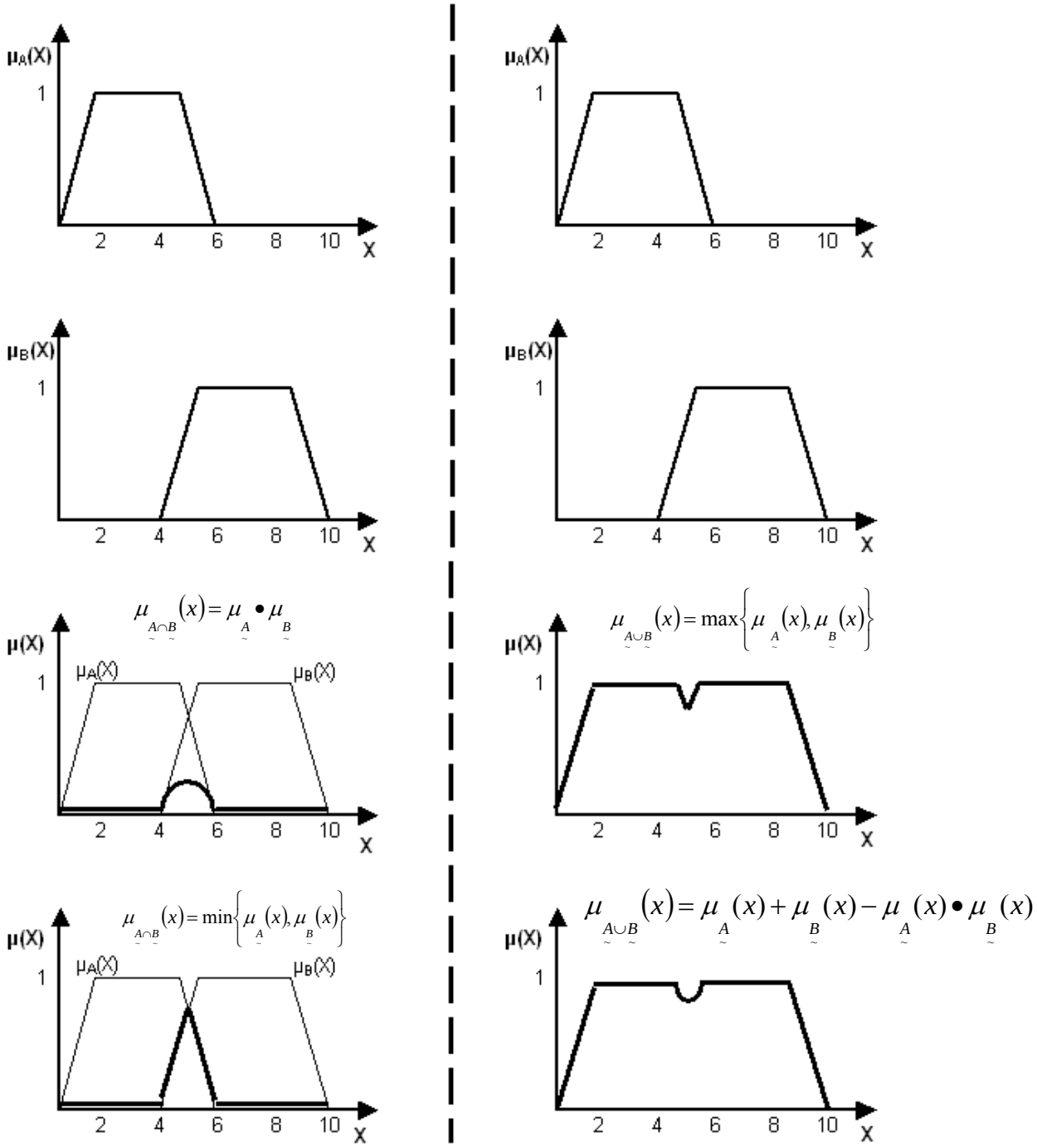


Figure App. 1.6 Standard fuzzy operations on membership functions

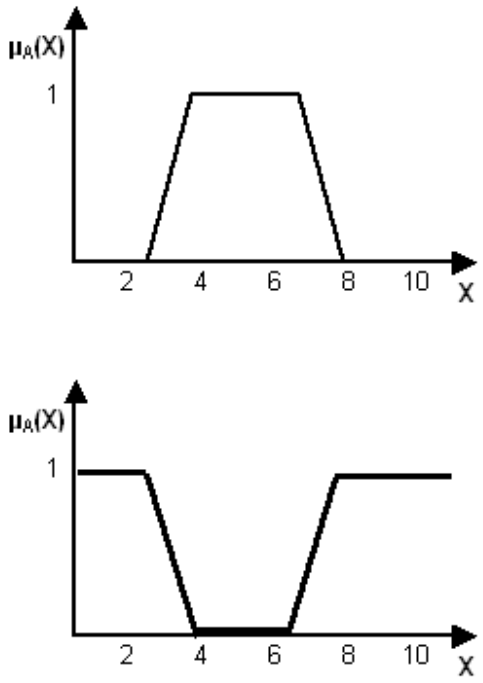


Figure App. 1.7 Complement operation on membership function $\mu_{\tilde{A}}(x)$

Other fuzzy set operations include linguistic hedges, concentration, dilation, intensification and t-norms operations. None of these operations are used in this work; instead the reader is referred to textbooks such as Zimmermann [52] or Ross [53].

Appendix 2 Program Listing: Deterministic and Stochastic Simulation

```
'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
'+ Module: Stochastic_Discrete_Simulation    (STOCHASTIC and DETERMINISTIC)     +
'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ DISCRETE EVENT SIMULATION of a Purified Water Distribution System comprised        @
'@ of a DI water distribution system feeding a WFI distribution system.               @
'@ Uncertainty is modelled via stochastic distributions.                              @
'@ The program can be run as a deterministic or a stochastic discrete-event           @
'@ simulation.                                                                        @
'@ The program was developed by Frank Riedewald.                                      @
'@                                                                                    @
'@ Copyright (C) <2010>  <Frank Riedewald>                                            @
'@                                                                                    @
'@ This program is free software: you can redistribute it and/or modify              @
'@ it under the terms of the GNU General Public License as published by              @
'@ the Free Software Foundation, either version 3 of the License, or                 @
'@ (at your option) any later version.                                               @
'@                                                                                    @
'@ This program is distributed WITHOUT ANY WARRANTY;                                  @
'@ without even the implied warranty of MERCHANTABILITY or FITNESS FOR A              @
'@ PARTICULAR PURPOSE.  See the GNU General Public License                            @
'@ <http://www.gnu.org/licenses/> for more details.                                   @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
‘
Option Explicit                 'Forces explicit variable declaration.
Option Base 1                   'Set default array subscripts to 1.
Option Private Module           'Indicates that module is private.
‘
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ Below are the Global Variables valid across all subroutines.          @
'@ Nomenclature for first letter of variables:                           @
'@ str = String          b = Boolean          L = Long                   @
'@ i = Integer           s = Single           v = Variable               @
'@ d = Date                                                              @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'GLOBAL INPUT VARIABLES
' sDIWaterDemandPerTap(Array) = Water Offtake DI per tap (see Event Table) [m3/h].
' sWFIWaterDemandPerTap(Array) = Water Offtake WFI per tap (see Event Table)[m3/h].
' sWFIWaterFromGen = WFI Water from WFI generation to storage tank [m3/h].
' sWFIStartVolumeTank = WFI storage tank [m3/h].
' sWFIMinVolumeTank = Minimum allowable water Volume in storage tank [m3].
' sWFIStartVolumeTank = Volume in storage tank at beginning of simulation [m3].
' sWFIGenBlowdown = Blowdown WFI Generation Plant [%].
' iColNumberDI = Number of Columns of DI Event Table [-].
' iRowNumberDI = Number Rows of DI Event Table [-].
' iColNumberWFI = Number of Columns of WFI Event Table [-].
' iRowNumberWFI = Number Rows of WFI Event Table [-].
' dTimeStart(Array) = Start Time tap opening as per Event Table WFI or DI [24 hours].
' dTimeEnd(Array) = End Time tap opening as per Event Table WFI or DI [24 hours].
' iResolution = Resolution for Simulation [s].
' bOptionNoUncertainty = Flag to in/exclude uncertainty [-].
' sNormalMean = Mean of Normal Distribution; see sheet "Normal Distribution" [-].
' iNumberofRepeats = Value of how often the simulation shall be repeated [-].
' iDayData = Value of how many days of data are available for the simulation [-].
' bOnlyOneLoop = Only the WFI loop will be simulated [-].
```

```
'
'GLOBAL INTERMEDIATE VARIABLES
' LRepeats = Index for Do-Loop; current number of repeats being simulated [-].
' LDayDataCounter = Index for Do-Loop; current number of simulated day being simulated [-].
' iProgress = Increases throughout calculation - for progress indicator [-].
' dNewTimeStart(Array) = New Start Time if uncertainty is added to dTimeStart [24 hours].
' dNewTimeEnd(Array) = New End Time if uncertainty is added to dTimeEnd [24 hours].
' sWFIWaterFromDI(Array) = DI Water to WFI Generator Array (per day, per second) [m3/s].
' sCalcUncertaintyTime(Array) = Calculated uncertainty time (from distribution) [s].
' sIncrement = For progress indicator; by how much indicator progresses per step [-].
'
'GLOBAL CALCULATED VALUES/OUTPUTS
' sWFIWaterConsumptionFromLoop(Array) =(iNumberofRepeats, 86401, iDayData) [m3/s].
' sDIWaterConsumptionFromLoop(Array) =(iNumberofRepeats, 86401, iDayData) [m3/s].
'
'
Dim sWFIWaterDemandPerTap() As Single, sDIWaterDemandPerTap() As Single
Dim sWFIWaterFromDI() As Single, sIncrement As Single
Dim sWFIWaterConsumptionFromLoop() As Single, sDIWaterConsumptionFromLoop() As Single
'
Dim dDICalcUncertaintyTime() As Date, dWFICalcUncertaintyTime() As Date
Dim dTimeStart() As Date, dTimeEnd() As Date
Dim dNewTimeStart() As Date, dNewTimeEnd() As Date
'
Dim iColNumberWFI As Integer, iRowNumberWFI As Integer
Dim iColNumberDI As Integer, iRowNumberDI As Integer
Dim iNumberofRepeats As Integer, iDayData As Integer
'
Dim LDayDataCounter As Long, LRepeats As Long
'
Dim iProgress As Integer, iResolution As Integer
Dim bOptionNoUncertainty As Boolean, bOnlyWFILoop As Boolean
Dim sWFIWaterFromGen As Single, sWFICalcVolumeTank As Single
Dim sWFIMinVolumeTank As Single, sWFIStartVolumeTank As Single, sWFIGenBlowdown As Single
'
'Maximum Row and Maximum Colum Number for sizing of array; increase if needed:
Const MaxRowNumber As Integer = 1000, MaxColNumber As Integer = 100
'
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ End Definition of the Global Variables.                    @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'
'
Sub Discrete_Event_Simulation()
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ This is the Master Subroutine controlling the flow of the simulation. @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'INPUTS
' iNumberofRepeats = Global variable [-].
' iResolution = Global variable [-].
' iRowNumberDI = Global variable [-].
' iColNumberDI = Global variable [-].
' iRowNumberWFI = Global variable [-].
' iColNumberWFI = Global variable [-].
' LRepeats = Global variable [-].
' iDayData = Global variable [-].
' bOnlyOneLoop = Global variable [-].
' bOptionNoUncertainty = Global variable [-].
'
'INTERMEDIATE VARIABLES
' dTStart = Real time start of calculation [time].
' dTEnd = Real time end of calculation [time].
' iSimulation = Counter for Do Loop [-].
```

```
'
'CALCULATED VALUES/OUTPUTS
' Real time the calculation/simulation took [min].
' Message "Calculation finished".
'
Dim dTStart As Date, dTEnd As Date
Dim strMsg As String, strStyle As String, strTitle As String, strResponse As String
Dim iCount As Integer
'
Application.ScreenUpdating = False  'Turn screen updating off to speed up macro code.
'
dTStart = Now()
'
'Display Progress Bar.
SimulationProgress.RedProgressLabel.Width = 0
SimulationProgress.Show (0)
'
Call UnProtect_Sheets_and_Workbook_No_Message
'
'Read in Data:
Sheets("Frontpage - WFI Loop Input Data").Select
bOptionNoUncertainty = Cells(19, 2)
iNumberofRepeats = Cells(13, 2)
iResolution = 10
iDayData = Cells(14, 2)         'No. of Days Event Table.
bOnlyWFILoop = Cells(24, 2)
'
Call Remove_Old_Data
'
If bOptionNoUncertainty = True Then
   iNumberofRepeats = 1
End If
'
ReDim dDICalcUncertaintyTime(MaxRowNumber, MaxColNumber)
ReDim dTimeStartWFI(MaxRowNumber, MaxColNumber)
ReDim dTimeEndWFI(MaxRowNumber, MaxColNumber)
'
ReDim sWFIWaterDemandPerTap(MaxRowNumber)
ReDim sWFIWaterConsumptionFromLoop(iNumberofRepeats, 100801, iDayData)
'100801 includes 2 hours on both sides.  86400 would be 24-hours in seconds.
ReDim sWFIWaterFromDI(iNumberofRepeats, 100801, iDayData)
'
ReDim dWFICalcUncertaintyTime(MaxRowNumber, MaxColNumber)
ReDim dTimeStartDI(MaxRowNumber, MaxColNumber)
ReDim dTimeEndDI(MaxRowNumber, MaxColNumber)
ReDim dNewTimeStart(MaxRowNumber, MaxColNumber)
ReDim dNewTimeEnd(MaxRowNumber, MaxColNumber)
'
ReDim sDIWaterDemandPerTap(MaxRowNumber)
ReDim sDIWaterConsumptionFromLoop(iNumberofRepeats, 100801, iDayData)
'
ReDim dTimeStart(MaxRowNumber, MaxColNumber), dTimeEnd(MaxRowNumber, MaxColNumber)
'
If bOnlyWFILoop = False Then
   Call DI_Input_Data_Check
End If
Call WFI_Input_Data_Check
'
'Progress indicator:
iProgress = 0
If bOnlyWFILoop = False Then
   sIncrement = 663 / (100 * iNumberofRepeats * iDayData)
Else
   sIncrement = 663 / (100 / 2 * iNumberofRepeats * iDayData)
End If
'
'Start WFI Distribution Simulation:
```

```
'Loop for Number of repeats of the simulation:
LRepeats = 1
Do While LRepeats < iNumberofRepeats + 1
   LDayDataCounter = 1
   Do While LDayDataCounter < iDayData + 1
      If bOptionNoUncertainty = False Then
         'Check if user does not want to include uncertainty in Simulation.
         Call Distribution_Choice_WFI
      End If
      Call Calculate_New_WFI_Start_And_End_Times
      Call WFI_Water_Demand_From_Loop
      Call WFI_Water_Demand_From_Generation_And_Tank_Volume
      LDayDataCounter = LDayDataCounter + 1
   Loop
   'End Simulation WFI Distribution.
   '
   'Start DI Distribution Simulation:
   If bOnlyWFILoop = False Then
      LDayDataCounter = 1
      Do While LDayDataCounter < iDayData + 1
         If bOptionNoUncertainty = False Then
            Call Distribution_Choice_DI
         End If
         Call Calculate_New_DI_Start_And_End_Times
         Call DI_Water_Demand_From_Loop
         LDayDataCounter = LDayDataCounter + 1
      Loop
   End If
   LRepeats = LRepeats + 1
Loop
'End Simulation DI Distribution.
'
Call Move_Data
Call Min_Allowable_Volume_And_Pump_Flowrate
'
If bOnlyWFILoop = False Then
   Call DI_Water_Demand_From_Generation_And_Tank_Volume
End If
'
Call Analysis_of_Data_WFI_Day_1
Call Cell_Select_A4
'
If bOnlyWFILoop = False Then
   Call Analysis_of_Data_DI_Day_1
End If
'
Call Pump_Analysis
'
Sheets("Frontpage - WFI Loop Input Data").Select
Range("K3").Select
dTEnd = Now()
'
'How long did the calculation take:
Cells(42, 2) = (dTEnd - dTStart)
'
Call Protect_Sheets_And_Workbook
'
Call Progress_Indicator(660)
Unload SimulationProgress
'
'Notify user that the calculation is finished:
strMsg = "Simulation Water Demand finished. See Analysis Sheets for results of simulation."
strStyle = vbOKOnly + vbInformation
strTitle = "WFI & DI Simulation"
strResponse = MsgBox(strMsg, strStyle, strTitle)
If strResponse = 1 Then
   End
```

```
    End If
    '
End Sub
'
'
'
Sub WFI_Input_Data_Check()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine checks the original data for time overlaps and some other    @
    '@ inconsistencies in the data and highlights such errors to the user.                @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' All Global variables are read in here as module can be used outside of main program.
    ' iColNumberWFI = Global variable [-].
    ' iRowNumberWFI = Global variable [-].
    ' dTimeStart = Global variable [-].
    ' dTimeEnd = Global variable [-].
    ' dMaxTimeUncertainty = See Input Table [-].
    ' dMaxTimeVariation = MaxTimeUncertainty added to EndTime [-].
    ' dMinTimeVariation = MaxTimeUncertainty added to EndTime [-].
    '
    'INTERMEDIATE VARIABLES
    ' iCwIndex = Index for Do While-Loop [-].
    ' iCount = Index for Do While-Loop [-].
    ' iCount2 = Index for Do While-Loop [-].
    ' dEnd1 = For check of time overlap [-].
    ' dStart2 = For check of time overlap [-].
    ' dStart1 = For check of time overlap [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' Various Messages that a problem with the input values have been detected. Program
    ' terminates if a problem has been detected.
    ' dTimeVariationPositive = Time variation added [date].
    ' dTimeVariationNegative = Time variation minus [date].
    ' strColumnIntegerConvert = Column number of problem identified [-].
    ' strRowIntegerConvert = Row number of problem identified [-].
    '
    Dim iCount As Integer, iCwIndex As Integer, iCount2 As Integer
    Dim dEnd1 As Date, dStart2 As Date, dStart1 As Date
    Dim strMsg As String, strColumnIntegerConvert As String, strRowIntegerConvert As String
    Dim strStyle As String, strTitle As String, strResponse As String, strMyString As String
    Dim dMaxTimeUncertainty As Date, dMaxTimeVariation As Date, dMinTimeVariation As Date
    Dim dTimeVariationPositive As Date, dTimeVariationNegative As Date
    '
    'Constants are maximum row and column data of Input Table:
    Const MaxRowNumber As Integer = 1000, MaxColNumber As Integer = 100
    '
    LDayDataCounter = 1
    Do While LDayDataCounter < iDayData + 1
        'Read in data:
        If LDayDataCounter = 1 Then
            Sheets("Frontpage - WFI Loop Input Data").Select
            iColNumberWFI = Cells(15, 2)
            iRowNumberWFI = Cells(16, 2)
        Else
            Sheets("Day " & LDayDataCounter & " WFI Loop Data").Select
            iColNumberWFI = Cells(15, 2)
            iRowNumberWFI = Cells(16, 2)
        End If
        'End read in data.
        '
        'Ensure that Time Start is not 00:00:00  :
        iCwIndex = 0
        Do While iCwIndex < iColNumberWFI + 1
            iCount = 5
```

```
        Do While iCount < iRowNumberWFI + 5
          dTimeStart(iCount - 4, iCwIndex + 1) = Cells(iCount, 12 + (4 * iCwIndex))
          '
          strRowIntegerConvert = iCount + 4
          strColumnIntegerConvert = iCwIndex + 1
          '
          If dTimeStart(iCount - 4, iCwIndex + 1) = Empty Then
            GoTo 10
          Else
            If dTimeStart(iCount - 4, iCwIndex + 1) = 0 Then
              strMsg = " Start Time cannot be 00:00:00.  Must at least be" _
              + " 00:00:01.  See row  " + strRowIntegerConvert _
              + " and column " + strColumnIntegerConvert _
              + " of Input Table.   Please check. Program cannot continue."
              strStyle = vbCritical
              strTitle = "A critical Message"
              strResponse = MsgBox(strMsg, strStyle, strTitle)
              Call Protect_Sheets_And_Workbook
              End
            End If
          End If
10          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
      Loop
      '
      iCwIndex = 0
      iCount = 5
      Do While iCwIndex < iColNumberWFI
        Do While iCount < iRowNumberWFI + 5
          'Time in 24-hour clock
          dTimeEnd(iCount - 4, iCwIndex + 1) = Cells(iCount, 13 + (4 * iCwIndex))
          iCount = iCount + 1
        Loop
        iCount = 5
        iCwIndex = iCwIndex + 1
      Loop
      '
      'Check precedence constrains:
      'Check if End Time is later than Start Time of next Input; if so terminate program.
      iCwIndex = 1
      Do While iCwIndex < iColNumberWFI + 1
        iCount = 1
        Do While iCount < iRowNumberWFI + 1
          '
          dEnd1 = dTimeEnd(iCount, iCwIndex)
          dStart2 = dTimeStart(iCount, iCwIndex + 1)
          '
          If dStart2 = Empty Then
            GoTo 20
          End If
          '
          If dEnd1 > dStart2 Then
            strRowIntegerConvert = iCount
            strColumnIntegerConvert = iCwIndex
            strMsg = "  Time overlap WFI Water Input Data: End Time is later than" + _
            "Start Time of previous task. See row  " + strRowIntegerConvert + _
            " and column " + strColumnIntegerConvert + " of Input Table." _
            + "  Please check.  Program cannot continue."
            strStyle = vbCritical
            strTitle = "A critical Message"
            strResponse = MsgBox(strMsg, strStyle, strTitle)
            Call Protect_Sheets_And_Workbook
            End
          End If
20          iCount = iCount + 1
        Loop
```

```
    iCwIndex = iCwIndex + 1
  Loop
  '
  'Check precedence constrains:
  'Check if Start Time is later than End Time and terminate program if so.
  iCwIndex = 1
  Do While iCwIndex < iRowNumberWFI + 1
    iCount = 1
    Do While iCount < iColNumberWFI + 1
      '
      dStart1 = dTimeStart(iCwIndex, iCount)
      dEnd1 = dTimeEnd(iCwIndex, iCount)
      '
      If dStart1 = Empty Then
        GoTo 30
      End If
      '
      If dEnd1 < dStart1 Then
        strRowIntegerConvert = iCwIndex
        strColumnIntegerConvert = iCount
        strMsg = "  Error WFI Water Input Data: Start Time is later than End" + _
        "Time. See row  " + strRowIntegerConvert + _
        " and column " + strColumnIntegerConvert + " of Input Table.  " _
        + "Please check.  Program cannot continue."
        strStyle = vbCritical
        strTitle = "A critical Message"
        strResponse = MsgBox(strMsg, strStyle, strTitle)
        Call Protect_Sheets_And_Workbook
        End
      End If
30      iCount = iCount + 1
    Loop
    iCwIndex = iCwIndex + 1
  Loop
  '
  'Check precedence constrains:
  'Check if time uncertainty added to EndTime1 might be later than StartTime2.
  iCwIndex = 1
  Do While iCwIndex < iColNumberWFI + 1
    iCount = 1
    Do While iCount < iRowNumberWFI + 1
      '
      dEnd1 = dTimeEnd(iCount, iCwIndex)
      dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
      '"N" specifies minute as interval.
      dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
      + 60 * Hour(dMaxTimeUncertainty)
      dMaxTimeVariation = DateAdd("N", dMaxTimeUncertainty, dEnd1)
      dStart2 = dTimeStart(iCount, iCwIndex + 1)
      '
      If dStart2 = Empty Then
        GoTo 40
      End If
      '
      If dMaxTimeVariation > dStart2 Then
        strRowIntegerConvert = iCount
        strColumnIntegerConvert = iCwIndex
        strMsg = "Time overlap WFI Water Input: End Time is later than Start" + _
        " Time of next Input if Max. Time Uncertainty is added to input." + _
        "  See row  " + strRowIntegerConvert + " and column " _
        + strColumnIntegerConvert + " of Input Table.  Please check. " + _
        "  Program cannot continue."
        strStyle = vbCritical
        strTitle = "A critical Message"
        strResponse = MsgBox(strMsg, strStyle, strTitle)
        Call Protect_Sheets_And_Workbook
        End
```

```
            End If
40          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
    'Check precedence constrains:
    'Check if time uncertainty subtracted from StartTime2 might be earlier than EndTime1.
    iCwIndex = 1
    Do While iCwIndex < iColNumberWFI + 1
        iCount = 1
        Do While iCount < iRowNumberWFI + 1
            '
            dStart2 = dTimeStart(iCount, iCwIndex + 1)
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * iCwIndex)
            '"N" specifies minute as interval.
            dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
            + 60 * Hour(dMaxTimeUncertainty)
            dMaxTimeVariation = DateAdd("N", (-1) * dMaxTimeUncertainty, dStart2)
            dEnd1 = dTimeEnd(iCount, iCwIndex)
            '
            If dStart2 = Empty Then
                GoTo 50
            End If
            '
            If dMaxTimeVariation < dEnd1 Then
                strRowIntegerConvert = iCount
                strColumnIntegerConvert = iCwIndex
                strMsg = "Time overlap WFI Water Input: Start Time might be earlier" + _
                " than End Time of next Input if Max. Time Uncertainty is subtracted" + _
                " from input.  See row  " + strRowIntegerConvert + " and column " _
                + strColumnIntegerConvert + " of Input Table.  " + _
                "Please check.  Program cannot continue."
                strStyle = vbCritical
                strTitle = "A critical Message"
                strResponse = MsgBox(strMsg, strStyle, strTitle)
                Call Protect_Sheets_And_Workbook
                End
            End If
50          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
    'Check precedence constrains:
    'Check if time uncertainty added to EndTime1 might be later than StartTime2 and
    'time uncertainty subtracted from it.
    iCwIndex = 1
    Do While iCwIndex < iColNumberWFI + 1
        iCount = 1
        Do While iCount < iRowNumberWFI + 1
            '
            dEnd1 = dTimeEnd(iCount, iCwIndex)
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
            '"N" specifies minute as interval.
            dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
            + 60 * Hour(dMaxTimeUncertainty)
            dEnd1 = DateAdd("N", dMaxTimeUncertainty, dEnd1)
            '
            dStart2 = dTimeStart(iCount, iCwIndex + 1)
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * iCwIndex)
            '"N" specifies minute as interval.
            dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
            + 60 * Hour(dMaxTimeUncertainty)
            dStart2 = DateAdd("N", (-1) * dMaxTimeUncertainty, dStart2)
            '
            If dStart2 = Empty Then
```

```
            GoTo 60
          End If
          '
          If dEnd1 > dStart2 Then
            strRowIntegerConvert = iCount
            strColumnIntegerConvert = iCwIndex
            strMsg = "Possible Time overlap.  If max time uncertainty 1 is " + _
            "added to Time 1 and max time uncertainty 2 is subtracted from " + _
            "Time 2, there would be a time overlap.  See " + _
            "row  " + strRowIntegerConvert + " and column " _
            + strColumnIntegerConvert + " of Input Table.  Please check. " _
            + "  Program cannot continue."
            strStyle = vbCritical
            strTitle = "A critical Message"
            strResponse = MsgBox(strMsg, strStyle, strTitle)
            Call Protect_Sheets_And_Workbook
            End
          End If
60          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
      Loop
      '
      'Check precedence constrains:
      'Check if time uncertainty subtracted from StartTime2 might be earlier than EndTime1.
      iCwIndex = 1
      Do While iCwIndex < iColNumberWFI + 1
        iCount = 1
        Do While iCount < iRowNumberWFI + 1
          '
          dStart2 = dTimeStart(iCount, iCwIndex + 1)
          dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * iCwIndex)
          '"N" specifies minute as interval.
          dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
          + 60 * Hour(dMaxTimeUncertainty)
          dMaxTimeVariation = DateAdd("N", (-1) * dMaxTimeUncertainty, dStart2)
          dEnd1 = dTimeEnd(iCount, iCwIndex)
          '
          If dStart2 = Empty Then
            GoTo 70
          End If
          '
          If dMaxTimeVariation <= dEnd1 Then
            strRowIntegerConvert = iCount
            strColumnIntegerConvert = iCwIndex
            strMsg = "Time overlap WFI Water Input: Start Time might be earlier" + _
            " than End Time of next Input if Max. Time Uncertainty is subtracted" + _
            " from input.  See row  " + strRowIntegerConvert + " and column " _
            + strColumnIntegerConvert + " of Input Table.  " + _
            "Please check.  Program cannot continue."
            strStyle = vbCritical
            strTitle = "A critical Message"
            strResponse = MsgBox(strMsg, strStyle, strTitle)
            Call Protect_Sheets_And_Workbook
            End
          End If
70          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
      Loop
      '
      'Check if time uncertainty added to EndTime1 is not later than 2:00:00 of next day.
      iCwIndex = 1
      Do While iCwIndex < iColNumberWFI + 1
        iCount = 1
        Do While iCount < iRowNumberWFI + 1
          '
```

```
            '30 December 1899 is reference date preset in Excel.
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
            '
            If dMaxTimeUncertainty = Empty Then
               GoTo 80
            End If
            '
            '30th December 1899 plus 2 hours.
            dMaxTimeVariation = DateSerial(1899, 12, 30) + 1 + 2 / 24
            dTimeVariationPositive = DateSerial(1899, 12, 30) _
            + dTimeEnd(iCount, iCwIndex) + dMaxTimeUncertainty
            '
            If dTimeVariationPositive > dMaxTimeVariation Then
               strRowIntegerConvert = iCount
               strColumnIntegerConvert = iCwIndex
               strMsg = "Max. Time Uncertainty added to Closing time of valve must not " _
               + "be later than 2:00:00 of the next day." + _
               " Time of next Input if Max. Time Uncertainty is added to input." + _
               "  See row  " + strRowIntegerConvert + " and column " _
               + strColumnIntegerConvert + " of Input Table.  Please check. " + _
               "  Program cannot continue."
               strStyle = vbCritical
               strTitle = "A critical Message"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
               End
               '
            End If
80          iCount = iCount + 1
         Loop
         iCwIndex = iCwIndex + 1
      Loop
      '
      'Check if time uncertainty minus StartTime1 is earlier than 22:00:00 of earlier day:
      iCwIndex = 1
      Do While iCwIndex < iColNumberWFI + 1
         iCount = 1
         Do While iCount < iRowNumberWFI + 1
            '
            'Negative whole numbers represent dates before 30 December 1899.
            '30 December 1899 is reference date preset in Excel.
            '
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
            '
            If dMaxTimeUncertainty = Empty Then
               GoTo 90
            End If
            '
            '29th December 1899 at 22:00:00.
            dMinTimeVariation = DateSerial(1899, 12, 29) - 22 / 24
            dTimeVariationNegative = DateSerial(1899, 12, 29) _
            - dTimeStart(iCount, iCwIndex) + dMaxTimeUncertainty - 1
            '
            'Times earlier than 30.12.1899 are negative values (integers)!
            If dTimeVariationNegative > dMinTimeVariation Then
               strRowIntegerConvert = iCount
               strColumnIntegerConvert = iCwIndex
               strMsg = "Max. Time Uncertainty minus opening time of valve must not " _
               + "be earlier than 22:00:00 of the earlier day." + _
               " Time of next Input if Max. Time Uncertainty is added to input." + _
               "  See row  " + strRowIntegerConvert + "  and column " _
               + strColumnIntegerConvert + " of Input Table.  Please check. " + _
               "  Program cannot continue."
               strStyle = vbCritical
               strTitle = "A critical Message"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
```

```
               End
               '
            End If
90             iCount = iCount + 1
         Loop
         iCwIndex = iCwIndex + 1
      Loop
      '
      'Check correct input of chosen distribution:
      iCount = 0
      Do While iCount < iRowNumberWFI
         iCount2 = 0
         Do While iCount2 < iColNumberWFI
            If Cells(iCount + 5, 13 + 4 * iCount2) = "b" _
               Or Cells(iCount + 5, 11 + 4 * iCount2) = "u" _
               Or Cells(iCount + 5, 11 + 4 * iCount2) = "n" _
               Or Cells(iCount + 5, 11 + 4 * iCount2) = Empty Then
            Else
               strMsg = "Input must either be 'u' for Uniform, 'n' for Normal, " _
               + " or 'b' for Beta."
               strStyle = vbOKOnly + vbCritical
               strTitle = "Critical Message Resolution: Sheet 3 - WFI Loop Input Data"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
               End
            End If
            iCount2 = iCount2 + 1
         Loop
         iCount = iCount + 1
      Loop
      LDayDataCounter = LDayDataCounter + 1
   Loop
   '
End Sub
'
'
'
Sub DI_Input_Data_Check()
   '
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '@ This subroutine checks the original data for time overlaps and some other       @
   '@ inconsistencies in the data and highlights such errors to the user.              @
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '
   'INPUTS
   ' All Global variables are read in here as module can be used outside of main program.
   ' iColNumberDI = Global variable [-].
   ' iRowNumberDI = Global variable [-].
   ' dTimeStart = Global variable [-].
   ' dTimeEnd = Global variable [-].
   ' dMaxTimeUncertainty = See Input Table [-].
   ' dMaxTimeVariation = MaxTimeUncertainty added to EndTime [-].
   ' dMinTimeVariation = MaxTimeUncertainty added to EndTime [-].
   '
   'INTERMEDIATE VARIABLES
   ' iCwIndex = Index for Do While-Loop [-].
   ' iCount = Index for Do While-Loop [-].
   ' iCount2 = Index for Do While-Loop [-].
   ' dEnd1 = For check of time overlap [-].
   ' dStart2 = For check of time overlap [-].
   ' dStart1 = For check of time overlap [-].
   '
   'CALCULATED VALUES/OUTPUTS
   ' Various Messages that a problem with the input values have been detected. Program
   ' terminates if a problem has been detected.
   ' dTimeVariationPositive = Time variation added [date].
   ' dTimeVariationNegative = Time variation minus [date].
```

```
' strColumnIntegerConvert = Column number of problem identified [-].
' strRowIntegerConvert = Row number of problem identified [-].
'
Dim iCount As Integer, iCwIndex As Integer, iCount2 As Integer
Dim dEnd1 As Date, dStart2 As Date, dStart1 As Date
Dim strMsg As String, strColumnIntegerConvert As String, strRowIntegerConvert As String
Dim strStyle As String, strTitle As String, strResponse As String, strMyString As String
Dim dMaxTimeUncertainty As Date, dMaxTimeVariation As Date, dMinTimeVariation As Date
Dim dTimeVariationPositive As Date, dTimeVariationNegative As Date
'
'Constants are maximum row and column data of Input Table:
Const MaxRowNumber As Integer = 1000, MaxColNumber As Integer = 100
'
LDayDataCounter = 1
Do While LDayDataCounter < iDayData + 1
   Sheets("Day " & LDayDataCounter & " DI Loop Data").Select
   iColNumberDI = Cells(15, 2)
   iRowNumberDI = Cells(16, 2)
   'End read in data.
   '
   'Ensure that Time Start is not 00:00:00  :
   iCwIndex = 0
   Do While iCwIndex < iColNumberDI + 1
      iCount = 5
      Do While iCount < iRowNumberDI + 5
         dTimeStart(iCount - 4, iCwIndex + 1) = Cells(iCount, 12 + (4 * iCwIndex))
         '
         strRowIntegerConvert = iCount + 4
         strColumnIntegerConvert = iCwIndex + 1
         '
         If dTimeStart(iCount - 4, iCwIndex + 1) = Empty Then
            GoTo 10
         Else
            If dTimeStart(iCount - 4, iCwIndex + 1) = 0 Then
               strMsg = " Start Time cannot be 00:00:00.  Must at least be" _
               + " 00:00:01.  See row  " + strRowIntegerConvert _
               + " and column " + strColumnIntegerConvert _
               + " of Input Table.   Please check. Program cannot continue."
               strStyle = vbCritical
               strTitle = "A critical Message"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
               End
            End If
         End If
10          iCount = iCount + 1
      Loop
      iCwIndex = iCwIndex + 1
   Loop
   '
   iCwIndex = 0
   iCount = 5
   Do While iCwIndex < iColNumberDI
      Do While iCount < iRowNumberDI + 5
         'Time in 24-hour clock
         dTimeEnd(iCount - 4, iCwIndex + 1) = Cells(iCount, 13 + (4 * iCwIndex))
         iCount = iCount + 1
      Loop
      iCount = 5
      iCwIndex = iCwIndex + 1
   Loop
   '
   'Check precedence constrains:
   'Check if End Time is later than Start Time of next Input; if so terminate program.
   iCwIndex = 1
   Do While iCwIndex < iColNumberDI + 1
      iCount = 1
```

```
        Do While iCount < iRowNumberDI + 1
            '
            dEnd1 = dTimeEnd(iCount, iCwIndex)
            dStart2 = dTimeStart(iCount, iCwIndex + 1)
            '
            If dStart2 = Empty Then
                GoTo 20
            End If
            '
            If dEnd1 > dStart2 Then
                strRowIntegerConvert = iCount
                strColumnIntegerConvert = iCwIndex
                strMsg = "  Time overlap DI Water Input Data: End Time is later than" + _
                "Start Time of previous task. See row  " + strRowIntegerConvert + _
                " and column " + strColumnIntegerConvert + " of Input Table." _
                + "  Please check.  Program cannot continue."
                strStyle = vbCritical
                strTitle = "A critical Message"
                strResponse = MsgBox(strMsg, strStyle, strTitle)
                Call Protect_Sheets_And_Workbook
                End
            End If
20          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
    'Check precedence constrains:
    'Check if Start Time is later than End Time and terminate program if so.
    iCwIndex = 1
    Do While iCwIndex < iRowNumberDI + 1
        iCount = 1
        Do While iCount < iColNumberDI + 1
            '
            dStart1 = dTimeStart(iCwIndex, iCount)
            dEnd1 = dTimeEnd(iCwIndex, iCount)
            '
            If dStart1 = Empty Then
                GoTo 30
            End If
            '
            If dEnd1 < dStart1 Then
                strRowIntegerConvert = iCwIndex
                strColumnIntegerConvert = iCount
                strMsg = "  Error DI Water Input Data: Start Time is later than End" + _
                "Time. See row  " + strRowIntegerConvert + _
                " and column " + strColumnIntegerConvert + " of Input Table.  " _
                + "Please check.  Program cannot continue."
                strStyle = vbCritical
                strTitle = "A critical Message"
                strResponse = MsgBox(strMsg, strStyle, strTitle)
                Call Protect_Sheets_And_Workbook
                End
            End If
30          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
    'Check precedence constrains:
    'Check if time uncertainty added to EndTime1 might be later than StartTime2.
    iCwIndex = 1
    Do While iCwIndex < iColNumberDI + 1
        iCount = 1
        Do While iCount < iRowNumberDI + 1
            '
            dEnd1 = dTimeEnd(iCount, iCwIndex)
```

```
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
            '"N" specifies minute as interval.
            dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
            + 60 * Hour(dMaxTimeUncertainty)
            dMaxTimeVariation = DateAdd("N", dMaxTimeUncertainty, dEnd1)
            dStart2 = dTimeStart(iCount, iCwIndex + 1)
            '
            If dStart2 = Empty Then
               GoTo 40
            End If
            '
            If dMaxTimeVariation > dStart2 Then
               strRowIntegerConvert = iCount
               strColumnIntegerConvert = iCwIndex
               strMsg = "Time overlap DI Water Input: End Time is later than Start" + _
               " Time of next Input if Max. Time Uncertainty is added to input." + _
               "  See row  " + strRowIntegerConvert + " and column " _
               + strColumnIntegerConvert + " of Input Table.  Please check. " + _
               "  Program cannot continue."
               strStyle = vbCritical
               strTitle = "A critical Message"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
               End
            End If
40          iCount = iCount + 1
         Loop
         iCwIndex = iCwIndex + 1
      Loop
      '
      'Check precedence constrains:
      'Check if time uncertainty substracted from StartTime2 might be earlier than EndTime1.
      iCwIndex = 1
      Do While iCwIndex < iColNumberDI + 1
         iCount = 1
         Do While iCount < iRowNumberDI + 1
            '
            dStart2 = dTimeStart(iCount, iCwIndex + 1)
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * iCwIndex)
            '"N" specifies minute as interval.
            dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
            + 60 * Hour(dMaxTimeUncertainty)
            dMaxTimeVariation = DateAdd("N", (-1) * dMaxTimeUncertainty, dStart2)
            dEnd1 = dTimeEnd(iCount, iCwIndex)
            '
            If dStart2 = Empty Then
               GoTo 50
            End If
            '
            If dMaxTimeVariation < dEnd1 Then
               strRowIntegerConvert = iCount
               strColumnIntegerConvert = iCwIndex
               strMsg = "Time overlap DI Water Input: Start Time might be earlier" + _
               " than End Time of next Input if Max. Time Uncertainty is subtracted" + _
               " from input.  See row  " + strRowIntegerConvert + " and column " _
               + strColumnIntegerConvert + " of Input Table.  " + _
               "Please check.  Program cannot continue."
               strStyle = vbCritical
               strTitle = "A critical Message"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
               End
            End If
50          iCount = iCount + 1
         Loop
         iCwIndex = iCwIndex + 1
      Loop
```

```
'
'Check precedence constrains:
'Check if time uncertainty added to EndTime1 might be later than StartTime2 and
'time uncertainty subtracted from it.
iCwIndex = 1
Do While iCwIndex < iColNumberDI + 1
   iCount = 1
   Do While iCount < iRowNumberDI + 1
      '
      dEnd1 = dTimeEnd(iCount, iCwIndex)
      dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
      '"N" specifies minute as interval.
      dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
      + 60 * Hour(dMaxTimeUncertainty)
      dEnd1 = DateAdd("N", dMaxTimeUncertainty, dEnd1)
      '
      dStart2 = dTimeStart(iCount, iCwIndex + 1)
      dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * iCwIndex)
      '"N" specifies minute as interval.
      dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
      + 60 * Hour(dMaxTimeUncertainty)
      dStart2 = DateAdd("N", (-1) * dMaxTimeUncertainty, dStart2)
      '
      If dStart2 = Empty Then
         GoTo 60
      End If
      '
      If dEnd1 > dStart2 Then
         strRowIntegerConvert = iCount
         strColumnIntegerConvert = iCwIndex
         strMsg = "Possible Time overlap.  If max time uncertainty 1 is " + _
         "added to Time 1 and max time uncertainty 2 is subtracted from " + _
         "Time 2, there would be a time overlap.  See " + _
         "row  " + strRowIntegerConvert + " and column " _
         + strColumnIntegerConvert + " of Input Table.  Please check. " _
         + "  Program cannot continue."
         strStyle = vbCritical
         strTitle = "A critical Message"
         strResponse = MsgBox(strMsg, strStyle, strTitle)
         Call Protect_Sheets_And_Workbook
         End
      End If
60       iCount = iCount + 1
   Loop
   iCwIndex = iCwIndex + 1
Loop
'
'Check precedence constrains:
'Check if time uncertainty substracted from StartTime2 might be earlier than EndTime1.
iCwIndex = 1
Do While iCwIndex < iColNumberDI + 1
   iCount = 1
   Do While iCount < iRowNumberDI + 1
      '
      dStart2 = dTimeStart(iCount, iCwIndex + 1)
      dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * iCwIndex)
      '"N" specifies minute as interval.
      dMaxTimeUncertainty = Minute(dMaxTimeUncertainty) _
      + 60 * Hour(dMaxTimeUncertainty)
      dMaxTimeVariation = DateAdd("N", (-1) * dMaxTimeUncertainty, dStart2)
      dEnd1 = dTimeEnd(iCount, iCwIndex)
      '
      If dStart2 = Empty Then
         GoTo 70
      End If
      '
      If dMaxTimeVariation <= dEnd1 Then
```

```
            strRowIntegerConvert = iCount
            strColumnIntegerConvert = iCwIndex
            strMsg = "Time overlap DI Water Input: Start Time might be earlier" + _
            " than End Time of next Input if Max. Time Uncertainty is subtracted" + _
            " from input.  See row  " + strRowIntegerConvert + " and column " _
            + strColumnIntegerConvert + " of Input Table.  " + _
            "Please check.  Program cannot continue."
            strStyle = vbCritical
            strTitle = "A critical Message"
            strResponse = MsgBox(strMsg, strStyle, strTitle)
            Call Protect_Sheets_And_Workbook
            End
          End If
70          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
    'Check if time uncertainty added to EndTime1 is not later than 2:00:00 of next day.
    iCwIndex = 1
    Do While iCwIndex < iColNumberDI + 1
        iCount = 1
        Do While iCount < iRowNumberDI + 1
            '
            '30 December 1899 is reference date preset in Excel.
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
            '
            If dMaxTimeUncertainty = Empty Then
                GoTo 80
            End If
            '
            '30th December 1899 plus 2 hours.
            dMaxTimeVariation = DateSerial(1899, 12, 30) + 1 + 2 / 24
            dTimeVariationPositive = DateSerial(1899, 12, 30) _
            + dTimeEnd(iCount, iCwIndex) + dMaxTimeUncertainty
            '
            If dTimeVariationPositive > dMaxTimeVariation Then
                strRowIntegerConvert = iCount
                strColumnIntegerConvert = iCwIndex
                strMsg = "Max. Time Uncertainty added to Closing time of valve must not " _
                + "be later than 2:00:00 of the next day." + _
                " Time of next Input if Max. Time Uncertainty is added to input." + _
                "  See row  " + strRowIntegerConvert + " and column " _
                + strColumnIntegerConvert + " of Input Table.  Please check. " + _
                "  Program cannot continue."
                strStyle = vbCritical
                strTitle = "A critical Message"
                strResponse = MsgBox(strMsg, strStyle, strTitle)
                Call Protect_Sheets_And_Workbook
                End
                '
            End If
80          iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
    'Check if time uncertainty minus StartTime1 is earlier than 22:00:00 of earlier day:
    iCwIndex = 1
    Do While iCwIndex < iColNumberDI + 1
        iCount = 1
        Do While iCount < iRowNumberDI + 1
            '
            'Negative whole numbers represent dates before 30 December 1899.
            '30 December 1899 is reference date preset in Excel.
            '
            dMaxTimeUncertainty = Cells(iCount + 4, 14 + 4 * (iCwIndex - 1))
```

```
            '
            If dMaxTimeUncertainty = Empty Then
               GoTo 90
            End If
            '
            '29th December 1899 at 22:00:00.
            dMinTimeVariation = DateSerial(1899, 12, 29) - 22 / 24
            dTimeVariationNegative = DateSerial(1899, 12, 29) _
            - dTimeStart(iCount, iCwIndex) + dMaxTimeUncertainty - 1
            '
            'Times earlier than 30.12.1899 are negative values (integers)!
            If dTimeVariationNegative > dMinTimeVariation Then
               strRowIntegerConvert = iCount
               strColumnIntegerConvert = iCwIndex
               strMsg = "Max. Time Uncertainty minus opening time of valve must not " _
               + "be earlier than 22:00:00 of the earlier day." + _
               " Time of next Input if Max. Time Uncertainty is added to input." + _
               "  See row  " + strRowIntegerConvert + "  and column " _
               + strColumnIntegerConvert + " of Input Table.  Please check. " + _
               "  Program cannot continue."
               strStyle = vbCritical
               strTitle = "A critical Message"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
               End
               '
            End If
90          iCount = iCount + 1
         Loop
         iCwIndex = iCwIndex + 1
      Loop
      '
      'Check correct input of chosen distribution:
      iCount = 0
      Do While iCount < iRowNumberDI
         iCount2 = 0
         Do While iCount2 < iColNumberDI
            If Cells(iCount + 5, 13 + 4 * iCount2) = "b" _
               Or Cells(iCount + 5, 11 + 4 * iCount2) = "u" _
               Or Cells(iCount + 5, 11 + 4 * iCount2) = "n" _
               Or Cells(iCount + 5, 11 + 4 * iCount2) = Empty Then
            Else
               strMsg = "Input must either be 'u' for Uniform, 'n' for Normal, " _
               + " or 'b' for Beta."
               strStyle = vbOKOnly + vbCritical
               strTitle = "Critical Message Resolution: Sheet 3 - DI Loop Input Data"
               strResponse = MsgBox(strMsg, strStyle, strTitle)
               Call Protect_Sheets_And_Workbook
               End
            End If
            iCount2 = iCount2 + 1
         Loop
         iCount = iCount + 1
      Loop
      LDayDataCounter = LDayDataCounter + 1
   Loop
   '
End Sub
'
'
```

```
'
Sub Distribution_Choice_WFI()
  '
  '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  '@ This subroutine chooses the appropriate distribution for WFI Water as per input  @
  '@ selection and calculates the uncertainty time, which will be added to the start       @
  '@ time in the subroutine "Calculate_Start_And_End_Times".                               @
  '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  '
  'INPUTS OF SUBROUTINE
  ' sNormalStanDev = Standard deviation of Normal Dist.; see sheet "Normal Distribution" [-].
  ' sAlphaBeta = Alpha parameter Beta distribution; see sheet "Beta Distribution" [-].
  ' sBetaBeta = Beta parameter Beta distribution; see sheet "Beta Distribution" [-].
  ' iColNumberWFI = Global variable [-].
  ' iRowNumberWFI = Global variable [-].
  ' dWFITimeUncertainty(Array) = Crisp Max. Time Uncertainty as per Input Table [time].
  ' sIncrement = Global variable [-].
  ' sWFIDescriptionTimeUncertainty(Array) = Distribution used (i.e. uniform [-].
  '
  'INTERMEDIATE VARIABLES
  ' iCount = Index for Do While-Loop [-].
  ' iCwIndex = Index for Do While-Loop [-].
  ' sRandomVariable = Random variable [-].
  '
  'CALCULATED VALUES/OUTPUTS
  ' dWFICalcUncertaintyTime() = Global variable [-].
  ' iProgress = Global variable [-].
  '
  Dim sNormalMean As Single, sNormalStanDev As Single
  Dim sAlphaBeta As Single, sBetaBeta As Single, sRandomVariable As Single
  Dim strWFIDescriptionTimeUncertainty() As String
  Dim dWFITimeUncertainty() As Date
  Dim iCount As Integer, iCwIndex As Integer
  '
  ReDim dWFITimeUncertainty(MaxRowNumber * iDayData, MaxColNumber * iDayData)
  ReDim sWFIDescriptionTimeUncertainty(MaxRowNumber * iDayData, MaxColNumber * iDayData)
  '
  'Start read in data:
  If LDayDataCounter = 1 Then
     Sheets("Frontpage - WFI Loop Input Data").Select
     iColNumberWFI = Cells(15, 2)
     iRowNumberWFI = Cells(16, 2)
  Else
     Sheets("Day " & LDayDataCounter & " WFI Loop Data").Select
     iColNumberWFI = Cells(15, 2)
     iRowNumberWFI = Cells(16, 2)
  End If
  '
  iCwIndex = 0
  iCount = 5
  Do While iCwIndex < iColNumberWFI
     Do While iCount < iRowNumberWFI + 5
        dWFITimeUncertainty(iCount - 4, iCwIndex + 1) = Cells(iCount, 14 + iCwIndex * 4)
        iCount = iCount + 1
     Loop
     iCount = 5
     iCwIndex = iCwIndex + 1
  Loop
  '
  iCwIndex = 0
  iCount = 5
  Do While iCwIndex < iColNumberWFI
     Do While iCount < iRowNumberWFI + 5
        sWFIDescriptionTimeUncertainty(iCount - 4, iCwIndex + 1) _
        = Cells(iCount, 15 + iCwIndex * 4)
        iCount = iCount + 1
     Loop
```

```
        iCount = 5
        iCwIndex = iCwIndex + 1
    Loop
    '
    Sheets("Sheet 12 - Normal Distribution").Select
    sNormalMean = Cells(3, 3)
    sNormalStanDev = Cells(4, 3)
    '
    Sheets("Sheet 13 - Beta Distribution").Select
    sAlphaBeta = Cells(3, 3)
    sBetaBeta = Cells(4, 3)
    'End reading in Data.
    '
    iCwIndex = 1
    iCount = 1
    Do While iCwIndex < iColNumberWFI + 1
        Do While iCount < iRowNumberWFI + 1
            '
            sRandomVariable = CSng(RandWH())
            '
            If sWFIDescriptionTimeUncertainty(iCount, iCwIndex) = "u" Then
                'Uniform Distribution.
                dWFICalcUncertaintyTime(iCount, iCwIndex) = sRandomVariable _
                * dWFITimeUncertainty(iCount, iCwIndex)
            End If
            '
            If sWFIDescriptionTimeUncertainty(iCount, iCwIndex) = "b" Then
                'Beta Distribution.
                dWFICalcUncertaintyTime(iCount, iCwIndex) _
                = Application.BetaDist(sRandomVariable, sAlphaBeta, sBetaBeta) _
                * dWFITimeUncertainty(iCount, iCwIndex)
            End If
            '
            If sWFIDescriptionTimeUncertainty(iCount, iCwIndex) = "n" Then
                'Normal Distribution.
                dWFICalcUncertaintyTime(iCount, iCwIndex) _
                = Application.NormDist(sRandomVariable, sNormalMean, sNormalStanDev, True) _
                * dWFITimeUncertainty(iCount, iCwIndex)
            End If
            '
            iCount = iCount + 1
        Loop
        iCount = 1
        iCwIndex = iCwIndex + 1
    Loop
    '
    iProgress = iProgress + sIncrement
    Call Progress_Indicator(iProgress)
    '
End Sub
'
'
'
Sub Distribution_Choice_DI()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine chooses the appropriate distribution for DI Water as per input     @
    '@ selection and calculates the uncertainty time, which will be added to the start       @
    '@ time in the subroutine "Calculate_Start_And_End_Times".                                   @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS OF SUBROUTINE
    ' sNormalStanDev = Standard deviation of Normal Dist.; see sheet "Normal Distribution" [-].
    ' sAlphaBeta = Alpha parameter Beta distribution; see sheet "Beta Distribution" [-].
    ' sBetaBeta = Beta parameter Beta distribution; see sheet "Beta Distribution" [-].
    ' iColNumberDI = Global variable [-].
    ' iRowNumberDI = Global variable [-].
```

```
' sIncrement = Global variable [-].
' sDIDescriptionTimeUncertainty(Array) = Distribution used (i.e. normal) [-].
' dDITimeUncertainty = Max. Time Uncertainty as per Input Table [time].
'
'INTERMEDIATE VARIABLES
' iCount = Index for Do While-Loop [-].
' iCwIndex = Index for Do While-Loop [-].
' sRandomVariable = Random variable [-].
'
'CALCULATED VALUES/OUTPUTS
' dDICalcUncertaintyTime() = Global variable [-].
' iProgress = Global variable [-].
'
Dim sNormalMean As Single, sNormalStanDev As Single
Dim sAlphaBeta As Single, sBetaBeta As Single, sRandomVariable As Single
Dim strDIDescriptionTimeUncertainty() As String
Dim dDITimeUncertainty() As Date
Dim iCount As Integer, iCwIndex As Integer
'
ReDim dDITimeUncertainty(MaxRowNumber * iDayData, MaxColNumber * iDayData)
ReDim sDIDescriptionTimeUncertainty(MaxRowNumber * iDayData, MaxColNumber * iDayData)
'
Sheets("Day " & LDayDataCounter & " DI Loop Data").Select
iColNumberDI = Cells(15, 2)
iRowNumberDI = Cells(16, 2)
'
iCwIndex = 0
Do While iCwIndex < iColNumberDI
   iCount = 5
   Do While iCount < iRowNumberDI + 5
      dDITimeUncertainty(iCount - 4, iCwIndex + 1) = Cells(iCount, 14 + iCwIndex * 4)
      iCount = iCount + 1
   Loop
   iCwIndex = iCwIndex + 1
Loop
'
iCwIndex = 0
Do While iCwIndex < iColNumberDI
   iCount = 5
   Do While iCount < iRowNumberDI + 5
      sDIDescriptionTimeUncertainty(iCount - 4, iCwIndex + 1) = _
      Cells(iCount, 15 + iCwIndex * 4)
      iCount = iCount + 1
   Loop
   iCwIndex = iCwIndex + 1
Loop
'
Sheets("Sheet 12 - Normal Distribution").Select
sNormalMean = Cells(3, 3)
sNormalStanDev = Cells(4, 3)
'
Sheets("Sheet 13 - Beta Distribution").Select
sAlphaBeta = Cells(3, 3)
sBetaBeta = Cells(4, 3)
'
iCwIndex = 1
Do While iCwIndex < iColNumberDI + 1
   iCount = 1
   Do While iCount < iRowNumberDI + 1
      '
      sRandomVariable = CSng(RandWH())
      '
      If sDIDescriptionTimeUncertainty(iCount, iCwIndex) = "u" Then
         'Uniform Distribution, valid between 0 and 1.
         dDICalcUncertaintyTime(iCount, iCwIndex) = sRandomVariable _
         * dDITimeUncertainty(iCount, iCwIndex)
      End If
```

```
            '
            If sDIDescriptionTimeUncertainty(iCount, iCwIndex) = "b" Then
               'Beta Distribution, valid between 0 and 1.
               dDICalcUncertaintyTime(iCount, iCwIndex) _
               = Application.BetaDist(sRandomVariable, sAlphaBeta, sBetaBeta) _
               * dDITimeUncertainty(iCount, iCwIndex)
            End If
            '
            If sDIDescriptionTimeUncertainty(iCount, iCwIndex) = "n" Then
               'Normal Distribution, valid between 0 and 1.
               dDICalcUncertaintyTime(iCount, iCwIndex) _
               = Application.NormDist(sRandomVariable, sNormalMean, sNormalStanDev, True) _
               * dDITimeUncertainty(iCount, iCwIndex)
            End If
            '
            iCount = iCount + 1
         Loop
         iCwIndex = iCwIndex + 1
      Loop
      '
      iProgress = iProgress + 1 * sIncrement
      Call Progress_Indicator(iProgress)
      '
End Sub
'
'
'
Sub Calculate_New_WFI_Start_And_End_Times()
   '
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '@ This subroutine calculates the new start and end times as a result          @
   '@ of the added or subtracted uncertainty time to the start time.              @
   '@ The program checks in Sub Routine "Check Times New Schedule"      @
   '@ if there is a problem (overlap "end time 1" with "start time 2"           @
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '
   'INPUTS
   ' dTimeStart = Global variable [-].
   ' dTimeEnd = Global variable [-].
   ' iColNumberWFI = Global variable [-].
   ' iRowNumberWFI = Global variable [-].
   ' bOptionNoUncertainty = Global variable [-].
   ' sIncrement = Global variable [-].
   '
   'INTERMEDIATE VARIABLES
   ' iCount = Index for Do While-Loop [-].
   ' iCwIndex = Index for Do While-Loop [-].
   ' sPositiveOrNegative = Random variable to determine negative or positive addition [-].
   '
   'CALCULATED VALUES/OUTPUTS
   ' dNewTimeStart =  Global variable [24 hours].
   ' dNewTimeEnd = Global variable [24 hours].
   ' iProgress = Global variable [-].
   '
   Dim sPositiveOrNegative As Single
   Dim iCount As Integer, iCwIndex As Integer
   Dim iOptionNoTimeUncertainty As Boolean
   '
   If LDayDataCounter = 1 Then
      Sheets("Frontpage - WFI Loop Input Data").Select
   Else
      Sheets("Day " & LDayDataCounter & " WFI Loop Data").Select
   End If
   '
   'Start read in data of Time Start and Time End:
   iCwIndex = 0
   iCount = 5
```

```
Do While iCwIndex < iColNumberWFI
   Do While iCount < iRowNumberWFI + 5
      dTimeStart(iCount - 4, iCwIndex + 1) = Cells(iCount, 12 + (4 * iCwIndex))
      iCount = iCount + 1
   Loop
   iCount = 5
   iCwIndex = iCwIndex + 1
Loop
'
iCwIndex = 0
iCount = 5
Do While iCwIndex < iColNumberWFI
   Do While iCount < iRowNumberWFI + 5
      dTimeEnd(iCount - 4, iCwIndex + 1) = Cells(iCount, 13 + (4 * iCwIndex))
      iCount = iCount + 1
   Loop
   iCount = 5
   iCwIndex = iCwIndex + 1
Loop
'End Read in Data.
'
'Check if no need to calculate time uncertainty:
If bOptionNoUncertainty = True Then
   iCount = 1
   iCwIndex = 1
   Do While iCwIndex < iColNumberWFI + 1
      Do While iCount < iRowNumberWFI + 1
         dNewTimeStart(iCount, iCwIndex) = dTimeStart(iCount, iCwIndex)
         dNewTimeEnd(iCount, iCwIndex) = dTimeEnd(iCount, iCwIndex)
         iCount = iCount + 1
      Loop
      iCount = 1
      iCwIndex = iCwIndex + 1
   Loop
   GoTo NoUncertainty
End If
'
'Add or subtract uncertainty time from Start Time and EndTime:
'sPositiveOrNegative decides of add or subtract.
iCount = 1
iCwIndex = 1
Do While iCwIndex < iColNumberWFI + 1
   Do While iCount < iRowNumberWFI + 1
      '
      sPositiveOrNegative = CSng(RandWH())
      '
      If sPositiveOrNegative <= 0.5 Then
         sPositiveOrNegative = (-1)
      Else
         sPositiveOrNegative = 1
      End If
      '
      dNewTimeStart(iCount, iCwIndex) = DateAdd("n", sPositiveOrNegative * _
      (Minute(dWFICalcUncertaintyTime(iCount, iCwIndex)) + _
      60 _
      * Hour(dWFICalcUncertaintyTime(iCount, iCwIndex))), dTimeStart(iCount, iCwIndex))
      '
      dNewTimeEnd(iCount, iCwIndex) = DateAdd("n", sPositiveOrNegative * _
      (Minute(dWFICalcUncertaintyTime(iCount, iCwIndex)) + _
      60 _
      * Hour(dWFICalcUncertaintyTime(iCount, iCwIndex))), dTimeEnd(iCount, iCwIndex))
      '
      iCount = iCount + 1
   Loop
   iCount = 1
   iCwIndex = iCwIndex + 1
Loop
```

```
    '
NoUncertainty:
    '
    iProgress = iProgress + 1 * sIncrement
    Call Progress_Indicator(iProgress)
    '
End Sub
'
'
'
Sub Calculate_New_DI_Start_And_End_Times()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates all actual (new) start and end times as a result      @
    '@ of the added or subtracted uncertainty time to the start time.                    @
    '@ The program checks in Sub Routine "Check Times New Schedule"                      @
    '@ if there is a problem (overlap "end time 1" with "start time 2".                  @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' dTimeStart = Global variable [-].
    ' dTimeEnd = Global variable [-].
    ' iColNumberDI = Global variable [-].
    ' iRowNumberDI = Global variable [-].
    ' sIncrement = Global variable [-].
    ' bOptionNoUncertainty = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' iCount = Index for Do While-Loop [-].
    ' iCwIndex = Index for Do While-Loop [-].
    ' sPositiveOrNegative = Random variable to determine negative or positive addition [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' dNewTimeStart= Global variable [24 hours].
    ' dNewTimeEnd = Global variable [24 hours].
    ' iProgress = Global variable [-].
    '
    Dim iCount As Integer, iCwIndex As Integer
    Dim sPositiveOrNegative As Single
    '
    Sheets("Day " & LDayDataCounter & " DI Loop Data").Select
    '
    iCwIndex = 0
    iCount = 5
    'Read in Data:
    Do While iCwIndex < iColNumberDI
        Do While iCount < iRowNumberDI + 5         'See Input Table.
            dTimeStart(iCount - 4, iCwIndex + 1) = Cells(iCount, 12 + (4 * iCwIndex))
            iCount = iCount + 1
        Loop
        iCount = 5
        iCwIndex = iCwIndex + 1
    Loop
    '
    iCwIndex = 0
    iCount = 5
    Do While iCwIndex < iColNumberDI
        Do While iCount < iRowNumberDI + 5
            dTimeEnd(iCount - 4, iCwIndex + 1) = Cells(iCount, 13 + (4 * iCwIndex))
            iCount = iCount + 1
        Loop
        iCount = 5
        iCwIndex = iCwIndex + 1
    Loop
    'End read in data.
    '
    'Check if no need to calculate time uncertainty:
```

```
    If bOptionNoUncertainty = True Then
        iCount = 1
        iCwIndex = 1
        Do While iCwIndex < iColNumberDI + 1
            Do While iCount < iRowNumberDI + 1
                dNewTimeStart(iCount, iCwIndex) = dTimeStart(iCount, iCwIndex)
                dNewTimeEnd(iCount, iCwIndex) = dTimeEnd(iCount, iCwIndex)
                iCount = iCount + 1
            Loop
            iCount = 1
            iCwIndex = iCwIndex + 1
        Loop
        GoTo NoUncertainty
    End If
    '
    'Add or subtract uncertainty time from Start Time and EndTime:
    'sPositiveOrNegative decides of add or subtract.
    iCwIndex = 1
    Do While iCwIndex < iColNumberDI + 1
        iCount = 1
        Do While iCount < iRowNumberDI + 1
            '
            sPositiveOrNegative = CSng(RandWH())
            '
            If sPositiveOrNegative <= 0.5 Then
                sPositiveOrNegative = (-1)
            Else
                sPositiveOrNegative = 1
            End If
            '
            dNewTimeStart(iCount, iCwIndex) = DateAdd("n", sPositiveOrNegative * _
            (Minute(dDICalcUncertaintyTime(iCount, iCwIndex)) + _
            60 * Hour(dDICalcUncertaintyTime(iCount, iCwIndex))), dTimeStart(iCount, iCwIndex))
            '
            dNewTimeEnd(iCount, iCwIndex) = DateAdd("n", sPositiveOrNegative * _
            (Minute(dDICalcUncertaintyTime(iCount, iCwIndex)) + _
            60 * Hour(dDICalcUncertaintyTime(iCount, iCwIndex))), dTimeEnd(iCount, iCwIndex))
            '
            iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
NoUncertainty:
    '
    iProgress = iProgress + 1 * sIncrement
    Call Progress_Indicator(iProgress)
    '
End Sub
'
'
'
Sub WFI_Water_Demand_From_Loop()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the WFI water demand from the consumers along the    @
    '@ loop; ignoring any consumption that might be caused by the tank being filled        @
    '@ by WFI as the WFI Tank may not full at the beginning of the simulation.              @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterDemandPerTap = Global variable [m3/h].
    ' iColNumberWFI = Global variable [-].
    ' iRowNumberWFI = Global variable [-].
    ' dNewTimeStart = Global variable [-].
    ' dNewTimeEnd = Global variable [-].
    ' LRepeats = Global variable [-].
```

```
' sIncrement = Global variable [-].
' bOptionNoUncertainty = Global variable [-].
'
'INTERMEDIATE VARIABLES
' sWFIWaterConsumptionPrior = [m3/s].
' sWFITimeStart = [24 hour clock].
' sWFITimeEnd = [24 hour clock].
' sWFITimeEndIndex = WFI Time End Index [-].
' sWFITimeStartIndex = Index start time [-].
' sWFITimeStartIndexCounter = Index time [-].
' sWFIWaterConsumptionPrior = WFI water consumption [m/s].
' LCount = Index for Do While-Loop [-].
' LCwIndex = Index for Do While-Loop [-].
' dMinTime = Minimum Time [date].
' dMaxTime = Maximum Time [date].
' dTimeVariationPositive =
' dNormalDayStart = Start of day 00:00:00 [date].
' iIntermediate = Intermediate value [-].
' dSimTimeCheck = For checking of time [date].
' dSimStartTime = Start value time [date].
' dSimEndTime = End value time [date].
'
'CALCULATED VALUES/OUTPUTS
' sWFIWaterConsumptionFromLoop() = Global variable [m3/s].
' iProgress = Global variable [-].
'
Dim sWFIWaterConsumptionPrior As Single
Dim LWFITimeEndIndex As Long, LWFITimeStartIndex As Long, LWFITimeStartIndexCounter As Long
Dim LOneDayInSeconds As Long
Dim iCwIndex As Integer, iCount As Integer, iIntermediate As Integer

Dim dMinTime As Date, dMaxTime As Date, dTimeVariationPositive As Date
Dim dSimStartTime As Date, dSimEndTime As Date, dNormalDayStart As Date
Dim dSimTimeCheck  As Date
'
'Read in water offtake from tap:
iCount = 5
Do While iCount < iRowNumberWFI + 5                    'See WFI Input Table.
   If LDayDataCounter = 1 Then
      Sheets("Frontpage - WFI Loop Input Data").Select
   Else
      Sheets("Day " & LDayDataCounter & " WFI Loop Data").Select
   End If
   '
   sWFIWaterDemandPerTap(iCount - 4) = Cells(iCount, 9)     'Cells(rwIndex, colIndex).
   '
   If bOptionNoUncertainty = False Then
      If sWFIWaterDemandPerTap(iCount - 4) > 0 Then
         Call WFIDemandUncertainty(iCount)
      End If
   End If
   iCount = iCount + 1
Loop
'
'Calculate Water Demand for every Water Demand Time Frame (per second):
'
'The three times below are starting times & days of the simulation:
dMinTime = DateSerial(1899, 12, 29) - 22 / 24   '29th Dec.1899 at 22:00:00.
dNormalDayStart = DateSerial(1899, 12, 30)
dMaxTime = DateSerial(1899, 12, 31) + 2 / 24    '30th Dec.1899 plus 2 hours.
'
LOneDayInSeconds = 86400 + 3600 * 4
iCwIndex = 1
iCount = 1
Do While iCwIndex < iColNumberWFI + 1
   Do While iCount < iRowNumberWFI + 1
      'Establish time start & end tap opening:
```

```
dSimStartTime = dNewTimeStart(iCount, iCwIndex)
'
'Select case finds start time values:
LWFITimeStartIndexCounter = 1
Do While LWFITimeStartIndexCounter < LOneDayInSeconds - 1
'Evaluate which simulated time (dSimTime) it is:
   Select Case LWFITimeStartIndexCounter
      Case 1 To 3600 * 2
         '29th Dec.1899 22:00:00 to 29th Dec.1899 24:00:00.
         'Timeserial is negative as all dates before 30th Dec.1899 are negative.
         dSimTimeCheck = dMinTime - TimeSerial(0, 0, LWFITimeStartIndexCounter)
         '
         If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
         And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
            GoTo StartTimeFound
         End If
      Case 3600 * 2 + 1 To 32400
         '30th Dec.1899 00:00:00 to 30th Dec.1899 07:00:00.
         dSimTimeCheck = dNormalDayStart _
         + TimeSerial(0, 0, LWFITimeStartIndexCounter - 7200)
         '
         If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
         And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
            GoTo StartTimeFound
         End If
      Case 32401 To 64800
         '07:00:01 to 16:00:00.
         dSimTimeCheck = dNormalDayStart + TimeSerial(0, 0, 32400 - 7200) _
         + TimeSerial(0, 0, LWFITimeStartIndexCounter - 32400)
         '
         If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
         And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
            GoTo StartTimeFound
         End If
      Case 64801 To 86400 + 3600 * 2 - 1
         '16:00:01 to 23:59:59.
         iIntermediate = LWFITimeStartIndexCounter - 64800
         dSimTimeCheck = dNormalDayStart + TimeSerial(16, 0, 0) _
         + TimeSerial(0, 0, iIntermediate)
         '
         If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
         And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
            GoTo StartTimeFound
         End If
      Case 93600 To 100800 - 1
         '31th Dec.1899 00:00:00 to 31th Dec.1899 2:00:00.
         iIntermediate _
         = CInt(LWFITimeStartIndexCounter - (86400 + 3600 * 2 - 1))
         dSimTimeCheck _
         = DateSerial(1899, 12, 31) + TimeSerial(0, 0, iIntermediate)
         '
         If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
         And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
            GoTo StartTimeFound
         End If
      Case Else         'Other values.
         Stop           'Error trap.
                        'LSecondsOfDaysCount cannot be lower than 1
                        'or higher than 100800-1.  Check program.
         End
   End Select
   '
   LWFITimeStartIndexCounter = LWFITimeStartIndexCounter + 1
Loop
'
StartTimeFound:   'Time of start valve opening found, now find the time of end valve opening:
'
```

```
        dSimEndTime = dNewTimeEnd(iCount, iCwIndex)
        '
        'Select case finds end time values:
        LWFITimeEndIndex = 1
        Do While LWFITimeEndIndex < LOneDayInSeconds - 1
        'Evaluate which simulated time (dSimTime) it is:
          Select Case LWFITimeEndIndex
            Case 1 To 3600 * 2
              '29th Dec.1899 22:00:00 to 29th Dec.1899 24:00:00.
              'Timeserial is negative as all dates before 30th Dec.1899 are negative.
              dSimTimeCheck = dMinTime - TimeSerial(0, 0, LWFITimeEndIndex)
              '
              If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
              And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                GoTo EndTimeFound
              End If
            Case 3600 * 2 + 1 To 32400
              '30th Dec.1899 00:00:00 to 30th Dec.1899 07:00:00.
              dSimTimeCheck _
              = dNormalDayStart + TimeSerial(0, 0, LWFITimeEndIndex - 7200)
              '
              If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
              And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                GoTo EndTimeFound
              End If
            Case 32401 To 64800
              '07:00:01 to 16:00:00.
              dSimTimeCheck = dNormalDayStart + TimeSerial(0, 0, 32400 - 7200) _
              + TimeSerial(0, 0, LWFITimeEndIndex - 32400)
              '
              If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
              And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                GoTo EndTimeFound
              End If
            Case 64801 To 86400 + 3600 * 2 - 1
              '16:00:01 to 23:59:59.
              iIntermediate = LWFITimeEndIndex - 64800
              dSimTimeCheck = dNormalDayStart + TimeSerial(16, 0, 0) _
              + TimeSerial(0, 0, iIntermediate)
              '
              If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
              And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                GoTo EndTimeFound
              End If
            Case 93600 To 100800 - 1
              '31th Dec.1899 00:00:00 to 31th Dec.1899 2:00:00.
              iIntermediate = CInt(LWFITimeEndIndex - (86400 + 3600 * 2 - 1))
              dSimTimeCheck _
              = DateSerial(1899, 12, 31) + TimeSerial(0, 0, iIntermediate)
              '
              If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
              And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                GoTo EndTimeFound
              End If
            Case Else         'Other values.
              Stop            'Error trap.
                              'LSecondsOfDaysCount cannot be lower than 1
                              'or higher than 100800-1.  Check program.
              End
          End Select
          '
          LWFITimeEndIndex = LWFITimeEndIndex + 1
        Loop
EndTimeFound: 'Time of end valve opening found.
        '
        'Find the water consumption from the loop:
        Do While LWFITimeStartIndexCounter < LWFITimeEndIndex
```

```
                sWFIWaterConsumptionPrior = sWFIWaterConsumptionFromLoop(LRepeats, _
                LWFITimeStartIndexCounter, LDayDataCounter)
                'Consumption in m3/s; This is the water consumption from the loop:
                sWFIWaterConsumptionFromLoop(LRepeats, LWFITimeStartIndexCounter, _
                LDayDataCounter) = sWFIWaterConsumptionPrior + _
                sWFIWaterDemandPerTap(iCount) / 3600
                LWFITimeStartIndexCounter = LWFITimeStartIndexCounter + 1
            Loop
            iCount = iCount + 1
        Loop
        iCount = 1
        iCwIndex = iCwIndex + 1
    Loop
    '
    iProgress = iProgress + 1 * sIncrement
    Call Progress_Indicator(iProgress)
    '
End Sub
'
'
'
Sub WFIDemandUncertainty(iCount)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine computes the variation of the WFI flowrate due to uncertainty.  @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' iCount = see calling subroutine [-].
    ' sNormalMean = Mean of Normal Distribution; see sheet "Normal Distribution" [-].
    ' sNormalStanDev = Standard deviation of Normal Dist.; see sheet "Normal Distribution"[-].
    ' sAlphaBeta = Alpha parameter Beta distribution; see sheet "Beta Distribution" [-].
    ' sBetaBeta = Beta parameter Beta distribution; see sheet "Beta Distribution" [-].
    ' strWFIVolumeUncertaintyDistribution = Description of distribution [-].
    '
    'INTERMEDIATE VARIABLES
    ' sInterval = Temporary variable [-].
    ' sRandomVariable = Random variable as per random function [-].
    ' sWFIWaterConsumptionPrior = WFI water consumption [m/s].
    ' sPositiveOrNegative = Random variable to determine negative or positive addition [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sWFIWaterDemandPerTap() = Global variable [m3/s].
    '
    Dim sNormalMean As Single, sNormalStanDev As Single, sPositiveOrNegative As Single
    Dim sAlphaBeta As Single, sBetaBeta As Single, sRandomVariable As Single
    Dim sWFIVolumeUncertainty As Single
    Dim strWFIVolumeUncertaintyDistribution As String
    Dim sInterval As Single
    '
    sWFIVolumeUncertainty = Cells(iCount, 10)
    strWFIVolumeUncertaintyDistribution = Cells(iCount, 11)
    '
    Sheets("Sheet 12 - Normal Distribution").Select
    sNormalMean = Cells(3, 3)
    sNormalStanDev = Cells(4, 3)
    '
    Sheets("Sheet 13 - Beta Distribution").Select
    sAlphaBeta = Cells(3, 3)
    sBetaBeta = Cells(4, 3)
    '
    sRandomVariable = CSng(RandWH())
    sPositiveOrNegative = CSng(RandWH())
    '
    If strWFIVolumeUncertaintyDistribution = "u" Then
        'Uniform Distribution:
        sInterval = sWFIWaterDemandPerTap(iCount - 4) * sWFIVolumeUncertainty / 100
```

```
        '
        'Determine if sInterval is positive or negative:
        If sPositiveOrNegative <= 0.5 Then
            sPositiveOrNegative = (-1)
        Else
            sPositiveOrNegative = 1
        End If
        '
        sWFIWaterDemandPerTap(iCount - 4) = sWFIWaterDemandPerTap(iCount - 4) _
        + sInterval * sRandomVariable * sPositiveOrNegative
    End If
    '
    If strWFIVolumeUncertaintyDistribution = "b" Then
        'Beta Distribution:
        sInterval = sWFIWaterDemandPerTap(iCount - 4) * sWFIVolumeUncertainty / 100
        '
        If sPositiveOrNegative <= 0.5 Then
            sPositiveOrNegative = (-1)
        Else
            sPositiveOrNegative = 1
        End If
        '
        sWFIWaterDemandPerTap(iCount - 4) = sWFIWaterDemandPerTap(iCount - 4) + sInterval _
        * Application.BetaDist(sRandomVariable, sAlphaBeta, sBetaBeta) * sPositiveOrNegative
    End If
    '
    If strWFIVolumeUncertaintyDistribution = "n" Then
        'Normal Distribution:
        sInterval = sWFIWaterDemandPerTap(iCount - 4) * sWFIVolumeUncertainty / 100
        '
        If sPositiveOrNegative <= 0.5 Then
            sPositiveOrNegative = (-1)
        Else
            sPositiveOrNegative = 1
        End If
        '
        sWFIWaterDemandPerTap(iCount - 4) = sWFIWaterDemandPerTap(iCount - 4) + sInterval _
        * Application.NormDist(sRandomVariable, sNormalMean, sNormalStanDev, True) _
        * sPositiveOrNegative
    End If
    '
End Sub
'
'
'
Sub DI_Water_Demand_From_Loop()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the DI water demand from the consumers along the   @
    '@ loop; ignoring any consumption that might be caused by the tank being filled     @
    '@ by DI as the DI Tank may not full at the beginning of the simulation.                @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIWaterDemandPerTap = Global variable [m3/h].
    ' iColNumberDI = Global variable [-].
    ' iRowNumberDI = Global variable [-].
    ' dNewTimeStart = Global variable [-].
    ' dNewTimeEnd = Global variable [-].
    ' LRepeats = Global variable [-].
    ' sIncrement = Global variable [-].
    ' bOptionNoUncertainty = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' sDIWaterConsumptionPrior = [m3/s].
    ' sDITimeStart = [24 hour clock].
    ' sDITimeEnd = [24 hour clock].
```

```
' sDITimeEndIndex = DI Time End Index [-].
' sDITimeStartIndex = Index start time [-].
' sDITimeStartIndexCounter = Index time [-].
' sDIWaterConsumptionPrior = DI water consumption [m/s].
' LCount = Index for Do While-Loop [-].
' LCwIndex = Index for Do While-Loop [-].
' dMinTime = Minimum Time [date].
' dMaxTime = Maximum Time [date].
' dTimeVariationPositive =
' dNormalDayStart = Start of day 00:00:00 [date].
' iIntermediate = Intermediate value [-].
' dSimTimeCheck = For checking of time [date].
' dSimStartTime = Start value time [date].
' dSimEndTime = End value time [date].
'
'CALCULATED VALUES/OUTPUTS
' sDIWaterConsumptionFromLoop() = Global variable [m3/s].
' iProgress = Global variable [-].
'
Dim sDIWaterConsumptionPrior As Single
Dim LDITimeEndIndex As Long, LDITimeStartIndex As Long, LDITimeStartIndexCounter As Long
Dim LOneDayInSeconds As Long, LCount As Long
Dim iCwIndex As Integer, iCount As Integer, iIntermediate As Integer, iTen As Integer
Dim dMinTime As Date, dMaxTime As Date, dTimeVariationPositive As Date
Dim dSimStartTime As Date, dSimEndTime As Date, dNormalDayStart As Date
Dim dSimTimeCheck  As Date
'
'Read in water offtake from tap:
iCount = 5
Do While iCount < iRowNumberDI + 5                      'See DI Input Table.
   Sheets("Day " & LDayDataCounter & " DI Loop Data").Select
   '
   sDIWaterDemandPerTap(iCount - 4) = Cells(iCount, 9)     'Cells(rwIndex, colIndex).
   '
   If bOptionNoUncertainty = False Then
      If sDIWaterDemandPerTap(iCount - 4) > 0 Then
         Call DIDemandUncertainty(iCount)
      End If
   End If
   iCount = iCount + 1
Loop
'
'Calculate Water Demand for every Water Demand Time Frame (per second):
'
'The three times below are starting times & days of the simulation:
dMinTime = DateSerial(1899, 12, 29) - 22 / 24   '29th Dec.1899 at 22:00:00.
dNormalDayStart = DateSerial(1899, 12, 30)
dMaxTime = DateSerial(1899, 12, 31) + 2 / 24    '30th Dec.1899 plus 2 hours.
'
LOneDayInSeconds = 86400 + 3600 * 4
iCwIndex = 1
iCount = 1
Do While iCwIndex < iColNumberDI + 1
   Do While iCount < iRowNumberDI + 1
      'Establish time start & end tap opening:
      dSimStartTime = dNewTimeStart(iCount, iCwIndex)
      '
      'Select case finds start time values:
      LDITimeStartIndexCounter = 1
      Do While LDITimeStartIndexCounter < LOneDayInSeconds - 1
      'Evaluate which simulated time (dSimTime) it is:
         Select Case LDITimeStartIndexCounter
            Case 1 To 3600 * 2
               '29th Dec.1899 22:00:00 to 29th Dec.1899 24:00:00.
               'Timeserial is negative as all dates before 30th Dec.1899 are negative.
               dSimTimeCheck = dMinTime - TimeSerial(0, 0, LDITimeStartIndexCounter)
               '
```

```
                If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
                And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                   GoTo StartTimeFound
                End If
             Case 3600 * 2 + 1 To 32400
                '30th Dec.1899 00:00:00 to 30th Dec.1899 07:00:00.
                dSimTimeCheck = dNormalDayStart _
                + TimeSerial(0, 0, LDITimeStartIndexCounter - 7200)
                '
                If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
                And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                   GoTo StartTimeFound
                End If
             Case 32401 To 64800
                '07:00:01 to 16:00:00.
                dSimTimeCheck = dNormalDayStart + TimeSerial(0, 0, 32400 - 7200) _
                + TimeSerial(0, 0, LDITimeStartIndexCounter - 32400)
                '
                If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
                And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                   GoTo StartTimeFound
                End If
             Case 64801 To 86400 + 3600 * 2 - 1
                '16:00:01 to 23:59:59.
                iIntermediate = LDITimeStartIndexCounter - 64800
                dSimTimeCheck = dNormalDayStart + TimeSerial(16, 0, 0) _
                + TimeSerial(0, 0, iIntermediate)
                '
                If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
                And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                   GoTo StartTimeFound
                End If
             Case 93600 To 100800 - 1
                '31th Dec.1899 00:00:00 to 31th Dec.1899 2:00:00.
                iIntermediate _
                = CInt(LDITimeStartIndexCounter - (86400 + 3600 * 2 - 1))
                dSimTimeCheck _
                = DateSerial(1899, 12, 31) + TimeSerial(0, 0, iIntermediate)
                '
                If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimStartTime _
                And dSimStartTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                   GoTo StartTimeFound
                End If
             Case Else         'Other values.
                Stop           'Error trap.
                               'LSecondsOfDaysCount cannot be lower than 1
                               'or higher than 100800-1.  Check program.
                End
          End Select
          '
          LDITimeStartIndexCounter = LDITimeStartIndexCounter + 1
       Loop
       '
StartTimeFound:   'Time of start valve opening found, now find the time of end valve opening:
       '
       dSimEndTime = dNewTimeEnd(iCount, iCwIndex)
       '
       'Select case finds end time values:
       LDITimeEndIndex = 1
       Do While LDITimeEndIndex < LOneDayInSeconds - 1
       'Evaluate which simulated time (dSimTime) it is:
          Select Case LDITimeEndIndex
             Case 1 To 3600 * 2
                '29th Dec.1899 22:00:00 to 29th Dec.1899 24:00:00.
                'Timeserial is negative as all dates before 30th Dec.1899 are negative.
                dSimTimeCheck = dMinTime - TimeSerial(0, 0, LDITimeEndIndex)
                '
```

```
                    If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
                    And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                       GoTo EndTimeFound
                    End If
                 Case 3600 * 2 + 1 To 32400
                    '30th Dec.1899 00:00:00 to 30th Dec.1899 07:00:00.
                    dSimTimeCheck _
                    = dNormalDayStart + TimeSerial(0, 0, LDITimeEndIndex - 7200)
                    '
                    If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
                    And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                       GoTo EndTimeFound
                    End If
                 Case 32401 To 64800
                    '07:00:01 to 16:00:00.
                    dSimTimeCheck = dNormalDayStart + TimeSerial(0, 0, 32400 - 7200) _
                    + TimeSerial(0, 0, LDITimeEndIndex - 32400)
                    '
                    If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
                    And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                       GoTo EndTimeFound
                    End If
                 Case 64801 To 86400 + 3600 * 2 - 1
                    '16:00:01 to 23:59:59.
                    iIntermediate = LDITimeEndIndex - 64800
                    dSimTimeCheck = dNormalDayStart + TimeSerial(16, 0, 0) _
                    + TimeSerial(0, 0, iIntermediate)
                    '
                    If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
                    And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                       GoTo EndTimeFound
                    End If
                 Case 93600 To 100800 - 1
                    '31th Dec.1899 00:00:00 to 31th Dec.1899 2:00:00.
                    iIntermediate = CInt(LDITimeEndIndex - (86400 + 3600 * 2 - 1))
                    dSimTimeCheck _
                    = DateSerial(1899, 12, 31) + TimeSerial(0, 0, iIntermediate)
                    '
                    If dSimTimeCheck - TimeSerial(0, 0, 1) <= dSimEndTime _
                    And dSimEndTime <= dSimTimeCheck + TimeSerial(0, 0, 1) Then
                       GoTo EndTimeFound
                    End If
                 Case Else      'Other values.
                    Stop        'Error trap.
                                'LSecondsOfDaysCount cannot be lower than 1
                                ' or higher than 100800-1.  Check program.
                    End
              End Select
              '
              LDITimeEndIndex = LDITimeEndIndex + 1
           Loop
EndTimeFound: 'Time of end valve opening found
           '
           'Find the water consumption from the loop:
           Do While LDITimeStartIndexCounter < LDITimeEndIndex
              sDIWaterConsumptionPrior = sDIWaterConsumptionFromLoop(LRepeats, _
              LDITimeStartIndexCounter, LDayDataCounter)
              'Consumption in m3/s; This is the water consumption from the loop:
              sDIWaterConsumptionFromLoop(LRepeats, LDITimeStartIndexCounter, _
              LDayDataCounter) = sDIWaterConsumptionPrior + _
              sDIWaterDemandPerTap(iCount) / 3600
              LDITimeStartIndexCounter = LDITimeStartIndexCounter + 1
           Loop
           iCount = iCount + 1
        Loop
        iCount = 1
        iCwIndex = iCwIndex + 1
```

```
    Loop
    '
    'Add WFI Demand to DI Demand establishing total DI Demand per second and write to
    'Sheet "Sheet 16 - Water Demand DI" for graphical representation of results:
    '
    iTen = 1
    LCount = 3600 * 2 + 1   'Disregard the first two hours of data.  Only use the data of one
                        'day starting at 00:00:00 and ending at 24:00:00.
    Do While LCount < 100800 - 3600 * 2 + 1 'Also disregard the last two hours of data.
       sDIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) _
       = sDIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) _
       + sWFIWaterFromDI(LRepeats, LCount, LDayDataCounter)
       '
       If iTen = 10 Then
          Sheets("Sheet 10 - Results DI Repeat " & LRepeats).Select
          Cells((LCount / 10 - 3600 * 2 / 10 + 3) _
          + 8640 * (LDayDataCounter - 1), LDayDataCounter) _
          = sDIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) * 3600 '[m3/h]
          iTen = 0
       End If
       iTen = iTen + 1
       LCount = LCount + 1
    Loop
    '
    iProgress = iProgress + 3 * sIncrement
    Call Progress_Indicator(iProgress)
    '
End Sub
'
'
'
Sub DIDemandUncertainty(LCount)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine computes the variation of the DI flowrate due to uncertainty.  @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' iCount = see calling subroutine [-].
    ' sNormalMean = Mean of Normal Distribution; see sheet "Normal Distribution" [-].
    ' sNormalStanDev = Standard deviation of Normal Dist.; see sheet "Normal Distribution" [-].
    ' sAlphaBeta = Alpha parameter Beta distribution; see sheet "Beta Distribution" [-].
    ' sBetaBeta = Beta parameter Beta distribution; see sheet "Beta Distribution" [-].
    ' strDIVolumeUncertaintyDistribution = Description of distribution [-].
    '
    'INTERMEDIATE VARIABLES
    ' sInterval = Temporary variable [-].
    ' sRandomVariable = Random variable as per random function [-].
    ' sDITimeEnd = [24 hour clock].
    ' sDIWaterConsumptionPrior = WFI water consumption [m/s].
    ' sPositiveOrNegative = Random variable to determine negative or positive addition [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sDIWaterDemandPerTap() = Global variable [m3/s].
    '
    Dim sNormalMean As Single, sNormalStanDev As Single, sPositiveOrNegative As Single
    Dim sAlphaBeta As Single, sBetaBeta As Single, sRandomVariable As Single
    Dim sDIVolumeUncertainty As Single
    Dim strDIVolumeUncertaintyDistribution As String
    Dim sInterval As Single
    '
    sDIVolumeUncertainty = Cells(LCount, 10)
    strDIVolumeUncertaintyDistribution = Cells(LCount, 11)
    '
    Sheets("Sheet 12 - Normal Distribution").Select
    sNormalMean = Cells(3, 3)
    sNormalStanDev = Cells(4, 3)
```

```
    '
    Sheets("Sheet 13 - Beta Distribution").Select
    sAlphaBeta = Cells(3, 3)
    sBetaBeta = Cells(4, 3)
    '
    sRandomVariable = CSng(RandWH())
    sPositiveOrNegative = CSng(RandWH())
    '
    If strDIVolumeUncertaintyDistribution = "u" Then
      'Uniform Distribution:
      sInterval = sDIWaterDemandPerTap(LCount - 4) * sDIVolumeUncertainty / 100
      '
      'Determine if sInterval is positive or negative:
      If sPositiveOrNegative <= 0.5 Then
        sPositiveOrNegative = (-1)
      Else
        sPositiveOrNegative = 1
      End If
      '
      sDIWaterDemandPerTap(LCount - 4) _
      = sDIWaterDemandPerTap(LCount - 4) + sInterval * sRandomVariable * sPositiveOrNegative
    End If
    '
    If strDIVolumeUncertaintyDistribution = "b" Then
      'Beta Distribution:
      sInterval = sDIWaterDemandPerTap(LCount - 4) * sDIVolumeUncertainty / 100
      '
      If sPositiveOrNegative <= 0.5 Then
        sPositiveOrNegative = (-1)
      Else
        sPositiveOrNegative = 1
      End If
      '
      sDIWaterDemandPerTap(LCount - 4) = sDIWaterDemandPerTap(LCount - 4) + sInterval _
      * Application.BetaDist(sRandomVariable, sAlphaBeta, sBetaBeta) * sPositiveOrNegative
    End If
    '
    If strDIVolumeUncertaintyDistribution = "n" Then
      'Normal Distribution:
      sInterval = sDIWaterDemandPerTap(LCount - 4) * sDIVolumeUncertainty / 100
      '
      If sPositiveOrNegative <= 0.5 Then
        sPositiveOrNegative = (-1)
      Else
        sPositiveOrNegative = 1
      End If
      '
      sDIWaterDemandPerTap(LCount - 4) = sDIWaterDemandPerTap(LCount - 4) + sInterval _
      * Application.NormDist(sRandomVariable, sNormalMean, sNormalStanDev, True) _
      * sPositiveOrNegative
    End If
    '
End Sub
'
'
'
Sub WFI_Water_Demand_From_Generation_And_Tank_Volume()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the Water volume of the WFI in WFI Storage Tank    @
    '@ as a function of time [m3]. Routine also calculates the WFI demand            @
    '@ from filling the WFI water tank if not full at the start of simulation.        @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterFromGen = Global variable  [m3/h].
    ' sWFIStartVolumeTank = Global variable [m3].
```

```
' sWFIMinVolumeTank = Global variable [m3].
' sWFIStartVolumeTank = Global variable [m3].
' sWFIWaterConsumption() = Global variable [m3/s].
' sWFIVolumeTank = WFI Volume in WFI storage tank [m3].
' sWFICalcVolumeTank = Global variable [m3].
' sIncrement = Global variable [-].
' Rho20 and Rho80 = Water density at 20 and 80 degrees [m3/kg].
'
'INTERMEDIATE VARIABLES
' LCount = Index for Do While-Loop [-].
' iTen = Writing every tenth value of the Volume to the sheet for graph etc. [-].
' iI = Index for Progress Indicator [-].
' sDelta = For Euler integration; step width [-].
' LDayDataCounter = Index for Do While-Loop [-].
'
'CALCULATED VALUES/OUTPUTS
' sWFIWaterConsumptionFromLoop(Array) = Global variable [m3/s].
' sWFIVolumeTank = Simulated volume of water in WFI storage tank [m3].
' Messages:  The simulation predicts there will be no WFI available during an
'                  offtake period.  The results of the simulation will be
'                  incorrect. Please change input data on worksheet.
' iProgress = Global variable [-].
'
Dim sDelta As Single, sWFIVolumeTank As Single
Dim iWFITankMinAlarm As Integer, iTen As Integer, iI As Integer, iAlarm As Integer
Dim LCount As Long
Dim strMsg As String, strStyle As String, strTitle As String, strResponse As String
Dim NewSheet As Variant
'
Const Rho20 = 998.21
Const Rho80 = 971.79
'
sDelta = 1
iI = 0
If LDayDataCounter = 1 Then
   Sheets("Sheet 2 - WFI Gen. & Tank Data").Select
   sWFIWaterFromGen = Cells(4, 6)                  '[m3/h].
   sWFICalcVolumeTank = Cells(7, 6)                '[m3].
   sWFIMinVolumeTank = Cells(11, 6)                '[m3].
   sWFIStartVolumeTank = Cells(9, 6)               '[m3].
   sWFIGenBlowdown = Cells(16, 6)                  '[%].
   '
   sWFIWaterFromGen = sWFIWaterFromGen / 3600      '[m3/h] to [m3/s].
   sWFIVolumeTank = sWFIStartVolumeTank            'Start condition.
Else
   Sheets("Sheet 8 - Results WFI Repeat " & LRepeats).Select
   sWFIVolumeTank = Cells(8643, 66 + LDayDataCounter - 1)          '[m3].
End If
'End read in Data.
'
LCount = 3600 * 2 + 1   'Disregard the first two hours of data.  Only use the data of one
                        'day starting at 00:00:00 and ending at 24:00:00.
iTen = 1
Do While LCount < 100800 - 3600 * 2 + 1         'Also disregard the last two hours of data.
   'Mass Balance WFI Storage Tank:
   'Integrate Mass balance below with explicit first order Euler.
   sWFIVolumeTank = sWFIVolumeTank + (sWFIWaterFromGen - _
   sWFIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter)) * sDelta
   '
   If sWFIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) <= _
   sWFIWaterFromGen _
   And _
   sWFIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) <> 0 Then
      'No change in Volume as removal is less than intake from still.
      sWFIWaterFromDI(LRepeats, LCount, LDayDataCounter) _
      = sWFIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) _
      + sWFIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) _
```

```
    * sWFIGenBlowdown / 100
End If
'
If sWFIVolumeTank < sWFICalcVolumeTank Then
    'Max flow of water from generation was added to storage tank:
    sWFIWaterFromDI(LRepeats, LCount, LDayDataCounter) = sWFIWaterFromGen _
    + sWFIGenBlowdown * sWFIWaterFromGen / 100
End If
If sWFIVolumeTank > sWFICalcVolumeTank Then
    'Water Volume in storage tank is at max:
    sWFIVolumeTank = sWFICalcVolumeTank
End If
If sWFIVolumeTank < sWFIMinVolumeTank Then
    'Water Volume is below min allowable:
    '
    Sheets("Sheet 4 - Report WFI Day 1").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available. Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available.  Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available.  Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("Sheet 5 - Report DI Day 1").Select
```

```
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available. Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available. Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("Sheet 7 - Report DI All Days").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available. Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    strMsg = "The simulation predicts there will be a period of no WFI being " + _
    "available. The results of the simulation will be " + _
    " incorrect. The simulation will therefore be terminated." + _
    "  Please change input data on worksheet of the WFI storage tank. "
    strStyle = vbOKOnly
    strTitle = "A critical Message"
    strResponse = MsgBox(strMsg, strStyle, strTitle)
    Call Protect_Sheets_And_Workbook
    End                                 'Terminate Simulation.
End If
'
If iTen = iResolution Then
    iTen = 0
    If LRepeats = 1 Then
        '
        Sheets("Sheet 8 - Results WFI Repeat 1").Select
        '
```

```
            'Write to sheet for graph of water volume in WFI Tank:
            'Multiply WFI water volume by temperature coefficient (Rho80/Rho20).
            Cells((LCount / 10 - 3600 * 2 / 10) + 3, 66 + LDayDataCounter) = _
            sWFIVolumeTank * Rho20 / Rho80                    '[m3]
            '
            'Write DI Water Demand (or WFI generation to Tank) from DI loop to
            'WFI Generation Plant to sheet:
            'No Volume adjustment for temperature required as water temp. is ambient:
            Cells((LCount / 10 - 3600 * 2 / 10) + 3, 33 + LDayDataCounter) = _
            sWFIWaterFromDI(LRepeats, LCount, LDayDataCounter) * 3600          '[m3/h]
            '
            'Write WFI consumption to sheet:
            Cells((LCount / 10 - 3600 * 2 / 10) + 3, LDayDataCounter) = _
            sWFIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) * 3600 '[m3/h]
            '
          Else
            '
            Sheets("Sheet 8 - Results WFI Repeat " & LRepeats).Select
            '
            Cells((LCount / 10 - 3600 * 2 / 10) + 3, 66 + LDayDataCounter) = _
            sWFIVolumeTank * Rho20 / Rho80                                  '[m3]
            '
            'Write DI Water Demand (or WFI generation to Tank) from DI loop to
            'WFI Generation Plant to sheet:
            'No Volume adjustment for temperature required as water temp. is ambient:
            Cells((LCount / 10 - 3600 * 2 / 10) + 3, 33 + LDayDataCounter) = _
            sWFIWaterFromDI(LRepeats, LCount, LDayDataCounter) * 3600          '[m3/h]
            '
            'Write WFI consumption to sheet:
            Cells((LCount / 10 - 3600 * 2 / 10) + 3, LDayDataCounter) = _
            sWFIWaterConsumptionFromLoop(LRepeats, LCount, LDayDataCounter) * 3600 '[m3/h]
            '
          End If
        End If
        '
        If iI = 3930 Then
          iI = 0
          iProgress = iProgress + sIncrement
          Call Progress_Indicator(iProgress)
        End If
        '
        iI = iI + 1
        iTen = iTen + 1
        LCount = LCount + 1
      Loop
      '
      End Sub
'
'
'
Sub DI_Water_Demand_From_Generation_And_Tank_Volume()
   '
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '@ This subroutine calculates the Water volume of the DI in DI Storage Tank     @
   '@ as a function of time [m3]. Routine also calculates the DI demand            @
   '@ from filling the DI water tank if not full at the start of simulation.       @
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '
   'INPUTS
   ' sDIWaterFromGen = DI Water from DI generation to DI storage tank [m3/h].
   ' sDIStartVolumeTank = DI storage tank at start of simulation [m3].
   ' sDIMinVolumeTank = Minimum allowable water Volume in storage tank [m3].
   ' sDIStartVolumeTank = Volume in storage tank at beginning of simulation [m3].
   ' sDIWaterConsumptionFromLoop() = Global variable [m3/s].
   ' sDICalcVolumeTank = DI Volume in storage tank [m3].
   ' sDIVolumeTank = Volume of DI storage tank [m3].
   ' iNumberofRepeats = Global variable [-].
```

```
' sIncrement = Global variable [-].
' iResolution = Global variable [-].
' iDayData = Global variable [-].
'
'INTERMEDIATE VARIABLES
' LCount = Index for Do While-Loop [-].
' iTen = Writing every tenth value of the Volume to the sheet for graph etc. [-].
' iI = Index for Progress Indicator [-].
' sDelta = For Euler integration [-].
' LRepeatsCount = Index for Do While-Loop [-].
' iTen = Writing every tenth value of the Volume to the sheet for graph etc. [-].
' iI = Index for Progress Indicator [-].
' LRepeatsCount = Index for Do While-Loop [-].
' LDayCounter = Index for Do While-Loop [-].
' iColumn = Index [-].
' sTemp = Temporary variable [-].
' LRow = Index for Do While-Loop [-].
' LCount = Index for Do While-Loop [-].
' sTempWaterGentoStorage = Temporary value [-].
' sTempWaterDemand = Temporary value [-].
' sTempWaterVolumeInStorage = Temporary value [-].
'
'CALCULATED VALUES/OUTPUTS
' sDIWaterToTank = DI water flow to storage tank [m3/s].
' iDITankMinAlarm = Alarm to indicate minimum volume of tank has been reached [-].
' Messages:  1. The simulation predicts there will be no DI available during an
'               offtake period.  The results of the simulation will be
'               incorrect. Please change input data on worksheet.
'            2. Simulation Result will be incorrect as there is a
'               period of no DI being available.
' iProgress = Global variable [-].
'
Dim sDIWaterFromGen As Single, sDIStartVolumeTank As Single, sDIMinVolumeTank As Single
Dim sDIWaterToTank() As Single, sTempWaterDemand As Single
Dim sTempWaterGentoStorage As Single, sTempWaterVolumeInStorage As Single
Dim sDICalcVolumeTank As Single, sDIVolumeTank As Single, sDelta As Single, sTemp As Single
Dim iTen As Integer, iI As Integer, LRepeatsCount As Integer, iColumn As Integer
Dim iTwo As Integer
Dim LCount As Long, LRow As Long, LDayCounter As Long
Dim strMsg As String, strStyle As String, strTitle As String, strResponse As String
'
ReDim sDIWaterToTank(iNumberofRepeats, 86400, iDayData)
'
sDelta = 1
LRepeatsCount = 1
iI = 0
Do While LRepeatsCount < iNumberofRepeats + 1
   LDayCounter = 1
   Do While LDayCounter < iDayData + 1
      'Read in Data for calculation of Volume in DI storage tank:
      If LDayCounter = 1 Then
         Sheets("Sheet 3 - DI Gen. & Tank Data").Select
         sDIWaterFromGen = Cells(4, 6)                    '[m3/h].
         sDIWaterFromGen = sDIWaterFromGen / 3600         '[m3/h] to [m3/s].
         sDIVolumeTank = Cells(7, 6)                      '[m3].
         sDIMinVolumeTank = Cells(11, 6)                  '[m3].
         sDIStartVolumeTank = Cells(9, 6)                 '[m3].
         sDICalcVolumeTank = sDIStartVolumeTank           'Start condition.
         'End Read in Data.
      Else
         'Read in the last value of "yesterdays" tank volume:
         Sheets("Sheet 10 - Results DI Repeat " & LRepeatsCount).Select
         sDICalcVolumeTank = Cells(8643, LDayCounter - 1)          '[m3].
         'End Read in Data.
      End If
      '
      LCount = 3600 * 2 + 1   'Disregard the first two hours of data.  Only use the data
```

```
            'of one day starting at 00:00:00 and ending at 24:00:00.
iTen = 1
Do While LCount < 100800 - 3600 * 2 + 1 'Also disregard the last two hours of data.
   'Mass Balance Tank [m3] (every second):
   'Integrate Mass Balance below with explicit first order Euler.
   sDICalcVolumeTank = sDICalcVolumeTank + (sDIWaterFromGen - _
   sDIWaterConsumptionFromLoop(LRepeatsCount, LCount, LDayCounter)) * sDelta
   '
   If (sDIWaterConsumptionFromLoop(LRepeatsCount, LCount, LDayCounter) <= _
      sDIWaterFromGen _
      And _
      sDIWaterConsumptionFromLoop(LRepeatsCount, LCount, LDayCounter) <> 0) _
      Then
      'No change in Volume, but water added [m3/s]:
      sDIWaterToTank(LRepeatsCount, LCount, LDayCounter) = _
      sDIWaterConsumptionFromLoop(LRepeatsCount, LCount, LDayCounter)
   End If
   '
   If sDICalcVolumeTank < sDIVolumeTank Then
      'Max flow of water from generation was added to storage tank [m3/s]:
      sDIWaterToTank(LRepeatsCount, LCount, LDayCounter) = sDIWaterFromGen
   End If
   '
   If sDICalcVolumeTank > sDIVolumeTank Then
      'Water Volume in storage tank at max [m3]:
      sDICalcVolumeTank = sDIVolumeTank
   End If
   '
   If sDICalcVolumeTank < sDIMinVolumeTank Then
      'Water Volume below min allowable [m3/s]:
      sDICalcVolumeTank = sDIMinVolumeTank
      sDIWaterToTank(LRepeatsCount, LCount, LDayCounter) = sDIWaterFromGen
      '
      Sheets("Sheet 5 - Report DI Day 1").Select
      Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
      " period of no DI water being available.  Change Input Data."
      With Selection.Font
         .Name = "Arial"
         .FontStyle = "Bold"
         .Size = 12
         .Strikethrough = False
         .Superscript = False
         .Subscript = False
         .OutlineFont = False
         .Shadow = False
         .Underline = xlUnderlineStyleNone
         .ColorIndex = 3
      End With
      Cells(4, 1).Select
      '
      Sheets("Sheet 7 - Report DI Any Day").Select
      Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
      " period of no DI water being available.  Change Input Data."
      With Selection.Font
         .Name = "Arial"
         .FontStyle = "Bold"
         .Size = 12
         .Strikethrough = False
         .Superscript = False
         .Subscript = False
         .OutlineFont = False
         .Shadow = False
         .Underline = xlUnderlineStyleNone
         .ColorIndex = 3
      End With
      Cells(4, 1).Select
      '
```

```
            Sheets("Sheet 4 - Report WFI Day 1").Select
            Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
            " period of no DI water being available.  Change Input Data."
            With Selection.Font
               .Name = "Arial"
               .FontStyle = "Bold"
               .Size = 12
               .Strikethrough = False
               .Superscript = False
               .Subscript = False
               .OutlineFont = False
               .Shadow = False
               .Underline = xlUnderlineStyleNone
               .ColorIndex = 3
            End With
            Cells(4, 1).Select
            '
            Sheets("Sheet 6 - Report WFI Any Day").Select
            Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
            " period of no DI water being available.  Change Input Data."
            With Selection.Font
               .Name = "Arial"
               .FontStyle = "Bold"
               .Size = 12
               .Strikethrough = False
               .Superscript = False
               .Subscript = False
               .OutlineFont = False
               .Shadow = False
               .Underline = xlUnderlineStyleNone
               .ColorIndex = 3
            End With
            Cells(4, 1).Select
            '
            strMsg = "The simulation predicts there will be a period of no DI " + _
            "being available. The results of the simulation will be " + _
            " incorrect. The simulation will therefore be terminated." + _
            " Please change input data on worksheet of the DI storage tank. "
            '
            strStyle = vbOKOnly
            strTitle = "A critical Message"
            strResponse = MsgBox(strMsg, strStyle, strTitle)
            Call Protect_Sheets_And_Workbook
            End                         'Terminate simulation.
         End If
         If iTen = iResolution Then
            'Write to sheet for graph of Water Volume in DI Tank:
            Sheets("Sheet 10 - Results DI Repeat " & LRepeatsCount).Select
            Cells((LCount / 10) - 3600 * 2 / 10 + 3, 66 + LDayCounter) _
            = sDICalcVolumeTank '[m3]
            iTen = 0
         End If
         If iI = 7600 Then
            iI = 0
            iProgress = iProgress + sIncrement
            Call Progress_Indicator(iProgress)
         End If
         '
         iI = iI + 1
         iTen = iTen + 1
         LCount = LCount + 1
      Loop
      LDayCounter = LDayCounter + 1
   Loop
   LRepeatsCount = LRepeatsCount + 1
Loop
'
```

```
'Write data DI Water to Tank to sheet:
LRepeatsCount = 1
iI = 0
Do While LRepeatsCount < iNumberofRepeats + 1
   Sheets("Sheet 10 - Results DI Repeat " & LRepeatsCount).Select
   LDayCounter = 1
   Do While LDayCounter < iDayData + 1
      '
      iTen = 0
      LCount = 1
      Do While LCount < 86400 + 1
         If iTen = 10 Then
            iTen = 0
            Cells((LCount / 10 + 3), 33 + LDayCounter) _
            = sDIWaterToTank(LRepeatsCount, LCount, LDayCounter) * 3600 '[m3/h]
         End If
         If iI = 7600 Then
            iI = 0
            iProgress = iProgress + sIncrement
            Call Progress_Indicator(iProgress)
         End If
         iI = iI + 1
         iTen = iTen + 1
         LCount = LCount + 1
      Loop
      LDayCounter = LDayCounter + 1
   Loop
   LRepeatsCount = LRepeatsCount + 1
Loop
'
'Write data to sheets with lower resolution to display graphs over a couple of days:
LRepeats = 1
Do While LRepeats < iNumberofRepeats + 1  'Loop for the repeat.
   iColumn = 1
   LRow = 1
   iTwo = 2
   'Write data from one sheet to other:
   Do While iColumn < iDayData + 1
      Do While LRow < 8640
         If LRow = 1 Then
            '
            Sheets("Sheet 10 - Results DI Repeat " & LRepeats).Select
            sTempWaterDemand = Cells((LRow + 3), iColumn)                    '[m3/h].
            sTempWaterGentoStorage = Cells((LRow + 3), iColumn + 33)         '[m3/h].
            sTempWaterVolumeInStorage = Cells((LRow + 3), iColumn + 66)   '[m3].
            '
            Sheets("11 Results DI - All - Repeat " & LRepeats).Select
            '
            Range("A4").Select
            ActiveCell.Value = sTempWaterDemand
            '
            Range("E4").Select
            ActiveCell.Value = sTempWaterGentoStorage
            '
            Range("I4").Select
            ActiveCell.Value = sTempWaterVolumeInStorage
            '
         End If
         '
         If iTwo = 2 Then       'Every 2nd datapoint only, as the Excel
                                'graphs cannot display more than 32,000
                                'datapoints.
            '
            Sheets("Sheet 10 - Results DI Repeat " & LRepeats).Select
            sTempWaterDemand = Cells((LRow + 3), iColumn)                    '[m3/h].
            sTempWaterGentoStorage = Cells((LRow + 3), iColumn + 33)         '[m3/h].
            sTempWaterVolumeInStorage = Cells((LRow + 3), iColumn + 66)   '[m3].
```

```
                '
                Sheets("11 Results DI - All - Repeat " & LRepeats).Select
                '
                Range("A3").Select
                'Find last entry into column
                Selection.End(xlDown).Select
                ActiveCell.Offset(rowOffset:=1, columnOffset:=0).Activate
                ActiveCell.Value = sTempWaterDemand                    '[m3/h].
                '
                Range("E3").Select
                'Find last entry into column
                Selection.End(xlDown).Select
                ActiveCell.Offset(rowOffset:=1, columnOffset:=0).Activate
                ActiveCell.Value = sTempWaterGentoStorage              '[m3/h].
                '
                Range("I3").Select
                'Find last entry into column
                Selection.End(xlDown).Select
                ActiveCell.Offset(rowOffset:=1, columnOffset:=0).Activate
                ActiveCell.Value = sTempWaterVolumeInStorage           '[m3].
                '
                iTwo = 0
            End If
            iTwo = iTwo + 1
            LRow = LRow + 1
        Loop
        iColumn = iColumn + 1
    Loop
    LRepeats = LRepeats + 1
Loop
'
End Sub
'
'
'
Sub Time_DI_Low()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This Subroutine calculates by how long the volume in the DI water storage tank     @
    '@ is below a user defined volume over the time horizon of the simulation.            @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIVolumeNotToBeBelow = Volume in Tank shall not be below this Volume [m3].
    ' iNumberofRepeat = Number of Repeats of Simulation [-].
    ' iNumberOfDays = Number of days of the simulation [-].
    '
    'INTERMEDIATE VARIABLES
    ' iCount = Index for Do While-Loop [-].
    ' iRow = Index for Do While-Loop [-].
    ' iRepeat = Index for Do While-Loop [-].
    ' iColumn = Index for Do While-Loop [-].
    ' LNumberOfVolumesBelow = Predicted number of Volumes below the user input [-].
    ' strRepeatSelected = Converted value [-].
    ' sWFIVolume = Simulated volume in DI Tank [m3].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sTimeBelowDI = Estimated time that Volume in tank is below user defined volume [-].
    '
    Dim sDIVolumeNotToBeBelow As Single, sTimeBelowDI As Single, sDIVolume As Single
    Dim LNumberOfEventsBelowMin As Long
    Dim iColumn As Integer, iRow As Integer, iCount As Integer
    Dim iRepeat As Integer
    Dim iNumberofRepeats As Integer, iNumberOfDays As Integer
    Dim strRepeatSelected As String
    '
    Application.ScreenUpdating = False
```

```
    '
    'Read in Data:
    Sheets("Sheet 7 - Report DI Any Day").Select
    sDIVolumeNotToBeBelow = Cells(48, 25)
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    iNumberofRepeats = Cells(13, 2)
    iNumberOfDays = Cells(14, 2)
    '
    'Calculate the average and minimum Volume in the DI Storage Tank [m3]:
    'Read in Data for Volume in DI Tank [m3]:
    LNumberOfEventsBelowMin = 0
    iCount = 1
    iRepeat = 1
    Do While iRepeat < iNumberofRepeats + 1
        '
        'Find the correct sheet:
        strRepeatSelected = CStr(iRepeat)
        Sheets("Sheet 10 - Results DI Repeat " & strRepeatSelected).Select
        '
        'Read in the Data:
        iColumn = 1
        Do While iColumn < iNumberOfDays + 1
            iRow = 1
            Do While iRow < 8640 + 1
                sDIVolume = Cells(iRow + 3, iCount + 66)
                If sDIVolume < sDIVolumeNotToBeBelow Then
                    LNumberOfEventsBelowMin = LNumberOfEventsBelowMin + 1
                    End If
                iRow = iRow + 1
            Loop
            iColumn = iColumn + 1
        Loop
        iRepeat = iRepeat + 1
    Loop
    'End Read in Data.
    '
    'Predicted time in minutes that level may be below user input:
    sTimeBelowDI = LNumberOfEventsBelowMin / iNumberOfDays / iNumberofRepeats * 10 / 60
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    Cells(50, 25) = sTimeBelowDI
    '
    Application.ScreenUpdating = True
    '
End Sub
'
'
'
Sub Time_WFI_Low()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This Subroutine calculates by how long the volume in the WFI water storage tank      @
    '@ is below a user defined volume over the time horizon of the simulation.              @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIVolumeNotToBeBelow = Volume in Tank shall not be below this Volume [m3].
    ' iNumberofRepeat = Number of Repeats of Simulation [-].
    ' iNumberOfDays = Number of days of the simulation [-].
    '
    'INTERMEDIATE VARIABLES
    ' iCount = Index for Do While-Loop [-].
    ' iRow = Index for Do While-Loop [-].
    ' iRepeat = Index for Do While-Loop [-].
    ' iColumn = Index for Do While-Loop [-].
    ' LNumberOfVolumesBelow = Predicted number of Volumes below the user input [-].
```

```
    ' strRepeatSelected = Converted value [-].
    ' sWFIVolume = Simulated volume in WFI Tank [m3].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sTimeBelowWFI = Estimated time that Volume in tank is below user defined volume [-].
    '
    Dim sWFIVolumeNotToBeBelow As Single, sTimeBelowWFI As Single, sWFIVolume As Single
    Dim LNumberOfEventsBelowMin As Long
    Dim iColumn As Integer, iRow As Integer, iCount As Integer
    Dim iRepeat As Integer
    Dim iNumberofRepeats As Integer, iNumberOfDays As Integer
    Dim strRepeatSelected As String
    '
    Application.ScreenUpdating = False
    '
    'Read in Data:
    Sheets("Sheet 6 - Report WFI Any Day").Select
    sWFIVolumeNotToBeBelow = Cells(48, 25)
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    iNumberofRepeats = Cells(13, 2)
    iNumberOfDays = Cells(14, 2)
    '
    'Calculate the average and minimum Volume in the WFI Storage Tank [m3]:
    'Read in Data for Volume in WFI Tank [m3]:
    LNumberOfEventsBelowMin = 0
    iCount = 1
    iRepeat = 1
    Do While iRepeat < iNumberofRepeats + 1
        '
        'Find the correct sheet:
        strRepeatSelected = CStr(iRepeat)
        Sheets("Sheet 8 - Results WFI Repeat " & strRepeatSelected).Select
        '
        'Read in the Data:
        iColumn = 1
        Do While iColumn < iNumberOfDays + 1
            iRow = 1
            Do While iRow < 8640 + 1
                sWFIVolume = Cells(iRow + 3, iCount + 66)
                If sWFIVolume < sWFIVolumeNotToBeBelow Then
                    LNumberOfEventsBelowMin = LNumberOfEventsBelowMin + 1
                    End If
                iRow = iRow + 1
            Loop
            iColumn = iColumn + 1
        Loop
        iRepeat = iRepeat + 1
    Loop
    'End Read in Data.
    '
    'Predicted time in minutes that level may be below user input:
    sTimeBelowWFI = LNumberOfEventsBelowMin / iNumberOfDays / iNumberofRepeats * 10 / 60
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    Cells(50, 25) = sTimeBelowWFI
    '
    Application.ScreenUpdating = True
    '
End Sub
'
'
```

```
'
Sub Analysis_of_Data_WFI_Day_1()
  '
  '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  '@ This subroutine analyses the simulation output for WFI data of Day 1 /          @
  '@ Repeat 1 and writes the data  to the Sheet "Sheet 8 - Results WFI Repeat 1".   @
  '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  '
  'INPUTS
  ' sWFIWaterFromGen = WFI from Generation Plant [m3/s].
  ' sWFICalcVolumeTank = Volume of WFI storage tank [m3].
  ' sWFIMinVolumeTank = Min. allowable volume of tank [m3].
  ' sWFIStartVolumeTank = Start volume in WFI storage tank [m3].
  '
  'INTERMEDIATE VARIABLES
  ' iRowNumber = Number of rows as per Input Table [-].
  ' iCount = Index for DoWhile Loop [-].
  ' iNCount = Calculation of WFI generation utilisation [-].
  ' sWFIVolume() = Volume in WFI storage tank [-].
  ' sValue = Calculation of WFI generation utilisation [-].
  ' sTemp = Temporary value [-].
  ' iLocationOfPercentile = For calculation of percentile [-].
  '
  'CALCULATED VALUES/OUTPUTS
  ' sMin = Minimum value [various].
  ' sMax = Maximum value [various].
  ' sAverage = Average value [various].
  ' sWFISum = Overall WFI consumption from loop excluding tank filling [m3].
  '
  Dim sWFIWaterFromGen As Single, sWFICalcVolumeTank As Single
  Dim sWFIStartVolumeTank As Single, sWFIMinVolumeTank As Single
  Dim sMin As Single, sMinTemp As Single, sMax As Single, sTemp As Single
  Dim sWFIVolume(8650) As Single, sWFILoopConsumption(8650) As Single
  Dim sAverage As Single, sValue As Single, sWFISum As Single
  Dim iNCount As Integer, iRowNumber As Integer, iCount As Integer
  Dim iLocationOfPercentile As Integer, sWFIFromGen(8650) As Single
  '
  'Read in Data:
  Sheets("Sheet 2 - WFI Gen. & Tank Data").Select
  sWFIWaterFromGen = Cells(4, 6)                  '[m3/h].
  sWFICalcVolumeTank = Cells(7, 6)                '[m3].
  sWFIMinVolumeTank = Cells(11, 6)                '[m3].
  sWFIStartVolumeTank = Cells(9, 6)               '[m3].
  'End Read in Data.
  '
  Sheets("Sheet 4 - Report WFI Day 1").Select
  Cells(4, 7) = sWFIWaterFromGen
  Cells(5, 7) = sWFICalcVolumeTank
  Cells(7, 7) = sWFIMinVolumeTank
  Cells(6, 7) = sWFIStartVolumeTank
  '
  'Read in Data for Volume in WFI Tank [m3]:
  Sheets("Sheet 8 - Results WFI Repeat 1").Select
  iCount = 1
  Do While iCount < 8640 + 1
    sWFIVolume(iCount) = Cells(iCount + 3, 67)
    iCount = iCount + 1
  Loop
  'End Read in Data.
  '
  'Calculate the minimum Volume in WFI Tank [m3]:
  sMin = sWFIVolume(1)
  iCount = 2
  Do While iCount < 8640 + 1
    sTemp = sWFIVolume(iCount)
    If sTemp < sMin Then
      sMin = sTemp
```

```
    End If
    iCount = iCount + 1
Loop
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q4") = sMin
'
'Find the maximum Volume in WFI Tank [m3].
sMax = sWFIVolume(1)
iCount = 2
Do While iCount < 8640 + 1
    sTemp = sWFIVolume(iCount)
    If sTemp > sMax Then
        sMax = sTemp
    End If
    iCount = iCount + 1
Loop
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q5") = sMax
'
'Calculate the range of the Volume in the storage tank [m3]:
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q6") = sMax - sMin
'
'Calculate the average Volume in the WFI Storage Tank [m3]:
Sheets("Sheet 8 - Results WFI Repeat 1").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, 67)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q8") = sAverage
'
'Calculate the average water flow from the WFI generation plant [m3/h]:
Sheets("Sheet 8 - Results WFI Repeat 1").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, 34)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q9") = sAverage
'
'Calculate the average offtake from the WFI loop [m3/h]:
Sheets("Sheet 8 - Results WFI Repeat 1").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, 1)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q10") = sAverage
'
'Read in Data for Volume in WFI Tank [m3]:
Sheets("Sheet 8 - Results WFI Repeat 1").Select
iCount = 1
Do While iCount < 8640 + 1
    sWFIVolume(iCount) = Cells(iCount + 3, 67)
    iCount = iCount + 1
Loop
'End Read in Data.
```

```
'
'Sort algorithm for sWFIVolume(iCount):
iCount = 2
Do While iCount < 8640 + 1
   sTemp = sWFIVolume(iCount)
   For iRowNumber = iCount - 1 To 1 Step -1
      If (sWFIVolume(iRowNumber) <= sTemp) Then
         GoTo 10
      End If
      sWFIVolume(iRowNumber + 1) = sWFIVolume(iRowNumber)
   Next iRowNumber
   iRowNumber = 0
10    sWFIVolume(iRowNumber + 1) = sTemp
iCount = iCount + 1
Loop
'
'Calculate the 90% percentile WFI Water Volume in Tank [m3]:
iLocationOfPercentile = Int(90 / 100 * 8640)
iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q12") = sWFIVolume(iLocationOfPercentile)
'
'Calculate the 60% percentile WFI Water Volume in Tank [m3]:
iLocationOfPercentile = Int(60 / 100 * 8640)
iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q13") = sWFIVolume(iLocationOfPercentile)
'
'Calculate the overall WFI Water Consumption from Loop excluding the tank [m3]:
Sheets("Sheet 8 - Results WFI Repeat 1").Select
sWFISum = 0
iCount = 4
Do While iCount < 8640 + 1
   sWFISum = sWFISum + Cells(iCount, 1) / 3600 * 10
   iCount = iCount + 1
Loop
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q15") = sWFISum
'
'Calculate the overall WFI Water to Storage Tank [m3]:
Sheets("Sheet 8 - Results WFI Repeat 1").Select
sWFISum = 0
iCount = 4
Do While iCount < 8340 + 1
   sWFISum = sWFISum + Cells(iCount, 34) / 3600 * 10
   iCount = iCount + 1
Loop
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q16") = sWFISum
'
'Calculate the WFI Generation utilisation [-]:
Sheets("Sheet 8 - Results WFI Repeat 1").Select
iNCount = 0
iCount = 4
Do While iCount < 8640 + 1
   sValue = Cells(iCount, 34)
   If sValue <> 0 Then
      iNCount = iNCount + 1
   End If
   iCount = iCount + 1
Loop
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("Q18") = iNCount / 8640
Range("B5").Select
'
End Sub
'
```

```
'
'
Sub Analysis_of_Data_DI_Day_1()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine analyses the simulation output for WFI data of Day 1          @
    '@ / Repeat 1 and writes the data to the Sheet "Sheet 10 - Results DI Repeat 1".   @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWDIaterFromGen = DI from Generation Plant [m3/s].
    ' sDIVolumeTank = Volume of DI storage tank [m3].
    ' sDIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sDIStartVolumeTank = Start volume in DI storage tank [m3].
    ' sDIVolume() = Volume in DI storage tank [m3].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' iNCount = Calculation of DI generation utilisation [-].
    ' sValue = Calculation of WFI generation utilisation [-].
    ' sTemp = Temporary value [various].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sMin = Minimum value [various].
    ' sMax = Maximum value [various].
    ' sAverage = Average value [various].
    ' sDISum = Overall WFI consumption from loop excluding tank filling [m3].
    ' iLocationOfPercentile = For calculation of 60 and 90 % percentile [m3].
    '
    Dim sDIWaterFromGen As Single, sDIVolumeTank As Single
    Dim sDIMinVolumeTank As Single, sDIStartVolumeTank As Single, sAverage As Single
    Dim iRowNumber As Integer, iCount As Integer, iNCount As Integer
    Dim sDIVolume(8640) As Single, sDIFlow(8640) As Single, sDISum As Single
    Dim sDILoopConsumption(8640) As Single, sDIFromGen(8650) As Single
    Dim sMin As Single, sMax As Single, sValue As Single, sTemp As Single
    Dim iLocationOfPercentile As Integer
    '
    'Read in Data:
    Sheets("Sheet 3 - DI Gen. & Tank Data").Select
    sDIWaterFromGen = Cells(4, 6)                '[m3/h].
    sDIVolumeTank = Cells(7, 6)                  '[m3].
    sDIMinVolumeTank = Cells(11, 6)              '[m3].
    sDIStartVolumeTank = Cells(9, 6)             '[m3].
    'End Read in Data.
    '
    Sheets("Sheet 5 - Report DI Day 1").Select
    '
    Cells(4, 7) = sDIWaterFromGen
    Cells(5, 7) = sDIVolumeTank
    Cells(6, 7) = sDIStartVolumeTank
    Cells(7, 7) = sDIMinVolumeTank
    '
    'Read in Data for Volume in DI Tank [m3]:
    Sheets("Sheet 10 - Results DI Repeat 1").Select
    iCount = 1
    Do While iCount < 8640 + 1
        sDIVolume(iCount) = Cells(iCount + 3, 67)
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Find the minimum Volume in DI Tank [m3]:
    sMin = sDIVolume(1)
    iCount = 1
    Do While iCount < 8640 + 1
        sTemp = sDIVolume(iCount)
```

```
    If sTemp < sMin Then
        sMin = sTemp
    End If
    iCount = iCount + 1
Loop
Sheets("Sheet 5 - Report DI Day 1").Select
Range("Q4") = sMin
'
'Find the maximum Volume in DI Tank [m3]:
sMax = sDIVolume(1)
iCount = 1
Do While iCount < 8640 + 1
    sTemp = sDIVolume(iCount)
    If sTemp > sMin Then
        sMax = sTemp
    End If
    iCount = iCount + 1
Loop
Sheets("Sheet 5 - Report DI Day 1").Select
Range("Q5") = sMax
'
'Calculate the range of the Volume in the storage tank [m3]:
Sheets("Sheet 5 - Report DI Day 1").Select
Range("Q6") = sMax - sMin
'
'Calculate the average Volume in DI Tank [m3]:
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + sDIVolume(iCount)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 5 - Report DI Day 1").Select
Range("Q8") = sAverage
'
'Calculate the average water flow from the DI generation plant [m3/h]:
Sheets("Sheet 10 - Results DI Repeat 1").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, 34)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 5 - Report DI Day 1").Select
Range("Q10") = sAverage
'
'Calculate the average offtake from the DI loop [m3/h]:
Sheets("Sheet 10 - Results DI Repeat 1").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, 1)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 5 - Report DI Day 1").Select
Range("Q9") = sAverage
'
'Sort algorithm for vDIVolume(iCount):
iCount = 2
Do While iCount < 8640 + 1
    sTemp = sDIVolume(iCount)
    For iRowNumber = iCount - 1 To 1 Step -1
        If (sDIVolume(iRowNumber) <= sTemp) Then
            GoTo 10
```

```
            End If
            sDIVolume(iRowNumber + 1) = sDIVolume(iRowNumber)
        Next iRowNumber
        iRowNumber = 0
10      sDIVolume(iRowNumber + 1) = sTemp
    iCount = iCount + 1
    Loop
    '
    'Calculate the 90% percentile DI Water Volume in Tank [m3]:
    iLocationOfPercentile = Int(90 / 100 * 8640)
    iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
    Sheets("Sheet 5 - Report DI Day 1").Select
    Range("Q12") = sDIVolume(iLocationOfPercentile)
    '
    'Calculate the 60% percentile DI Water Volume in Tank [m3]:
    iLocationOfPercentile = Int(60 / 100 * 8640)
    iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
    Sheets("Sheet 5 - Report DI Day 1").Select
    Range("Q13") = sDIVolume(iLocationOfPercentile)
    '
    'Read in Data for DI Water Loop Consumption [m3/h]:
    Sheets("Sheet 10 - Results DI Repeat 1").Select
    iCount = 1
    Do While iCount < 8640 + 1
        sDILoopConsumption(iCount) = Cells(iCount + 3, 1)
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Sort algorithm for sDILoopConsumption(iCount) [m3/s]:
    iCount = 2
    Do While iCount < 8640 + 1
        sTemp = sDILoopConsumption(iCount)
        For iRowNumber = iCount - 1 To 1 Step -1
            If (sDILoopConsumption(iRowNumber) <= sTemp) Then
                GoTo 20
            End If
            sDILoopConsumption(iRowNumber + 1) = sDILoopConsumption(iRowNumber)
        Next iRowNumber
        iRowNumber = 0
20      sDILoopConsumption(iRowNumber + 1) = sTemp
    iCount = iCount + 1
    Loop
    '
    'Read in Data for DI Water from Generation Plant [m3/h]:
    Sheets("Sheet 10 - Results DI Repeat 1").Select
    iCount = 1
    Do While iCount < 8640 + 1
        sDIFromGen(iCount) = Cells(iCount + 3, 34)
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Sort algorithm for sDILoopConsumption(iCount) [m3/h]:
    iCount = 2
    Do While iCount < 8640 + 1
        sTemp = sDIFromGen(iCount)
        For iRowNumber = iCount - 1 To 1 Step -1
            If (sDIFromGen(iRowNumber) <= sTemp) Then
                GoTo 30
            End If
            sDIFromGen(iRowNumber + 1) = sDIFromGen(iRowNumber)
        Next iRowNumber
        iRowNumber = 0
30      sDIFromGen(iRowNumber + 1) = sTemp
    iCount = iCount + 1
    Loop
```

```
    '
    'Calculate the overall DI Water Consumption from Loop excluding the tank [m3]:
    Sheets("Sheet 10 - Results DI Repeat 1").Select
    sDISum = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sDISum = sDISum + Cells(iCount, 1) / 3600 * 10
        iCount = iCount + 1
    Loop
    Sheets("Sheet 5 - Report DI Day 1").Select
    Range("Q15") = sDISum
    '
    'Calculate the overall DI Water to Storage Tank [m3]:
    Sheets("Sheet 10 - Results DI Repeat 1").Select
    sDISum = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sDISum = sDISum + Cells(iCount, 34) / 3600 * 10
        iCount = iCount + 1
    Loop
    Sheets("Sheet 5 - Report DI Day 1").Select
    Range("Q16") = sDISum
    '
    'Calculate the DI Generation utilisation [-]:
    Sheets("Sheet 10 - Results DI Repeat 1").Select
    iNCount = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sValue = Cells(iCount, 34)
        If sValue <> 0 Then
            iNCount = iNCount + 1
        End If
        iCount = iCount + 1
    Loop
    Sheets("Sheet 5 - Report DI Day 1").Select
    Range("Q18") = iNCount / 8640
    Range("B5").Select
    '
End Sub
'
'
'
Sub Analysis_WFI_Other_Days_and_Repeats()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine analyses all data of the WFI data for the day and repeat as        @
    '@ selected by the user.                                                              @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterFromGen = WFI from Generation Plant [m3/s].
    ' sWFICalcVolumeTank = Volume of WFI storage tank [m3].
    ' sWFIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sWFIStartVolumeTank = Start volume in WFI storage tank [m3].
    ' sWFISimulatedDay = Number of simulated day [-].
    ' sWFINumberOfRepeat = Number of repeat [-].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' iNCount = Calculation of WFI generation utilisation [-].
    ' sWFIVolume() = Volume in WFI storage tank [-].
    ' LUtilcount = Calculation of WFI generation utilisation [-].
    ' sTemp = Temporary value [-].
    ' strRepeatSelected = String value of sWFINumberOfRepeat [-].
    '
    'CALCULATED VALUES/OUTPUTS
```

```
' sMin = Minimum value [various].
' sMedian = Median value [various].
' sMax = Maximum value [various].
' sAverage = Average value [various].
' s90Volume = 90% percentile of tank volume [m3].
' sDISum = Overall WFI consumption from loop excluding tank filling [m3].
'
Dim strRepeatSelected As String
Dim sMin As Single, sValue As Single
Dim sMax As Single
Dim sAverage As Single, sTemp As Single
Dim sWFIVolume(8640) As Single, sWFIGenToStorage(8640) As Single
Dim sWFIDemand(8640) As Single
Dim sWFISum As Single, sWFIWaterFromGen As Single, sWFIStartVolumeTank As Single
Dim sWFICalcVolumeTank As Single, sWFIMinVolumeTank As Single
Dim iRowNumber As Integer, iCount As Integer
Dim iDaySelected As Integer, iTemp As Integer
Dim sWFISimulatedDay As Single, sWFINumberOfRepeat As Single
Dim LCount As Long, LRowNumber As Long, LLocationOfPercentile As Long, LUtilcount As Long
'
Application.ScreenUpdating = False
'
Call UnProtect_Sheets_and_Workbook_No_Message
'
'Read in Data:
Sheets("Sheet 2 - WFI Gen. & Tank Data").Select
sWFIWaterFromGen = Cells(4, 6)
sWFICalcVolumeTank = Cells(7, 6)
sWFIMinVolumeTank = Cells(11, 6)
sWFIStartVolumeTank = Cells(9, 6)
'
Sheets("Sheet 6 - Report WFI Any Day").Select
sWFISimulatedDay = Cells(39, 25)
sWFINumberOfRepeat = Cells(41, 25)
'
Range("G4") = sWFIWaterFromGen
Range("G5") = sWFICalcVolumeTank
Range("G6") = sWFIMinVolumeTank
Range("G7") = sWFIStartVolumeTank
'
strRepeatSelected = CStr(sWFINumberOfRepeat)
'
'Make header of sheet:
Range("A1:AD1").Select
ActiveCell.FormulaR1C1 = "Results of the WFI Simulation Day: " & sWFISimulatedDay & _
"; Repeat No.: " & strRepeatSelected
With ActiveCell.Characters(Start:=1, Length:=36).Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 20
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
'
Sheets("Sheet 8 - Results WFI Repeat " & strRepeatSelected).Select
'Calculate the average and minimum Volume in the WFI Storage Tank [m3]:
'Read in Data for Volume in WFI Tank [m3]:
iCount = 1
Do While iCount < 8640 + 1
    sWFIDemand(iCount) = Cells(iCount + 3, sWFISimulatedDay)
    sWFIGenToStorage(iCount) = Cells(iCount + 3, 33 + sWFISimulatedDay)
    sWFIVolume(iCount) = Cells(iCount + 3, 66 + sWFISimulatedDay)
```

```
    iCount = iCount + 1
Loop
'End Read in Data.
'
'Write Data to sheet for graphical display:
Sheets("Sheet 6 - Report WFI Any Day").Select
'
iCount = 1
Do While iCount < 8640 + 1
    Cells(iCount + 3, 53) = sWFIDemand(iCount)
    Cells(iCount + 3, 57) = sWFIGenToStorage(iCount)
    Cells(iCount + 3, 61) = sWFIVolume(iCount)
    iCount = iCount + 1
Loop
'End write data for graphical display.
'
'Calculate the minimum and average Volume in the WFI Tank [m3]:
sMin = sWFIVolume(1)
sAverage = 0
iCount = 2
Do While iCount < 8640 + 1
    sTemp = sWFIVolume(iCount)
    If sTemp < sMin Then
        sMin = sTemp
    End If
    sAverage = sAverage + sWFIVolume(iCount)
    iCount = iCount + 1
Loop
'
Sheets("Sheet 6 - Report WFI Any Day").Select
Range("O4") = sMin
Range("Y60") = sMin
sAverage = sAverage / CSng(8640)
Range("O6") = sAverage
Range("Y64") = sAverage
'
'Find the maximum Volume in the WFI Tank [m3]:
sMax = sWFIVolume(1)
iCount = 2
Do While iCount < 8640 + 1
    sTemp = sWFIVolume(iCount)
    If sTemp > sMax Then
        sMax = sTemp
    End If
    iCount = iCount + 1
Loop
'
Sheets("Sheet 6 - Report WFI Any Day").Select
Range("O5") = sMax
Range("Y61") = sMax
'
'Calculate the 90% percentile WFI Water Volume in Tank [m3]:
'Read in Data:
Sheets("Sheet 8 - Results WFI Repeat " & strRepeatSelected).Select
iCount = 1
Do While iCount < 8640 + 1
    sWFIVolume(iCount) = Cells(iCount + 3, 66 + sWFISimulatedDay)
    iCount = iCount + 1
Loop
'End read in data.
'
'Sort algorithm for sWFIVolume(iCount):
LCount = 2
Do While LCount < CLng(8640)
    sTemp = sWFIVolume(LCount)
    For LRowNumber = LCount - 1 To 1 Step -1
        If (sWFIVolume(LRowNumber) <= sTemp) Then
```

```
            GoTo 10
          End If
          sWFIVolume(LRowNumber + 1) = sWFIVolume(LRowNumber)
        Next LRowNumber
        LRowNumber = 0
10      sWFIVolume(LRowNumber + 1) = sTemp
        LCount = LCount + 1
    Loop
    LLocationOfPercentile = Int(90 / 100 * CLng(8640))
    LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    Range("O8") = sWFIVolume(LLocationOfPercentile)
    Range("Y68") = sWFIVolume(LLocationOfPercentile)
    '
    'Calculate the 60% percentile WFI Water Volume in Tank [m3]:
    LLocationOfPercentile = Int(60 / 100 * CLng(8640))
    LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    Range("O7") = sWFIVolume(LLocationOfPercentile)
    Range("Y69") = sWFIVolume(LLocationOfPercentile)
    '
    'Calculate the WFI Generation utilisation [-]:
    'Read in data and calculate:
    Sheets("Sheet 8 - Results WFI Repeat " & strRepeatSelected).Select
    LUtilcount = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sValue = Cells(iCount, 33 + sWFISimulatedDay)
        If sValue <> 0 Then
            LUtilcount = LUtilcount + 1
        End If
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    Range("V8") = LUtilcount / (CLng(8640))
    Range("Y74") = LUtilcount / (CLng(8640))
    Range("A5").Select
    '
    'Calculate the average offtake from the WFI Generation [m3/s]:
    sAverage = 0
    iCount = 1
    Do While iCount < 8640 + 1
        sAverage = sAverage + sWFIGenToStorage(iCount)
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    sAverage = sAverage / CSng(8640)
    Range("V4") = sAverage
    Range("Y66") = sAverage
    '
    'Calculate the average offtake from the WFI Distribution System [m3/s]:
    sAverage = 0
    iCount = 1
    Do While iCount < 8640 + 1
        sAverage = sAverage + sWFIDemand(iCount)
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    sAverage = sAverage / CSng(8640)
    Range("V4") = sAverage
    Range("Y65") = sAverage
    '
```

```
    Range("B5").Select
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    '
    Call Protect_Sheets_And_Workbook
    '
End Sub '
'
'
Sub Analysis_DI_Other_Days_and_Repeats()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine analyses all data of the DI data for the day and repeat as         @
    '@ selected by the user.                                                               @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIWaterFromGen = DI from Generation Plant [m3/s].
    ' sDICalcVolumeTank = Volume of DI storage tank [m3].
    ' sDIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sDIStartVolumeTank = Start volume in DI storage tank [m3].
    ' sDISimulatedDay = Number of simulated day [-].
    ' sDINumberOfRepeat = Number of repeat [-].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' sDIVolume() = Volume in DI storage tank [-].
    ' sDIFlow() = DI flow to storage tank [-].
    ' LUtilcount = Calculation of DI generation utilisation [-].
    ' sTemp = Temporary value [-].
    ' strRepeatSelected = String value of sDINumberOfRepeat [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sMin = Minimum value.
    ' sMedian = Median value.
    ' sMax = Maximum value.
    ' sAverage = Average value.
    ' sDISum = Overall DI consumption from loop excluding tank filling [m3].
    '
    Dim strRepeatSelected As String
    Dim sMin As Single, sValue As Single
    Dim sMax As Single
    Dim sAverage As Single, sTemp As Single
    Dim sDIVolume(8640) As Single, sDIGenToStorage(8640) As Single
    Dim sDIDemand(8640) As Single
    Dim sDISum As Single, sDIWaterFromGen As Single, sDIStartVolumeTank As Single
    Dim sDICalcVolumeTank As Single, sDIMinVolumeTank As Single
    Dim iRowNumber As Integer, iCount As Integer
    Dim iDaySelected As Integer, iTemp As Integer
    Dim sDISimulatedDay As Single, sDINumberOfRepeat As Single
    Dim LCount As Long, LRowNumber As Long, LLocationOfPercentile As Long, LUtilcount As Long
    '
    Application.ScreenUpdating = False
    '
    Call UnProtect_Sheets_and_Workbook_No_Message
    '
    'Read in Data:
    Sheets("Sheet 3 - DI Gen. & Tank Data").Select
    sDIWaterFromGen = Cells(4, 6)
    sDICalcVolumeTank = Cells(7, 6)
    sDIMinVolumeTank = Cells(11, 6)
    sDIStartVolumeTank = Cells(9, 6)
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    sDISimulatedDay = Cells(39, 25)
    sDINumberOfRepeat = Cells(41, 25)
```

```
'
Range("G4") = sDIWaterFromGen
Range("G5") = sDICalcVolumeTank
Range("G6") = sDIMinVolumeTank
Range("G7") = sDIStartVolumeTank
'
strRepeatSelected = CStr(sDINumberOfRepeat)
'
'Make header of sheet:
Range("A1:AD1").Select
ActiveCell.FormulaR1C1 = "Results of the DI Simulation Day: " & sDISimulatedDay & _
"; Repeat No.: " & strRepeatSelected
With ActiveCell.Characters(Start:=1, Length:=36).Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 20
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
'
Sheets("Sheet 10 - Results DI Repeat " & strRepeatSelected).Select
'Calculate the average and minimum Volume in the DI Storage Tank [m3]:
'Read in Data for Volume in DI Tank [m3]:
iCount = 1
Do While iCount < 8640 + 1
    sDIDemand(iCount) = Cells(iCount + 3, sDISimulatedDay)
    sDIGenToStorage(iCount) = Cells(iCount + 3, 33 + sDISimulatedDay)
    sDIVolume(iCount) = Cells(iCount + 3, 66 + sDISimulatedDay)
    iCount = iCount + 1
Loop
'End Read in Data.
'
'Write Data to sheet for graphical display:
Sheets("Sheet 7 - Report DI Any Day").Select
'
iCount = 1
Do While iCount < 8640 + 1
    Cells(iCount + 3, 53) = sDIDemand(iCount)
    Cells(iCount + 3, 57) = sDIGenToStorage(iCount)
    Cells(iCount + 3, 61) = sDIVolume(iCount)
    iCount = iCount + 1
Loop
'End write data for graphical display.
'
'Calculate the minimum and average Volume in the DI Tank [m3]:
sMin = sDIVolume(1)
sAverage = 0
iCount = 2
Do While iCount < 8640 + 1
    sTemp = sDIVolume(iCount)
    If sTemp < sMin Then
        sMin = sTemp
    End If
    sAverage = sAverage + sDIVolume(iCount)
    iCount = iCount + 1
Loop
'
Sheets("Sheet 7 - Report DI Any Day").Select
Range("O4") = sMin
sAverage = sAverage / CSng(8640)
Range("O6") = sAverage
'
```

```
'Find the maximum Volume in the DI Tank [m3]:
sMax = sDIVolume(1)
iCount = 2
Do While iCount < 8640 + 1
   sTemp = sDIVolume(iCount)
   If sTemp > sMax Then
      sMax = sTemp
   End If
   iCount = iCount + 1
Loop
'
Sheets("Sheet 7 - Report DI Any Day").Select
Range("O5") = sMax
'
'Calculate the 90% percentile DI Water Volume in Tank [m3]:
'Read in Data:
Sheets("Sheet 10 - Results DI Repeat " & strRepeatSelected).Select
iCount = 1
Do While iCount < 8640 + 1
   sDIVolume(iCount) = Cells(iCount + 3, 66 + sDISimulatedDay)
   iCount = iCount + 1
Loop
'End read in data.
'
'Sort algorithm for sDIVolume(iCount):
LCount = 2
Do While LCount < CLng(8640)
   sTemp = sDIVolume(LCount)
   For LRowNumber = LCount - 1 To 1 Step -1
      If (sDIVolume(LRowNumber) <= sTemp) Then
         GoTo 10
      End If
      sDIVolume(LRowNumber + 1) = sDIVolume(LRowNumber)
   Next LRowNumber
   LRowNumber = 0
10      sDIVolume(LRowNumber + 1) = sTemp
   LCount = LCount + 1
Loop
LLocationOfPercentile = Int(90 / 100 * CLng(8640))
LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
'
Sheets("Sheet 7 - Report DI Any Day").Select
Range("O8") = sDIVolume(LLocationOfPercentile)
'
'Calculate the 60% percentile DI Water Volume in Tank [m3]:
LLocationOfPercentile = Int(60 / 100 * CLng(8640))
LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
'
Sheets("Sheet 7 - Report DI Any Day").Select
Range("O7") = sDIVolume(LLocationOfPercentile)
'
'Calculate the DI Generation utilisation [-]:
'Read in data and calculate:
Sheets("Sheet 10 - Results DI Repeat " & strRepeatSelected).Select
LUtilcount = 0
iCount = 4
Do While iCount < 8640 + 1
   sValue = Cells(iCount, 33 + sDISimulatedDay)
   If sValue <> 0 Then
      LUtilcount = LUtilcount + 1
   End If
   iCount = iCount + 1
Loop
'
Sheets("Sheet 7 - Report DI Any Day").Select
Range("V8") = LUtilcount / (CLng(8640))
Range("A5").Select
```

```
    '
    'Calculate the average offtake from the DI Generation [m3/s]:
    sAverage = 0
    iCount = 1
    Do While iCount < 8640 + 1
        sAverage = sAverage + sDIGenToStorage(iCount)
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    sAverage = sAverage / CSng(8640)
    Range("V4") = sAverage
    '
    'Calculate the average offtake from the DI Distribution System [m3/s]:
    sAverage = 0
    iCount = 1
    Do While iCount < 8640 + 1
        sAverage = sAverage + sDIDemand(iCount)
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    sAverage = sAverage / CSng(8640)
    Range("V6") = sAverage
    '
    Range("B5").Select
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    '
    Call Protect_Sheets_And_Workbook
    '
End Sub
'
'
'
Sub Analysis_WFI_All_Days_and_Repeats()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine analyses all data of the WFI data for the day and repeat as      @
    '@ selected by the user.                                                            @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterFromGen = WFI from Generation Plant [m3/s].
    ' sWFICalcVolumeTank = Volume of WFI storage tank [m3].
    ' sWFIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sWFIStartVolumeTank = Start volume in WFI storage tank [m3].
    ' sWFISimulatedDay = Number of simulated day [-].
    ' sWFINumberOfRepeat = Number of repeat [-].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' sWFIVolume() = Volume in WFI storage tank [-].
    ' sWFIFlow() = WFI flow to storage tank [-].
    ' LUtilcount = Calculation of WFI generation utilisation [-].
    ' sTemp = Temporary value [-].
    ' strRepeatSelected = String value of sWFINumberOfRepeat [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sMin = Minimum value [various].
    ' sMedian = Median value [various].
    ' sMax = Maximum value [various].
    ' sAverage = Average value [various].
    ' sWFISum = Overall WFI consumption from loop excluding tank filling [m3].
    '
    Dim strRepeatSelected As String
```

```
Dim sMin As Single, sValue As Single
Dim sMax As Single
Dim sAverage As Single, sTemp As Single
Dim sWFIVolume(10800) As Single, sWFIGenToStorage(10800) As Single
Dim sWFIDemand(10800) As Single
Dim sWFISum As Single, sWFIWaterFromGen As Single, sWFIStartVolumeTank As Single
Dim sWFICalcVolumeTank As Single, sWFIMinVolumeTank As Single
Dim iRowNumber As Integer, iCount As Integer
Dim iDaySelected As Integer, iTemp As Integer
Dim sWFINumberOfRepeat As Single
Dim LCount As Long, LRowNumber As Long, LLocationOfPercentile As Long, LUtilcount As Long
'
Application.ScreenUpdating = False
'
Call UnProtect_Sheets_and_Workbook_No_Message
'
'Read in Data:
Sheets("Sheet 2 - WFI Gen. & Tank Data").Select
sWFIWaterFromGen = Cells(4, 6)
sWFICalcVolumeTank = Cells(7, 6)
sWFIMinVolumeTank = Cells(11, 6)
sWFIStartVolumeTank = Cells(9, 6)
'
Sheets("Sheet 6 - Report WFI All Days").Select
sWFINumberOfRepeat = Cells(41, 25)
'
Range("G4") = sWFIWaterFromGen
Range("G5") = sWFICalcVolumeTank
Range("G6") = sWFIMinVolumeTank
Range("G7") = sWFIStartVolumeTank
'
strRepeatSelected = CStr(sWFINumberOfRepeat)
'
'Make header of sheet:
Range("A1:AD1").Select
ActiveCell.FormulaR1C1 = "Results of the WFI Simulation All Simulated Days Repeat No.: " _
& strRepeatSelected
With ActiveCell.Characters(Start:=1, Length:=36).Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 20
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
'
Sheets("9 Results WFI - All - Repeat " & strRepeatSelected).Select
'Calculate the average and minimum Volume in the WFI Storage Tank [m3]:
'Read in Data for Volume in WFI Tank [m3]:
iCount = 1
Do While iCount < 10750 + 1
    sWFIDemand(iCount) = Cells(iCount + 3, 1)
    sWFIGenToStorage(iCount) = Cells(iCount + 3, 5)
    sWFIVolume(iCount) = Cells(iCount + 3, 9)
    iCount = iCount + 1
Loop
'End Read in Data.
'
'Write Data to sheet for graphical display:
Sheets("Sheet 6 - Report WFI All Days").Select
'
iCount = 1
Do While iCount < 10750 + 1
```

```
        Cells(iCount + 3, 53) = sWFIDemand(iCount)
        Cells(iCount + 3, 57) = sWFIGenToStorage(iCount)
        Cells(iCount + 3, 61) = sWFIVolume(iCount)
        iCount = iCount + 1
    Loop
    'End write data for graphical display.
    '
    'Calculate the minimum and average Volume in the WFI Tank [m3]:
    sMin = sWFIVolume(1)
    sAverage = 0
    iCount = 2
    Do While iCount < 10750 + 1
        sTemp = sWFIVolume(iCount)
        If sTemp < sMin Then
            sMin = sTemp
        End If
        sAverage = sAverage + sWFIVolume(iCount)
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    Range("O4") = sMin
    sAverage = sAverage / CSng(8640)
    Range("O6") = sAverage
    '
    'Find the maximum Volume in the WFI Tank [m3]:
    sMax = sWFIVolume(1)
    iCount = 2
    Do While iCount < 10750 + 1
        sTemp = sWFIVolume(iCount)
        If sTemp > sMax Then
            sMax = sTemp
        End If
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    Range("O5") = sMax
    '
    'Calculate the 90% percentile WFI Water Volume in Tank [m3]:
    'Read in Data:
    Sheets("9 Results WFI - All - Repeat " & strRepeatSelected).Select
    iCount = 1
    Do While iCount < 10750 + 1
        sWFIVolume(iCount) = Cells(iCount + 3, 66)
        iCount = iCount + 1
    Loop
    'End read in data.
    '
    'Sort algorithm for sWFIVolume(iCount):
    LCount = 2
    Do While LCount < CLng(10750)
        sTemp = sWFIVolume(LCount)
        For LRowNumber = LCount - 1 To 1 Step -1
            If (sWFIVolume(LRowNumber) <= sTemp) Then
                GoTo 10
            End If
            sWFIVolume(LRowNumber + 1) = sWFIVolume(LRowNumber)
        Next LRowNumber
        LRowNumber = 0
10      sWFIVolume(LRowNumber + 1) = sTemp
        LCount = LCount + 1
    Loop
    LLocationOfPercentile = Int(90 / 100 * CLng(10750))
    LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
```

```
    Range("O8") = sWFIVolume(LLocationOfPercentile)
    '
    'Calculate the 60% percentile WFI Water Volume in Tank [m3]:
    LLocationOfPercentile = Int(60 / 100 * CLng(10750))
    LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    Range("O7") = sWFIVolume(LLocationOfPercentile)
    '
    'Calculate the WFI Generation utilisation [-]:
    'Read in data and calculate:
    Sheets("9 Results WFI - All - Repeat " & strRepeatSelected).Select
    LUtilcount = 0
    iCount = 4
    Do While iCount < 10750 + 1
       sValue = Cells(iCount, 33)
       If sValue <> 0 Then
          LUtilcount = LUtilcount + 1
       End If
       iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    Range("V8") = LUtilcount / (CLng(10750))
    Range("A5").Select
    '
    'Calculate the average offtake from the WFI Generation [m3/s]:
    sAverage = 0
    iCount = 1
    Do While iCount < 10750 + 1
       sAverage = sAverage + sWFIGenToStorage(iCount)
       iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    sAverage = sAverage / CSng(10750)
    Range("V4") = sAverage
    '
    'Calculate the average offtake from the WFI Distribution System [m3/s]:
    sAverage = 0
    iCount = 1
    Do While iCount < 10750 + 1
       sAverage = sAverage + sWFIDemand(iCount)
       iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    sAverage = sAverage / CSng(10750)
    Range("V6") = sAverage
    '
    Range("B5").Select
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    '
    Call Protect_Sheets_And_Workbook
    '
End Sub
'
'
'
```

```
Sub Analysis_DI_All_Days_and_Repeats()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine analyses all data of the DI data for the day and repeat as          @
    '@ selected by the user.                                                               @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIWaterFromGen = DI from Generation Plant [m3/s].
    ' sDICalcVolumeTank = Volume of DI storage tank [m3].
    ' sDIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sDIStartVolumeTank = Start volume in DI storage tank [m3].
    ' sDISimulatedDay = Number of simulated day [-].
    ' sDINumberOfRepeat = Number of repeat [-].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' iNCount = Calculation of DI generation utilisation [-].
    ' sDIVolume() = Volume in DI storage tank [-].
    ' sDIFlow() = DI flow to storage tank [-].
    ' LUtilcount = Calculation of DI generation utilisation [-].
    ' sTemp = Temporary value [-].
    ' strRepeatSelected = String value of sDINumberOfRepeat [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sMin = Minimum value.
    ' sMedian = Median value.
    ' sMax = Maximum value.
    ' sAverage = Average value.
    ' sDISum = Overall DI consumption from loop excluding tank filling [m3].
    '
    Dim strRepeatSelected As String
    Dim sMin As Single, sValue As Single
    Dim sMax As Single
    Dim sAverage As Single, sTemp As Single
    Dim sDIVolume(10800) As Single, sDIGenToStorage(10800) As Single
    Dim sDIDemand(10800) As Single
    Dim sDISum As Single, sDIWaterFromGen As Single, sDIStartVolumeTank As Single
    Dim sDICalcVolumeTank As Single, sDIMinVolumeTank As Single
    Dim iRowNumber As Integer, iCount As Integer
    Dim iDaySelected As Integer, iTemp As Integer
    Dim sDINumberOfRepeat As Single
    Dim LCount As Long, LRowNumber As Long, LLocationOfPercentile As Long, LUtilcount As Long
    '
    Application.ScreenUpdating = False
    '
    Call UnProtect_Sheets_and_Workbook_No_Message
    '
    'Read in Data:
    Sheets("Sheet 3 - DI Gen. & Tank Data").Select
    sDIWaterFromGen = Cells(4, 6)
    sDICalcVolumeTank = Cells(7, 6)
    sDIMinVolumeTank = Cells(11, 6)
    sDIStartVolumeTank = Cells(9, 6)
    '
    Sheets("Sheet 7 - Report DI All Days").Select
    sDINumberOfRepeat = Cells(41, 25)
    '
    Range("G4") = sDIWaterFromGen
    Range("G5") = sDICalcVolumeTank
    Range("G6") = sDIMinVolumeTank
    Range("G7") = sDIStartVolumeTank
    '
    strRepeatSelected = CStr(sDINumberOfRepeat)
    '
```

```
'Make header of sheet:
Range("A1:AD1").Select
ActiveCell.FormulaR1C1 = "Results of the DI Simulation All Simulated Days Repeat No.: " _
& strRepeatSelected
With ActiveCell.Characters(Start:=1, Length:=36).Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 20
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
'
Sheets("11 Results DI - All - Repeat " & strRepeatSelected).Select
'Calculate the average and minimum Volume in the DI Storage Tank [m3]:
'Read in Data for Volume in DI Tank [m3]:
iCount = 1
Do While iCount < 10750 + 1
    sDIDemand(iCount) = Cells(iCount + 3, 1)
    sDIGenToStorage(iCount) = Cells(iCount + 3, 5)
    sDIVolume(iCount) = Cells(iCount + 3, 9)
    iCount = iCount + 1
Loop
'End Read in Data.
'
'Write Data to sheet for graphical display:
Sheets("Sheet 7 - Report DI All Days").Select
'
iCount = 1
Do While iCount < 10750 + 1
    Cells(iCount + 3, 53) = sDIDemand(iCount)
    Cells(iCount + 3, 57) = sDIGenToStorage(iCount)
    Cells(iCount + 3, 61) = sDIVolume(iCount)
    iCount = iCount + 1
Loop
'End write data for graphical display.
'
'Calculate the minimum and average Volume in the DI Tank [m3]:
sMin = sDIVolume(1)
sAverage = 0
iCount = 2
Do While iCount < 10750 + 1
    sTemp = sDIVolume(iCount)
    If sTemp < sMin Then
        sMin = sTemp
    End If
    sAverage = sAverage + sDIVolume(iCount)
    iCount = iCount + 1
Loop
'
Sheets("Sheet 7 - Report DI All Days").Select
Range("O4") = sMin
sAverage = sAverage / CSng(8640)
Range("O6") = sAverage
'
'Find the maximum Volume in the DI Tank [m3]:
sMax = sDIVolume(1)
iCount = 2
Do While iCount < 10750 + 1
    sTemp = sDIVolume(iCount)
    If sTemp > sMax Then
        sMax = sTemp
    End If
```

```
    iCount = iCount + 1
  Loop
  '
  Sheets("Sheet 7 - Report DI All Days").Select
  Range("O5") = sMax
  '
  'Calculate the 90% percentile DI Water Volume in Tank [m3]:
  'Read in Data:
  Sheets("11 Results DI - All - Repeat " & strRepeatSelected).Select
  iCount = 1
  Do While iCount < 10750 + 1
    sDIVolume(iCount) = Cells(iCount + 3, 66)
    iCount = iCount + 1
  Loop
  'End read in data.
  '
  'Sort algorithm for sDIVolume(iCount):
  LCount = 2
  Do While LCount < CLng(10750)
    sTemp = sDIVolume(LCount)
    For LRowNumber = LCount - 1 To 1 Step -1
      If (sDIVolume(LRowNumber) <= sTemp) Then
        GoTo 10
      End If
      sDIVolume(LRowNumber + 1) = sDIVolume(LRowNumber)
    Next LRowNumber
    LRowNumber = 0
10      sDIVolume(LRowNumber + 1) = sTemp
    LCount = LCount + 1
  Loop
  LLocationOfPercentile = Int(90 / 100 * CLng(10750))
  LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
  '
  Sheets("Sheet 7 - Report DI All Days").Select
  Range("O8") = sDIVolume(LLocationOfPercentile)
  '
  'Calculate the 60% percentile DI Water Volume in Tank [m3]:
  LLocationOfPercentile = Int(60 / 100 * CLng(10750))
  LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
  '
  Sheets("Sheet 7 - Report DI All Days").Select
  Range("O7") = sDIVolume(LLocationOfPercentile)
  '
  'Calculate the DI Generation utilisation [-]:
  'Read in data and calculate:
  Sheets("11 Results DI - All - Repeat " & strRepeatSelected).Select
  LUtilcount = 0
  iCount = 4
  Do While iCount < 10750 + 1
    sValue = Cells(iCount, 33)
    If sValue <> 0 Then
      LUtilcount = LUtilcount + 1
    End If
    iCount = iCount + 1
  Loop
  '
  Sheets("Sheet 7 - Report DI All Days").Select
  Range("V8") = LUtilcount / (CLng(10750))
  Range("A5").Select
  '
  'Calculate the average offtake from the DI Generation [m3/s]:
  sAverage = 0
  iCount = 1
  Do While iCount < 10750 + 1
    sAverage = sAverage + sDIGenToStorage(iCount)
    iCount = iCount + 1
  Loop
```

```
    '
    Sheets("Sheet 7 - Report DI All Days").Select
    sAverage = sAverage / CSng(10750)
    Range("V4") = sAverage
    '
    'Calculate the average offtake from the DI Distribution System [m3/s]:
    sAverage = 0
    iCount = 1
    Do While iCount < 10750 + 1
        sAverage = sAverage + sDIDemand(iCount)
        iCount = iCount + 1
    Loop
    '
    Sheets("Sheet 7 - Report DI All Days").Select
    sAverage = sAverage / CSng(10750)
    Range("V6") = sAverage
    '
    Range("B5").Select
    '
    Sheets("Sheet 7 - Report DI All Days").Select
    '
    Call Protect_Sheets_And_Workbook
    '
End Sub
    '
    '
    '
Sub Protect_Sheets_And_Workbook()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine protects all sheets in the workbook.      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim wSheet As Worksheet
    '
    Application.ScreenUpdating = False
    '
    ActiveWorkbook.Protect Structure:=True, Windows:=False, _
    Password:="Discrete Event Simulation 12345"
    For Each wSheet In Worksheets
        wSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True, _
        Password:="Discrete Event Simulation 12345"
    Next wSheet
    '
    Application.ScreenUpdating = True
    '
End Sub
'
'
'
Sub UnProtect_Sheets_and_Workbook_No_Message()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine unprotect all Sheets and the Workbook.   @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim wSheet As Worksheet
    '
    Application.ScreenUpdating = False
    For Each wSheet In Worksheets
        wSheet.Unprotect Password:="Discrete Event Simulation 12345"
    Next wSheet
    '
    ActiveWorkbook.Unprotect Password:="Discrete Event Simulation 12345"
    '
End Sub
'
```

```
'
'
Sub UnProtect_Sheets_and_Workbook()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine unprotect all Sheets and the Workbook.   @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim strMsg As String, strStyle As String, strTitle As String, strResponse As String
    Dim strMyString As String
    Dim wSheet As Worksheet
    '
    strMsg = "Deleting or adding rows, columns or cells may corrupt the spreadsheet. " + _
    "  It may no longer work.  Do you want to continue ?"
    strStyle = vbYesNo + vbCritical + vbDefaultButton2
    strTitle = "A Warning"
    strResponse = MsgBox(strMsg, strStyle, strTitle)
    If strResponse = vbYes Then
        strMyString = "Yes"
    Else
        End
    End If
    '
    Application.ScreenUpdating = False
    '
    For Each wSheet In Worksheets
        wSheet.Unprotect Password:="Discrete Event Simulation 12345"
    Next wSheet
    ActiveWorkbook.Unprotect Password:="Discrete Event Simulation 12345"
    '
    Application.ScreenUpdating = True
    '
End Sub
'
'
'
Sub Delete_All_Input_Data()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine deletes all data from the active Input Tables.  @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim strMsg As String, strStyle As String, strTitle As String, strResponse As String
    '
    strMsg = "This action will delete all data on this Event Table. Do you wish" + _
    " to continue? Press 'Yes' to continue or 'No' to cancel this action."
    strStyle = vbYesNo + vbDefaultButton2
    strTitle = "A critical Message"
    strResponse = MsgBox(strMsg, strStyle, strTitle)
    '
    If strResponse = vbNo Then
        End
    End If
    '
    ActiveSheet.Unprotect Password:="Discrete Event Simulation 12345"
    Range("G5:IK204").Select
    Selection.ClearContents
    Range("A2").Select
    '
End Sub
'
'
```

```
'
Sub Progress_Indicator(Progress)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine displays the process indicator.   @
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '
    SimulationProgress.RedProgressLabel.Width = Progress
    DoEvents
    '
End Sub
'
'
'
Sub Cell_Select_A4()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine moves the active cell to A1.          @
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim iCount As Integer
    '
    On Error Resume Next
    '
    iCount = 1
    Do While iCount < 50    'Increase if more than 50 repeats are needed.
        '
        Sheets("Sheet 8 - Results WFI Repeat " & iCount).Select
        Range("A1").Select
        '
        Sheets("9 Results WFI - All - Repeat " & iCount).Select
        Range("A1").Select
        '
        Sheets("Sheet 10 - Results DI Repeat " & iCount).Select
        Range("A1").Select
        '
        Sheets("11 Results DI - All - Repeat " & iCount).Select
        Range("A1").Select
        '
        iCount = iCount + 1
    Loop
    '
    On Error GoTo 0     'Disables error handling in the current procedure.
    '
    Sheets("Sheet 4 - Report WFI Day 1").Select
    Range("A1").Select
    '
    Sheets("Sheet 5 - Report DI Day 1").Select
    Range("A1").Select
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    Range("A1").Select
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    Range("A1").Select
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    Range("A1").Select
    '
    Sheets("Analysis WFI Exceeding Pump").Select
    Range("AG1").Select
    '
End Sub
'
'
'
Sub Move_Data()
```

```
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ This routine collates the data from the various simulated days into one  @
'@ overall graph.                                                           @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'INPUTS
' iNumberofRepeats = Global variable [-].
' iDayData = Global variable [-].
' sDIMinVolumeTank = Minimum allowable water Volume in DI storage tank [m3].
' sWFIMinVolumeTank = Minimum allowable water Volume in WFI storage tank [m3].
'
'INTERMEDIATE VARIABLES
' iTwo = Number of rows as per Input Table [-].
' LRow = Index for Do While Loop [-].
' LRepeatsCount = Index for Do While Loop [-].
' iColumn = Index for Do While Loop [-].
' sTempWaterDemand = Temporary variable, Water Demand.
' sTempWaterGentoStorage = Temporary variable, WFI flow to storage tank.
' sTempWaterVolumeInStorage = Temporary variable, Water in Tank [m3].
'
Dim iTwo As Integer, LRepeatsCount As Integer, iColumn As Integer
Dim LRow As Long
Dim sTempWaterDemand As Single, sTempWaterGentoStorage As Single
Dim sTempWaterVolumeInStorage As Single, sDIMinVolumeTank As Single
'
'Write data to sheets with lower resolution to display graphs over a couple of days:
LRepeats = 1
Do While LRepeats < iNumberofRepeats + 1  'Loop for the repeat.
   iColumn = 1
   LRow = 1
   iTwo = 2
   'Write data from one sheet to other:
   Do While iColumn < iDayData + 1
      Do While LRow < 8640
         '
         If iTwo = 2 Then        'Every 2nd datapoint only, as the Excel
                                 'graphs cannot display more than 32,000
                                 'datapoints.
            '
            Sheets("Sheet 8 - Results WFI Repeat " & LRepeats).Select
            sTempWaterDemand = Cells((LRow + 3), iColumn)
            sTempWaterGentoStorage = Cells((LRow + 3), iColumn + 33)
            sTempWaterVolumeInStorage = Cells((LRow + 3), iColumn + 66)
            '
            If LRow = 1 Then
               Sheets("9 Results WFI - All - Repeat " & LRepeats).Select
               '
               Range("A4").Select
               ActiveCell.Value = sTempWaterDemand
               '
               Range("E4").Select
               ActiveCell.Value = sTempWaterGentoStorage
               '
               Range("I4").Select
               ActiveCell.Value = sTempWaterVolumeInStorage
               '
            Else
               Sheets("9 Results WFI - All - Repeat " & LRepeats).Select
               '
               Range("A3").Select
               'Find last entry into column
               Selection.End(xlDown).Select
               ActiveCell.Offset(rowOffset:=1, columnOffset:=0).Activate
               ActiveCell.Value = sTempWaterDemand
               '
               Range("E3").Select
```

```
                    'Find last entry into column
                    Selection.End(xlDown).Select
                    ActiveCell.Offset(rowOffset:=1, columnOffset:=0).Activate
                    ActiveCell.Value = sTempWaterGentoStorage
                    '
                    Range("I3").Select
                    'Find last entry into column
                    Selection.End(xlDown).Select
                    ActiveCell.Offset(rowOffset:=1, columnOffset:=0).Activate
                    ActiveCell.Value = sTempWaterVolumeInStorage
                    '
                End If
                '
                iTwo = 0
            End If
            iTwo = iTwo + 1
            LRow = LRow + 1
        Loop
        iColumn = iColumn + 1
    Loop
    LRepeats = LRepeats + 1
Loop
'
'Move minimum volumes to sheets for display of graphs:
'Read in Data:
Sheets("Sheet 3 - DI Gen. & Tank Data").Select
sDIMinVolumeTank = Cells(11, 6)                   '[m3].
'End read in data.
'
End Sub
'
'
'
Sub Delete_Output_Data()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine deletes all the output data residing on the spreadsheets.    @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim iCount As Integer
    '
    Application.ScreenUpdating = False
    '
    Call UnProtect_Sheets_and_Workbook_No_Message
    '
    On Error Resume Next
    '
    iCount = 1
    Do While iCount < 50    'Increase if more than 50 repeats are needed.
        '
        Sheets("Sheet 8 - Results WFI Repeat " & iCount).Select
        Range("A4:IV65536").Select
        Selection.ClearContents
        Range("A4").Select
        '
        Sheets("9 Results WFI - All - Repeat " & iCount).Select
        Range("A4:IV65536").Select
        Selection.ClearContents
        Range("A4").Select
        '
        Sheets("Sheet 10 - Results DI Repeat " & iCount).Select
        Range("A4:IV65536").Select
        Selection.ClearContents
        Range("A4").Select
        '
        Sheets("11 Results DI - All - Repeat " & iCount).Select
        Range("A4:IV65536").Select
```

```
    Selection.ClearContents
    Range("A4").Select
    '
    iCount = iCount + 1
Loop
'
On Error GoTo 0     'Disables error handling in the current procedure.
'
Sheets("Sheet 4 - Report WFI Day 1").Select
Range("G4:G7").Select
Selection.ClearContents
Range("Q4:Q18").Select
Selection.ClearContents
Range("A4").Select
'
Sheets("Sheet 5 - Report DI Day 1").Select
Range("G4:G7").Select
Selection.ClearContents
Range("Q4:Q18").Select
Selection.ClearContents
Range("A4").Select
'
Sheets("Sheet 6 - Report WFI Any Day").Select
Range("G4:G7").Select
Selection.ClearContents
Range("O4:O8").Select
Selection.ClearContents
Range("V4:V8").Select
Selection.ClearContents
Range("Y39").Select
Selection.ClearContents
Range("Y41").Select
Selection.ClearContents
Range("Y48").Select
Selection.ClearContents
Range("Y50").Select
Selection.ClearContents
Range("A4").Select
Range("BA4:BM9000").Select
Selection.ClearContents
'
Sheets("Sheet 7 - Report DI Any Day").Select
Range("G4:G7").Select
Selection.ClearContents
Range("O4:O8").Select
Selection.ClearContents
Range("V4:V8").Select
Selection.ClearContents
Range("Y41").Select
Selection.ClearContents
Range("Y43").Select
Selection.ClearContents
Range("Y50").Select
Selection.ClearContents
Range("Y52").Select
Selection.ClearContents
Range("A4").Select
Range("BA4:BM9000").Select
Selection.ClearContents
'
Sheets("Frontpage - WFI Loop Input Data").Select
Range("B42").Select
Selection.ClearContents
Range("A4").Select
'
Sheets("Analysis WFI Exceeding Pump").Select
Range("A4:AD12000").Select
```

```
    Selection.ClearContents
    Range("AH5:BK7").Select
    Selection.ClearContents
    Range("AH9").Select
    Selection.ClearContents
    '
    Call Protect_Sheets_And_Workbook
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    Range("A5").Select
    '
    Application.ScreenUpdating = True
    '
End Sub
'
'
'
```

```
Static Function RandWH() As Double
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine is the Random Number Generator.  Code by A.J. Steel and R.J. Douglas  @
    '@ Institute for National Measurements Standards, Canada,                                @
    '@ http://inms-ienm.nrc-cnrc.gc.ca/qde/montecarlo/choosedownloads.html                   @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    ' User function to implement the Hill-Wichmann pseudorandom number generator.
    '
    ' Key statements from Fortran code in original article.
    ' Wichmann, B.A. and I.D. Hill, Algorithm AS 183: An Efficient and Portable
    ' Pseudo-Random Number Generator, Applied Statistics, 31, 188-190, 1982.
    ' C   IX, IY, IZ SHOULD BE SET TO INTEGER VALUES BETWEEN 1 AND 30000 BEFORE FIRST ENTRY.
    ' IX = MOD(171 * IX, 30269)
    ' IY = MOD(172 * IY, 30307)
    ' IZ = MOD(170 * IZ, 30323)
    ' RANDOM = AMOD(FLOAT(IX) / 30269# + FLOAT(IY) / 30307# + FLOAT(IZ) / 30323#, 1#)
    '
    ' Note: the FORTRAN function AMOD(A,P) = A - (INT(A / P) * P)
    ' VBA Mod function rounds non-integer values of A,P FIRST, so it won't work in the last
    ' line!
    '
    Dim ix As Long, iy As Long, iz As Long
    Dim sx As Single, sy As Single, sz As Single
    Dim sum As Single
    Dim seeded As Boolean ' initialized to 'False' during first call to function.
    '
    If seeded = False Then
        ix = 123      ' use a fixed initial seed.
        iy = 234
        iz = 345
        seeded = True ' static variable set true so we don't 're-seed' every time.
    End If
    '
    ' the guts of the linear congruential algorithm is only three lines long.
    ix = 171 * ix Mod 30269
    iy = 172 * iy Mod 30307
    iz = 170 * iz Mod 30323
    '
    ' convert the integer values to single precision reals for arithmetic.
    sx = CSng(ix) / 30269! ' use ! to ensure denominator is also single precision real.
    sy = CSng(iy) / 30307!
    sz = CSng(iz) / 30323!
    sum = sx + sy + sz
    '
    ' take the fractional part of the sum as the random number.
    RandWH = sum - Int(sum)
    '
    ' some cleaning here to be sure we stay 'inside the lines'.
    If RandWH <= 0# Then RandWH = 0.0000003
    If RandWH >= 1# Then RandWH = 1# - 0.0000003
    '
End Function
'
'
```

```
'
Sub Remove_Old_Data()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This routine removes all old data still residing on the spreadsheet   @
    '@ before starting a new calculation.                                    @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim iCount As Integer
    '
    On Error Resume Next
    '
    iCount = 1
    Do While iCount < 50    'Increase if more than 50 repeats are needed.
        '
        Sheets("Sheet 8 - Results WFI Repeat " & iCount).Select
        Range("A4:IV65536").Select
        Selection.ClearContents
        Range("A4").Select
        '
        Sheets("9 Results WFI - All - Repeat " & iCount).Select
        Range("A4:IV65536").Select
        Selection.ClearContents
        Range("A4").Select
        '
        Sheets("Sheet 10 - Results DI Repeat " & iCount).Select
        Range("A4:IV65536").Select
        Selection.ClearContents
        Range("A4").Select
        '
        Sheets("11 Results DI - All - Repeat " & iCount).Select
        Range("A4:IV65536").Select
        Selection.ClearContents
        Range("A4").Select
        '
        iCount = iCount + 1

    Loop
    '
    iCount = 2
    Do While iCount < 50
        '
        Sheets("Day " & iCount & " WFI Loop Data").Select
        Range("C50").Select
        Selection.ClearContents
        '
        iCount = iCount + 1
    Loop
    '
    iCount = 1
    Do While iCount < 50
        '
        Sheets("Day " & iCount & " DI Loop Data").Select
        Range("C50").Select
        Selection.ClearContents
        '
        iCount = iCount + 1
    Loop
    '
    On Error GoTo 0     'Disables error handling in the current procedure.
    '
    Sheets("Sheet 4 - Report WFI Day 1").Select
    Range("G4:G7").Select
    Selection.ClearContents
    Range("Q4:Q18").Select
    Selection.ClearContents
    Range("A2").Select
```

```
Selection.ClearContents
Range("A4").Select
'
Sheets("Sheet 5 - Report DI Day 1").Select
Range("G4:G7").Select
Selection.ClearContents
Range("Q4:Q18").Select
Selection.ClearContents
Range("A2").Select
Selection.ClearContents
Range("A4").Select
'
Sheets("Sheet 6 - Report WFI Any Day").Select
Range("G4:G7").Select
Selection.ClearContents
Range("O4:O8").Select
Selection.ClearContents
Range("V4:V8").Select
Selection.ClearContents
Range("Y39").Select
Selection.ClearContents
Range("Y41").Select
Selection.ClearContents
Range("Y48").Select
Selection.ClearContents
Range("Y50").Select
Selection.ClearContents
Range("A4").Select
Range("BA4:BQ9000").Select
Selection.ClearContents
Range("A2").Select
Selection.ClearContents
'
Sheets("Sheet 7 - Report DI Any Day").Select
Range("G4:G7").Select
Selection.ClearContents
Range("O4:O8").Select
Selection.ClearContents
Range("V4:V8").Select
Selection.ClearContents
Range("Y41").Select
Selection.ClearContents
Range("Y43").Select
Selection.ClearContents
Range("Y50").Select
Selection.ClearContents
Range("Y52").Select
Selection.ClearContents
Range("A4").Select
Range("BA4:BQ9000").Select
Selection.ClearContents
Range("A2").Select
Selection.ClearContents
'
Sheets("Sheet 6 - Report WFI All Days").Select
Range("G4:G7").Select
Selection.ClearContents
Range("O4:O8").Select
Selection.ClearContents
Range("V4:V8").Select
Selection.ClearContents
Range("Y41").Select
Selection.ClearContents
Range("A4").Select
Range("BA4:BQ12000").Select
Selection.ClearContents
Range("A2").Select
```

```
    Selection.ClearContents
    '
    Sheets("Sheet 7 - Report DI All Days").Select
    Range("G4:G7").Select
    Selection.ClearContents
    Range("O4:O8").Select
    Selection.ClearContents
    Range("V4:V8").Select
    Selection.ClearContents
    Range("Y41").Select
    Selection.ClearContents
    Range("A4").Select
    Range("BA4:BQ12000").Select
    Selection.ClearContents
    Range("A2").Select
    Selection.ClearContents
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    Range("B42").Select
    Selection.ClearContents
    Range("C50").Select
    Selection.ClearContents
    '
    Sheets("Analysis WFI Exceeding Pump").Select
    Range("AH5:BK7").Select
    Selection.ClearContents
    Range("AH9:AH11").Select
    Selection.ClearContents
    Range("AI27:AK56").Select
    Selection.ClearContents
    '
    iProgress = 2
    Call Progress_Indicator(iProgress)
    '
End Sub
'
'
'
Sub Min_Allowable_Volume_And_Pump_Flowrate()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This routine writes the appropriate "minimum allowable Tank Volume" values   @
    '@ and WFI,DI Pump Flowrates to all appropriate sheets.                          @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIMinVolumeTank = Global variable [-].
    ' sDIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sWFIPump = WFI pump flowrate [m3/h].
    ' sDIPump = DI pump flowrate [m3/h].
    ' iNumberofRepeats = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' LRepeats = Index for DoWhile Loop [-].
    '
    Dim LRepeats As Long
    Dim sDIMinVolumeTank As Single, sDIPump As Single, sWFIPump As Single
    '
    'Read data in:
    If bOnlyWFILoop = False Then
        Sheets("Sheet 3 - DI Gen. & Tank Data").Select
        sDIMinVolumeTank = Cells(11, 6)                 '[m3].
        sDIPump = Cells(19, 6)                          '[m3/h].
    End If
    '
    Sheets("Sheet 2 - WFI Gen. & Tank Data").Select
    sWFIPump = Cells(19, 6)                             '[m3/h].
```

```
'
'Write minimum volume and Pump Flowrate to the WFI & DI Repeat Sheets:
LRepeats = 1
Do While LRepeats < iNumberofRepeats + 1  'Loop for the repeat.
    '
    Sheets("9 Results WFI - All - Repeat " & LRepeats).Select
    '
    Cells(4, 13) = sWFIMinVolumeTank
    Cells(4, 14) = sWFIPump
    '
    Range("M4").Select
    Selection.Copy
    Range("M5:M8645").Select
    ActiveSheet.Paste
    '
    Range("N4").Select
    Selection.Copy
    Range("N5:N8645").Select
    ActiveSheet.Paste
    '
    Sheets("Sheet 8 - Results WFI Repeat " & LRepeats).Select
    '
    Cells(4, 100) = sWFIMinVolumeTank
    Cells(4, 101) = sWFIPump
    '
    Range("CV4").Select
    Selection.Copy
    Range("CV5:CV8645").Select
    ActiveSheet.Paste
    '
    Range("CW4").Select
    Selection.Copy
    Range("CW5:CW8645").Select
    ActiveSheet.Paste
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    '
    Cells(4, 69) = sWFIPump
    Cells(4, 65) = sWFIMinVolumeTank
    '
    Range("BM4").Select
    Selection.Copy
    Range("BM5:BM8645").Select
    ActiveSheet.Paste
    '
    Range("BQ4").Select
    Selection.Copy
    Range("BQ5:BQ8645").Select
    ActiveSheet.Paste
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    '
    Cells(4, 69) = sWFIPump
    Cells(4, 65) = sWFIMinVolumeTank
    '
    Range("BM4").Select
    Selection.Copy
    Range("BM5:BM8645").Select
    ActiveSheet.Paste
    '
    Range("BQ4").Select
    Selection.Copy
    Range("BQ5:BQ8645").Select
    ActiveSheet.Paste
    '
    If bOnlyWFILoop = False Then
        '
```

```
            Sheets("11 Results DI - All - Repeat " & LRepeats).Select
            '
            Cells(4, 13) = sDIMinVolumeTank
            Cells(4, 14) = sDIPump
            '
            Range("M4").Select
            Selection.Copy
            Range("M5:M8645").Select
            ActiveSheet.Paste
            '
            Range("N4").Select
            Selection.Copy
            Range("N5:N8645").Select
            ActiveSheet.Paste
            '
            Sheets("Sheet 7 - Report DI Any Day").Select
            '
            Cells(4, 69) = sWFIPump
            Cells(4, 65) = sWFIMinVolumeTank
            '
            Range("BM4").Select
            Selection.Copy
            Range("BM5:BM8645").Select
            ActiveSheet.Paste
            '
            Range("BQ4").Select
            Selection.Copy
            Range("BQ5:BQ8645").Select
            ActiveSheet.Paste
            '
            Sheets("Sheet 10 - Results DI Repeat " & LRepeats).Select
            '
            Cells(4, 100) = sDIMinVolumeTank
            Cells(4, 101) = sDIPump
            '
            Range("CV4").Select
            Selection.Copy
            Range("CV5:CV8645").Select
            ActiveSheet.Paste
            '
            Range("CW4").Select
            Selection.Copy
            Range("CW5:CW8645").Select
            ActiveSheet.Paste
            '
            Sheets("Sheet 7 - Report DI All Days").Select
            '
            Cells(4, 69) = sWFIPump
            Cells(4, 65) = sWFIMinVolumeTank
            '
            Range("BM4").Select
            Selection.Copy
            Range("BM5:BM8645").Select
            ActiveSheet.Paste
            '
            Range("BQ4").Select
            Selection.Copy
            Range("BQ5:BQ8645").Select
            ActiveSheet.Paste
            '
        End If
    '
    LRepeats = LRepeats + 1
  Loop
  '
End Sub
'
```

```
'
'
Sub Remove_Headers()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine removes the row and column headers.      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim iCount As Integer
    '
    Application.ScreenUpdating = False
    '
    Call UnProtect_Sheets_and_Workbook_No_Message
    '
    'On Error Resume Next
    '
    iCount = 1
    Do While iCount < 2    'Increase if more than 50 repeats are needed.
        '
        Sheets("Sheet 8 - Results WFI Repeat " & iCount).Select
        ActiveSheet.DisplayHeadings = False
        '
        Sheets("9 Results WFI - All - Repeat " & iCount).Select
        ActiveSheet.DisplayHeadings = False
        '
        Sheets("Sheet 10 - Results DI Repeat " & iCount).Select
        ActiveSheet.DisplayHeadings = False
        '
        Sheets("11 Results DI - All - Repeat " & iCount).Select
        ActiveSheet.DisplayHeadings = False
        '
        iCount = iCount + 1
    Loop
    '
    'On Error GoTo 0     'Disables error handling in the current procedure.
    '
    Sheets("Sheet 4 - Report WFI Day 1").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 5 - Report DI Day 1").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 6 - Report WFI Any Day").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 7 - Report DI Any Day").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 4 - DI Gen. & Tank Data").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 2 - WFI Gen. & Tank Data").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 12 - Normal Distribution").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 13 - Beta Distribution").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 6 - Report WFI All Days").Select
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Sheet 7 - Report DI All Days").Select
```

```
    ActiveSheet.DisplayHeadings = False
    '
    Sheets("Analysis WFI Exceeding Pump").Select
    ActiveSheet.DisplayHeadings = False
    '
    Call Protect_Sheets_And_Workbook
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    Range("C31").Select
    ActiveWindow.LargeScroll Down:=-1
    '
End Sub
'
'
'
Sub Volume_Calculation()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the water consumption over one day to    @
    '@ help sizing the WFI generation plant.  Uncertainty is ignored.      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' dTimeStart = Start time of tap opening [time].
    ' dTimeEnd = End time of tap opening [time].
    ' iColNumber = Column number of event table [-].
    ' iRowNumber = Row number of event table [-].
    ' sWaterDemandPerTap = Water demand of individual tap [-].
    '
    'INTERMEDIATE VARIABLES
    ' dTimeDiff = Time differential between opening and closing of valve [time].
    ' iCount = Index for DoWhile Loop [-].
    ' iCwIndex = Index for DoWhile Loop [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sOverallWaterVolume = One day water demand from loop excluding tank filling [m3].
    '
    Dim iCount As Integer, iCwIndex As Integer
    Dim iColNumber As Integer, iRowNumber As Integer
    Dim dTimeStart() As Date, dTimeEnd() As Date, dTimeDiff() As Date
    Dim sWaterDemandPerTap() As Single, sOverallWaterVolume As Single
    '
    ReDim dTimeStart(1000, 100), dTimeEnd(1000, 100), dTimeDiff(1000, 100)
    ReDim sWFIWaterDemandPerTap(1000)
    '
    'Start read in data:
    'Read in data from sheet calling the routine only:
    '
    iColNumber = Cells(15, 2)
    iRowNumber = Cells(16, 2)
    '
    iCwIndex = 0
    iCount = 5
    Do While iCwIndex < iColNumber
        Do While iCount < iRowNumber + 5
            dTimeStart(iCount - 4, iCwIndex + 1) = Cells(iCount, 12 + (4 * iCwIndex))
            dTimeEnd(iCount - 4, iCwIndex + 1) = Cells(iCount, 13 + (4 * iCwIndex))
            dTimeDiff(iCount - 4, iCwIndex + 1) = dTimeEnd(iCount - 4, iCwIndex + 1) _
            - dTimeStart(iCount - 4, iCwIndex + 1)
            iCount = iCount + 1
        Loop
        iCount = 5
        iCwIndex = iCwIndex + 1
    Loop
    '
    iCount = 5
    Do While iCount < iRowNumber + 5
```

```
        sWFIWaterDemandPerTap(iCount - 4) = Cells(iCount, 9)
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Calculation of overall volume:
    sOverallWaterVolume = 0
    iCwIndex = 1
    Do While iCwIndex < iColNumber + 1
        iCount = 1
        Do While iCount < iRowNumber + 1
            sOverallWaterVolume = sWFIWaterDemandPerTap(iCount) _
            * Hour(dTimeDiff(iCount, iCwIndex)) _
            + sWFIWaterDemandPerTap(iCount) / 60 * Minute(dTimeDiff(iCount, iCwIndex)) _
            + sWFIWaterDemandPerTap(iCount) / 3600 * Second(dTimeDiff(iCount, iCwIndex)) _
            + sOverallWaterVolume
            iCount = iCount + 1
        Loop
        iCwIndex = iCwIndex + 1
    Loop
    '
    Range("C50") = sOverallWaterVolume
    '
End Sub
'
'
'
Sub Pump_Analysis()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates how long the distribution pump flowrate   @
    '@ sWFIPump has been exceeded.                                          @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIPump = Distribution Pump Flowrate [m3/h].
    ' iNumberofRepeats = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' sWFIWaterDemand = Water Demand from Loop [m3/h].
    ' iCount = Index for DoWhile Loop [-].
    ' iCol = Index for DoWhile Loop [-].
    ' sExceedPump = Time flowrate of distribution pump has been exceeded [min].
    '
    'CALCULATED VALUES/OUTPUTS
    ' Pump_Analysis = Time flowrate of distribution pump has been exceeded [min].
    '
    Dim iCount As Integer, iCol As Integer
    Dim sWFIWaterDemand As Single, sWFIPump As Single, sMin As Single, sMax As Single
    Dim sExceedPump As Single, sAverage As Single, sTemp As Single
    '
    'Read in Data:
    Sheets("Frontpage - WFI Loop Input Data").Select
    iNumberofRepeats = Cells(13, 2)
    '
    Sheets("Sheet 2 - WFI Gen. & Tank Data").Select
    sWFIPump = Cells(19, 6)                    '[m3/h].
    '
    iCol = 1
    '
    Sheets("Analysis WFI Exceeding Pump").Select
    '
    'Calculate how often WFI demand exceeds WFI pump flowrate:
    Do While iCol < iNumberofRepeats + 1
        iCount = 1
        sExceedPump = 0
        Do While iCount < 8640 + 1
```

```
        '
        sWFIWaterDemand = Cells(iCount + 3, iCol)       'Row, Column.
        '
        If sWFIWaterDemand > sWFIPump Then
           sExceedPump = sExceedPump + 1
        End If
        iCount = iCount + 1
    Loop
    '
    Cells(5, 33 + iCol) = sWFIPump                      '[m3/h].
    sExceedPump = sExceedPump * 10 / 60                 '[min].
    Cells(7, 33 + iCol) = sExceedPump                   '[min].
    '
    iCol = iCol + 1
Loop
'
'Find the minimum value:
iCount = 1
sMin = Cells(7, 34)
'
Do While iCount < iNumberofRepeats
    sTemp = Cells(7, 34 + iCount)
    If sTemp < sMin Then
        sMin = sTemp
    End If
    iCount = iCount + 1
Loop
'
'Find the maximum value:
iCount = 1
sMax = Cells(7, 34)
'
Do While iCount < iNumberofRepeats
    sTemp = Cells(7, 34 + iCount)
    If sTemp > sMax Then
        sMax = sTemp
    End If
    iCount = iCount + 1
Loop
'
Cells(10, 34) = sMin
Cells(11, 34) = sMax
'
'Calculation of average:
iCount = 1
sAverage = 0
Do While iCount < iNumberofRepeats
    sAverage = sAverage + Cells(7, 33 + iCount)
    iCount = iCount + 1
Loop
'
sAverage = sAverage / iNumberofRepeats
Cells(9, 34) = sAverage
'
'Procedure Histogram:
'Bins:
Cells(27, 35) = sMin
'
iCount = 1
Do While iCount < iNumberofRepeats + 1
    Cells(27 + iCount, 35) = (sMax - sMin) / iNumberofRepeats _
    + Cells((27 + iCount - 1), 35)
    iCount = iCount + 1
Loop
'
'Count:
Range("AJ27:AJ56").Select
```

```
    Selection.FormulaArray = "=FREQUENCY(R7C34:R7C63,RC[-1]:R[29]C[-1])"
    '
    'SCALED:
    iCount = 0
    Do While iCount < iNumberofRepeats
        Cells(27 + iCount, 37) = (Cells(27 + iCount, 36) / iNumberofRepeats) * (sMax - sMin)
        iCount = iCount + 1
    Loop
    '
    'End Histogram.
    '
End Sub
'
'
'
Sub Statistical_Analysis()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine performs a statistical analysis on the results          @
    '@ finding the statistical point measures over all Monte Carlo runs.     @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIVolume = Water volume in tank [m3].
    ' sWFIGen = Water from generation plant to storage tank [m3/h].
    ' sWFIDemand = Water demand from loop [m3/h].
    ' LNumberofRepeat = Number of how often calculation is repeated (Monte Carlo) [-].
    ' LNumberofDays = Number of days avialable (7 = max) [-].
    '
    'INTERMEDIATE VARIABLES/CALCULATED VALUES/OUTPUTS
    ' sWFIWaterDemand = Water Demand from Loop [m3/h].
    ' iColCount = Index for DoWhile Loop [-].
    ' LCount = Index for DoWhile Loop [-].
    ' LCol = Index for DoWhile Loop [-].
    ' LRowNumber = Index for DoWhile Loop [-].
    ' sMin = Minimum value [-].
    ' sMax = Maximum value [-].
    ' sAverage = Average value [-].
    ' sTemp = Temporary value [-].
    ' strRepeatSelected = LNumberofRepeat [-].
    '
    Dim LCount As Long, LCol As Long, LRowNumber As Long, LNumberofDays As Long
    Dim LNumberofRepeat As Long
    Dim iColCount As Integer
    Dim sMin As Single, sMax As Single, sAverage As Single, sTemp As Single
    Dim strRepeatSelected As String
    '
    Const LDataPoint As Long = 1776810   '(8461 * 30 * 7)
    Dim sWFIVolume(LDataPoint) As Single, sWFIGen(LDataPoint) As Single
    Dim sWFIDemand(LDataPoint) As Single
    '
    'Read in Data:
    Sheets("Frontpage - WFI Loop Input Data").Select
    LNumberofRepeat = Cells(13, 2)
    LNumberofDays = Cells(14, 2)
    '
    'Read in Data for Volume in WFI Tank [m3]:
    LCol = 1
    Do While LCol < LNumberofRepeat + 1
        strRepeatSelected = CStr(LCol)
        Sheets("Sheet 8 - Results WFI Repeat " & LCol).Select
        '
        LCount = 1
        iColCount = 1
        Do While iColCount < LNumberofDays + 1
            Do While LCount < 8640 + 1
                sWFIVolume(1 + (LCol - 1) * LNumberofDays * 8640) _
```

```
            = Cells(LCount + 3, 67 + iColCount - 1)
            LCount = LCount + 1
        Loop
        iColCount = iColCount + 1
    Loop
    LCol = LCol + 1
Loop
'End Read in Data.
'
'Calculate the minimum Volume in WFI Tank [m3]:
LCount = 1
    Do While LCount < 8640 * LNumberofRepeat * LNumberofDays
        sTemp = sWFIVolume(LCount)
        If sTemp < sMin Then
            sMin = sTemp
        End If
        LCount = LCount + 1
    Loop
Sheets("Sheet 4 - Report WFI All Days").Select
Range("Q4") = sMin
'
'Find the maximum Volume in WFI Tank [m3].
'sMax = sWFIVolume(1)
LCount = 2
Do While LCount < 8640 * iNumberofRepeats * LNumberofDays
    sTemp = sWFIVolume(LCount)
    If sTemp > sMax Then
        sMax = sTemp
    End If
    LCount = LCount + 1
Loop
Sheets("Sheet 4 - Report WFI All Days").Select
Range("Q5") = sMax
'
'Calculate the range of the Volume in the storage tank [m3]:
Sheets("Sheet 4 - Report WFI All Days").Select
Range("Q6") = sMax - sMin
'
'Calculate the average Volume in the WFI Storage Tank [m3]:
sAverage = 0
LCount = 1
Do While LCount < 8640 * LNumberofRepeat * LNumberofDays
    sAverage = sAverage + sWFIVolume(LCount)
    LCount = LCount + 1
Loop
sAverage = sAverage / (8640 * LNumberofRepeat * LNumberofDays)
Sheets("Sheet 4 - Report WFI All Days").Select
Range("Q8") = sAverage
'
'Calculate the average water flow from the WFI generation plant [m3/h]:
'Read in Data water flow from the WFI generation plant [m3/h]:
LCol = 1
Do While LCol < LNumberofRepeat + 1
    strRepeatSelected = CStr(LCol)
    Sheets("Sheet 8 - Results WFI Repeat " & LCol).Select
    '
    LCount = 1
    iColCount = 1
    Do While iColCount < LNumberofDays + 1
        Do While LCount < 8640 + 1
            sWFIGen(1 + (LCol - 1) * LNumberofDays * 8640) _
            = Cells(LCount + 3, 34 + iColCount - 1)
            LCount = LCount + 1
        Loop
        iColCount = iColCount + 1
    Loop
    LCol = LCol + 1
```

```
    Loop
    'End Read in Data.
    '
    'Calculate the average water flow from the WFI generation plant [m3/h]:
    sAverage = 0
    LCount = 1
    Do While LCount < 8640 * LNumberofRepeat * LNumberofDays
        sAverage = sAverage + sWFIGen(LCount)
        LCount = LCount + 1
    Loop
    sAverage = sAverage / (8640 * LNumberofRepeat * LNumberofDays)
    Sheets("Sheet 4 - Report WFI All Days").Select
    Range("Q10") = sAverage
    '
    'Calculate the average offtake from the WFI loop [m3/h]:
    'Read in Data water flow from the WFI loop [m3/h]:
    LCol = 1
    Do While LCol < LNumberofRepeat + 1
        strRepeatSelected = CStr(LCol)
        Sheets("Sheet 8 - Results WFI Repeat " & LCol).Select
        '
        LCount = 1
        iColCount = 1
        Do While iColCount < LNumberofDays + 1
            Do While LCount < 8640 + 1
                sWFIDemand(1 + (LCol - 1) * LNumberofDays * 8640) _
                = Cells(LCount + 3, 1 + iColCount - 1)
                LCount = LCount + 1
            Loop
            iColCount = iColCount + 1
        Loop
        LCol = LCol + 1
    Loop
    'End Read in Data.
    '
    'Calculate the average water flow from the WFI loop [m3/h]:
    sAverage = 0
    LCount = 1
    Do While LCount < 8640 * LNumberofRepeat * LNumberofDays
        sAverage = sAverage + sWFIDemand(LCount)
        LCount = LCount + 1
    Loop
    sAverage = sAverage / (8640 * LNumberofRepeat * LNumberofDays)
    Sheets("Sheet 4 - Report WFI All Days").Select
    Range("Q9") = sAverage
    '
End Sub
'
'
'
'@@@@@@@@ END OF PROGRAM STOCHASTIC DISCRETE EVENT SIMULATION @@@@@@@@@
```

```
    '+++++++++++++++++++++++++++++++++++++++++++++++++++++
    '+ Modules for Sheet "Day 2 WFI Loop Data"              +
    '+++++++++++++++++++++++++++++++++++++++++++++++++++++
    '
    Option Explicit             ' Force explicit variable declaration.
    Option Base 1               ' Set default array subscripts to 1.
'
'
'
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This routines checks if the an incorrect description for a distribution has   @
    '@ been entered and highlights such errors to the user.                          @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim Msg, style, title, response, MyString
    Dim iCount As Integer, iCount2 As Integer
    '
    iCount = 0
    Do While iCount < 200
      iCount2 = 0
      Do While iCount2 < 59
        If Cells(iCount + 5, 11 + 4 * iCount2) = "b" _
        Or Cells(iCount + 5, 11 + 4 * iCount2) = "u" _
        Or Cells(iCount + 5, 11 + 4 * iCount2) = "n" _
        Or Cells(iCount + 5, 11 + 4 * iCount2) = Empty Then
        Else
          Msg = "Input must either be 'u' for Uniform, 'n' for Normal, " _
          + "or 'b' for Beta."
          style = vbOKOnly + vbCritical
          title = "Critical Message Resolution: Frontpage - DI Loop Input Data"
          response = MsgBox(Msg, style, title)
          End
        End If
        iCount2 = iCount2 + 1
      Loop
      iCount = iCount + 1
    Loop
End Sub
'
''
'                                       END OF PROGRAM
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

```
'++++++++++++++++++++++++++++++++++++++++++++++++
'+ Modules for "This Workbook"                    +
'++++++++++++++++++++++++++++++++++++++++++++++++
'
Option Explicit                 ' Force explicit variable declaration.
Option Base 1                   ' Set default array subscripts to 1.
'
'
Private Sub Workbook_Open()
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ Display the Intro Sheet after opening the workbook.          @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
Greetings.Show
'
Application.ScreenUpdating = False
'
Sheets("Frontpage - WFI Loop Input Data").Select
Range("A1").Select
'
End Sub
'
'                                                   END OF PROGRAM
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

Appendix 3 Program Listing: Fuzzy Discrete Event Simulation

```
'
'++++++++++++++++++++++++++++++++++++++++++++++++++++
'+ Module: Discrete_Simulation_Fuzzy                            +
'++++++++++++++++++++++++++++++++++++++++++++++++++++
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ DISCRETE EVENT SIMULATION of a Purified Water Distribution System comprised        @
'@ of a DI water distribution system feeding a WFI distribution system.               @
'@ Schedule Uncertainty is modelled via fuzzy logic.                                  @
'@                                                                                    @
'@ Copyright (C) <2010>  <Frank Riedewald>                                            @
'@                                                                                    @
'@ This program is free software: you can redistribute it and/or modify               @
'@ it under the terms of the GNU General Public License as published by               @
'@ the Free Software Foundation, either version 3 of the License, or                  @
'@ any later version.                                                                 @
'@                                                                                    @
'@ This program is distributed WITHOUT ANY WARRANTY; without even the implied         @
'@ warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the          @
'@ GNU General Public License for more details.                                       @
'@                                                                                    @
'@ For a copy of the GNU General Public License;                                      @
'@ please see <http://www.gnu.org/licenses/>.                                         @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
Option Explicit               'Forces explicit variable declaration.
Option Base 1                 'Set default array subscripts to 1.
Option Private Module        'Indicates that module is private.
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ Below are the Global Variables valid across all subroutines.        @
'@ Nomenclature for first letter of variables:                         @
'@ str = String          b = Boolean          L = Long                 @
'@ i = Integer           s = Single           v = Variable             @
'@ d = Date                                                            @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'GLOBAL INPUT VARIABLES
' iNumberOfOperators = Number of Operators available throughout the day [-].
' dWFITimeFuzzy1 = Fuzzy Trapezoidal number, operator setup time WFI [date].
' dWFITimeFuzzy2 = Fuzzy Trapezoidal number, operator setup time WFI [date].
' dWFITimeFuzzy3 = Fuzzy Trapezoidal number, operator setup time WFI [date].
' dWFITimeFuzzy4 = Fuzzy Trapezoidal number, operator setup time WFI [date].
' dWFITimeFuzzy1(Array) = Fuzzy trapezoid time parameter WFI [date].
' dWFITimeFuzzy2(Array) = Fuzzy trapezoid time parameter WFI [date].
' dWFITimeFuzzy3(Array) = Fuzzy trapezoid time parameter WFI [date].
' dWFITimeFuzzy4(Array) = Fuzzy trapezoid time parameter WFI [date].
' dWFIStart(Array) = Start of WFI valve opening time (crisp value) [date].
' dWFIEnd(Array) = End of WFI valve opening time (crisp value) [date].
' iWFIPriority(Array) = Priority rules for WFI [-].
' sFuzzy1Shift(Array) = Fuzzy time parameter for 1st shift pattern [-].
' sFuzzy2Shift(Array) = Fuzzy time parameter for 2nd shift pattern [-].
' sFuzzy3Shift(Array) = Fuzzy time parameter for 3rd shift pattern [-].
' sFuzzy1Output (Array) = Fuzzy values of output fuzzy set [-].
' sFuzzy2Output (Array) = Fuzzy values of output fuzzy set [-].
' sFuzzy3Output (Array) = Fuzzy values of output fuzzy set [-].
' sFuzzy4Output (Array) = Fuzzy values of output fuzzy set [-].
' dFuzzy1ShiftBreak(Array) = Fuzzy trapezoid time parameter for breaks [date].
' dFuzzy2ShiftBreak(Array) = Fuzzy trapezoid time parameter for breaks [date].
```

```
' dFuzzy3ShiftBreak(Array) = Fuzzy trapezoid time parameter for breaks [date].
' sFuzzy1ShiftBreak(Array) = Fuzzy trapezoid single parameter for breaks [-].
' sFuzzy2ShiftBreak(Array) = Fuzzy trapezoid single parameter for breaks [-].
' sFuzzy3ShiftBreak(Array) = Fuzzy trapezoid single parameter for breaks [-].
' dFuzzyTired(Array) = Fuzzy time parameter for tiredness of operator [date].
' sFuzzyTired(Array) = Fuzzy single parameter for tiredness of operator [-].
' dDITimeFuzzy1 = Fuzzy Trapezoidal time parameter, operator setup time DI [date].
' dDITimeFuzzy2 = Fuzzy Trapezoidal time parameter, operator setup time DI [date].
' dDITimeFuzzy3 = Fuzzy Trapezoidal time parameter, operator setup time DI [date].
' dDITimeFuzzy4 = Fuzzy Trapezoidal time parameter, operator setup time DI [date].
' dDITimeFuzzy1(Array) = Fuzzy trapezoid time parameter DI [date].
' dDITimeFuzzy2(Array) = Fuzzy trapezoid time parameter DI [date].
' dDITimeFuzzy3(Array) = Fuzzy trapezoid time parameter DI [date].
' dDITimeFuzzy4(Array) = Fuzzy trapezoid time parameter DI [date].
' dDIStart(Array) = Start of DI valve opening time (crisp value) [date].
' dDIEnd(Array) = End of DI valve opening time (crisp value) [date].
' iDIPriority(Array) = Priority rules for DI [-].
' sDIWaterDemandPerTap(Array) = Water Offtake WFI per tap (see Event Table) [m3/h].
' sWFIWaterDemandPerTap(Array) = Water Offtake DI per tap (see Event Table) [m3/h].
' sWFIWaterFromGen = WFI Water from WFI generation to storage tank [m3/h].
' sWFIStartVolumeTank = WFI storage tank [m3/h].
' sWFIMinVolumeTank = Minimum allowable water Volume in storage tank [m3].
' sWFIStartVolumeTank = Volume in storage tank at beginning of simulation [m3].
' sWFIGenBlowDown = WFI blowdown to drain [%].
' iColNumberDI = Number of Colums of DI Event Table [-].
' iRowNumberDI = Number Rows of DI Event Table [-].
' iColNumberWFI = Number of Colums of WFI Event Table [-].
' iRowNumberWFI = Number Rows of WFI Event Table [-].
' dTimeStart(Array) = Start Time tap opening as per Event Table WFI or DI [24 hours].
' dTimeEnd(Array) = End Time tap opening as per Event Table WFI or DI [24 hours].
' bOptionNoTimeUncertainty = Flag to in/exclude schedule uncertainty [-].
' iDayData = Value of how many days of data are available for the simulation [-].
' bOnlyOneLoop = Only the WFI loop will be simulated [-].
'
'GLOBAL INTERMEDIATE VARIABLES
' LDayDataCounter = Index for Do-Loop; current number of day being simulated [-].
' sProgress = Increases throughout calculation - for progress indicator [-].
' sWFIWaterFromDI(Array) = DI Water to WFI Generator Array (per day, per second) [m3/s].
' sIncrement = For progress indicator; indicator progresses per step [-].
' Wait() = Indicates if a task has to wait as an operator is not available [-].
' sDefuzzifier = Crisp value from defuzzifier [-].
' Rule1Activated = Has Rule 1 been activated? [-].
' Rule4Activated = Has Rule 4 been activated? [-].
' Rule5Activated = Has Rule 5 been activated? [-].
' Rule1DelayAfterBreak = Start of task shifts to time after break [date].
'
'GLOBAL CALCULATED VALUES/OUTPUTS
' sWFIWaterConsumptionFromLoop(Array) =(86401, iDayData) [m3/s] (1 day = 86,400 s).
' sDIWaterConsumptionFromLoop(Array) =(86401, iDayData) [m3/s].
' sMuiFuzzy1Shift = Fuzzy membership 1st shift time from Sheets("Fuzzy Shift") [date].
' sMuiFuzzy2Shift = Fuzzy membership 2nd shift time from Sheets("Fuzzy Shift") [date].
' sMuiFuzzy3Shift = Fuzzy membership 3rd shift time from Sheets("Fuzzy Shift") [date].
' sMuiFuzzyBreak = Fuzzy membership shift time from Sheets("Fuzzy Break") [date].
' sMuiFuzzyTired = Fuzzy membership tired time from Sheets("Fuzzy Tired") [date].
'
Dim sWFIWaterDemandPerTap() As Single, sDIWaterDemandPerTap() As Single
Dim sWFIWaterFromDI() As Single, sIncrement As Single, sWFIGenBlowDown As Single
Dim sWFIPumpFlowrate As Single
Dim sWFIWaterConsumptionFromLoop() As Single, sDIWaterConsumptionFromLoop() As Single
Dim iColNumberWFI As Integer, iRowNumberWFI As Integer
Dim iColNumberDI As Integer, iRowNumberDI As Integer
Dim iDayData As Integer
Dim sProgress As Single
Dim LDayDataCounter As Long
Dim iRowNumberDIMax As Integer, iColNumberDIMax As Integer
Dim iRowNumberWFIMax As Integer, iColNumberWFIMax As Integer
Dim bOptionNoTimeUncertainty As Boolean, bOnlyWFILoop As Boolean
Dim sWFIWaterFromGen As Single, sWFIVolumeTank As Single
Dim sWFIMinVolumeTank As Single, sWFIStartVolumeTank As Single
```

```
'
Dim dWFITimeFuzzy1() As Date, dWFITimeFuzzy2() As Date
Dim dWFITimeFuzzy3() As Date, dWFITimeFuzzy4() As Date
Dim dWFIStart() As Date, dWFIEnd() As Date
Dim dDITimeFuzzy1() As Date, dDITimeFuzzy2() As Date
Dim dDITimeFuzzy3() As Date, dDITimeFuzzy4() As Date
Dim dDIStart() As Date, dDIEnd() As Date
Dim iWFIPriority() As Integer, iDIPriority() As Integer, iNumberOfOperators As Integer
Dim dFuzzy1Shift() As Date, dFuzzy2Shift() As Date, dFuzzy3Shift() As Date
Dim sFuzzy1Shift() As Single, sFuzzy2Shift() As Single, sFuzzy3Shift() As Single
Dim dFuzzy1ShiftBreak() As Date, dFuzzy2ShiftBreak() As Date, dFuzzy3ShiftBreak() As Date
Dim Rule1DelayAfterBreak As Date
'
Dim sFuzzy1ShiftBreak() As Single, sFuzzy2ShiftBreak() As Single
Dim sFuzzy3ShiftBreak() As Single
Dim sFuzzy0Output() As Single, sFuzzy1Output() As Single, sFuzzy2Output() As Single
Dim sFuzzy3Output() As Single, sFuzzy4Output() As Single, sFuzzy5Output() As Single
Dim sFuzzy6Output() As Single, sFuzzy7Output() As Single, sFuzzy8Output() As Single
Dim sFuzzy9Output() As Single, sFuzzy10Output() As Single
Dim dFuzzyTired() As Date
Dim sMuiFuzzy1Shift As Single, sMuiFuzzy2Shift As Single, sMuiFuzzy3Shift As Single
Dim sMuiFuzzyBreak As Single, sMuiFuzzyTired As Single, sFuzzyTired() As Single
Dim sDefuzzifier As Single
'
Dim Rule1Activated As String, Rule4Activated As String, Rule5Activated As String
Dim Wait() As String
'
Const LOneDayInSeconds As Long = 86400 'Number of seconds of one day.
'
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ End Definition of Global Variables.                    @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'
'
Sub Fuzzy_Discrete_Event_Simulation()
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ This is the Master Subroutine controlling the flow of the simulation.     @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'INPUTS
' iRowNumberDI = Global variable [-].
' iColNumberDI = Global variable [-].
' iRowNumberWFI = Global variable [-].
' iColNumberWFI = Global variable [-].
' iDayData = Global variable [-].
' bOnlyOneLoop = Global variable [-].
'
'INTERMEDIATE VARIABLES
' dTStart = Real time start of calculation [date].
' dTEnd = Real time end of calculation [date].
'
'CALCULATED VALUES/OUTPUTS
' Real time the calculation/simulation took [min].
' Message "Calculation finished".
'
Dim dTStart As Date, dTEnd As Date
Dim Msg As String, Style As String, Title As String, Response As String
Dim NumberOfTasksWaiting As Integer, iCWIndex As Integer, iCount As Integer
'
ActiveWorkbook.Date1904 = False
Application.ScreenUpdating = False  'Turn screen updating off to speed up macro code.
'
dTStart = Now()
'
'Display Progress Bar.
SimulationProgress.RedProgressLabel.Width = 0
```

```
SimulationProgress.Show (0)
'
Call Remove_Old_Data
'
'Read in Data:
Sheets("Frontpage - WFI Loop Input Data").Select
iNumberOfOperators = Cells(17, 2)
iDayData = Cells(14, 2)         'No. of Days available Event Table.
bOnlyWFILoop = Cells(19, 2)
'
iColNumberDIMax = 100
iColNumberWFIMax = 100
iRowNumberDIMax = 1000
iRowNumberWFIMax = 1000
'
'Progress indicator:
sProgress = 0
If bOnlyWFILoop = False Then
   sIncrement = 663 / (1000 * iDayData)
Else
   sIncrement = 663 / (1000 / 2 * iDayData)
End If
'
ReDim dTimeStartWFI(iRowNumberWFIMax, iColNumberWFIMax)
ReDim dTimeEndWFI(iRowNumberWFIMax, iColNumberWFIMax)
ReDim sWFIWaterDemandPerTap(iRowNumberWFIMax)
ReDim sWFIWaterConsumptionFromLoop(86401, iDayData)
ReDim sWFIWaterFromDI(86401, iDayData)
ReDim sDIWaterDemandPerTap(iRowNumberDIMax)
ReDim sDIWaterConsumptionFromLoop(86401, iDayData)
ReDim dWFITimeFuzzy1(iRowNumberWFIMax, iColNumberWFIMax)
ReDim dWFITimeFuzzy2(iRowNumberWFIMax, iColNumberWFIMax)
ReDim dWFITimeFuzzy3(iRowNumberWFIMax, iColNumberWFIMax)
ReDim dDITimeFuzzy1(iRowNumberDIMax, iColNumberDIMax)
ReDim dDITimeFuzzy2(iRowNumberDIMax, iColNumberDIMax)
ReDim dDITimeFuzzy3(iRowNumberDIMax, iColNumberDIMax)
ReDim dWFITimeFuzzy4(iRowNumberWFIMax, iColNumberWFIMax)
ReDim dDITimeFuzzy4(iRowNumberDIMax, iColNumberDIMax)
ReDim dWFIStart(iRowNumberWFIMax, iColNumberWFIMax)
ReDim dWFIEnd(iRowNumberWFIMax, iColNumberWFIMax)
ReDim dDIStart(iRowNumberDIMax, iColNumberDIMax)
ReDim dDIEnd(iRowNumberDIMax, iColNumberDIMax)
ReDim iWFIPriority(iRowNumberWFIMax, iColNumberWFIMax)
ReDim iDIPriority(iRowNumberDIMax, iColNumberDIMax)
ReDim dFuzzy1Shift(10, 10), dFuzzy2Shift(10, 10), dFuzzy3Shift(10, 10)
ReDim sFuzzy1Shift(10, 10), sFuzzy2Shift(10, 10), sFuzzy3Shift(10, 10)
ReDim dFuzzy1ShiftBreak(10, 10), dFuzzy2ShiftBreak(10, 10), dFuzzy3ShiftBreak(10, 10)
ReDim sFuzzy1ShiftBreak(10, 10), sFuzzy2ShiftBreak(10, 10), sFuzzy3ShiftBreak(10, 10)
ReDim Wait(iRowNumberWFIMax, iRowNumberWFIMax)
ReDim dFuzzyTired(25, 25), sFuzzyTired(25, 25)
ReDim sFuzzy0Output(3, 3), sFuzzy1Output(3, 3), sFuzzy2Output(3, 3), sFuzzy3Output(3, 3)
ReDim sFuzzy4Output(3, 3), sFuzzy5Output(3, 3), sFuzzy6Output(3, 3), sFuzzy7Output(3, 3)
ReDim sFuzzy8Output(3, 3), sFuzzy9Output(3, 3), sFuzzy10Output(3, 3)
'
Call Fuzzy_Data_Input
'
'Loop for WFI Simulation:
LDayDataCounter = 1
Do While LDayDataCounter < iDayData + 1
   Call WFI_WorkSheet_Data_Input
   Call Discrete_Time_Events_Fuzzy_WFI
   Call WFI_Water_Demand_From_Loop
   Call WFI_Water_Demand_From_Generation_And_Tank_Volume
   LDayDataCounter = LDayDataCounter + 1
Loop
Call Analysis_of_Data_WFI_Day_1
'End Simulation WFI Distribution.
'
If bOnlyWFILoop = False Then
```

```
        'Loop for DI Loop Simulation:
        LDayDataCounter = 1
        Do While LDayDataCounter < iDayData + 1
            Call DI_WorkSheet_Data_Input
            Call Discrete_Time_Events_Fuzzy_DI
            Call DI_Water_Demand_From_Loop
            LDayDataCounter = LDayDataCounter + 1
        Loop
        'End Simulation DI Distribution.
        Call DI_Water_Demand_From_Generation_And_Tank_Volume
    End If
    '
    Call Analysis_of_Data_DI_Day_1
    Call Analysis_of_Data_WFI_and_DI_all_Days
    Call Cell_Select_A4
    '
    NumberOfTasksWaiting = 0
    iCWIndex = 1
    'Calculate number of waiting operators:
    Do While iCWIndex < iColNumberWFI + 1
        iCount = 1
        Do While iCount < iRowNumberWFI + 1
            If Wait(iCount, iCWIndex) = "Waiting" Then
                NumberOfTasksWaiting = NumberOfTasksWaiting + 1
            End If
            iCount = iCount + 1
        Loop
        iCWIndex = iCWIndex + 1
    Loop
    '
    Call Min_Allowable_Volume_And_Pump_Flowrate
    '
    Sheets("Frontpage - WFI Loop Input Data").Select
    '
    Cells(7, 2) = NumberOfTasksWaiting
    '
    dTEnd = Now()
    '
    'How long did the calculation take:
    Cells(42, 2) = dTEnd - dTStart
    Range("K1").Select
    '
    Call Progress_Indicator(615)
    Unload SimulationProgress
    '
    'Notify user that the calculation is finished:
    Msg = "Simulation Water Demand finished. See Analysis Sheets for results of simulation."
    Style = vbOKOnly + vbInformation
    Title = "WFI & DI Simulation"
    Response = MsgBox(Msg, Style, Title)
    If Response = 1 Then
        End
    End If
    '
End Sub
'
'
'
Sub WFI_WorkSheet_Data_Input()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine reads in Start and End Time of Tap opening; Fuzzy Trapezoidal    @
    '@ numbers, operator setup time and Priority.                                       @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS OF SUBROUTINE
    ' iColNumberWFI = Global variable [-].
    ' iRowNumberWFI = Global variable [-].
    ' dWFITimeFuzzy1(Array) = Global variable [date].
```

```
' dWFITimeFuzzy2(Array) = Global variable [date].
' dWFITimeFuzzy3(Array) = Global variable [date].
' dWFITimeFuzzy4(Array) = Global variable [date].
' dWFIStart(Array)= Global variable [date].
' dWFIEnd(Array) = Global variable [date].
' iWFIPriority(Array) = Global variable [-].
' sIncrement = Global variable [-].
'
'INTERMEDIATE VARIABLES
' iCount = Index for Do While-Loop [-].
' iCwIndex = Index for Do While-Loop [-].
'
'CALCULATED VALUES/OUTPUTS
' sProgress = Global variable [-].
'
Dim iCount As Integer, iCWIndex As Integer
'
'Start read in data:
If LDayDataCounter = 1 Then
   Sheets("Frontpage - WFI Loop Input Data").Select
   iColNumberWFI = Cells(16, 2)
   iRowNumberWFI = Cells(15, 2)
Else
   Sheets("Day " & LDayDataCounter & " WFI Loop Data").Select
   iRowNumberWFI = Cells(15, 2)
   iColNumberWFI = Cells(16, 2)
End If
'
'Read in Start and End Time of Tap opening:
iCWIndex = 1
iCount = 5
Do While iCWIndex < iColNumberWFI + 1
   Do While iCount < iRowNumberWFI + 5
      dWFIStart(iCount - 4, iCWIndex) = Cells(iCount, 10 + (iCWIndex - 1) * 7)
      dWFIEnd(iCount - 4, iCWIndex) = Cells(iCount, 11 + (iCWIndex - 1) * 7)
      iCount = iCount + 1
   Loop
   iCount = 5
   iCWIndex = iCWIndex + 1
Loop
'
'Read in Fuzzy Trapezoidal number, operator setup time:
iCWIndex = 1
iCount = 5
Do While iCWIndex < iColNumberWFI + 1
   Do While iCount < iRowNumberWFI + 5
      dWFITimeFuzzy1(iCount - 4, iCWIndex) = Cells(iCount, 12 + (iCWIndex - 1) * 7)
      dWFITimeFuzzy2(iCount - 4, iCWIndex) = Cells(iCount, 13 + (iCWIndex - 1) * 7)
      dWFITimeFuzzy3(iCount - 4, iCWIndex) = Cells(iCount, 14 + (iCWIndex - 1) * 7)
      dWFITimeFuzzy4(iCount - 4, iCWIndex) = Cells(iCount, 15 + (iCWIndex - 1) * 7)
      iCount = iCount + 1
   Loop
   iCount = 5
   iCWIndex = iCWIndex + 1
Loop
'
'Read in Priority:
iCWIndex = 1
iCount = 5
Do While iCWIndex < iColNumberWFI + 1
   Do While iCount < iRowNumberWFI + 5
      iWFIPriority(iCount - 4, iCWIndex) = Cells(iCount, 16 + (iCWIndex - 1) * 7)
      iCount = iCount + 1
   Loop
   iCount = 5
   iCWIndex = iCWIndex + 1
Loop
'
'End reading in Data.
```

```
    sProgress = sProgress + 1 * sIncrement
    Call Progress_Indicator(sProgress)
    '
End Sub
'
'
'
Sub DI_WorkSheet_Data_Input()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine reads in Start and End Time of Tap opening; Fuzzy Trapezoidal    @
    '@ numbers, operator setup time and Priority.                                        @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS OF SUBROUTINE
    ' iColNumberDI = Global variable [-].
    ' iRowNumberDI = Global variable [-].
    ' dDITimeFuzzy1(Array) = Global variable [date].
    ' dDITimeFuzzy2(Array) = Global variable [date].
    ' dDITimeFuzzy3(Array) = Global variable [date].
    ' dDITimeFuzzy4(Array) = Global variable [date].
    ' dDIStart(Array) = Global variable [date].
    ' dDIEnd(Array) = Global variable [date].
    ' iDIPriority(Array) = Global variable [-].
    ' sIncrement = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' iCount = Index for Do While-Loop [-].
    ' iCwIndex = Index for Do While-Loop [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sProgress = Global variable [-].
    '
    Dim iCount As Integer, iCWIndex As Integer
    '
    'Start read in data:
    Sheets("Day " & LDayDataCounter & " DI Loop Data").Select
    iRowNumberDI = Cells(13, 2)
    iColNumberDI = Cells(14, 2)
    '
    'Read in Start and End Time of Tap opening:
    iCWIndex = 1
    iCount = 5
    Do While iCWIndex < iColNumberDI + 1
        Do While iCount < iRowNumberDI + 5
            dDIStart(iCount - 4, iCWIndex) = Cells(iCount, 10 + (iCWIndex - 1) * 7)
            dDIEnd(iCount - 4, iCWIndex) = Cells(iCount, 11 + (iCWIndex - 1) * 7)
            iCount = iCount + 1
        Loop
        iCount = 5
        iCWIndex = iCWIndex + 1
    Loop
    '
    'Read in Fuzzy Trapezoidal number, operator setup time:
    iCWIndex = 1
    iCount = 5
    Do While iCWIndex < iColNumberDI + 1
        Do While iCount < iRowNumberDI + 5
            dDITimeFuzzy1(iCount - 4, iCWIndex) = Cells(iCount, 12 + (iCWIndex - 1) * 7)
            dDITimeFuzzy2(iCount - 4, iCWIndex) = Cells(iCount, 13 + (iCWIndex - 1) * 7)
            dDITimeFuzzy3(iCount - 4, iCWIndex) = Cells(iCount, 14 + (iCWIndex - 1) * 7)
            dDITimeFuzzy4(iCount - 4, iCWIndex) = Cells(iCount, 15 + (iCWIndex - 1) * 7)
            iCount = iCount + 1
        Loop
        iCount = 5
        iCWIndex = iCWIndex + 1
    Loop
    '
    'Read in Priority:
```

```
    iCWIndex = 1
    iCount = 5
    Do While iCWIndex < iColNumberDI + 1
        Do While iCount < iRowNumberDI + 5
            iDIPriority(iCount - 4, iCWIndex) = Cells(iCount, 16 + (iCWIndex - 1) * 7)
            iCount = iCount + 1
        Loop
        iCount = 5
        iCWIndex = iCWIndex + 1
    Loop
    'End reading in Data.
    '
    sProgress = sProgress + 1 * sIncrement
    Call Progress_Indicator(sProgress)
    '
End Sub
'
'
'
Sub Fuzzy_Data_Input()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine reads in the Fuzzy Shift, Fuzzy Break and Fuzzy Tired data.      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS OF SUBROUTINE
    ' dFuzzy1Shift (Array) = Global variable [date].
    ' dFuzzy2Shift (Array) = Global variable [date].
    ' dFuzzy3Shift (Array) = Global variable [time-].
    ' sFuzzy1Shift (Array) = Global variable [-].
    ' sFuzzy2Shift (Array) = Global variable [-].
    ' sFuzzy3Shift (Array) = Global variable [-].
    ' sFuzzy1Output (Array) = Global variable [-].
    ' sFuzzy2Output (Array) = Global variable [-].
    ' sFuzzy3Output (Array) = Global variable [-].
    ' sFuzzy4Output (Array) = Global variable [-].
    ' dFuzzy1ShiftBreak (Array) = Global variable [time-].
    ' dFuzzy2ShiftBreak (Array) = Global variable [time-].
    ' dFuzzy3ShiftBreak (Array) = Global variable [time-].
    ' sFuzzy1ShiftBreak (Array) = Global variable [-].
    ' sFuzzy2ShiftBreak (Array) = Global variable [-].
    ' sFuzzy3ShiftBreak (Array) = Global variable [-].
    ' dFuzzyTired(Array) = Global variable [time-].
    ' sFuzzyTired(Array) = Global variable [-].
    ' iColNumberWFI = Global variable [-].
    ' iRowNumberWFI = Global variable [-].
    ' dWFITimeFuzzy1(Array) = Global variable [time-].
    ' dWFITimeFuzzy2(Array) = Global variable [time-].
    ' dWFITimeFuzzy3(Array) = Global variable [time-].
    ' dWFITimeFuzzy4(Array) = Global variable [time-].
    ' dWFIStart(Array) = Global variable [date].
    ' dWFIEnd(Array) = Global variable [date].
    ' iWFIPriority(Array) = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' iCount = Index for Do While-Loop [-].
    ' iCwIndex = Index for Do While-Loop [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sProgress = Global variable [-].
    '
    Dim iCount As Integer, iCWIndex As Integer
    '
    'Start Read in Data:
    Sheets("Fuzzy Shift").Select
    iCount = 1
    Do While iCount < 4 + 1
        dFuzzy1Shift(1, iCount) = Cells(6 + iCount - 1, 2)
        sFuzzy1Shift(2, iCount) = Cells(6 + iCount - 1, 3)
```

```
    '
    dFuzzy2Shift(1, iCount) = Cells(13 + iCount - 1, 2)
    sFuzzy2Shift(2, iCount) = Cells(13 + iCount - 1, 3)
    '
    dFuzzy3Shift(1, iCount) = Cells(20 + iCount - 1, 2)
    sFuzzy3Shift(2, iCount) = Cells(20 + iCount - 1, 3)
    iCount = iCount + 1
Loop
'
'Read in Fuzzy Break Data:
Sheets("Fuzzy Breaks").Select
iCount = 1
Do While iCount < 8 + 1
    If iCount < 5 Then
        dFuzzy1ShiftBreak(1, iCount) = Cells(5 + iCount - 1, 2)
        sFuzzy1ShiftBreak(2, iCount) = Cells(5 + iCount - 1, 3)
    Else
        dFuzzy1ShiftBreak(1, iCount) = Cells(12 + iCount - 5, 2)
        sFuzzy1ShiftBreak(2, iCount) = Cells(12 + iCount - 5, 3)
    End If
    iCount = iCount + 1
Loop
'
iCount = 1
Do While iCount < 8 + 1
    If iCount < 5 Then
        dFuzzy2ShiftBreak(1, iCount) = Cells(19 + iCount - 1, 2)
        sFuzzy2ShiftBreak(2, iCount) = Cells(19 + iCount - 1, 3)
    Else
        dFuzzy2ShiftBreak(1, iCount) = Cells(26 + iCount - 5, 2)
        sFuzzy2ShiftBreak(2, iCount) = Cells(26 + iCount - 5, 3)
    End If
    iCount = iCount + 1
Loop
'
iCount = 1
Do While iCount < 8 + 1
    If iCount < 5 Then
        dFuzzy3ShiftBreak(1, iCount) = Cells(33 + iCount - 1, 2)
        sFuzzy3ShiftBreak(2, iCount) = Cells(33 + iCount - 1, 3)
    Else
        dFuzzy3ShiftBreak(1, iCount) = Cells(40 + iCount - 5, 2)
        sFuzzy3ShiftBreak(2, iCount) = Cells(40 + iCount - 5, 3)
    End If
    iCount = iCount + 1
Loop
'
'Read in Fuzzy Tired data:
Sheets("Fuzzy Tired").Select
iCount = 1
Do While iCount < 25 + 1
    dFuzzyTired(1, iCount) = Cells(5 + iCount - 1, 2)
    sFuzzyTired(2, iCount) = Cells(5 + iCount - 1, 3)
    iCount = iCount + 1
Loop
'
Sheets("Fuzzy Output").Select
iCount = 1
Do While iCount < 3 + 1
    sFuzzy0Output(1, iCount) = Cells(6 + iCount - 1, 2)
    sFuzzy0Output(2, iCount) = Cells(6 + iCount - 1, 3)
    '
    sFuzzy1Output(1, iCount) = Cells(12 + iCount - 1, 2)
    sFuzzy1Output(2, iCount) = Cells(12 + iCount - 1, 3)
    '
    sFuzzy2Output(1, iCount) = Cells(18 + iCount - 1, 2)
    sFuzzy2Output(2, iCount) = Cells(18 + iCount - 1, 3)
    '
    sFuzzy3Output(1, iCount) = Cells(24 + iCount - 1, 2)
```

```
        sFuzzy3Output(2, iCount) = Cells(24 + iCount - 1, 3)
        '
        sFuzzy4Output(1, iCount) = Cells(30 + iCount - 1, 2)
        sFuzzy4Output(2, iCount) = Cells(30 + iCount - 1, 3)
        '
        sFuzzy5Output(1, iCount) = Cells(36 + iCount - 1, 2)
        sFuzzy5Output(2, iCount) = Cells(36 + iCount - 1, 3)
        '
        sFuzzy6Output(1, iCount) = Cells(42 + iCount - 1, 2)
        sFuzzy6Output(2, iCount) = Cells(42 + iCount - 1, 3)
        '
        sFuzzy7Output(1, iCount) = Cells(48 + iCount - 1, 2)
        sFuzzy7Output(2, iCount) = Cells(48 + iCount - 1, 3)
        '
        sFuzzy8Output(1, iCount) = Cells(54 + iCount - 1, 2)
        sFuzzy8Output(2, iCount) = Cells(54 + iCount - 1, 3)
        '
        sFuzzy9Output(1, iCount) = Cells(60 + iCount - 1, 2)
        sFuzzy9Output(2, iCount) = Cells(60 + iCount - 1, 3)
        '
        sFuzzy10Output(1, iCount) = Cells(66 + iCount - 1, 2)
        sFuzzy10Output(2, iCount) = Cells(66 + iCount - 1, 3)
        '
        iCount = iCount + 1
    Loop
    '
    sProgress = sProgress + 1 * sIncrement
    Call Progress_Indicator(sProgress)
    '
End Sub
'
'
'
Sub Discrete_Time_Events_Fuzzy_WFI()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine modifies the WFIStart and WFIEnd times by the fuzzy times      @
    '@ as are calculated here.  This routine uses the priority rules and calls       @
    '@ on the rule base if required.                                                 @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' dTimeStart(Array) = Global variable [date].
    ' dTimeEnd(Array) = Global variable [date].
    ' iColNumberWFI = Global variable [-].
    ' iRowNumberWFI = Global variable [-].
    ' bOptionNoTimeUncertainty = Global variable [-].
    ' sIncrement = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' LSecondsOfDaysCount = Index for Do While-Loop (seconds per day) [-].
    ' iCount = Index for Do While-Loop [-].
    ' iCwIndex = Index for Do While-Loop [-].
    ' iCwIndex2 = Index for Do While-Loop [-].
    ' iCount1 = Index for Do While-Loop [-].
    ' iCount2 = Index for Do While-Loop [-].
    ' LSecondsOfDaysCount = Index for Do While-Loop [-].
    ' LCount = Index for Do While-Loop [-].
    ' dSimTime = Current simulated time [date].
    ' dTime = Current simulated time [date].
    ' dTimeDiff = Time differential between two time frames [date].
    ' iCounterEarliestTask1 = To remember the earliest task found for priority 1 [-].
    ' iCounterEarliestTask2 = To remember the earliest task found for priority 2 [-].
    ' iCounterEarliestTask3 = To remember the earliest task found for priority 3 [-].
    ' dEarliestTask1 = Time index of earliest task 1 found [date].
    ' dEarliestTask2 = Time index of earliest task 2 found [date].
    ' dEarliestTask3 = Time index of earliest task 3 found [date].
    ' iNumOperatorsOccupied = Number of operators occupied with tasks [-].
    ' iIntermediate = Variable for evaluation of simulated time [-].
```

```
' dMinDurationOfTask = Minimum time operator is occupied with a task [date].
' dDelayTask = Calculated delay for operator for a given task [date].
' Wait = Global variable [-].
' Rule1DelayAfterBreak = Global variable [-].
'
'CALCULATED VALUES/OUTPUTS
' dWFIStart(Array) = Global variable [date].
' dWFIEnd(Array) = Global variable [date].
'
Dim iCount As Integer, iCWIndex As Integer, iNumOperatorsOccupied As Integer
Dim iCount1 As Integer, iCount2 As Integer, iCwIndex2 As Integer, iIntermediate As Integer
Dim iCounterEarliestTask1(2) As Integer, iCounterEarliestTask2(2) As Integer
Dim iCounterEarliestTask3(2) As Integer
Dim LSecondsOfDaysCount As Long, LCount As Long
Dim dSimTime As Date, dDelayTask As Date, dTimeDiff As Date, dMinDurationOfTask As Date
Dim dEarliestTask1 As Date, dEarliestTask2 As Date, dEarliestTask3 As Date
'
Const dZeroTime As Date = 0 'Time frame 00:00:00 [date].
'
'Start values:
LSecondsOfDaysCount = 1
'
Do While LSecondsOfDaysCount < LOneDayInSeconds + 1
   'Evaluate which simulated time (dSimTime) it is:
   Select Case LSecondsOfDaysCount
   Case 1 To 32400    'Number between 1 and 32400, inclusive.
      dSimTime = TimeSerial(0, 0, LSecondsOfDaysCount)
   Case 32401 To 64800
      iIntermediate = LSecondsOfDaysCount - 32400
      dSimTime = TimeSerial(9, 0, iIntermediate)
   Case 64801 To 86400
      iIntermediate = LSecondsOfDaysCount - 64800
      dSimTime = TimeSerial(18, 0, iIntermediate)
   Case Else          'Other values.
      Stop            'Error trap.
                 'LSecondsOfDaysCount cannot be lower than zero or higher than
                 '86400.  Check program.
      End
   End Select
   '
   iCWIndex = 1
   Do While iCWIndex < iColNumberWFI + 1
      iCount = 1
      Do While iCount < iRowNumberWFI + 1
         '
         '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         'Condition: An operator is no longer occupied with a task.  The operator is
         'available to start another task which is "waiting" (see code below).
         '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         '
         If (dSimTime = dWFIEnd(iCount, iCWIndex) _
         And (Wait(iCount, iCWIndex) = "Operating") _
         And iWFIPriority(iCount, iCWIndex) > 0) _
         Then
            '
            iNumOperatorsOccupied = iNumOperatorsOccupied - 1
            '
            iCounterEarliestTask1(1) = 0
            iCounterEarliestTask1(2) = 0
            iCounterEarliestTask2(1) = 0
            iCounterEarliestTask2(2) = 0
            iCounterEarliestTask3(1) = 0
            iCounterEarliestTask3(2) = 0
            dEarliestTask3 = CSng(86400)
            iCwIndex2 = 1
            '
            'Search only for tasks which are "Waiting" and have Priority = 3:
            Do While iCwIndex2 < iColNumberWFI + 1
               iCount2 = 1
```

```
    Do While iCount2 < iRowNumberWFI + 1
      If Wait(iCount2, iCwIndex2) = "Waiting" _
      And iWFIPriority(iCount2, iCwIndex2) = 3 Then
        If (CSng(dWFIStart(iCount2, iCwIndex2)) _
        < (CSng(dEarliestTask3))) Then
          dEarliestTask3 = dWFIStart(iCount2, iCwIndex2)
          iCounterEarliestTask3(1) = iCount2
          iCounterEarliestTask3(2) = iCwIndex2
        End If
      End If
      iCount2 = iCount2 + 1
    Loop
    iCwIndex2 = iCwIndex2 + 1
Loop
'
'Search only for tasks which are "Waiting" and have Priority = 2:
dEarliestTask2 = CSng(86400)
iCwIndex2 = 1
Do While iCwIndex2 < iColNumberWFI + 1
    iCount2 = 1
    Do While iCount2 < iRowNumberWFI + 1
      If Wait(iCount2, iCwIndex2) = "Waiting" _
      And iWFIPriority(iCount2, iCwIndex2) = 2 Then
        If (CSng(dWFIStart(iCount2, iCwIndex2)) _
        < (CSng(dEarliestTask2))) Then
          dEarliestTask2 = dWFIStart(iCount2, iCwIndex2)
          iCounterEarliestTask2(1) = iCount2
          iCounterEarliestTask2(2) = iCwIndex2
        End If
      End If
      iCount2 = iCount2 + 1
    Loop
    iCwIndex2 = iCwIndex2 + 1
Loop
'
'Search only for tasks which are "Waiting" and have Priority = 1:
dEarliestTask1 = CSng(86400)
iCwIndex2 = 1
Do While iCwIndex2 < iColNumberWFI + 1
    iCount2 = 1
    Do While iCount2 < iRowNumberWFI + 1
      If Wait(iCount2, iCwIndex2) = "Waiting" _
      And iWFIPriority(iCount2, iCwIndex2) = 1 Then
        If (CSng(dWFIStart(iCount2, iCwIndex2)) _
        < (CSng(dEarliestTask1))) Then
          dEarliestTask1 = dWFIStart(iCount2, iCwIndex2)
          iCounterEarliestTask1(1) = iCount2
          iCounterEarliestTask1(2) = iCwIndex2
        End If
      End If
      iCount2 = iCount2 + 1
    Loop
    iCwIndex2 = iCwIndex2 + 1
Loop
'
'First: Priority = 3 (highest priority).
'If no Priority 3 tasks jump to Priority 2:
If iCounterEarliestTask3(1) = 0 Or iCounterEarliestTask3(2) = 0 Then
    GoTo NoPriority3
End If
'
If Wait(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) = "Waiting" _
And iWFIPriority(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
= 3 Then
    Call Fuzzification(LSecondsOfDaysCount)
    'Assign new dWFIStart and dWFIEnd time for this process.
    Call Rule_Base_WFI(iCounterEarliestTask3(1), _
    iCounterEarliestTask3(2), dSimTime)
    '
```

```
                dTimeDiff = dWFIEnd(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2)) - dWFIStart(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2))
                '
                'Calculate the "Minimum Duration of Task":
                dMinDurationOfTask = dWFITimeFuzzy3(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2)) - dWFITimeFuzzy2(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2))
                '
                'Calculate the delay of the task:
                dDelayTask = sDefuzzifier / 100 _
                * (dWFITimeFuzzy2(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2)) - dWFITimeFuzzy1(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2)) + dWFITimeFuzzy4(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2)) - dWFITimeFuzzy3(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2)))
                '
                If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
                Or Rule5Activated = "Yes" Then
                   '
                   If Rule1Activated = "Yes" Then
                      'Below allocate new start and end times to tasks:
                      dWFIStart(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                      = Rule1DelayAfterBreak + dMinDurationOfTask + dDelayTask
                      dWFIEnd(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                      = dWFIStart(iCounterEarliestTask3(1), _
                      iCounterEarliestTask3(2)) + dTimeDiff
                      Wait(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                      = "Operating"
                   End If
                   '
                   If Rule4Activated = "Yes" Then
                      'Operator on break while process becomes available (Rule 4):
                      Call Operator_Is_On_Break_WFI(iCounterEarliestTask3(1), _
                      iCounterEarliestTask3(2), dSimTime, _
                      dMinDurationOfTask, dDelayTask, dTimeDiff)
                   End If
                   '
                   If Rule5Activated = "Yes" Then
                      'Operator delays batch deliberately (Rule 5):
                      Call Operator_Delays_Batch_WFI(iCounterEarliestTask3(1), _
                      iCounterEarliestTask3(2), dSimTime, dMinDurationOfTask, _
                      dDelayTask, dTimeDiff)
                   End If
                   '
                Else
                   'Case: Operator was not on a break when task started.
                   'Below allocate new start and end times to tasks:
                   dWFIStart(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                   = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
                   + dMinDurationOfTask + dDelayTask
                   dWFIEnd(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                   = dWFIStart(iCounterEarliestTask3(1), _
                   iCounterEarliestTask3(2)) + dTimeDiff
                   Wait(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                   = "Operating"
                End If
                '
                Call WFI_Time_Check(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2), LSecondsOfDaysCount)
                '
                iNumOperatorsOccupied = iNumOperatorsOccupied + 1
                '
                GoTo FoundTask  'Exit the loop, found the most urgent task.
             End If
             '
             'Second: Priority = 2 (median priority).
NoPriority3:
             If iCounterEarliestTask2(1) = 0 Or iCounterEarliestTask2(2) = 0 Then
```

```
    GoTo NoPriority2
End If
'
If Wait(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
= "Waiting" And iWFIPriority(iCounterEarliestTask2(1), _
iCounterEarliestTask2(2)) = 2 Then
    '
    Call Fuzzification(LSecondsOfDaysCount)
    '
    Call Rule_Base_WFI(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2), dSimTime)
    '
    dTimeDiff = dWFIEnd(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2)) - dWFIStart(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2))
    '
    'Calculate the "Minimum Duration of Task":
    dMinDurationOfTask = dWFITimeFuzzy3(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2)) - dWFITimeFuzzy2(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2))
    '
    'Calculate the delay of the task:
    dDelayTask = sDefuzzifier / 100 _
    * (dWFITimeFuzzy2(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2)) - dWFITimeFuzzy1(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2)) + dWFITimeFuzzy4(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2)) - dWFITimeFuzzy3(iCounterEarliestTask2(1), _
    iCounterEarliestTask2(2)))
    '
    If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
    Or Rule5Activated = "Yes" Then
        '
        If Rule1Activated = "Yes" Then
            'Below allocate new start and end times to tasks:
            dWFIStart(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
            = Rule1DelayAfterBreak + dMinDurationOfTask + dDelayTask
            dWFIEnd(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
            = dWFIStart(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2)) + dTimeDiff
            Wait(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) = _
            "Operating"
        End If
        '
        If Rule4Activated = "Yes" Then
            'Operator on break while process becomes available (Rule 4):
            Call Operator_Is_On_Break_WFI(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2), dSimTime, dMinDurationOfTask, _
            dDelayTask, dTimeDiff)
        End If
        '
        If Rule5Activated = "Yes" Then
            'Operator delays batch deliberately (Rule 5):
            Call Operator_Delays_Batch_WFI(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2), dSimTime, dMinDurationOfTask, _
            dDelayTask, dTimeDiff)
        End If
        '
    Else
        'Case: Operator was not on a break when task started.
        'Below allocate new start and end times to tasks:
        dWFIStart(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
        = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
        + dMinDurationOfTask + dDelayTask
        dWFIEnd(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
        = dWFIStart(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
        + dTimeDiff
        Wait(iCounterEarliestTask2(1), iCWIndex) = "Operating"
    End If
    '
```

```
                Call WFI_Time_Check(iCounterEarliestTask2(1), _
                iCounterEarliestTask2(2), LSecondsOfDaysCount)
                '
                iNumOperatorsOccupied = iNumOperatorsOccupied + 1
                '
                GoTo FoundTask
            End If
            '
            'Third: Priority = 1 (lowest priority).  Condition: no more tasks to do.
NoPriority2:
            If iCounterEarliestTask1(1) = 0 Or iCounterEarliestTask1(2) = 0 Then
                GoTo NoMoreTasks    'Exit the loop, found the most urgent task.
            End If
            '
            If Wait(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) = "Waiting" _
            And iWFIPriority(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
            = 1 Then
                '
                Call Fuzzification(LSecondsOfDaysCount)
                '
                Call Rule_Base_WFI(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2), dSimTime)
                '
                dTimeDiff = dWFIEnd(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2)) - dWFIStart(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2))
                '
                'Calculate the "Minimum Duration of Task":
                dMinDurationOfTask = dWFITimeFuzzy3(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2)) - dWFITimeFuzzy2(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2))
                '
                'Calculate the delay of the task:
                dDelayTask = sDefuzzifier / 100 _
                * (dWFITimeFuzzy2(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2)) - dWFITimeFuzzy1(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2)) + dWFITimeFuzzy4(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2)) - dWFITimeFuzzy3(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2)))
                '
                If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
                Or Rule5Activated = "Yes" Then
                    '
                    If Rule1Activated = "Yes" Then
                        'Below allocate new start and end times to tasks:
                        dWFIStart(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                        = Rule1DelayAfterBreak + dMinDurationOfTask + dDelayTask
                        dWFIEnd(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                        = dWFIStart(iCounterEarliestTask1(1), _
                        iCounterEarliestTask1(2)) + dTimeDiff
                        Wait(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) = _
                        "Operating"
                    End If
                    '
                    If Rule4Activated = "Yes" Then
                        'Operator on break while process becomes available (Rule 4):
                        Call Operator_Is_On_Break_WFI(iCounterEarliestTask1(1), _
                        iCounterEarliestTask1(2), dSimTime, dMinDurationOfTask, _
                        dDelayTask, dTimeDiff)
                    End If
                    '
                    If Rule5Activated = "Yes" Then
                        'Operator delays batch deliberately (Rule 5):
                        Call Operator_Delays_Batch_WFI(iCounterEarliestTask1(1), _
                        iCounterEarliestTask1(2), dSimTime, dMinDurationOfTask, _
                        dDelayTask, dTimeDiff)
                    End If
                    '
                Else
```

```
                    'Case: Operator was not on a break when task started.
                    'Below allocate new start and end times to tasks:
                    dWFIStart(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                    = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
                    + dMinDurationOfTask + dDelayTask
                    dWFIEnd(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                    = dWFIStart(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                    + dTimeDiff
                    Wait(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) = _
                    "Operating"
                End If
                '
                Call WFI_Time_Check(iCounterEarliestTask1(1), _
                iCounterEarliestTask1(2), LSecondsOfDaysCount)
                '
                iNumOperatorsOccupied = iNumOperatorsOccupied + 1
                '
                GoTo FoundTask  'Exit the loop, found the most urgent task.
              End If
            End If
            '
NoMoreTasks:
            iCount = iCount + 1
          Loop
          iCWIndex = iCWIndex + 1
        Loop
        '
FoundTask:
        iCWIndex = 1
        '
        '#########################################################################################
        'The code below is activated if a task starts, which has not been previously delayed
        '(see code above). In addition this code assigns "waiting" should no operator
        ' be available to start a task.
        '#########################################################################################
        '
        Do While iCWIndex < iColNumberWFI + 1
          iCount = 1
          Do While iCount < iRowNumberWFI + 1
            '
            'If Block to check how many operators are working at any one time.
            If (CSng(dSimTime) = CSng(dWFIStart(iCount, iCWIndex)) _
            And ((Wait(iCount, iCWIndex) = "Waiting") _
            Or (Wait(iCount, iCWIndex) = "") _
            And iWFIPriority(iCount, iCWIndex) > 0)) Then
              '
              iNumOperatorsOccupied = iNumOperatorsOccupied + 1
              '
              If iNumOperatorsOccupied > iNumberOfOperators Then
                '
                Wait(iCount, iCWIndex) = "Waiting"
                '
                iNumOperatorsOccupied = iNumOperatorsOccupied - 1
              Else
                Wait(iCount, iCWIndex) = "Operating"
                '
                Call Fuzzification(LSecondsOfDaysCount)
                '
                Call Rule_Base_WFI(iCount, iCWIndex, dSimTime)
                '
                dTimeDiff = dWFIEnd(iCount, iCWIndex) - dWFIStart(iCount, iCWIndex)
                '
                'Calculate the "Minimum Duration of Task":
                dMinDurationOfTask = dWFITimeFuzzy3(iCount, iCWIndex) _
                - dWFITimeFuzzy2(iCount, iCWIndex)
                '
                'Calculate the delay of the task:
                dDelayTask = sDefuzzifier / 100 * (dWFITimeFuzzy2(iCount, iCWIndex) _
                - dWFITimeFuzzy1(iCount, iCWIndex) + dWFITimeFuzzy4(iCount, iCWIndex) _
```

```
                    - dWFITimeFuzzy3(iCount, iCWIndex))
                    '
                    If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
                    Or Rule5Activated = "Yes" Then
                       '
                       If Rule1Activated = "Yes" Then
                          'Below allocates new start and end times to tasks:
                          dWFIStart(iCount, iCWIndex) = Rule1DelayAfterBreak _
                          + dMinDurationOfTask + dDelayTask
                          dWFIEnd(iCount, iCWIndex) _
                          = dWFIStart(iCount, iCWIndex) + dTimeDiff
                          Wait(iCount, iCWIndex) = "Operating"
                       End If
                       '
                       If Rule4Activated = "Yes" Then
                          'Operator on break while process becomes available (Rule 4):
                          Call Operator_Is_On_Break_WFI(iCount, iCWIndex, dSimTime, _
                          dMinDurationOfTask, dDelayTask, dTimeDiff)
                       End If
                       '
                       If Rule5Activated = "Yes" Then
                          'Operator delays batch deliberately (Rule 5):
                          Call Operator_Delays_Batch_WFI(iCount, _
                          iCWIndex, dSimTime, dMinDurationOfTask, dDelayTask, dTimeDiff)
                       End If
                       '
                    Else
                       'Case: Operator was not on a break when task started.
                       'Below allocate new start and end times to tasks:
                       dWFIStart(iCount, iCWIndex) _
                       = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
                       + dMinDurationOfTask + dDelayTask
                       dWFIEnd(iCount, iCWIndex) _
                       = dWFIStart(iCount, iCWIndex) + dTimeDiff
                       Wait(iCount, iCWIndex) = "Operating"
                    End If
                    '
                    Call WFI_Time_Check(iCount, iCWIndex, LSecondsOfDaysCount)
                    '
                 End If
              End If
              iCount = iCount + 1
           Loop
           iCWIndex = iCWIndex + 1
        Loop
        LSecondsOfDaysCount = LSecondsOfDaysCount + 1
     Loop
     '
     '
     'The Do-Loop below sets the tasks the operators were not able to work on to empty.
     'This ensures these tasks will not be included in the calculation of the demand
     'from the WFI loop.
     '
     iCWIndex = 1
     Do While iCWIndex < iColNumberWFI + 1
        iCount = 1
        Do While iCount < iRowNumberWFI + 1
           If Wait(iCount, iCWIndex) = "Waiting" Then
              dWFIStart(iCount, iCWIndex) = Empty
              dWFIEnd(iCount, iCWIndex) = Empty
           End If
           iCount = iCount + 1
        Loop
        iCWIndex = iCWIndex + 1
     Loop
     '
     End Sub'
'
'
```

```
'
Sub Operator_Delays_Batch_WFI(iCount, iCWIndex, dSimTime, dMinTimeTorTask, _
dDelayTask, dTimeDiff)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine modifies the WFIStart and WFIEnd times if operator deliberately  @
    '@ delays a batch due to break (Rule No. 5).                                        @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' iCount = Index for Do While-Loop, see calling subroutine [-].
    ' iCwIndex = Index for Do While-Loop, see calling subroutine [-].
    ' dWFIStart(Array) = Global variable [date].
    ' dWFIEnd(Array) = Global variable [date].
    ' dSimTime = Global variable, see calling subroutine [date].
    ' dDelayTask = Global variable, see calling subroutine [date].
    ' dTimeDiff = Global variable, see calling subroutine [date].
    ' dMinTimeTorTask = Global variable, see calling subroutine [date].
    '
    'CALCULATED VALUES/OUTPUTS
    ' Wait = New status allocated. Global variable [-].
    ' dWFIStart(Array) = New time allocated. Global variable [date].
    ' dWFIEnd(Array) = New time allocated. Global variable [date].
    '
    dDelayTask = 0 'Rule 5: Operator at max. efficiency.
    '
    Select Case dSimTime
    Case dFuzzy1ShiftBreak(1, 1) To dFuzzy1ShiftBreak(1, 4)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
        + dFuzzy1ShiftBreak(1, 3)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy1ShiftBreak(1, 5) To dFuzzy1ShiftBreak(1, 8)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
        + dFuzzy1ShiftBreak(1, 7)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy2ShiftBreak(1, 1) To dFuzzy2ShiftBreak(1, 4)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
        + dFuzzy2ShiftBreak(1, 3)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy2ShiftBreak(1, 5) To dFuzzy2ShiftBreak(1, 8)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
        + dFuzzy2ShiftBreak(1, 7)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy3ShiftBreak(1, 1) To dFuzzy3ShiftBreak(1, 4)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
        + dFuzzy3ShiftBreak(1, 3)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy3ShiftBreak(1, 5) To dFuzzy3ShiftBreak(1, 8)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
        + dFuzzy3ShiftBreak(1, 7)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case Else
        Stop        'Error trap. dSimTime wrong.  Check program.
        End
    End Select
    '
End Sub
'
'
'
```

```
Sub Operator_Delays_Batch_DI(iCount, iCWIndex, dSimTime, dMinTimeTorTask, _
dDelayTask, dTimeDiff)
   '
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '@ This subroutine modifies the DIStart and DIEnd times if operator deliberately     @
   '@ delays a batch due to break (Rule No. 5).                                         @
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '
   'INPUTS
   ' iCount = Index for Do While-Loop, see calling subroutine [-].
   ' iCwIndex = Index for Do While-Loop, see calling subroutine [-].
   ' dDIStart(Array) = Global variable [date].
   ' dDIEnd(Array) = Global variable [date].
   ' dSimTime = Global variable, see calling subroutine [date].
   ' dDelayTask = Global variable, see calling subroutine [date].
   ' dTimeDiff = Global variable, see calling subroutine [date].
   ' dMinTimeTorTask = Global variable, see calling subroutine [date].
   '
   'CALCULATED VALUES/OUTPUTS
   ' Wait = New status allocated. Global variable [-].
   ' dDIStart(Array) = New time allocated. Global variable [date].
   ' dDIEnd(Array) = New time allocated. Global variable [date].
   '
   dDelayTask = 0 'Rule 5: Operator at max. efficiency.
   '
   Select Case dSimTime
   Case dFuzzy1ShiftBreak(1, 1) To dFuzzy1ShiftBreak(1, 4)
      dDIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
      + dFuzzy1ShiftBreak(1, 3)
      dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
      Wait(iCount, iCWIndex) = "Operating"
   Case dFuzzy1ShiftBreak(1, 5) To dFuzzy1ShiftBreak(1, 8)
      dDIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
      + dFuzzy1ShiftBreak(1, 7)
      dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
      Wait(iCount, iCWIndex) = "Operating"
   Case dFuzzy2ShiftBreak(1, 1) To dFuzzy2ShiftBreak(1, 4)
      dDIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
      + dFuzzy2ShiftBreak(1, 3)
      dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
      Wait(iCount, iCWIndex) = "Operating"
   Case dFuzzy2ShiftBreak(1, 5) To dFuzzy2ShiftBreak(1, 8)
      dDIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
      + dFuzzy2ShiftBreak(1, 7)
      dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
      Wait(iCount, iCWIndex) = "Operating"
   Case dFuzzy3ShiftBreak(1, 1) To dFuzzy3ShiftBreak(1, 4)
      dDIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
      + dFuzzy3ShiftBreak(1, 3)
      dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
      Wait(iCount, iCWIndex) = "Operating"
   Case dFuzzy3ShiftBreak(1, 5) To dFuzzy3ShiftBreak(1, 8)
      dDIStart(iCount, iCWIndex) = dDelayTask + dMinTimeTorTask _
      + dFuzzy3ShiftBreak(1, 7)
      dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
      Wait(iCount, iCWIndex) = "Operating"
   Case Else          'Other values.
      Stop          'Error trap.  dSimTime wrong.  Check program.
      End
   End Select
   '
End Sub
'
'
'
```

```
Sub Operator_Is_On_Break_WFI(iCount, iCWIndex, dSimTime, dMinDurationOfTask, _
dDelayTask, dTimeDiff)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine modifies the WFIStart and WFIEnd times for the condition     @
    '@ that operator was on a break while task was suppose to begin.                @
    '@ Operator will delay task to end of break & 20% delay as his efficiency is    @
    '@ not as its maximum yet. (Rule 4)                                             @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' iCount = Index for Do While-Loop, see calling subroutine [-].
    ' iCwIndex = Index for Do While-Loop, see calling subroutine [-].
    ' dWFIStart(Array) = Global variable [date].
    ' dWFIEnd(Array) = Global variable [date].
    ' dSimTime = Global variable, see calling subroutine [date].
    ' dDelayTask = Global variable, see calling subroutine [date].
    ' dTimeDiff = Global variable, see calling subroutine [date].
    ' dMinDurationOfTask = Global variable, see calling subroutine [date].
    '
    'CALCULATED VALUES/OUTPUTS
    ' Wait = New status allocated. Global variable [-].
    ' dWFIStart(Array) = New time allocated. Global variable [date].
    ' dWFIEnd(Array) = New time allocated. Global variable [date].
    '
    Select Case dSimTime
    Case dFuzzy1ShiftBreak(1, 2) To dFuzzy1ShiftBreak(1, 3)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy1ShiftBreak(1, 3)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy1ShiftBreak(1, 6) To dFuzzy1ShiftBreak(1, 7)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy1ShiftBreak(1, 7)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy2ShiftBreak(1, 2) To dFuzzy2ShiftBreak(1, 3)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy2ShiftBreak(1, 3)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy2ShiftBreak(1, 6) To dFuzzy2ShiftBreak(1, 7)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy2ShiftBreak(1, 7)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy3ShiftBreak(1, 2) To dFuzzy3ShiftBreak(1, 3)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy3ShiftBreak(1, 3)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy3ShiftBreak(1, 6) To dFuzzy3ShiftBreak(1, 7)
        dWFIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy3ShiftBreak(1, 7)
        dWFIEnd(iCount, iCWIndex) = dWFIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case Else           'Other values.
        Stop            'Error trap. dSimTime wrong.  Check program.
        End
    End Select
    '
End Sub
'
'
'
```

```
Sub Operator_Is_On_Break_DI(iCount, iCWIndex, dSimTime, dDelayTask, dTimeDiff)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine modifies the DIStart and DIEnd times for the condition        @
    '@ that operator was on a break while task was suppose to begin.                 @
    '@ Operator will delay task to end of break & 20% delay as his efficiency is     @
    '@ not as its maximum yet. (Rule 4)                                              @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' iCount = Index for Do While-Loop, see calling subroutine [-].
    ' iCwIndex = Index for Do While-Loop, see calling subroutine [-].
    ' dDIStart(Array) = Global variable [date].
    ' dDIEnd(Array) = Global variable [date].
    ' dSimTime = Global variable, see calling subroutine [date].
    ' dDelayTask = Global variable, see calling subroutine [date].
    ' dTimeDiff = Global variable, see calling subroutine [date].
    ' dMinDurationOfTask = Global variable, see calling subroutine [date].
    '
    'CALCULATED VALUES/OUTPUTS
    ' Wait = New status allocated. Global variable [-].
    ' dDIStart(Array) = New time allocated. Global variable [date].
    ' dDIEnd(Array) = New time allocated. Global variable [date].
    '
    Select Case dSimTime
    Case dFuzzy1ShiftBreak(1, 2) To dFuzzy1ShiftBreak(1, 3)
        dDIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy1ShiftBreak(1, 3)
        dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy1ShiftBreak(1, 6) To dFuzzy1ShiftBreak(1, 7)
        dDIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy1ShiftBreak(1, 7)
        dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy2ShiftBreak(1, 2) To dFuzzy2ShiftBreak(1, 3)
        dDIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy2ShiftBreak(1, 3)
        dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy2ShiftBreak(1, 6) To dFuzzy2ShiftBreak(1, 7)
        dDIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy2ShiftBreak(1, 7)
        dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy3ShiftBreak(1, 2) To dFuzzy3ShiftBreak(1, 3)
        dDIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy3ShiftBreak(1, 3)
        dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case dFuzzy3ShiftBreak(1, 6) To dFuzzy3ShiftBreak(1, 7)
        dDIStart(iCount, iCWIndex) = dDelayTask + dMinDurationOfTask + _
        dFuzzy3ShiftBreak(1, 7)
        dDIEnd(iCount, iCWIndex) = dDIStart(iCount, iCWIndex) + dTimeDiff
        Wait(iCount, iCWIndex) = "Operating"
    Case Else          'Other values.
        Stop           'Error trap.  dSimTime wrong.  Check program.
        End
    End Select
    '
End Sub
'
'
'
```

```
Sub Discrete_Time_Events_Fuzzy_DI()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine modifies the DIStart and DIEnd times by the fuzzy times     @
    '@ as are calculated here.  This routine uses the priority rules and calls             @
    '@ on the rule base if required.                                                                      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' dTimeStart(Array) = Global variable [date].
    ' dTimeEnd(Array) = Global variable [date].
    ' iColNumberDI = Global variable [-].
    ' iRowNumberDI = Global variable [-].
    ' bOptionNoTimeUncertainty = Global variable [-].
    ' sIncrement = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' LSecondsOfDaysCount = Index for Do While-Loop (seconds per day) [-].
    ' iCount = Index for Do While-Loop [-].
    ' iCwIndex = Index for Do While-Loop [-].
    ' iCwIndex2 = Index for Do While-Loop [-].
    ' iCount1 = Index for Do While-Loop [-].
    ' iCount2 = Index for Do While-Loop [-].
    ' LSecondsOfDaysCount = Index for Do While-Loop [-].
    ' LCount = Index for Do While-Loop [-].
    ' dSimTime = Current simulated time [date].
    ' dTime = Current simulated time [date].
    ' dTimeDiff = Time differential between two time frames [date].
    ' iCounterEarliestTask1 = To remember the earliest task found for priority 1 [-].
    ' iCounterEarliestTask2 = To remember the earliest task found for priority 2 [-].
    ' iCounterEarliestTask3 = To remember the earliest task found for priority 3 [-].
    ' dEarliestTask1 = Time index of earliest task 1 found [date].
    ' dEarliestTask2 = Time index of earliest task 2 found [date].
    ' dEarliestTask3 = Time index of earliest task 3 found [date].
    ' iNumOperatorsOccupied = Number of operators occupied with tasks [-].
    ' iIntermediate = Variable for evaluation of simulated time [-].
    ' dMinDurationOfTask = Minimum time operator is occupied with a task [date].
    ' dDelayTask = Calculated delay for operator for a given task [date].
    ' Wait = Global variable [-].
    ' Rule1DelayAfterBreak = Global variable [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' dDIStart(Array) = Global variable [date].
    ' dDIEnd(Array) = Global variable [date].
    '
    Dim iCount As Integer, iCWIndex As Integer, iNumOperatorsOccupied As Integer
    Dim iCount1 As Integer, iCount2 As Integer, iCwIndex2 As Integer, iIntermediate As Integer
    Dim iCounterEarliestTask1(2) As Integer, iCounterEarliestTask2(2) As Integer
    Dim iCounterEarliestTask3(2) As Integer
    Dim LSecondsOfDaysCount As Long, LCount As Long
    Dim dSimTime As Date, dDelayTask As Date, dTimeDiff As Date, dMinDurationOfTask As Date
    Dim dEarliestTask1 As Date, dEarliestTask2 As Date, dEarliestTask3 As Date
    '
    Const dZeroTime As Date = 0 'Time frame 00:00:00 [date].
    '
    'Start values:
    LSecondsOfDaysCount = 1
    '
    Do While LSecondsOfDaysCount < LOneDayInSeconds + 1
        'Evaluate which simulated time (dSimTime) it is:
        Select Case LSecondsOfDaysCount
        Case 1 To 32400    'Number between 1 and 32400, inclusive.
            dSimTime = TimeSerial(0, 0, LSecondsOfDaysCount)
        Case 32401 To 64800
            iIntermediate = LSecondsOfDaysCount - 32400
            dSimTime = TimeSerial(9, 0, iIntermediate)
        Case 64801 To 86400
            iIntermediate = LSecondsOfDaysCount - 64800
            dSimTime = TimeSerial(18, 0, iIntermediate)
```

```
Case Else          'Other values.
   Stop            'Error trap.
               'LSecondsOfDaysCount cannot be lower than zero or higher than
               '86400.  Check program.
   End
End Select
'
iCWIndex = 1
Do While iCWIndex < iColNumberDI + 1
   iCount = 1
   Do While iCount < iRowNumberDI + 1
      '
      '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      'Condition: An operator is no longer occupied with a task.  The operator is
      'available to start another task which is "waiting" (see code below).
      '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      '
      If (dSimTime = dDIStart(iCount, iCWIndex) _
      And (Wait(iCount, iCWIndex) = "Operating") _
      And iDIPriority(iCount, iCWIndex) > 0) Then
         '
         iNumOperatorsOccupied = iNumOperatorsOccupied - 1
         '
         iCounterEarliestTask1(1) = 0
         iCounterEarliestTask1(2) = 0
         iCounterEarliestTask2(1) = 0
         iCounterEarliestTask2(2) = 0
         iCounterEarliestTask3(1) = 0
         iCounterEarliestTask3(2) = 0
         dEarliestTask3 = CSng(86400)
         iCwIndex2 = 1
         '
         'Search only for tasks which are "Waiting" and have Priority = 3:
         Do While iCwIndex2 < iColNumberDI + 1
            iCount2 = 1
            Do While iCount2 < iRowNumberDI + 1
               If Wait(iCount2, iCwIndex2) = "Waiting" _
               And iDIPriority(iCount2, iCwIndex2) = 3 Then
                  If (CSng(dDIStart(iCount2, iCwIndex2)) _
                  < (CSng(dEarliestTask3))) Then
                     dEarliestTask3 = dDIStart(iCount2, iCwIndex2)
                     iCounterEarliestTask3(1) = iCount2
                     iCounterEarliestTask3(2) = iCwIndex2
                  End If
               End If
               iCount2 = iCount2 + 1
            Loop
            iCwIndex2 = iCwIndex2 + 1
         Loop
         '
         'Search only for tasks which are "Waiting" and have Priority = 2:
         dEarliestTask2 = CSng(86400)
         iCwIndex2 = 1
         Do While iCwIndex2 < iColNumberDI + 1
            iCount2 = 1
            Do While iCount2 < iRowNumberDI + 1
               If Wait(iCount2, iCwIndex2) = "Waiting" _
               And iDIPriority(iCount2, iCwIndex2) = 2 Then
                  If (CSng(dDIStart(iCount2, iCwIndex2)) _
                  < (CSng(dEarliestTask2))) Then
                     dEarliestTask2 = dDIStart(iCount2, iCwIndex2)
                     iCounterEarliestTask2(1) = iCount2
                     iCounterEarliestTask2(2) = iCwIndex2
                  End If
               End If
               iCount2 = iCount2 + 1
            Loop
            iCwIndex2 = iCwIndex2 + 1
         Loop
```

```
'
'Search only for tasks which are "Waiting" and have Priority = 1:
dEarliestTask1 = CSng(86400)
iCwIndex2 = 1
Do While iCwIndex2 < iColNumberDI + 1
   iCount2 = 1
   Do While iCount2 < iRowNumberDI + 1
      If Wait(iCount2, iCwIndex2) = "Waiting" _
      And iDIPriority(iCount2, iCwIndex2) = 1 Then
         If (CSng(dDIStart(iCount2, iCwIndex2)) _
         < (CSng(dEarliestTask1))) Then
            dEarliestTask1 = dDIStart(iCount2, iCwIndex2)
            iCounterEarliestTask1(1) = iCount2
            iCounterEarliestTask1(2) = iCwIndex2
         End If
      End If
      iCount2 = iCount2 + 1
   Loop
   iCwIndex2 = iCwIndex2 + 1
Loop
'
'First: Priority = 3 (highest priority).
'If no Priority 3 tasks jump to Priority 2:
If iCounterEarliestTask3(1) = 0 Or iCounterEarliestTask3(2) = 0 Then
   GoTo NoPriority3
End If
'
If Wait(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) = "Waiting" _
And iDIPriority(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
= 3 Then
   Call Fuzzification(LSecondsOfDaysCount)
   'Assign new dDIStart and dDIEnd time for this process.
   Call Rule_Base_DI(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2), dSimTime)
   '
   dTimeDiff = dDIEnd(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2)) - dDIStart(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2))
   '
   'Calculate the "Minimum Duration of Task":
   dMinDurationOfTask = dDITimeFuzzy3(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2)) - dDITimeFuzzy2(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2))
   '
   'Calculate the delay of the task:
   dDelayTask = sDefuzzifier / 100 _
   * (dDITimeFuzzy2(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2)) - dDITimeFuzzy1(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2)) + dDITimeFuzzy4(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2)) - dDITimeFuzzy3(iCounterEarliestTask3(1), _
   iCounterEarliestTask3(2)))
   '
   If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
   Or Rule5Activated = "Yes" Then
      '
      If Rule1Activated = "Yes" Then
         'Below allocate new start and end times to tasks:
         dDIStart(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
         = Rule1DelayAfterBreak + dMinDurationOfTask + dDelayTask
         dDIEnd(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
         = dDIStart(iCounterEarliestTask3(1), _
         iCounterEarliestTask3(2)) + dTimeDiff
         Wait(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
         = "Operating"
      End If
      '
      If Rule4Activated = "Yes" Then
         'Operator on break while process becomes available (Rule 4):
         Call Operator_Is_On_Break_DI(iCounterEarliestTask3(1), _
```

```
                    iCounterEarliestTask3(2), dSimTime, _
                    dMinDurationOfTask, dDelayTask, dTimeDiff)
                End If
                '
                If Rule5Activated = "Yes" Then
                    'Operator delays batch deliberately (Rule 5):
                    Call Operator_Delays_Batch_DI(iCounterEarliestTask3(1), _
                    iCounterEarliestTask3(2), dSimTime, dMinDurationOfTask, _
                    dDelayTask, dTimeDiff)
                End If
                '
            Else
                'Case: Operator was not on a break when task started.
                'Below allocate new start and end times to tasks:
                dDIStart(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
                + dMinDurationOfTask + dDelayTask
                dDIEnd(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                = dDIStart(iCounterEarliestTask3(1), _
                iCounterEarliestTask3(2)) + dTimeDiff
                Wait(iCounterEarliestTask3(1), iCounterEarliestTask3(2)) _
                = "Operating"
            End If
            '
            Call DI_Time_Check(iCounterEarliestTask3(1), _
            iCounterEarliestTask3(2), LSecondsOfDaysCount)
            '
            iNumOperatorsOccupied = iNumOperatorsOccupied + 1
            GoTo FoundTask  'Exit the loop, found the most urgent task.
            '
        End If
        '
        'Second: Priority = 2 (median priority).
NoPriority3:
        If iCounterEarliestTask2(1) = 0 Or iCounterEarliestTask2(2) = 0 Then
            GoTo NoPriority2
        End If
        '
        If Wait(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
        = "Waiting" And iDIPriority(iCounterEarliestTask2(1), _
        iCounterEarliestTask2(2)) = 2 Then
            '
            Call Fuzzification(LSecondsOfDaysCount)
            '
            Call Rule_Base_DI(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2), dSimTime)
            '
            dTimeDiff = dDIEnd(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2)) - dDIStart(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2))
            '
            'Calculate the "Minimum Duration of Task":
            dMinDurationOfTask = dDITimeFuzzy3(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2)) - dDITimeFuzzy2(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2))
            '
            'Calculate the delay of the task:
            dDelayTask = sDefuzzifier / 100 _
            * (dDITimeFuzzy2(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2)) - dDITimeFuzzy1(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2)) + dDITimeFuzzy4(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2)) - dDITimeFuzzy3(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2)))
            '
            If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
            Or Rule5Activated = "Yes" Then
                '
                If Rule1Activated = "Yes" Then
                    'Below allocate new start and end times to tasks:
```

```
                    dDIStart(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
                    = Rule1DelayAfterBreak + dMinDurationOfTask + dDelayTask
                    dDIEnd(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
                    = dDIStart(iCounterEarliestTask2(1), _
                    iCounterEarliestTask2(2)) + dTimeDiff
                    Wait(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) = _
                    "Operating"
                End If
                '
                If Rule4Activated = "Yes" Then
                    'Operator on break while process becomes available (Rule 4):
                    Call Operator_Is_On_Break_DI(iCounterEarliestTask2(1), _
                    iCounterEarliestTask2(2), dSimTime, dMinDurationOfTask, _
                    dDelayTask, dTimeDiff)
                End If
                '
                If Rule5Activated = "Yes" Then
                    'Operator delays batch deliberately (Rule 5):
                    Call Operator_Delays_Batch_DI(iCounterEarliestTask2(1), _
                    iCounterEarliestTask2(2), dSimTime, dMinDurationOfTask, _
                    dDelayTask, dTimeDiff)
                End If
                '
            Else
                'Case: Operator was not on a break when task started.
                'Below allocate new start and end times to tasks:
                dDIStart(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
                = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
                + dMinDurationOfTask + dDelayTask
                dDIEnd(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
                = dDIStart(iCounterEarliestTask2(1), iCounterEarliestTask2(2)) _
                + dTimeDiff
                Wait(iCounterEarliestTask2(1), iCWIndex) = "Operating"
            End If
            '
            Call DI_Time_Check(iCounterEarliestTask2(1), _
            iCounterEarliestTask2(2), LSecondsOfDaysCount)
            '
            iNumOperatorsOccupied = iNumOperatorsOccupied + 1
            GoTo FoundTask
        End If
        '
        'Third: Priority = 1 (lowest priority).  Condition: no more tasks to do.
NoPriority2:
        If iCounterEarliestTask1(1) = 0 Or iCounterEarliestTask1(2) = 0 Then
            GoTo NoMoreTasks    'Exit the loop, found the most urgent task.
        End If
        '
        If Wait(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) = "Waiting" _
        And iDIPriority(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
        = 1 Then
            '
            Call Fuzzification(LSecondsOfDaysCount)
            '
            Call Rule_Base_DI(iCounterEarliestTask1(1), _
            iCounterEarliestTask1(2), dSimTime)
            '
            dTimeDiff = dDIEnd(iCounterEarliestTask1(1), _
            iCounterEarliestTask1(2)) - dDIStart(iCounterEarliestTask1(1), _
            iCounterEarliestTask1(2))
            '
            'Calculate the "Minimum Duration of Task":
            dMinDurationOfTask = dDITimeFuzzy3(iCounterEarliestTask1(1), _
            iCounterEarliestTask1(2)) - dDITimeFuzzy2(iCounterEarliestTask1(1), _
            iCounterEarliestTask1(2))
            '
            'Calculate the delay of the task:
            dDelayTask = sDefuzzifier / 100 _
            * (dDITimeFuzzy2(iCounterEarliestTask1(1), _
```

```
                    iCounterEarliestTask1(2)) - dDITimeFuzzy1(iCounterEarliestTask1(1), _
                    iCounterEarliestTask1(2)) + dDITimeFuzzy4(iCounterEarliestTask1(1), _
                    iCounterEarliestTask1(2)) - dDITimeFuzzy3(iCounterEarliestTask1(1), _
                    iCounterEarliestTask1(2)))
                    '
                    If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
                    Or Rule5Activated = "Yes" Then
                       '
                       If Rule1Activated = "Yes" Then
                          'Below allocate new start and end times to tasks:
                          dDIStart(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                          = Rule1DelayAfterBreak + dMinDurationOfTask + dDelayTask
                          dDIEnd(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                          = dDIStart(iCounterEarliestTask1(1), _
                          iCounterEarliestTask1(2)) + dTimeDiff
                          Wait(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) = _
                          "Operating"
                       End If
                       '
                       If Rule4Activated = "Yes" Then
                          'Operator on break while process becomes available (Rule 4):
                          Call Operator_Is_On_Break_DI(iCounterEarliestTask1(1), _
                          iCounterEarliestTask1(2), dSimTime, dMinDurationOfTask, _
                          dDelayTask, dTimeDiff)
                       End If
                       '
                       If Rule5Activated = "Yes" Then
                          'Operator delays batch deliberately (Rule 5):
                          Call Operator_Delays_Batch_DI(iCounterEarliestTask1(1), _
                          iCounterEarliestTask1(2), dSimTime, dMinDurationOfTask, _
                          dDelayTask, dTimeDiff)
                       End If
                       '
                    Else
                       'Case: Operator was not on a break when task started.
                       'Below allocate new start and end times to tasks:
                       dDIStart(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                       = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
                       + dMinDurationOfTask + dDelayTask
                       dDIEnd(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                       = dDIStart(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) _
                       + dTimeDiff
                       Wait(iCounterEarliestTask1(1), iCounterEarliestTask1(2)) = _
                       "Operating"
                    End If
                    '
                    Call DI_Time_Check(iCounterEarliestTask1(1), _
                    iCounterEarliestTask1(2), LSecondsOfDaysCount)
                    '
                    iNumOperatorsOccupied = iNumOperatorsOccupied + 1
                    GoTo FoundTask  'Exit the loop, found the most urgent task.
                    '
                 End If
              End If
              '
NoMoreTasks:
              iCount = iCount + 1
           Loop
           iCWIndex = iCWIndex + 1
        Loop
        '
FoundTask:
        iCWIndex = 1
        '
        '#########################################################################
        'The code below is activated if a task starts, which has not been previously delayed
        '(see code above). In addition this code assigns "waiting" should no operator
        ' be available to start a task.
        '#########################################################################
```

```
'
Do While iCWIndex < iColNumberDI + 1
   iCount = 1
   Do While iCount < iRowNumberDI + 1
      'If Block to check how many operators are working at any one time.
      If (CSng(dSimTime) = CSng(dDIStart(iCount, iCWIndex)) _
      And ((Wait(iCount, iCWIndex) = "Waiting") _
      Or (Wait(iCount, iCWIndex) = "") _
      And iDIPriority(iCount, iCWIndex) > 0)) Then
         '
         iNumOperatorsOccupied = iNumOperatorsOccupied + 1
         '
         If iNumOperatorsOccupied > iNumberOfOperators Then
            '
            'All operators are occupied and task duration is not zero,
            'task has to wait.  Tasks with task duration zero will be served
            'regardless.
            '
            Wait(iCount, iCWIndex) = "Waiting"
            '
            iNumOperatorsOccupied = iNumOperatorsOccupied - 1
         Else
            Wait(iCount, iCWIndex) = "Operating"
            '
            Call Fuzzification(LSecondsOfDaysCount)
            '
            Call Rule_Base_DI(iCount, iCWIndex, dSimTime)
            '
            dTimeDiff = dDIEnd(iCount, iCWIndex) - dDIStart(iCount, iCWIndex)
            '
            'Calculate the "Minimum Duration of Task":
            dMinDurationOfTask = dDITimeFuzzy3(iCount, iCWIndex) _
            - dDITimeFuzzy2(iCount, iCWIndex)
            '
            'Calculate the delay of the task:
            dDelayTask = sDefuzzifier / 100 * (dDITimeFuzzy2(iCount, iCWIndex) _
            - dDITimeFuzzy1(iCount, iCWIndex) + dDITimeFuzzy4(iCount, iCWIndex) _
            - dDITimeFuzzy3(iCount, iCWIndex))
            '
            If Rule1Activated = "Yes" Or Rule4Activated = "Yes" _
            Or Rule5Activated = "Yes" Then
               '
               If Rule1Activated = "Yes" Then
                  'Below allocates new start and end times to tasks:
                  dDIStart(iCount, iCWIndex) = Rule1DelayAfterBreak _
                  + dMinDurationOfTask + dDelayTask
                  dDIEnd(iCount, iCWIndex) _
                  = dDIStart(iCount, iCWIndex) + dTimeDiff
                  Wait(iCount, iCWIndex) = "Operating"
               End If
               '
               If Rule4Activated = "Yes" Then
                  'Operator on break while process becomes available (Rule 4):
                  Call Operator_Is_On_Break_DI(iCount, iCWIndex, dSimTime, _
                  dMinDurationOfTask, dDelayTask, dTimeDiff)
               End If
               '
               If Rule5Activated = "Yes" Then
                  'Operator delays batch deliberately (Rule 5):
                  Call Operator_Delays_Batch_DI(iCount, _
                  iCWIndex, dSimTime, dMinDurationOfTask, dDelayTask, dTimeDiff)
               End If
               '
            Else
               'Case: Operator was not on a break when task started.
               'Below allocate new start and end times to tasks:
               dDIStart(iCount, iCWIndex) _
               = DateAdd("s", LSecondsOfDaysCount, dZeroTime) _
               + dMinDurationOfTask + dDelayTask
```

```
                    dDIEnd(iCount, iCWIndex) _
                    = dDIStart(iCount, iCWIndex) + dTimeDiff
                    Wait(iCount, iCWIndex) = "Operating"
                End If
                '
                Call DI_Time_Check(iCount, iCWIndex, LSecondsOfDaysCount)
                '
                If dDITimeFuzzy3(iCount, iCWIndex) = 0 _
                And dDITimeFuzzy4(iCount, iCWIndex) = 0 Then
                    iNumOperatorsOccupied = iNumOperatorsOccupied - 1
                End If
                '
            End If
        End If
        iCount = iCount + 1
    Loop
    iCWIndex = iCWIndex + 1
  Loop
  LSecondsOfDaysCount = LSecondsOfDaysCount + 1
Loop
'
End Sub
'
'
'
Sub Fuzzification(LSecondsOfDaysCount)
'
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'@ This subroutine fuzzifies all fuzzy input variables.    @
'@@@@@@@@@@@@@@@@@@@@@@@@@@@@
'
'INPUTS
' LSecondsOfDaysCount = Time of simulation (see calling subroutine) [-].
'
'INTERMEDIATE VARIABLES
' x = x value for calculation of linear equation: y = m*x + b [date].
' y = y value for calculation of linear equation: y = m*x + b [-].
' m = m value for calculation of linear equation: y = m*x + b [-].
' b = b value for calculation of linear equation: y = m*x + b [-].
' dTimeDiff = Time difference between two time frames [date].
' iIntermediate = Intermediate value for time calculation [-].
' sTimeDiff = Time difference between two time frames [s].[-].
' dSimTime = Current simulated time [date].
' d2Shift = Additional time to get time to 2nd shift [date].
' d3Shift = Additional time to get time to 3rd shift [date].
'
'CALCULATED VALUES/OUTPUTS
' sMuiFuzzy1Shift(Array) = Global variable [-].
' sMuiFuzzy2Shift(Array) = Global variable [-].
' sMuiFuzzy3Shift(Array) = Global variable [-].
' sMuiFuzzyBreak(Array) = Global variable [-].
' sMuiFuzzyTired(Array) = Global variable [-].
'
Dim x As Variant, y As Variant, m As Variant, b As Variant
Dim d2Shift As Date, d3Shift As Date, dTimeDiff As Date, dSimTime As Date
Dim iIntermediate As Integer
Dim sTimeDiff As Single
'
'Evaluate simulated time:
Select Case LSecondsOfDaysCount
Case 1 To 32400
  dSimTime = TimeSerial(0, 0, LSecondsOfDaysCount)
Case 32401 To 64800
  iIntermediate = LSecondsOfDaysCount - 32400
  dSimTime = TimeSerial(9, 0, iIntermediate)
Case 64801 To 86400
  iIntermediate = LSecondsOfDaysCount - 64800
  dSimTime = TimeSerial(18, 0, iIntermediate)
Case Else
```

```
    Stop          'Error Trap for LSecondsOfDaysCount.
    End
End Select
'
'Reset all calculated values:
sMuiFuzzy1Shift = 0
sMuiFuzzy2Shift = 0
sMuiFuzzy3Shift = 0
sMuiFuzzyBreak = 0
sMuiFuzzyTired = 0
'
'Parameters to calculate time of 2nd and 3rd shift pattern:
d2Shift = TimeSerial(8, 0, 0)
d3Shift = TimeSerial(16, 0, 0)
'
'-------------------------------------------------------------------------------
'
'Start Fuzzyfication Fuzzy Shift.   From: Sheets("Fuzzy Shift").
'1st Shift:
If dSimTime > dFuzzy1Shift(1, 1) And dSimTime <= dFuzzy1Shift(1, 2) Then
    'Linear equation: y = m*x + b
    x = DateDiff("s", dFuzzy1Shift(1, 1), dFuzzy1Shift(1, 2))
    y = sFuzzy1Shift(2, 2) - sFuzzy1Shift(2, 1)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1Shift(1, 1), dSimTime)
    sMuiFuzzy1Shift = m * sTimeDiff + sFuzzy1Shift(2, 1)
    '
    'In addition: sMuiFuzzy3Shift must be calculated (not shown on spreadsheet).
    x = DateDiff("s", dFuzzy1Shift(1, 3), dFuzzy1Shift(1, 4))
    y = sFuzzy1Shift(2, 4) - sFuzzy1Shift(2, 3)
    m = y / x
    'Factor TimeSerial(7, 0, 0) is in to adjust time to end of 1st Shift.
    sTimeDiff = DateDiff("s", dFuzzy1Shift(1, 3), (dSimTime + TimeSerial(8, 0, 0)))
    sMuiFuzzy3Shift = m * sTimeDiff + sFuzzy1Shift(2, 3)
End If
If dSimTime > dFuzzy1Shift(1, 2) And dSimTime <= dFuzzy1Shift(1, 3) Then
    x = DateDiff("s", dFuzzy1Shift(1, 2), dFuzzy1Shift(1, 3))
    y = sFuzzy1Shift(2, 3) - sFuzzy1Shift(2, 2)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1Shift(1, 2), dSimTime)
    sMuiFuzzy1Shift = m * sTimeDiff + sFuzzy1Shift(2, 2)
End If
If dSimTime > dFuzzy1Shift(1, 3) And dSimTime <= dFuzzy1Shift(1, 4) Then
    x = DateDiff("s", dFuzzy1Shift(1, 3), dFuzzy1Shift(1, 4))
    y = sFuzzy1Shift(2, 4) - sFuzzy1Shift(2, 3)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1Shift(1, 3), dSimTime)
    sMuiFuzzy1Shift = m * sTimeDiff + sFuzzy1Shift(2, 3)
End If
'
'2nd Shift:
If dSimTime > dFuzzy2Shift(1, 1) And dSimTime <= dFuzzy2Shift(1, 2) Then
    x = DateDiff("s", dFuzzy2Shift(1, 1), dFuzzy2Shift(1, 2))
    y = sFuzzy2Shift(2, 2) - sFuzzy2Shift(2, 1)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy2Shift(1, 1), dSimTime)
    sMuiFuzzy2Shift = m * sTimeDiff + sFuzzy2Shift(2, 1)
End If
If dSimTime > dFuzzy2Shift(1, 2) And dSimTime <= dFuzzy2Shift(1, 3) Then
    x = DateDiff("s", dFuzzy2Shift(1, 2), dFuzzy2Shift(1, 3))
    y = sFuzzy2Shift(2, 3) - sFuzzy2Shift(2, 2)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy2Shift(1, 2), dSimTime)
    sMuiFuzzy2Shift = m * sTimeDiff + sFuzzy2Shift(2, 2)
End If
If dSimTime > dFuzzy2Shift(1, 3) And dSimTime <= dFuzzy2Shift(1, 4) Then
    x = DateDiff("s", dFuzzy2Shift(1, 3), dFuzzy2Shift(1, 4))
    y = sFuzzy2Shift(2, 4) - sFuzzy2Shift(2, 3)
    m = y / x
```

```
    sTimeDiff = DateDiff("s", dFuzzy2Shift(1, 3), dSimTime)
    sMuiFuzzy2Shift = m * sTimeDiff + sFuzzy2Shift(2, 3)
End If
'
'3rd Shift:
If dSimTime > dFuzzy3Shift(1, 1) And dSimTime <= dFuzzy3Shift(1, 2) Then
    x = DateDiff("s", dFuzzy3Shift(1, 1), dFuzzy3Shift(1, 2))
    y = sFuzzy3Shift(2, 2) - sFuzzy3Shift(2, 1)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy3Shift(1, 1), dSimTime)
    sMuiFuzzy3Shift = m * sTimeDiff + sFuzzy3Shift(2, 1)
End If
If dSimTime > dFuzzy3Shift(1, 2) And dSimTime <= dFuzzy3Shift(1, 3) Then
    x = DateDiff("s", dFuzzy3Shift(1, 2), dFuzzy3Shift(1, 3))
    y = sFuzzy3Shift(2, 3) - sFuzzy3Shift(2, 2)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy3Shift(1, 2), dSimTime)
    sMuiFuzzy3Shift = m * sTimeDiff + sFuzzy3Shift(2, 2)
End If
If dSimTime > dFuzzy3Shift(1, 3) And dSimTime <= TimeSerial(23, 59, 59) Then
    x = DateDiff("s", dFuzzy3Shift(1, 3), TimeSerial(23, 59, 59))
    y = sFuzzy3Shift(2, 4) - sFuzzy3Shift(2, 3)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy3Shift(1, 3), dSimTime)
    sMuiFuzzy3Shift = m * sTimeDiff + sFuzzy3Shift(2, 3)
End If
'End Fuzzification Fuzzy Shift.
'
'-------------------------------------------------------------------------------
'
'Start Fuzzyfication Fuzzy Breaks.   From: Sheets("Fuzzy Breaks").
Select Case dSimTime
'1st Shift:
Case dFuzzy1ShiftBreak(1, 1) To dFuzzy1ShiftBreak(1, 2)
    x = DateDiff("s", dFuzzy1ShiftBreak(1, 1), dFuzzy1ShiftBreak(1, 2))
    y = sFuzzy1ShiftBreak(2, 2) - sFuzzy1ShiftBreak(2, 1)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1ShiftBreak(1, 1), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy1ShiftBreak(2, 1)
Case dFuzzy1ShiftBreak(1, 2) To dFuzzy1ShiftBreak(1, 3)
    x = DateDiff("s", dFuzzy1ShiftBreak(1, 2), dFuzzy1ShiftBreak(1, 3))
    y = sFuzzy3ShiftBreak(2, 3) - sFuzzy3ShiftBreak(2, 2)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1ShiftBreak(1, 2), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy1ShiftBreak(2, 2)
Case dFuzzy1ShiftBreak(1, 3) To dFuzzy1ShiftBreak(1, 4)
    x = DateDiff("s", dFuzzy1ShiftBreak(1, 3), dFuzzy1ShiftBreak(1, 4))
    y = sFuzzy1ShiftBreak(2, 4) - sFuzzy1ShiftBreak(2, 3)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1ShiftBreak(1, 3), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy1ShiftBreak(2, 3)
Case dFuzzy1ShiftBreak(1, 4) To dFuzzy1ShiftBreak(1, 5)
    x = DateDiff("s", dFuzzy1ShiftBreak(1, 4), dFuzzy1ShiftBreak(1, 5))
    y = sFuzzy1ShiftBreak(2, 5) - sFuzzy1ShiftBreak(2, 4)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1ShiftBreak(1, 4), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy1ShiftBreak(2, 4)
Case dFuzzy1ShiftBreak(1, 5) To dFuzzy1ShiftBreak(1, 6)
    x = DateDiff("s", dFuzzy1ShiftBreak(1, 5), dFuzzy1ShiftBreak(1, 6))
    y = sFuzzy1ShiftBreak(2, 6) - sFuzzy1ShiftBreak(2, 5)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1ShiftBreak(1, 5), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy1ShiftBreak(2, 5)
Case dFuzzy1ShiftBreak(1, 6) To dFuzzy1ShiftBreak(1, 7)
    x = DateDiff("s", dFuzzy1ShiftBreak(1, 6), dFuzzy1ShiftBreak(1, 7))
    y = sFuzzy1ShiftBreak(2, 7) - sFuzzy1ShiftBreak(2, 6)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy1ShiftBreak(1, 6), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy1ShiftBreak(2, 6)
```

```
Case dFuzzy1ShiftBreak(1, 7) To dFuzzy1ShiftBreak(1, 8)
   x = DateDiff("s", dFuzzy1ShiftBreak(1, 7), dFuzzy1ShiftBreak(1, 8))
   y = sFuzzy1ShiftBreak(2, 8) - sFuzzy1ShiftBreak(2, 7)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy1ShiftBreak(1, 7), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy1ShiftBreak(2, 7)
   '
'2nd Shift:
Case dFuzzy2ShiftBreak(1, 1) To dFuzzy2ShiftBreak(1, 2)
   x = DateDiff("s", dFuzzy2ShiftBreak(1, 1), dFuzzy2ShiftBreak(1, 2))
   y = sFuzzy2ShiftBreak(2, 2) - sFuzzy2ShiftBreak(2, 1)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy2ShiftBreak(1, 1), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy2ShiftBreak(2, 1)
Case dFuzzy2ShiftBreak(1, 2) To dFuzzy2ShiftBreak(1, 3)
   x = DateDiff("s", dFuzzy2ShiftBreak(1, 2), dFuzzy2ShiftBreak(1, 3))
   y = sFuzzy2ShiftBreak(2, 3) - sFuzzy2ShiftBreak(2, 2)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy2ShiftBreak(1, 2), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy2ShiftBreak(2, 2)
Case dFuzzy2ShiftBreak(1, 3) To dFuzzy2ShiftBreak(1, 4)
   x = DateDiff("s", dFuzzy2ShiftBreak(1, 3), dFuzzy2ShiftBreak(1, 4))
   y = sFuzzy2ShiftBreak(2, 4) - sFuzzy2ShiftBreak(2, 3)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy2ShiftBreak(1, 3), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy2ShiftBreak(2, 3)
Case dFuzzy2ShiftBreak(1, 4) To dFuzzy2ShiftBreak(1, 5)
   x = DateDiff("s", dFuzzy2ShiftBreak(1, 4), dFuzzy2ShiftBreak(1, 5))
   y = sFuzzy2ShiftBreak(2, 5) - sFuzzy2ShiftBreak(2, 4)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy2ShiftBreak(1, 4), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy2ShiftBreak(2, 4)
Case dFuzzy2ShiftBreak(1, 5) To dFuzzy2ShiftBreak(1, 6)
   x = DateDiff("s", dFuzzy2ShiftBreak(1, 5), dFuzzy2ShiftBreak(1, 6))
   y = sFuzzy2ShiftBreak(2, 6) - sFuzzy2ShiftBreak(2, 5)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy2ShiftBreak(1, 5), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy2ShiftBreak(2, 5)
Case dFuzzy2ShiftBreak(1, 6) To dFuzzy2ShiftBreak(1, 7)
   x = DateDiff("s", dFuzzy2ShiftBreak(1, 6), dFuzzy2ShiftBreak(1, 7))
   y = sFuzzy2ShiftBreak(2, 7) - sFuzzy2ShiftBreak(2, 6)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy2ShiftBreak(1, 6), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy2ShiftBreak(2, 6)
Case dFuzzy2ShiftBreak(1, 7) To dFuzzy2ShiftBreak(1, 8)
   x = DateDiff("s", dFuzzy2ShiftBreak(1, 7), dFuzzy2ShiftBreak(1, 8))
   y = sFuzzy2ShiftBreak(2, 8) - sFuzzy2ShiftBreak(2, 7)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy2ShiftBreak(1, 7), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy2ShiftBreak(2, 7)
   '
'3rd Shift
Case dFuzzy3ShiftBreak(1, 1) To dFuzzy3ShiftBreak(1, 2)
   x = DateDiff("s", dFuzzy3ShiftBreak(1, 1), dFuzzy3ShiftBreak(1, 2))
   y = sFuzzy3ShiftBreak(2, 2) - sFuzzy3ShiftBreak(2, 1)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy3ShiftBreak(1, 1), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy3ShiftBreak(2, 1)
Case dFuzzy3ShiftBreak(1, 2) To dFuzzy3ShiftBreak(1, 3)
   x = DateDiff("s", dFuzzy3ShiftBreak(1, 2), dFuzzy3ShiftBreak(1, 3))
   y = sFuzzy3ShiftBreak(2, 3) - sFuzzy3ShiftBreak(2, 2)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy3ShiftBreak(1, 2), dSimTime)
   sMuiFuzzyBreak = m * sTimeDiff + sFuzzy3ShiftBreak(2, 2)
Case dFuzzy3ShiftBreak(1, 3) To dFuzzy3ShiftBreak(1, 4)
   x = DateDiff("s", dFuzzy3ShiftBreak(1, 3), dFuzzy3ShiftBreak(1, 4))
   y = sFuzzy3ShiftBreak(2, 4) - sFuzzy3ShiftBreak(2, 3)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzy3ShiftBreak(1, 3), dSimTime)
```

```
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy3ShiftBreak(2, 3)
Case dFuzzy3ShiftBreak(1, 4) To dFuzzy3ShiftBreak(1, 5)
    x = DateDiff("s", dFuzzy3ShiftBreak(1, 4), dFuzzy3ShiftBreak(1, 5))
    y = sFuzzy3ShiftBreak(2, 5) - sFuzzy3ShiftBreak(2, 4)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy3ShiftBreak(1, 4), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy3ShiftBreak(2, 4)
Case dFuzzy3ShiftBreak(1, 5) To dFuzzy3ShiftBreak(1, 6)
    x = DateDiff("s", dFuzzy3ShiftBreak(1, 5), dFuzzy3ShiftBreak(1, 6))
    y = sFuzzy3ShiftBreak(2, 6) - sFuzzy3ShiftBreak(2, 5)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy3ShiftBreak(1, 5), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy3ShiftBreak(2, 5)
Case dFuzzy3ShiftBreak(1, 6) To dFuzzy3ShiftBreak(1, 7)
    x = DateDiff("s", dFuzzy3ShiftBreak(1, 6), dFuzzy3ShiftBreak(1, 7))
    y = sFuzzy3ShiftBreak(2, 7) - sFuzzy3ShiftBreak(2, 6)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy3ShiftBreak(1, 6), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy3ShiftBreak(2, 6)
Case dFuzzy3ShiftBreak(1, 7) To dFuzzy3ShiftBreak(1, 8)
    x = DateDiff("s", dFuzzy3ShiftBreak(1, 7), dFuzzy3ShiftBreak(1, 8))
    y = sFuzzy3ShiftBreak(2, 8) - sFuzzy3ShiftBreak(2, 7)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzy3ShiftBreak(1, 7), dSimTime)
    sMuiFuzzyBreak = m * sTimeDiff + sFuzzy3ShiftBreak(2, 7)
Case Else                   'Other values.
    sMuiFuzzyBreak = 0      'Operator is not on a break.
End Select
'End Fuzzification Fuzzy Break.
'
'-------------------------------------------------------------------------------
'
'Start Fuzzyfication Fuzzy Tired.
If dSimTime > dFuzzyTired(1, 1) And dSimTime <= dFuzzyTired(1, 2) Then
    x = DateDiff("s", dFuzzyTired(1, 1), dFuzzyTired(1, 2))
    y = sFuzzyTired(2, 2) - sFuzzyTired(2, 1)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 1), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 1)
End If
If dSimTime > dFuzzyTired(1, 2) And dSimTime <= dFuzzyTired(1, 3) Then
    x = DateDiff("s", dFuzzyTired(1, 2), dFuzzyTired(1, 3))
    y = sFuzzyTired(2, 3) - sFuzzyTired(2, 2)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 2), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 2)
End If
If dSimTime > dFuzzyTired(1, 3) And dSimTime <= dFuzzyTired(1, 4) Then
    x = DateDiff("s", dFuzzyTired(1, 3), dFuzzyTired(1, 4))
    y = sFuzzyTired(2, 4) - sFuzzyTired(2, 3)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 3), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 3)
End If
If dSimTime > dFuzzyTired(1, 4) And dSimTime <= dFuzzyTired(1, 5) Then
    x = DateDiff("s", dFuzzyTired(1, 4), dFuzzyTired(1, 5))
    y = sFuzzyTired(2, 5) - sFuzzyTired(2, 4)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 4), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 4)
End If
If dSimTime > dFuzzyTired(1, 5) And dSimTime <= dFuzzyTired(1, 6) Then
    x = DateDiff("s", dFuzzyTired(1, 5), dFuzzyTired(1, 6))
    y = sFuzzyTired(2, 6) - sFuzzyTired(2, 5)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 5), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 5)
End If
If dSimTime > dFuzzyTired(1, 6) And dSimTime <= dFuzzyTired(1, 7) Then
```

```
    x = DateDiff("s", dFuzzyTired(1, 6), dFuzzyTired(1, 7))
    y = sFuzzyTired(2, 7) - sFuzzyTired(2, 6)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 6), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 6)
End If
If dSimTime > dFuzzyTired(1, 7) And dSimTime <= dFuzzyTired(1, 8) Then
    x = DateDiff("s", dFuzzyTired(1, 7), dFuzzyTired(1, 8))
    y = sFuzzyTired(2, 8) - sFuzzyTired(2, 7)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 7), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 7)
End If
If dSimTime > dFuzzyTired(1, 8) And dSimTime <= dFuzzyTired(1, 9) Then
    x = DateDiff("s", dFuzzyTired(1, 8), dFuzzyTired(1, 9))
    y = sFuzzyTired(2, 9) - sFuzzyTired(2, 8)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 8), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 8)
End If
If dSimTime > dFuzzyTired(1, 9) And dSimTime <= dFuzzyTired(1, 10) Then
    x = DateDiff("s", dFuzzyTired(1, 9), dFuzzyTired(1, 10))
    y = sFuzzyTired(2, 10) - sFuzzyTired(2, 9)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 9), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 9)
End If
If dSimTime > dFuzzyTired(1, 10) And dSimTime <= dFuzzyTired(1, 11) Then
    x = DateDiff("s", dFuzzyTired(1, 10), dFuzzyTired(1, 11))
    y = sFuzzyTired(2, 11) - sFuzzyTired(2, 10)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 10), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 10)
End If
If dSimTime > dFuzzyTired(1, 11) And dSimTime <= dFuzzyTired(1, 12) Then
    x = DateDiff("s", dFuzzyTired(1, 11), dFuzzyTired(1, 12))
    y = sFuzzyTired(2, 12) - sFuzzyTired(2, 11)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 11), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 11)
End If
If dSimTime > dFuzzyTired(1, 12) And dSimTime <= dFuzzyTired(1, 13) Then
    x = DateDiff("s", dFuzzyTired(1, 12), dFuzzyTired(1, 13))
    y = sFuzzyTired(2, 13) - sFuzzyTired(2, 12)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 12), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 12)
End If
If dSimTime > dFuzzyTired(1, 13) And dSimTime <= dFuzzyTired(1, 14) Then
    x = DateDiff("s", dFuzzyTired(1, 13), dFuzzyTired(1, 14))
    y = sFuzzyTired(2, 14) - sFuzzyTired(2, 13)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 13), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 13)
End If
If dSimTime > dFuzzyTired(1, 14) And dSimTime <= dFuzzyTired(1, 15) Then
    x = DateDiff("s", dFuzzyTired(1, 14), dFuzzyTired(1, 15))
    y = sFuzzyTired(2, 15) - sFuzzyTired(2, 15)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 14), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 14)
End If
If dSimTime > dFuzzyTired(1, 15) And dSimTime <= dFuzzyTired(1, 16) Then
    x = DateDiff("s", dFuzzyTired(1, 15), dFuzzyTired(1, 16))
    y = sFuzzyTired(2, 16) - sFuzzyTired(2, 15)
    m = y / x
    sTimeDiff = DateDiff("s", dFuzzyTired(1, 15), dSimTime)
    sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 15)
End If
```

```
If dSimTime > dFuzzyTired(1, 16) And dSimTime <= dFuzzyTired(1, 17) Then
   x = DateDiff("s", dFuzzyTired(1, 16), dFuzzyTired(1, 17))
   y = sFuzzyTired(2, 17) - sFuzzyTired(2, 16)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 16), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 16)
End If
If dSimTime > dFuzzyTired(1, 17) And dSimTime <= dFuzzyTired(1, 18) Then
   x = DateDiff("s", dFuzzyTired(1, 17), dFuzzyTired(1, 18))
   y = sFuzzyTired(2, 18) - sFuzzyTired(2, 17)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 17), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 17)
End If
If dSimTime > dFuzzyTired(1, 18) And dSimTime <= dFuzzyTired(1, 19) Then
   x = DateDiff("s", dFuzzyTired(1, 18), dFuzzyTired(1, 19))
   y = sFuzzyTired(2, 19) - sFuzzyTired(2, 18)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 18), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 18)
End If
If dSimTime > dFuzzyTired(1, 19) And dSimTime <= dFuzzyTired(1, 20) Then
   x = DateDiff("s", dFuzzyTired(1, 19), dFuzzyTired(1, 20))
   y = sFuzzyTired(2, 20) - sFuzzyTired(2, 19)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 19), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 19)
End If
If dSimTime > dFuzzyTired(1, 20) And dSimTime <= dFuzzyTired(1, 21) Then
   x = DateDiff("s", dFuzzyTired(1, 20), dFuzzyTired(1, 21))
   y = sFuzzyTired(2, 21) - sFuzzyTired(2, 20)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 20), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 20)
End If
If dSimTime > dFuzzyTired(1, 21) And dSimTime <= dFuzzyTired(1, 22) Then
   x = DateDiff("s", dFuzzyTired(1, 21), dFuzzyTired(1, 22))
   y = sFuzzyTired(2, 22) - sFuzzyTired(2, 21)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 21), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 21)
End If
If dSimTime > dFuzzyTired(1, 22) And dSimTime <= dFuzzyTired(1, 23) Then
   x = DateDiff("s", dFuzzyTired(1, 22), dFuzzyTired(1, 23))
   y = sFuzzyTired(2, 23) - sFuzzyTired(2, 22)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 22), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 22)
End If
If dSimTime > dFuzzyTired(1, 23) And dSimTime <= dFuzzyTired(1, 24) Then
   x = DateDiff("s", dFuzzyTired(1, 23), dFuzzyTired(1, 24))
   y = sFuzzyTired(2, 24) - sFuzzyTired(2, 23)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 23), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 23)
End If
If dSimTime > dFuzzyTired(1, 24) And dSimTime <= TimeSerial(23, 59, 59) Then
   x = DateDiff("s", dFuzzyTired(1, 25), dFuzzyTired(1, 24))
   y = sFuzzyTired(2, 25) - sFuzzyTired(2, 24)
   m = y / x
   sTimeDiff = DateDiff("s", dFuzzyTired(1, 24), dSimTime)
   sMuiFuzzyTired = m * sTimeDiff + sFuzzyTired(2, 24)
End If
'End Fuzzyfication Fuzzy Tired.
'
'--------------------------------------------------------------------------------
'
'Error trap:
If (sMuiFuzzy1Shift < 0 Or sMuiFuzzy2Shift < 0 Or sMuiFuzzy3Shift < 0 Or sMuiFuzzyBreak _
```

```
        < 0 Or sMuiFuzzyTired < 0) Then
        Stop 'Error trap.  None of these parameters should be negative.
        End
    End If
    If (sMuiFuzzy1Shift > 1 Or sMuiFuzzy2Shift > 1 Or sMuiFuzzy3Shift > 1 Or sMuiFuzzyBreak _
        > 1 Or sMuiFuzzyTired > 1) Then
        Stop 'Error trap.  None of these parameters should be greater than unity.
        End
    End If
    '
NoUncertainty:
    sProgress = sProgress + 1 * sIncrement
    Call Progress_Indicator(sProgress)
    '
End Sub
'
'
'
Sub WFI_Time_Check(iCount, iCWIndex, LSecondsOfDaysCount)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine ensures that the time allocation to WFIStart                          @
    '@ and WFI End is correct for entire day.  Without this code, the program may         @
    '@ not find all WFIStart and WFIEnd.  This procedure may introduce an error           @
    '@ of +/- one second.                                                                             @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' iCount = Index of current time array; see calling subroutine [-].
    ' iCWIndex = Index of current time array; see calling subroutine [-].
    ' LSecondsOfDaysCount = Simulated time; see calling subroutine [date].
    ' iWFIPriority = Global variable [-].
    ' dWFIStart(Array) = Global variable [-].
    ' dWFIEnd(Array) = Global variable [-].
    ' LOneDayInSeconds = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' iIntermediate = for time calculation [-].
    ' dTime = For time calculation [date].
    '
    'CALCULATED VALUES/OUTPUTS
    ' dWFIStart(Array) = Global variable [24 hours].
    ' dWFIEnd(Array) = Global variable [24 hours].
    '
    Dim dTime As Date
    Dim iIntermediate As Integer
    Dim LCount As Long
    '
    LCount = LSecondsOfDaysCount
    Do While LCount < LOneDayInSeconds + 1
        Select Case LCount
        Case 1 To 32400
            dTime = TimeSerial(0, 0, LCount)
        Case 32401 To 64800
            iIntermediate = LCount - 32400
            dTime = TimeSerial(9, 0, iIntermediate)
        Case 64801 To 86400
            iIntermediate = LCount - 64800
            dTime = TimeSerial(18, 0, iIntermediate)
        Case Else
            Stop            'Error Trap for LCount.
            End
        End Select
        If dTime > dWFIStart(iCount, iCWIndex) Then
            dWFIStart(iCount, iCWIndex) = dTime
            Exit Do
        End If
        LCount = LCount + 1
    Loop
```

```
    '
    LCount = LSecondsOfDaysCount
    Do While LCount < LOneDayInSeconds + 1
        Select Case LCount
        Case 1 To 32400
            dTime = TimeSerial(0, 0, LCount)
        Case 32401 To 64800
            iIntermediate = LCount - 32400
            dTime = TimeSerial(9, 0, iIntermediate)
        Case 64801 To 86400
            iIntermediate = LCount - 64800
            dTime = TimeSerial(18, 0, iIntermediate)
        Case Else
            Stop            'Error Trap for LCount.
            End
        End Select
        If dTime > dWFIEnd(iCount, iCWIndex) Then
            dWFIEnd(iCount, iCWIndex) = dTime
            Exit Do
        End If
        LCount = LCount + 1
    Loop
    '
End Sub
'
'
'
Sub DI_Time_Check(iCount, iCWIndex, LSecondsOfDaysCount)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine ensures that the time allocation to DIStart                 @
    '@ and DI End is correct for entire day.  Without this code, the program may   @
    '@ not find all DIStart and DIEnd.  This procedure may introduce an error      @
    '@ of +/- one second.                                                          @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' iCount = Index of current time array; see calling subroutine [-].
    ' iCWIndex = Index of current time array; see calling subroutine [-].
    ' LSecondsOfDaysCount = Simulated time; see calling subroutine [date].
    ' iDIPriority = Global variable [-].
    ' dDIStart(Array) = Global variable [-].
    ' dDIEnd(Array) = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' iIntermediate = for time calculation [-].
    ' LCount = Index for Do While-Loop [-].
    ' dTime = For time calculation [date].
    '
    'CALCULATED VALUES/OUTPUTS
    ' dDIStart(Array) = Global variable [24 hours].
    ' dDIEnd(Array) = Global variable [24 hours].
    '
    Dim dTime As Date
    Dim iIntermediate As Integer
    Dim LCount As Long
    '
    LCount = LSecondsOfDaysCount
    Do While LCount < LOneDayInSeconds + 1
        Select Case LCount
        Case 1 To 32400
            dTime = TimeSerial(0, 0, LCount)
        Case 32401 To 64800
            iIntermediate = LCount - 32400
            dTime = TimeSerial(9, 0, iIntermediate)
        Case 64801 To 86400
            iIntermediate = LCount - 64800
            dTime = TimeSerial(18, 0, iIntermediate)
        Case Else
```

```
            Stop          'Error Trap for LCount.
            End
        End Select
        If dTime > dDIStart(iCount, iCWIndex) Then
            dDIStart(iCount, iCWIndex) = dTime
            Exit Do
        End If
        LCount = LCount + 1
    Loop
    '
    LCount = LSecondsOfDaysCount
    Do While LCount < LOneDayInSeconds + 1
        Select Case LCount
        Case 1 To 32400
            dTime = TimeSerial(0, 0, LCount)
        Case 32401 To 64800
            iIntermediate = LCount - 32400
            dTime = TimeSerial(9, 0, iIntermediate)
        Case 64801 To 86400
            iIntermediate = LCount - 64800
            dTime = TimeSerial(18, 0, iIntermediate)
        Case Else
            Stop          'Error Trap for LCount.
            End
        End Select
        If dTime > dDIEnd(iCount, iCWIndex) Then
            dDIEnd(iCount, iCWIndex) = dTime
            Exit Do
        End If
        LCount = LCount + 1
    Loop
    '
End Sub
'
'
'
Sub Rule_Base_WFI(iCount, iCWIndex, dSimTime)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine contains the rule-base for the entire WFI simulation.        @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sMuiFuzzy1Shift = Global variable [-].
    ' sMuiFuzzy2Shift = Global variable [-].
    ' sMuiFuzzy3Shift = Global variable [-].
    ' sMuiFuzzyBreak = Global variable [-].
    ' sMuiFuzzyTired = Global variable [-].
    ' dWFITimeFuzzy2(Array) = Global variable [-].
    ' dWFITimeFuzzy3(Array) = Global variable [-].
    ' dFuzzy1ShiftBreak(Array) = Global variable [-].
    ' iCount = Index of current time array; see calling subroutine [-].
    ' iCWIndex = Index of current time array; see calling subroutine [-].
    ' dSimTime = Simulated time; see calling subroutine [date].
    '
    'INTERMEDIATE VARIABLES
    ' iSearch = Index for Do While Loop [-].
    ' dWillItBeInABreak = Intermediate variable [-].
    ' STemp = Intermediate variable [-].
    ' sMax1 = Intermediate variable [-].
    ' sMax2 = Intermediate variable [-].
    ' sFuzzyOutput1 = Intermediate variable for calculation of sFuzzy1 [-].
    ' sFuzzyOutput2 = Intermediate variable for calculation of sFuzzy2 [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' Rule1Activated = Was Rule 1 activated or not ? [-].
    ' Rule4Activated = Was Rule 4 activated or not ? [-].
    ' Rule5Activated = Was Rule 5 activated or not ? [-].
    '
```

```
Dim iSearch As Integer
Dim sFuzzyDelayShift As Single, sFuzzyOutput1() As Single, sFuzzyOutput2() As Single
Dim sTemp As Single, sMax1 As Single, sMax2 As Single
Dim sFuzzy1 As Single, sFuzzy2 As Single
Dim dWillItBeInABreak As Date
'
ReDim sFuzzyOutput1(10, 10), sFuzzyOutput2(10, 10)
'
Rule1Activated = "No"
Rule4Activated = "No"
Rule5Activated = "No"
'
'-----------------------------------------------------------------------------------
'
'Rule 1: If a break is about to begin and priority of task is less than max. (less than 3)
'and operator cannot finish the task before (core break = 1) break, he will delay batch
'to after break.
'Operator efficiency for first task off the break is at max efficiency + 10% delay.
'
If iWFIPriority(iCount, iCWIndex) < 3 Then
   'Check if Wait + operation setup may endup in a break:
   dWillItBeInABreak = dSimTime + dWFITimeFuzzy4(iCount, iCWIndex)
   '
   'If blocks below checks if operator might have to work during a break.
   If dFuzzy1ShiftBreak(1, 2) <= dWillItBeInABreak And dWillItBeInABreak _
   <= dFuzzy1ShiftBreak(1, 3) Then
      Wait(iCount, iCWIndex) = "Operating"
      sFuzzyOutput1(1, 1) = 10
      sFuzzyOutput1(1, 2) = 1
      Rule1Activated = "Yes"
      Rule1DelayAfterBreak = dFuzzy1ShiftBreak(1, 3)
   End If
   '
   If dFuzzy1ShiftBreak(1, 6) <= dWillItBeInABreak _
   And dWillItBeInABreak <= dFuzzy1ShiftBreak(1, 7) Then
      Wait(iCount, iCWIndex) = "Operating"
      sFuzzyOutput1(1, 1) = 10
      sFuzzyOutput1(1, 2) = 1
      Rule1Activated = "Yes"
      Rule1DelayAfterBreak = dFuzzy1ShiftBreak(1, 7)
   End If
   '
   If dFuzzy2ShiftBreak(1, 2) <= dWillItBeInABreak _
   And dWillItBeInABreak <= dFuzzy2ShiftBreak(1, 3) Then
      Wait(iCount, iCWIndex) = "Operating"
      sFuzzyOutput1(1, 1) = 10
      sFuzzyOutput1(1, 2) = 1
      Rule1Activated = "Yes"
      Rule1DelayAfterBreak = dFuzzy2ShiftBreak(1, 3)
   End If
   '
   If dFuzzy2ShiftBreak(1, 6) <= dWillItBeInABreak _
   And dWillItBeInABreak <= dFuzzy2ShiftBreak(1, 7) Then
      Wait(iCount, iCWIndex) = "Operating"
      sFuzzyOutput1(1, 1) = 10
      sFuzzyOutput1(1, 2) = 1
      Rule1Activated = "Yes"
      Rule1DelayAfterBreak = dFuzzy2ShiftBreak(1, 7)
   End If
   '
   If dFuzzy3ShiftBreak(1, 2) <= dWillItBeInABreak _
   And dWillItBeInABreak <= dFuzzy3ShiftBreak(1, 3) Then
      Wait(iCount, iCWIndex) = "Operating"
      sFuzzyOutput1(1, 1) = 10
      sFuzzyOutput1(1, 2) = 1
      Rule1Activated = "Yes"
      Rule1DelayAfterBreak = dFuzzy3ShiftBreak(1, 3)
   End If
   '
```

```
    If dFuzzy3ShiftBreak(1, 6) <= dWillItBeInABreak _
    And dWillItBeInABreak <= dFuzzy3ShiftBreak(1, 7) Then
       Wait(iCount, iCWIndex) = "Operating"
       sFuzzyOutput1(1, 1) = 10
       sFuzzyOutput1(1, 2) = 1
       Rule1Activated = "Yes"
       Rule1DelayAfterBreak = dFuzzy3ShiftBreak(1, 7)
    End If
    '
    If Rule1Activated = "Yes" Then
       sFuzzyOutput2(1, 1) = 10
       GoTo RulesDone
    End If
 End If
 'End Rule 1.
 '
 '-------------------------------------------------------------------------------------
 '
 'Rule 2: Evaluate shift pattern of operator and use rules to select delay of operations.
 If (sMuiFuzzy1Shift < 1 And sMuiFuzzy2Shift < 1) _
 And (sMuiFuzzy1Shift > 0 And sMuiFuzzy2Shift > 0) Then
    '
    'Two operators are available.  Delay is assumed to be the average of both.
    sFuzzyDelayShift = CSng((sMuiFuzzy1Shift + sMuiFuzzy2Shift) / 2)
    '
    Select Case sFuzzyDelayShift
    Case 0.9 To 1    'Number between 1 and 0.9 inclusive.
       'Rule: If 0.9 < sFuzzyDelayShift < 1 then add 10 % delay from max delay to the
       'no delay figure.  Operator is beginning to get tired.
       sFuzzyOutput1(2, 1) = 10
       sFuzzyOutput1(2, 2) = 1
    Case 0.8 To 0.9
       'Rule: If 0.8 < sFuzzyDelayShift < 0.9 then add 20% delay.
       sFuzzyOutput1(2, 1) = 20
       sFuzzyOutput1(2, 2) = 1
    Case 0.6 To 0.8
       'Rule: If 0.8 < sFuzzyDelayShift < 0.6 then add 50% delay.
       sFuzzyOutput1(2, 1) = 50
       sFuzzyOutput1(2, 2) = 1
    Case 0.4 To 0.6
       'Rule: If 0.6 < sFuzzyDelayShift < 0.4 then add 70% delay.
       sFuzzyOutput1(2, 1) = 70
       sFuzzyOutput1(2, 2) = 1
    Case 0 To 0.4
       'Rule: If sFuzzyDelayShift < 0.4 then max delay.
       sFuzzyOutput1(2, 1) = 100
       sFuzzyOutput1(2, 2) = 1
    Case Else
       Stop          'Error Trap. sFuzzyDelayShift cannot be larger 1 or below 0.
       End
    End Select
 End If
 '
 If (sMuiFuzzy2Shift < 1 And sMuiFuzzy3Shift < 1) _
 And (sMuiFuzzy2Shift > 0 And sMuiFuzzy3Shift > 0) Then
    '
    sFuzzyDelayShift = CSng((sMuiFuzzy2Shift + sMuiFuzzy3Shift) / 2)
    '
    Select Case sFuzzyDelayShift
    Case 0.9 To 1    'Number between 1 and 0.9 inclusive.
       'Rule: If 0.9 < sFuzzyDelayShift < 1 then add 10 % delay.
       'delay figure.
       sFuzzyOutput1(2, 1) = 10
       sFuzzyOutput1(2, 2) = 1
    Case 0.8 To 0.9
       'Rule: If 0.8 < sFuzzyDelayShift < 0.9 then add 20% delay.
       sFuzzyOutput1(2, 1) = 20
       sFuzzyOutput1(2, 2) = 1
    Case 0.6 To 0.8
```

```
      'Rule: If 0.8 < sFuzzyDelayShift < 0.6 then add 50% delay.
      sFuzzyOutput1(2, 1) = 50
      sFuzzyOutput1(2, 2) = 1
   Case 0.4 To 0.6
      'Rule: If 0.6 < sFuzzyDelayShift < 0.4 then add 70% delay.
      sFuzzyOutput1(2, 1) = 70
      sFuzzyOutput1(2, 2) = 1
  Case 0 To 0.4
      'Rule: If sFuzzyDelayShift < 0.4 then max delay.
      sFuzzyOutput1(2, 1) = 100
      sFuzzyOutput1(2, 2) = 1
   Case Else          'Other Values; i.e.
      Stop          'Error Trap.   sFuzzyDelayShift cannot be larger 1 or below 0.
      End
   End Select
End If
'
If (sMuiFuzzy1Shift < 1 And sMuiFuzzy3Shift < 1) _
And (sMuiFuzzy1Shift > 0 And sMuiFuzzy3Shift > 0) Then
   '
   sFuzzyDelayShift = (sMuiFuzzy3Shift + sMuiFuzzy1Shift) / 2
   '
   Select Case sFuzzyDelayShift
   Case 0.9 To 1    'Number between 1 and 0.9 inclusive.
      'Rule: If 0.9 < sFuzzyDelayShift < 1 then add 10 % delay.
      'delay figure.
      sFuzzyOutput1(2, 1) = 10
      sFuzzyOutput1(2, 2) = 1
   Case 0.8 To 0.9
      'Rule: If 0.8 < sFuzzyDelayShift < 0.9 then add 20% delay.
      sFuzzyOutput1(2, 1) = 20
      sFuzzyOutput1(2, 2) = 1
   Case 0.6 To 0.8
      'Rule: If 0.8 < sFuzzyDelayShift < 0.6 then add 50% delay.
      sFuzzyOutput1(2, 1) = 50
      sFuzzyOutput1(2, 2) = 1
   Case 0.4 To 0.6
      'Rule: If 0.6 < sFuzzyDelayShift < 0.4 then add 70% delay.
      sFuzzyOutput1(2, 1) = 70
      sFuzzyOutput1(2, 2) = 1
   Case 0 To 0.4
      'Rule: If sFuzzyDelayShift < 0.4 then add max delay.
      sFuzzyOutput1(2, 1) = 100
      sFuzzyOutput1(2, 2) = 1
   Case Else
      Stop          'Error Trap. sFuzzyDelayShift cannot be larger 1 or below 0.
      End
   End Select
End If
'End of Rule 2.
'
'-------------------------------------------------------------------------------------
'
'Rule 3: Evaluate how tired the operator is and use rules to select delay of operations
'due to tiredness:
Select Case sMuiFuzzyTired
Case 0 To 0.15 'Number between 0 and 0.15 inclusive.
   'Rule: Operator is not tired.  No delay.
   sFuzzyOutput1(3, 1) = 0
   sFuzzyOutput1(3, 2) = 1
Case 0.15 To 0.3
   'Rule: If 0.15 < FuzzyTired < 0.3 then add 5 % delay.
   sFuzzyOutput1(3, 1) = 0
   sFuzzyOutput1(3, 2) = 1
   sFuzzyOutput2(3, 1) = 10
   sFuzzyOutput2(3, 2) = 1
Case 0.3 To 0.35
   'Rule: If 0.3 < FuzzyTired< 0.35 then add 10% delay.
   sFuzzyOutput1(3, 1) = 10
```

```
      sFuzzyOutput1(3, 2) = 1
   Case 0.35 To 0.4
      'Rule: If 0.35 < FuzzyTired < 0.4 then add 15% delay.
      sFuzzyOutput1(3, 1) = 10
      sFuzzyOutput1(3, 2) = 1
      sFuzzyOutput2(3, 1) = 20
      sFuzzyOutput2(3, 2) = 1
   Case 0.4 To 0.5
      'Rule: If 0.4 < FuzzyTired < 0.5 then add 20% delay.
      sFuzzyOutput1(3, 1) = 20
      sFuzzyOutput1(3, 2) = 1
   Case 0.5 To 0.6
      'Rule: If 0.5 < FuzzyTired < 0.6 then add 25% delay.
      sFuzzyOutput1(3, 1) = 20
      sFuzzyOutput1(3, 2) = 1
      sFuzzyOutput2(3, 1) = 30
      sFuzzyOutput2(3, 2) = 1
   Case 0.6 To 0.7
      'Rule: If 0.6 < FuzzyTired < 0.7 then add 30% delay.
      sFuzzyOutput1(3, 1) = 30
      sFuzzyOutput1(3, 2) = 1
   Case 0.7 To 1
      'Rule: If 0.7 < FuzzyTired < 1 then 35% delay.
      sFuzzyOutput1(3, 1) = 30
      sFuzzyOutput1(3, 2) = 1
      sFuzzyOutput2(3, 1) = 40
      sFuzzyOutput2(3, 2) = 1
   Case Else
      Stop          'Error Trap.  sMuiFuzzyTired cannot be larger 1 or below 0.
      End
   End Select
   'End Rule3.
   '
   '-------------------------------------------------------------------------------------
   '
   'Rule 4: Evaluate case that operator is on a break, while process becomes available.
   'If operator comes off a break his efficiency is down by 20% for the first task.
   If sMuiFuzzyBreak = 1 Then
      sFuzzyOutput1(4, 1) = 20
      sFuzzyOutput1(4, 2) = 1
      Rule4Activated = "Yes"
   End If
   'End Rule 4.
   '
   '-------------------------------------------------------------------------------------
   '
   'Rule 5: If sMuiFuzzyBreak > 0.6 and Priority = 3 (no delay allowed) operator will delay
   'batch to after break.  Only valid for time before core break, not after it
   '(hence "And Not", which is the time after the break.)
   'Operator effiency is at maximum as he feels guilty having deliberately delayed the batch.
   If sMuiFuzzyBreak > 0.6 And sMuiFuzzyBreak < 1 And iWFIPriority(iCount, iCWIndex) = 3 _
      And Not ((dSimTime > dFuzzy1ShiftBreak(1, 2) And dSimTime < dFuzzy1ShiftBreak(1, 4)) _
      Or (dSimTime > dFuzzy1ShiftBreak(1, 6) And dSimTime < dFuzzy1ShiftBreak(1, 8)) _
      Or (dSimTime > dFuzzy2ShiftBreak(1, 2) And dSimTime < dFuzzy2ShiftBreak(1, 4)) _
      Or (dSimTime > dFuzzy2ShiftBreak(1, 6) And dSimTime < dFuzzy2ShiftBreak(1, 8)) _
      Or (dSimTime > dFuzzy3ShiftBreak(1, 2) And dSimTime < dFuzzy3ShiftBreak(1, 4)) _
      Or (dSimTime > dFuzzy3ShiftBreak(1, 6) And dSimTime < dFuzzy3ShiftBreak(1, 8))) Then
      '
      sFuzzyOutput1(5, 1) = 100
      sFuzzyOutput1(5, 2) = 1
      Rule5Activated = "Yes"
   End If
   'End Rule 5.
   '
RulesDone:
   '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   'Aggregation:
   ' Aggregation rule is to only take the maximum delay and discard the rest.
   ' For example: sFuzzyOutput1(1, 1) = 10 will be discarded if there is a
```

```
    ' sFuzzyOutput1(5, 1) = 50.
    '
    'Find the maximum sFuzzyOutput1:
    sMax1 = sFuzzyOutput1(1, 1)
    iSearch = 2
    Do While iSearch < 5 + 1
       sTemp = sFuzzyOutput1(iSearch, 1)
       If sTemp > sMax1 Then
          sMax1 = sTemp
       End If
       iSearch = iSearch + 1
    Loop
    '
    'Find the maximum sFuzzyOutput2:
    sMax2 = sFuzzyOutput2(1, 1)
    iSearch = 2
    Do While iSearch < 5 + 1
       sTemp = sFuzzyOutput2(iSearch, 1)
       If sTemp > sMax2 Then
          sMax2 = sTemp
       End If
       iSearch = iSearch + 1
    Loop
    '
    'Take only the larger of the two figures from sMax1 or sMax2 if they the two figures are
    'not more apart than one membership function. For example sMax1=10 and sMax2=20 will
    'be aggregated and the defuzzification will be 15, but sMax1=10 and sMax2=30 will not
    'and only the 30 figure will be defuzzified to 30.
    If sMax1 > sMax2 Then
       If (sMax1 - 10 = sMax2) Then
          sFuzzy1 = sMax2
          sFuzzy2 = sMax1
       Else
          sFuzzy1 = sMax1
          sFuzzy2 = 666
       End If
    Else
       If (sMax2 - 10 = sMax1) Then
          sFuzzy1 = sMax2
          sFuzzy2 = sMax1
       Else
          sFuzzy1 = sMax2
          sFuzzy2 = 666
       End If
    End If
    'End Aggregation.
    '
    Call Defuzzifier_WFI(sFuzzy1, sFuzzy2)
    '
End Sub
'
'
'
```

```
Sub Defuzzifier_WFI(sFuzzy1, sFuzzy2)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine contains the WFI Defuzzifier. @
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS OF SUBROUTINE
    ' sFuzzy1 = Global variable [-].
    ' sFuzzy2 = Global variable [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sDefuzzifier = Global variable [24 hours].
    '
    'Defuzzifier:
    If sFuzzy1 = 666 Or sFuzzy2 = 666 Then
        'Discard 666 and take the other figure.
        If sFuzzy1 = 666 Then
            sDefuzzifier = sFuzzy2
        Else
            sDefuzzifier = sFuzzy1
        End If
    Else
        'Take the average:
        sDefuzzifier = (sFuzzy1 + sFuzzy2) / 2
    End If
    '
End Sub
'
'
'
Sub Defuzzifier_DI(sFuzzy1, sFuzzy2)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine contains the DI Defuzzifier.      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS OF SUBROUTINE
    ' sFuzzy1 = Global variable [-].
    ' sFuzzy2 = Global variable [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sDefuzzifier = Global variable [24 hours].
    '
    'Defuzzifier:
    If sFuzzy1 = 666 Or sFuzzy2 = 666 Then
        'Discard 666 and take the other figure.
        If sFuzzy1 = 666 Then
            sDefuzzifier = sFuzzy2
        Else
            sDefuzzifier = sFuzzy1
        End If
    Else
        'Take the average:
        sDefuzzifier = (sFuzzy1 + sFuzzy2) / 2
    End If
    '
End Sub
'
'
```

```
'
Sub Rule_Base_DI(iCount, iCWIndex, dSimTime)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine contains the rule-base for the entire DI simulation.      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sMuiFuzzy1Shift = Global variable [-].
    ' sMuiFuzzy2Shift = Global variable [-].
    ' sMuiFuzzy3Shift = Global variable [-].
    ' sMuiFuzzyBreak = Global variable [-].
    ' sMuiFuzzyTired = Global variable [-].
    ' dDITimeFuzzy2(Array) = Global variable [-].
    ' dDITimeFuzzy3(Array) = Global variable [-].
    ' dFuzzy1ShiftBreak(Array) = Global variable [-].
    ' iCount = Index of current time array; see calling subroutine [-].
    ' iCWIndex = Index of current time array; see calling subroutine [-].
    ' dSimTime = Simulated time; see calling subroutine [date].
    '
    'INTERMEDIATE VARIABLES
    ' iSearch = Index for Do While Loop [-].
    ' dWillItBeInABreak = Intermediate variable [-].
    ' STemp = Intermediate variable [-].
    ' sMax1 = Intermediate variable [-].
    ' sMax2 = Intermediate variable [-].
    ' sFuzzyOutput1 = Intermediate variable for calculation of sFuzzy1 [-].
    ' sFuzzyOutput2 = Intermediate variable for calculation of sFuzzy2 [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' Rule1Activated = Was Rule 1 activated or not ? [-].
    ' Rule4Activated = Was Rule 4 activated or not ? [-].
    ' Rule5Activated = Was Rule 5 activated or not ? [-].
    '
    Dim iSearch As Integer
    Dim sFuzzyDelayShift As Single, sFuzzyOutput1() As Single, sFuzzyOutput2() As Single
    Dim sTemp As Single, sMax1 As Single, sMax2 As Single
    Dim sFuzzy1 As Single, sFuzzy2 As Single
    Dim dWillItBeInABreak As Date
    '
    ReDim sFuzzyOutput1(10, 10), sFuzzyOutput2(10, 10)
    '
    Rule1Activated = "No"
    Rule4Activated = "No"
    Rule5Activated = "No"
    '
    '-------------------------------------------------------------------------------------
    '
    'Rule 1: If a break is about to begin and priority of task is less than max. (less than 3)
    'and operator cannot finish the task before (core break = 1) break, he will delay batch
    'to after break.
    'Operator efficiency for first task off the break is at max efficiency + 10% delay.
    '
    If iDIPriority(iCount, iCWIndex) < 3 Then
        'Check if Wait + operation setup may endup in a break:
        dWillItBeInABreak = dSimTime + dDITimeFuzzy3(iCount, iCWIndex)        '
        'If blocks below checks if operator might have to work during a break.
        If dFuzzy1ShiftBreak(1, 2) <= dWillItBeInABreak And dWillItBeInABreak _
        <= dFuzzy1ShiftBreak(1, 3) Then
            Wait(iCount, iCWIndex) = "Operating"
            sFuzzyOutput1(1, 1) = 10
            sFuzzyOutput1(1, 2) = 1
            Rule1Activated = "Yes"
            Rule1DelayAfterBreak = dFuzzy1ShiftBreak(1, 3)
        End If
        '
        If dFuzzy1ShiftBreak(1, 6) <= dWillItBeInABreak _
        And dWillItBeInABreak <= dFuzzy1ShiftBreak(1, 7) Then
            Wait(iCount, iCWIndex) = "Operating"
```

```
        sFuzzyOutput1(1, 1) = 10
        sFuzzyOutput1(1, 2) = 1
        Rule1Activated = "Yes"
        Rule1DelayAfterBreak = dFuzzy1ShiftBreak(1, 7)
    End If
    '
    If dFuzzy2ShiftBreak(1, 2) <= dWillItBeInABreak _
    And dWillItBeInABreak <= dFuzzy2ShiftBreak(1, 3) Then
        Wait(iCount, iCWIndex) = "Operating"
        sFuzzyOutput1(1, 1) = 10
        sFuzzyOutput1(1, 2) = 1
        Rule1Activated = "Yes"
        Rule1DelayAfterBreak = dFuzzy2ShiftBreak(1, 3)
    End If
    '
    If dFuzzy2ShiftBreak(1, 6) <= dWillItBeInABreak _
    And dWillItBeInABreak <= dFuzzy2ShiftBreak(1, 7) Then
        Wait(iCount, iCWIndex) = "Operating"
        sFuzzyOutput1(1, 1) = 10
        sFuzzyOutput1(1, 2) = 1
        Rule1Activated = "Yes"
        Rule1DelayAfterBreak = dFuzzy2ShiftBreak(1, 7)
    End If
    '
    If dFuzzy3ShiftBreak(1, 2) <= dWillItBeInABreak _
    And dWillItBeInABreak <= dFuzzy3ShiftBreak(1, 3) Then
        Wait(iCount, iCWIndex) = "Operating"
        sFuzzyOutput1(1, 1) = 10
        sFuzzyOutput1(1, 2) = 1
        Rule1Activated = "Yes"
        Rule1DelayAfterBreak = dFuzzy3ShiftBreak(1, 3)
    End If
    '
    If dFuzzy3ShiftBreak(1, 6) <= dWillItBeInABreak _
    And dWillItBeInABreak <= dFuzzy3ShiftBreak(1, 7) Then
        Wait(iCount, iCWIndex) = "Operating"
        sFuzzyOutput1(1, 1) = 10
        sFuzzyOutput1(1, 2) = 1
        Rule1Activated = "Yes"
        Rule1DelayAfterBreak = dFuzzy3ShiftBreak(1, 7)
    End If
    '
    If Rule1Activated = "Yes" Then
        sFuzzyOutput2(1, 1) = 10
        GoTo RulesDone
    End If
End If
'End Rule 1.
'
'-------------------------------------------------------------------------------
'
'Rule 2: Evaluate shift pattern of operator and use rules to select delay of operations.
If (sMuiFuzzy1Shift < 1 And sMuiFuzzy2Shift < 1) _
And (sMuiFuzzy1Shift > 0 And sMuiFuzzy2Shift > 0) Then
    '
    'Two operators are available.  Delay is assumed to be the average of both.
    sFuzzyDelayShift = CSng((sMuiFuzzy1Shift + sMuiFuzzy2Shift) / 2)
    '
    Select Case sFuzzyDelayShift
    Case 0.9 To 1    'Number between 1 and 0.9 inclusive.
        'Rule: If 0.9 < sFuzzyDelayShift < 1 then add 10 % delay from max delay to the
        'no delay figure.  Operator is beginning to get tired.
        sFuzzyOutput1(2, 1) = 10
        sFuzzyOutput1(2, 2) = 1
    Case 0.8 To 0.9
        'Rule: If 0.8 < sFuzzyDelayShift < 0.9 then add 20% delay.
        sFuzzyOutput1(2, 1) = 20
        sFuzzyOutput1(2, 2) = 1
    Case 0.6 To 0.8
```

```
      'Rule: If 0.8 < sFuzzyDelayShift < 0.6 then add 50% delay.
      sFuzzyOutput1(2, 1) = 50
      sFuzzyOutput1(2, 2) = 1
   Case 0.4 To 0.6
      'Rule: If 0.6 < sFuzzyDelayShift < 0.4 then add 70% delay.
      sFuzzyOutput1(2, 1) = 70
      sFuzzyOutput1(2, 2) = 1
   Case 0 To 0.4
      'Rule: If sFuzzyDelayShift < 0.4 then max delay.
      sFuzzyOutput1(2, 1) = 100
      sFuzzyOutput1(2, 2) = 1
   Case Else
      Stop          'Error Trap. sFuzzyDelayShift cannot be larger 1 or below 0.
      End
   End Select
End If
'
If (sMuiFuzzy2Shift < 1 And sMuiFuzzy3Shift < 1) _
And (sMuiFuzzy2Shift > 0 And sMuiFuzzy3Shift > 0) Then
   '
   sFuzzyDelayShift = CSng((sMuiFuzzy2Shift + sMuiFuzzy3Shift) / 2)
   '
   Select Case sFuzzyDelayShift
   Case 0.9 To 1    'Number between 1 and 0.9 inclusive.
      'Rule: If 0.9 < sFuzzyDelayShift < 1 then add 10 % delay.
      'delay figure.
      sFuzzyOutput1(2, 1) = 10
      sFuzzyOutput1(2, 2) = 1
   Case 0.8 To 0.9
      'Rule: If 0.8 < sFuzzyDelayShift < 0.9 then add 20% delay.
      sFuzzyOutput1(2, 1) = 20
      sFuzzyOutput1(2, 2) = 1
   Case 0.6 To 0.8
      'Rule: If 0.8 < sFuzzyDelayShift < 0.6 then add 50% delay.
      sFuzzyOutput1(2, 1) = 50
      sFuzzyOutput1(2, 2) = 1
   Case 0.4 To 0.6
      'Rule: If 0.6 < sFuzzyDelayShift < 0.4 then add 70% delay.
      sFuzzyOutput1(2, 1) = 70
      sFuzzyOutput1(2, 2) = 1
  Case 0 To 0.4
      'Rule: If sFuzzyDelayShift < 0.4 then max delay.
      sFuzzyOutput1(2, 1) = 100
      sFuzzyOutput1(2, 2) = 1
   Case Else          'Other Values; i.e.
      Stop          'Error Trap.   sFuzzyDelayShift cannot be larger 1 or below 0.
      End
   End Select
End If
'
If (sMuiFuzzy1Shift < 1 And sMuiFuzzy3Shift < 1) _
And (sMuiFuzzy1Shift > 0 And sMuiFuzzy3Shift > 0) Then
   '
   sFuzzyDelayShift = (sMuiFuzzy3Shift + sMuiFuzzy1Shift) / 2
   '
   Select Case sFuzzyDelayShift
   Case 0.9 To 1    'Number between 1 and 0.9 inclusive.
      'Rule: If 0.9 < sFuzzyDelayShift < 1 then add 10 % delay.
      'delay figure.
      sFuzzyOutput1(2, 1) = 10
      sFuzzyOutput1(2, 2) = 1
   Case 0.8 To 0.9
      'Rule: If 0.8 < sFuzzyDelayShift < 0.9 then add 20% delay.
      sFuzzyOutput1(2, 1) = 20
      sFuzzyOutput1(2, 2) = 1
   Case 0.6 To 0.8
      'Rule: If 0.8 < sFuzzyDelayShift < 0.6 then add 50% delay.
      sFuzzyOutput1(2, 1) = 50
      sFuzzyOutput1(2, 2) = 1
```

```
    Case 0.4 To 0.6
       'Rule: If 0.6 < sFuzzyDelayShift < 0.4 then add 70% delay.
       sFuzzyOutput1(2, 1) = 70
       sFuzzyOutput1(2, 2) = 1
    Case 0 To 0.4
       'Rule: If sFuzzyDelayShift < 0.4 then add max delay.
       sFuzzyOutput1(2, 1) = 100
       sFuzzyOutput1(2, 2) = 1
    Case Else
       Stop          'Error Trap. sFuzzyDelayShift cannot be larger 1 or below 0.
       End
    End Select
End If
'End of Rule 2.
'
'-------------------------------------------------------------------------------------
'
'Rule 3: Evaluate how tired the operator is and use rules to select delay of operations
'due to tiredness:
Select Case sMuiFuzzyTired
Case 0 To 0.15 'Number between 0 and 0.15 inclusive.
    'Rule: Operator is not tired.  No delay.
    sFuzzyOutput1(3, 1) = 0
    sFuzzyOutput1(3, 2) = 1
Case 0.15 To 0.3
    'Rule: If 0.15 < FuzzyTired < 0.3 then add 5 % delay.
    sFuzzyOutput1(3, 1) = 0
    sFuzzyOutput1(3, 2) = 1
    sFuzzyOutput2(3, 1) = 10
    sFuzzyOutput2(3, 2) = 1
Case 0.3 To 0.35
    'Rule: If 0.3 < FuzzyTired< 0.35 then add 10% delay.
    sFuzzyOutput1(3, 1) = 10
    sFuzzyOutput1(3, 2) = 1
Case 0.35 To 0.4
    'Rule: If 0.35 < FuzzyTired < 0.4 then add 15% delay.
    sFuzzyOutput1(3, 1) = 10
    sFuzzyOutput1(3, 2) = 1
    sFuzzyOutput2(3, 1) = 20
    sFuzzyOutput2(3, 2) = 1
Case 0.4 To 0.5
    'Rule: If 0.4 < FuzzyTired < 0.5 then add 20% delay.
    sFuzzyOutput1(3, 1) = 20
    sFuzzyOutput1(3, 2) = 1
Case 0.5 To 0.6
    'Rule: If 0.5 < FuzzyTired < 0.6 then add 25% delay.
    sFuzzyOutput1(3, 1) = 20
    sFuzzyOutput1(3, 2) = 1
    sFuzzyOutput2(3, 1) = 30
    sFuzzyOutput2(3, 2) = 1
Case 0.6 To 0.7
    'Rule: If 0.6 < FuzzyTired < 0.7 then add 30% delay.
    sFuzzyOutput1(3, 1) = 30
    sFuzzyOutput1(3, 2) = 1
Case 0.7 To 1
    'Rule: If 0.7 < FuzzyTired < 1 then 35% delay.
    sFuzzyOutput1(3, 1) = 30
    sFuzzyOutput1(3, 2) = 1
    sFuzzyOutput2(3, 1) = 40
    sFuzzyOutput2(3, 2) = 1
Case Else
    Stop          'Error Trap.  sMuiFuzzyTired cannot be larger 1 or below 0.
    End
End Select
'End Rule3.
'
'-------------------------------------------------------------------------------------
'
'Rule 4: Evaluate case that operator is on a break, while process becomes available.
```

```
    'If operator comes off a break his efficiency is down by 20% for the first task.
    If sMuiFuzzyBreak = 1 Then
        sFuzzyOutput1(4, 1) = 20
        sFuzzyOutput1(4, 2) = 1
        Rule4Activated = "Yes"
    End If
    'End Rule 4.
    '
    '-------------------------------------------------------------------------------------------
    '
    'Rule 5: If sMuiFuzzyBreak > 0.6 and Priority = 3 (no delay allowed) operator will delay
    'batch to after break.  Only valid for time before core break, not after it
    '(hence "And Not", which is the time after the break.)
    'Operator effiency is at maximum as he feels guilty having deliberately delayed the batch.
    If sMuiFuzzyBreak > 0.6 And sMuiFuzzyBreak < 1 And iDIPriority(iCount, iCWIndex) = 3 _
        And Not ((dSimTime > dFuzzy1ShiftBreak(1, 2) And dSimTime < dFuzzy1ShiftBreak(1, 4)) _
        Or (dSimTime > dFuzzy1ShiftBreak(1, 6) And dSimTime < dFuzzy1ShiftBreak(1, 8)) _
        Or (dSimTime > dFuzzy2ShiftBreak(1, 2) And dSimTime < dFuzzy2ShiftBreak(1, 4)) _
        Or (dSimTime > dFuzzy2ShiftBreak(1, 6) And dSimTime < dFuzzy2ShiftBreak(1, 8)) _
        Or (dSimTime > dFuzzy3ShiftBreak(1, 2) And dSimTime < dFuzzy3ShiftBreak(1, 4)) _
        Or (dSimTime > dFuzzy3ShiftBreak(1, 6) And dSimTime < dFuzzy3ShiftBreak(1, 8))) Then
        '
        sFuzzyOutput1(5, 1) = 100
        sFuzzyOutput1(5, 2) = 1
        Rule5Activated = "Yes"
    End If
    'End Rule 5.
    '
RulesDone:
    '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    'Aggregation:
    ' Aggregation rule is to only take the maximum delay and discard the rest.
    ' For example: sFuzzyOutput1(1, 1) = 10 will be discarded if there is a
    ' sFuzzyOutput1(5, 1) = 50.
    '
    'Find the maximum sFuzzyOutput1:
    sMax1 = sFuzzyOutput1(1, 1)
    iSearch = 2
    Do While iSearch < 5 + 1
        sTemp = sFuzzyOutput1(iSearch, 1)
        If sTemp > sMax1 Then
            sMax1 = sTemp
        End If
        iSearch = iSearch + 1
    Loop
    '
    'Find the maximum sFuzzyOutput2:
    sMax2 = sFuzzyOutput2(1, 1)
    iSearch = 2
    Do While iSearch < 5 + 1
        sTemp = sFuzzyOutput2(iSearch, 1)
        If sTemp > sMax2 Then
            sMax2 = sTemp
        End If
        iSearch = iSearch + 1
    Loop
    '
    'Take only the larger of the two figures from sMax1 or sMax2 if they the two figures are
    'not more apart than one membership function. For example sMax1=10 and sMax2=20 will
    'be aggregated and the defuzzification will be 15, but sMax1=10 and sMax2=30 will not
    'and only the 30 figure will be defuzzified to 30.
    If sMax1 > sMax2 Then
        If (sMax1 - 10 = sMax2) Then
            sFuzzy1 = sMax2
            sFuzzy2 = sMax1
        Else
            sFuzzy1 = sMax1
            sFuzzy2 = 666
        End If
```

```
    Else
        If (sMax2 - 10 = sMax1) Then
            sFuzzy1 = sMax2
            sFuzzy2 = sMax1
        Else
            sFuzzy1 = sMax2
            sFuzzy2 = 666
        End If
    End If
    'End Aggregation.
    '
    Call Defuzzifier_DI(sFuzzy1, sFuzzy2)
    '
End Sub
'
'
'
Sub WFI_Water_Demand_From_Loop()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the WFI water demand from the consumers along the     @
    '@ loop; ignoring any consumption that might be caused by the tank being filled     @
    '@ by WFI as the WFI Tank may not full at the beginning of the simulation.          @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterDemandPerTap = Global variable [m3/h].
    ' dWFIStart(Array) = Global variable [date].
    ' dWFIEnd(Array) = Global variable [date].
    ' iColNumberWFI = Global variable [-].
    ' iRowNumberWFI = Global variable [-].
    ' dNewTimeStart = Global variable [-].
    ' dNewTimeEnd = Global variable [-].
    ' LRepeats = Global variable [-].
    ' sIncrement = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' sWFITimeStart = [24 hour clock].
    ' sWFITimeEnd = [24 hour clock].
    ' sWFITimeEndIndex = WFI Time End Index [-].
    ' sWFITimeStartIndex = Index start time [-].
    ' sWFITimeStartIndexCounter = Index time [-].
    ' sWFITimeStartIndexHour = Index start time [h].
    ' sWFITimeEndIndexHour = Index end time [h].
    ' sWFITimeStartIndexMinute = Index start time [min].
    ' sWFITimeEndIndexMinute = Index end time [min].
    ' sWFITimeStartIndexSeconds = Index start time [s].
    ' sWFITimeEndIndexSeconds = Index end time [s].
    ' sWFIWaterConsumptionPrior = WFI water consumption [m/s].
    ' LCount = Index for Do While-Loop [-].
    ' LCwIndex = Index for Do While-Loop [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sWFIWaterConsumptionFromLoop() = Global variable [m3/s].
    ' sProgress = Global variable [-].
    '
    Dim sWFIWaterConsumptionPrior As Single
    Dim sWFITimeStart As Single, sWFITimeEnd As Single, sWFITimeEndIndex As Single
    Dim sWFITimeStartIndex As Single, sWFITimeStartIndexCounter As Single
    Dim sWFITimeStartIndexHour As Single, sWFITimeEndIndexHour As Single
    Dim sWFITimeStartIndexMinute As Single, sWFITimeEndIndexMinute As Single
    Dim sWFITimeStartIndexSeconds As Single, sWFITimeEndIndexSeconds As Single
    Dim iCWIndex As Integer, iCount As Integer
    '
    If LDayDataCounter = 1 Then
        Sheets("Frontpage - WFI Loop Input Data").Select
    Else
        Sheets("Day " & LDayDataCounter & " WFI Loop Data").Select
    End If
```

```
    '
    'Read in water offtake from tap.
    If LDayDataCounter = 1 Then     '"1" prevents reading in data more than once.
       iCount = 5
       Do While iCount < iRowNumberWFI + 5                'See WFI Input Table.
          sWFIWaterDemandPerTap(iCount - 4) = Cells(iCount, 9)
          iCount = iCount + 1
       Loop
    End If
    '
    'Calculate Water Demand for every Water Demand Time Frame.
    iCWIndex = 1
    iCount = 1
    Do While iCWIndex < iColNumberWFI + 1
       Do While iCount < iRowNumberWFI + 1
          'Establish time start & end tap opening.
          sWFITimeStartIndexHour = Hour(dWFIStart(iCount, iCWIndex)) * 60 * 60
          sWFITimeEndIndexHour = Hour(dWFIEnd(iCount, iCWIndex)) * 60 * 60
          sWFITimeStartIndexMinute = Minute(dWFIStart(iCount, iCWIndex)) * 60
          sWFITimeEndIndexMinute = Minute(dWFIEnd(iCount, iCWIndex)) * 60
          sWFITimeStartIndexSeconds = Second(dWFIStart(iCount, iCWIndex))
          sWFITimeEndIndexSeconds = Second(dWFIEnd(iCount, iCWIndex))
          '
          sWFITimeStartIndex = sWFITimeStartIndexHour + sWFITimeStartIndexMinute _
          + sWFITimeStartIndexSeconds
          sWFITimeEndIndex = sWFITimeEndIndexHour + sWFITimeEndIndexMinute _
          + sWFITimeEndIndexSeconds
          sWFITimeStartIndexCounter = sWFITimeStartIndex
          '
          Do While sWFITimeStartIndexCounter < sWFITimeEndIndex
             sWFIWaterConsumptionPrior = sWFIWaterConsumptionFromLoop( _
             sWFITimeStartIndexCounter, LDayDataCounter)
             'Consumption in m3/s; This is the water consumption from the loop.
             sWFIWaterConsumptionFromLoop(sWFITimeStartIndexCounter, LDayDataCounter) = _
             sWFIWaterConsumptionPrior + sWFIWaterDemandPerTap(iCount) / 3600
             sWFITimeStartIndexCounter = sWFITimeStartIndexCounter + 1
          Loop
          iCount = iCount + 1
       Loop
       iCount = 1
       iCWIndex = iCWIndex + 1
    Loop
    '
End Sub
'
'
'
Sub DI_Water_Demand_From_Loop()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the DI Water demand from the DI Water distribution system.   @
    '@ The routine also adds the DI consumption to the WFI generator plant as was              @
    '@ calculated in the Subroutine WFI_Demand_From_Generation_And_Tank_Volume.               @
    '@ This routine does not include any water consumption from initial filling of the DI      @
    '@ water storage tank.                                                                      @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIWaterDemandPerTap = Global variable [m3/h].
    ' sWFIWaterFromDI = Global variable [m3/h].
    ' sDIWaterDemandPerTap = Global variable [-].
    ' iColNumberDI = Global variable [-].
    ' iRowNumberDI = Global variable [-].
    ' dNewTimeStart = Global variable [-].
    ' dNewTimeEnd = Global variable [-].
    ' LRepeats = Global variable [-].
    ' iResolution = Global variable [-].
    ' sIncrement = Global variable [-].
    '
```

```
'INTERMEDIATE VARIABLES
' sDIWaterConsumptionPrior = DI water consumption [m/s].
' sDITimeStart = [24 hour clock].
' sDITimeEnd = [24 hour clock].
' sDITimeEndIndex = DI Time End Index [-].
' sDITimeStartIndex = Index start time [-].
' sDITimeStartIndexCounter = Index time [-].
' sDITimeStartIndexHour = Index start time [h].
' sDITimeEndIndexHour = Index end time [h].
' sDITimeStartIndexMinute = Index start time [min].
' sDITimeEndIndexMinute = Index end time [min].
' sDITimeStartIndexSeconds = Index start time [s].
' sDITimeEndIndexSeconds = Index end time [s].
' LCount = Index for Do While-Loop [-].
' LCwIndex = Index for Do While-Loop [-].
'
'CALCULATED VALUES/OUTPUTS
' sDIWaterConsumptionFromLoop() = Global variable [m3/s].
' sProgress = Global variable [-].
'
Dim sWaterConsumptionPrior As Single
Dim sTimeStart As Single, sTimeEnd As Single, sTimeEndIndex As Single
Dim sTimeStartIndex As Single, sTimeStartIndexCounter As Single
Dim sTimeStartIndexHour As Single, sTimeEndIndexHour As Single, sTemp As Single
Dim sTimeStartIndexMinute As Single, sTimeEndIndexMinute As Single
Dim sTimeStartIndexSeconds As Single, sTimeEndIndexSeconds As Single
Dim LCwIndex As Long, LCount As Long
Dim iTen As Integer, iI As Integer, iV As Integer
'
Sheets("Day " & LDayDataCounter & " DI Loop Data").Select
'
iRowNumberDI = Cells(13, 2)
iColNumberDI = Cells(14, 2)
'
'Minimising ColNumber - see Input Tables.
iColNumberDI = (iColNumberDI - 13) / 4 + 1
'
'Read in water offtake from tap.
LCount = 5
Do While LCount < iRowNumberDI + 5
   sDIWaterDemandPerTap(LCount - 4) = Cells(LCount, 9)
   LCount = LCount + 1
Loop
'
'Calculate Water Demand for every Water Demand Time Frame.
LCwIndex = 1
LCount = 1
Do While LCwIndex < iColNumberDI + 1
   Do While LCount < iRowNumberDI + 1
      'Establish time start & end tap opening.
      sTimeStartIndexHour = Hour(dDIStart(LCount, LCwIndex)) * 60 * 60
      sTimeEndIndexHour = Hour(dDIEnd(LCount, LCwIndex)) * 60 * 60
      sTimeStartIndexMinute = Minute(dDIStart(LCount, LCwIndex)) * 60
      sTimeEndIndexMinute = Minute(dDIEnd(LCount, LCwIndex)) * 60
      sTimeStartIndexSeconds = Second(dDIStart(LCount, LCwIndex))
      sTimeEndIndexSeconds = Second(dDIEnd(LCount, LCwIndex))
      '
      sTimeStartIndex = sTimeStartIndexHour + sTimeStartIndexMinute + _
      sTimeStartIndexSeconds
      sTimeEndIndex = sTimeEndIndexHour + sTimeEndIndexMinute + sTimeEndIndexSeconds
      sTimeStartIndexCounter = sTimeStartIndex
      '
      Do While sTimeStartIndexCounter < sTimeEndIndex
         sWaterConsumptionPrior = sDIWaterConsumptionFromLoop( _
         sTimeStartIndexCounter, LDayDataCounter)
         'Consumption in m3/s; This is the water consumption from the loop.
         sDIWaterConsumptionFromLoop(sTimeStartIndexCounter, LDayDataCounter) = _
         sWaterConsumptionPrior + sDIWaterDemandPerTap(LCount) / 3600
         sTimeStartIndexCounter = sTimeStartIndexCounter + 1
```

```
        Loop
        LCount = LCount + 1
    Loop
    LCount = 1
    LCwIndex = LCwIndex + 1
Loop
'
'Add WFI Demand to DI Demand establishing total DI Demand per second and write to
'Sheet "Sheet 16 - Water Demand DI" for graphical representation of results.
Sheets("Sheet 16 - Water Demand DI").Select
iTen = 1
LCount = 1
Do While LCount < LOneDayInSeconds + 1
    sDIWaterConsumptionFromLoop(LCount, LDayDataCounter) _
    = sDIWaterConsumptionFromLoop(LCount, LDayDataCounter) _
    + sWFIWaterFromDI(LCount, LDayDataCounter)                'WFI Demand from DI loop.
    '
    If iTen = 10 Then    'Every tenth is for EXCEL 2003.
        Sheets("Sheet 16 - Water Demand DI").Select
        Cells((LCount / 10 + 3) + 8640 * (LDayDataCounter - 1), LDayDataCounter) _
        = sDIWaterConsumptionFromLoop(LCount, LDayDataCounter)
        iTen = 0
    End If
    iTen = iTen + 1
    LCount = LCount + 1
Loop
'
End Sub
'
'
'
Sub WFI_Water_Demand_From_Generation_And_Tank_Volume()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the Water volume of the WFI in WFI Storage Tank     @
    '@ as a function of time [m3]. The routine also calculates the WFI demand              @
    '@ from filling the WFI water tank if tank was not full at the start of simulation.       @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterFromGen = Global variable  [m3/h].
    ' sWFIStartVolumeTank = Global variable [m3].
    ' sWFIMinVolumeTank = Global variable [m3].
    ' sWFIStartVolumeTank = Global variable [m3].
    ' sWFIWaterConsumption(Array) = Global variable [m3/s].
    ' sWFIVolumeTank = Global variable [m3].
    ' sIncrement = Global variable [-].
    ' LDayDataCounter = Global variable [-].
    ' sWFIGenBlowDown = Global variable [%].
    ' sWFIPumpFlowrate = Global variable [m3/h].
    '
    'INTERMEDIATE VARIABLES
    ' LCount = Index for Do While-Loop [-].
    ' iTen = Writing every tenth value of the Volume to the sheet for graph etc.
    ' iI = Index for Progress Indicator [-].
    ' sDelta = For Euler integration; step width [-].
    ' iNCount = Counter for writing results to sheet [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sWFICalcVolumeTank = Calculated WFI volume in WFI storage tank [m3].
    ' sWFIWaterFromDI(Array) = Water added from DI loop [m3/s].
    ' sWFIWaterConsumptionFromLoop(Array) = Global variable [m3/s].
    ' Messages:  The simulation predicts there will be no WFI available during an
    '                 offtake period.  The results of the simulation will be
    '                 incorrect. Please change input data on worksheet.
    ' sProgress = Global variable [-].
    '
    '
    Dim sDelta As Single, sWFICalcVolumeTank As Single
```

```
Dim iWFITankMinAlarm As Integer, iTen As Integer, iI As Integer, iNCount As Integer
Dim LCount As Long
Dim Msg As String, Style As String, Title As String, Response As String
'
Const Rho20 = 998.21    'Water density at 20 degrees Celsius [m3/kg].
Const Rho80 = 971.79    'Water density at 80 degrees Celsius [m3/kg].
'
sDelta = 1
iI = 0
If LDayDataCounter = 1 Then
   Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
   sWFIWaterFromGen = Cells(4, 6)
   sWFIVolumeTank = Cells(7, 6)
   sWFIMinVolumeTank = Cells(11, 6)
   sWFIStartVolumeTank = Cells(9, 6)
   sWFIGenBlowDown = Cells(13, 6)
   sWFIWaterFromGen = sWFIWaterFromGen / 3600                  '[m3/h] to [m3/s].
   sWFICalcVolumeTank = sWFIStartVolumeTank                     'Start condition.
   sWFIPumpFlowrate = Cells(15, 6)
Else
   Sheets("Sheet 21 - WFI Volume in Tank").Select
   sWFICalcVolumeTank = Cells(8643, LDayDataCounter - 1)       '[m3].
End If
'End read in Data.
'
LCount = 1
iTen = 1
Do While LCount < LOneDayInSeconds + 1
   'Mass Balance WFI Storage Tank:
   'Integrate Mass balance below with explicit first order Euler.
   sWFICalcVolumeTank = sWFICalcVolumeTank + (sWFIWaterFromGen - _
   sWFIWaterConsumptionFromLoop(LCount, LDayDataCounter)) * sDelta
   '
   If sWFIWaterConsumptionFromLoop(LCount, LDayDataCounter) <= _
   sWFIWaterFromGen _
   And _
   sWFIWaterConsumptionFromLoop(LCount, LDayDataCounter) <> 0 Then
      'No change in Volume as removal is less than intake from still:
      sWFIWaterFromDI(LCount, LDayDataCounter) _
      = sWFIWaterConsumptionFromLoop(LCount, LDayDataCounter) _
      + sWFIWaterConsumptionFromLoop(LCount, LDayDataCounter) _
      * sWFIGenBlowDown / 100
   End If
   '
   If sWFICalcVolumeTank < sWFIVolumeTank Then
      'Max flow of water from generation was added to storage tank:
      sWFIWaterFromDI(LCount, LDayDataCounter) = sWFIWaterFromGen _
      + sWFIWaterFromGen * sWFIGenBlowDown / 100
   End If
   If sWFICalcVolumeTank > sWFIVolumeTank Then
      'Water Volume in storage tank is at max.:
      sWFICalcVolumeTank = sWFIVolumeTank
   End If
   If sWFICalcVolumeTank < sWFIMinVolumeTank Then
      'Water Volume is below min allowable:
      '
      Sheets("Sheet 9 - Report Page WFI Day 1").Select
      Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
      " period of no WFI being available. Change Input Data."
      With Selection.Font
         .Name = "Arial"
         .FontStyle = "Bold"
         .Size = 12
         .Strikethrough = False
         .Superscript = False
         .Subscript = False
         .OutlineFont = False
         .Shadow = False
         .Underline = xlUnderlineStyleNone
```

```
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("11 - Report Page WFI All Days").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available.  Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("Sheet 6 - Report Page DI Day 1").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available. Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Sheets("8 - Report Page DI All Days").Select
    Cells(2, 1) = "Simulation Result will be incorrect as there is a" + _
    " period of no WFI being available. Change Input Data."
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = 3
    End With
    Cells(4, 1).Select
    '
    Msg = "The simulation predicts there will be a period of no WFI being " + _
    "available. The results of the simulation will be " + _
    " incorrect. The simulation will therefore be terminated." + _
    "  Please change input data on worksheet of the WFI storage tank. "
    Style = vbOKOnly
    Title = "A critical Message"
    Response = MsgBox(Msg, Style, Title)
    End                             'Terminate Simulation.
End If
If iTen = 10 Then
    iTen = 0
    'Write to sheet for graph of Water Volume in WFI Tank:
    'Multiply WFI water volume by density correction coefficient (Rho80/Rho20).
    Sheets("Sheet 21 - WFI Volume in Tank").Select
```

```
      Cells((LCount / 10) + 3, LDayDataCounter) = sWFICalcVolumeTank * Rho20 / Rho80
   End If
   '
   If iI = 1200 Then
      iI = 0
      sProgress = sProgress + sIncrement
      Call Progress_Indicator(sProgress)
   End If
   iI = iI + 1
   '
   iTen = iTen + 1
   LCount = LCount + 1
Loop
'
'Write WFI Water Generation to Storage Tank to sheet:
Sheets("Sheet 20 - WFI Gen. to Tank").Select
iTen = 1
iI = 0
LCount = 1
Do While LCount < 86400 + 1
   If iTen = 10 Then
      iTen = 0
      Cells(LCount / 10 + 3, LDayDataCounter) _
      = sWFIWaterFromDI(LCount, LDayDataCounter) * 3600 '[m3/h].
   End If
   '
   If iI = 1200 Then
      iI = 0
      sProgress = sProgress + sIncrement
      Call Progress_Indicator(sProgress)
   End If
   iI = iI + 1
   '
   iTen = iTen + 1
   LCount = LCount + 1
Loop
'
'Write Water Generation Data to sheet "Water Demand WFI":
Sheets("Sheet 19 - Water Demand WFI").Select
iTen = 0
iI = 0
LCount = 1
Do While LCount < 86400 + 1
   If iTen = 10 Then
      iTen = 0
      Cells(LCount / 10 + 3, LDayDataCounter) = _
      sWFIWaterConsumptionFromLoop(LCount, LDayDataCounter) * 3600    '[m3/h].
   End If
   '
   If iI = 1200 Then
      iI = 0
      sProgress = sProgress + sIncrement
      Call Progress_Indicator(sProgress)
   End If
   iI = iI + 1
   '
   iTen = iTen + 1
   LCount = LCount + 1
Loop
'
'Write min. water Volume in tank to sheet:
Sheets("Sheet 21 - WFI Volume in Tank").Select
iI = 0
LCount = 1
Do While LCount < 8640 + 1
   Cells(LCount + 3, 11) = sWFIMinVolumeTank
   '
   If iI = 1200 Then
      iI = 0
```

```
        sProgress = sProgress + sIncrement
        Call Progress_Indicator(sProgress)
    End If
    iI = iI + 1
    '
    LCount = LCount + 1
Loop
'
'Write Water Demand WFI for all simulated days into one Column for overall graph:
Sheets("Sheet 19 - Water Demand WFI").Select
LCount = 1
iNCount = 0
iI = 0
Do While LCount < 8640 + 1
    If iNCount = 2 Then
        Cells(Int(LCount / 2) + 3, 14) = Cells(LCount + 3, 1)
        Cells(Int(8643 / 2) + Int(LCount / 2), 14) = Cells(LCount + 3, 2)
        Cells(CSng(2) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 3)
        Cells(CSng(3) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 4)
        Cells(CSng(4) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 5)
        Cells(CSng(5) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 6)
        Cells(CSng(6) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 7)
        iNCount = 0
    End If
    '
    If iI = 150 Then
        iI = 0
        sProgress = sProgress + sIncrement
        Call Progress_Indicator(sProgress)
    End If
    iI = iI + 1
    '
    iNCount = iNCount + 1
    LCount = LCount + 1
Loop
'
'Write Water Demand WFI for all simulated days into one Column for overall graph:
Sheets("Sheet 20 - WFI Gen. to Tank").Select
LCount = 1
iNCount = 0
iI = 0
Do While LCount < 8640 + 1
    If iNCount = 2 Then
        Cells(Int(LCount / 2) + 3, 14) = Cells(LCount + 3, 1)
        Cells(Int(8643 / 2) + Int(LCount / 2), 14) = Cells(LCount + 3, 2)
        Cells(CSng(2) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 3)
        Cells(CSng(3) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 4)
        Cells(CSng(4) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 5)
        Cells(CSng(5) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 6)
        Cells(CSng(6) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 7)
        iNCount = 0
    End If
    '
    If iI = 150 Then
        iI = 0
        sProgress = sProgress + sIncrement
        Call Progress_Indicator(sProgress)
    End If
    iI = iI + 1
    '
    iNCount = iNCount + 1
    LCount = LCount + 1
Loop
'
'Write Water Demand WFI for all simulated days into one Column for overall graph:
Sheets("Sheet 21 - WFI Volume in Tank").Select
LCount = 1
iNCount = 0
iI = 0
```

```
    Do While LCount < 8640 + 1
       If iNCount = 2 Then
          Cells(Int(LCount / 2) + 3, 14) = Cells(LCount + 3, 1)
          Cells(Int(8643 / 2) + Int(LCount / 2), 14) = Cells(LCount + 3, 2)
          Cells(CSng(2) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 3)
          Cells(CSng(3) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 4)
          Cells(CSng(4) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 5)
          Cells(CSng(5) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 6)
          Cells(CSng(6) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 7)
          iNCount = 0
       End If
       '
       If iI = 150 Then
          iI = 0
          sProgress = sProgress + sIncrement
          Call Progress_Indicator(sProgress)
       End If
       iI = iI + 1
       '
       iNCount = iNCount + 1
       LCount = LCount + 1
    Loop
    '
    'Write WFI distribution pump flowrate to sheet:
    Sheets("Sheet 19 - Water Demand WFI").Select
    LCount = 1
    Do While LCount < 8640 + 1
       Cells(LCount + 3, 15) = sWFIPumpFlowrate
       LCount = LCount + 1
    Loop
    '
    End Sub
'
'
'
Sub DI_Water_Demand_From_Generation_And_Tank_Volume()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine calculates the Water volume of the DI in DI Storage Tank          @
    '@ as a function of time [m3]. Routine also calculates the DI demand                 @
    '@ from filling the DI water tank if not full at the start of simulation.            @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIWaterFromGen = DI Water from DI generation to DI storage tank [m3/h].
    ' sDIStartVolumeTank = DI storage tank at start of simulation [m3].
    ' sDIMinVolumeTank = Minimum allowable water Volume in storage tank [m3].
    ' sDIStartVolumeTank = Volume in storage tank at beginning of simulation [m3].
    ' sDICalcVolumeTank = DI Volume in storage tank [m3].
    ' sDIVolumeTank = Volume of DI storage tank [m3].
    ' sIncrement = Global variable [-].
    ' iResolution = Global variable [-].
    ' iDayData = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' LCount = Index for Do While-Loop [-].
    ' sVolumeTank = Volume in storage tank for every second [m3].
    ' iTen = Writing every tenth value of the Volume to the sheet for graph etc.
    ' iI = Index for Progress Indicator [-].
    ' sDelta = For Euler integration [-].
    ' LRepeatsCount = Index for Do While-Loop [-].
    ' iNCount = Counter for writing results to sheets [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sDIWaterToTank = [m3/h].
    ' iDITankMinAlarm = [m3].
    ' sWaterConsumption(Array) = Global variable [m3/s].
    ' sVolumeTank = [m3].
    ' Messages:  1. The simulation predicts there will be no WFI available during an
```

```
'                   offtake period.  The results of the simulation will be
'                   incorrect. Please change input data on worksheet.
'               2. Simulation Result will be incorrect as there is a
'                   period of no WFI being available.
' sProgress = Global variable [-].
'
Dim sDIWaterFromGen As Single, sDIStartVolumeTank As Single, sDIMinVolumeTank As Single
Dim iCheckStart As Integer, iCheckEnd As Integer
Dim sDIWaterToTank() As Single
Dim sDICalcVolumeTank As Single, sDIVolumeTank As Single, sDelta As Single, sTemp As Single
Dim iTen As Integer, iI As Integer, iNCount As Integer
Dim LCount As Long
Dim Msg As String, Style As String, Title As String, Response As String
'
ReDim sDIWaterToTank(86400, iDayData)
'
sDelta = 1
iI = 0
LDayDataCounter = 1
Do While LDayDataCounter < iDayData + 1
   'Read in Data for calculation of Volume in DI storage tank:
   If LDayDataCounter = 1 Then
      Sheets("Sheet 4 - DI Gen. & Tank Data").Select
      sDIWaterFromGen = Cells(4, 6)                    '[m3/h].
      sDIWaterFromGen = sDIWaterFromGen / 3600         '[m3/h] to [m3/s].
      sDIVolumeTank = Cells(7, 6)                      '[m3].
      sDIMinVolumeTank = Cells(11, 6)                  '[m3].
      sDIStartVolumeTank = Cells(9, 6)                 '[m3].
      sDICalcVolumeTank = sDIStartVolumeTank           'Start condition.
      'End Read in Data.
      Sheets("Sheet 17 - DI Volume in Tank").Select
   Else
      Sheets("Sheet 17 - DI Volume in Tank").Select
      sDICalcVolumeTank = Cells(8643, LDayDataCounter - 1)          '[m3].
      'End Read in Data.
   End If
   '
   LCount = 1
   iTen = 1
   Do While LCount < LOneDayInSeconds + 1
      'Mass Balance Tank [m3] (every second):
      'Integrate Mass Balance below with explicit first order Euler.
      sDICalcVolumeTank = sDICalcVolumeTank + (sDIWaterFromGen - _
      sDIWaterConsumptionFromLoop(LCount, LDayDataCounter)) * sDelta
      '
      If (sDIWaterConsumptionFromLoop(LCount, LDayDataCounter) <= _
         sDIWaterFromGen _
         And _
         sDIWaterConsumptionFromLoop(LCount, LDayDataCounter) <> 0) _
         Then
         'No change in Volume, but water added [m3/s].
         sDIWaterToTank(LCount, LDayDataCounter) = _
         sDIWaterConsumptionFromLoop(LCount, LDayDataCounter)
      End If
      '
      If sDICalcVolumeTank < sDIVolumeTank Then
         'Max flow of water from generation was added to storage tank [m3/s]:
         sDIWaterToTank(LCount, LDayDataCounter) = sDIWaterFromGen
      End If
      '
      If sDICalcVolumeTank > sDIVolumeTank Then
         'Water Volume in storage tank at max [m3]:
         sDICalcVolumeTank = sDIVolumeTank
      End If
      '
      If sDICalcVolumeTank < sDIMinVolumeTank Then
         'Water Volume below min allowable [m3/s]:
         sDICalcVolumeTank = sDIMinVolumeTank
         sDIWaterToTank(LCount, LDayDataCounter) = sDIWaterFromGen
```

```
'
Sheets("Sheet 6 - Report Page DI Day 1").Select
Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
" period of no DI water being available.  Change Input Data."
With Selection.Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = 3
End With
Cells(4, 1).Select
'
Sheets("8 - Report Page DI All Days").Select
Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
" period of no DI water being available.  Change Input Data."
With Selection.Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = 3
End With
Cells(4, 1).Select
'
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
" period of no DI water being available.  Change Input Data."
With Selection.Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = 3
End With
Cells(4, 1).Select
'
Sheets("11 - Report Page WFI All Days").Select
Cells(2, 1) = "Simulation Result will be incorrect as there is a " + _
" period of no DI water being available.  Change Input Data."
With Selection.Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = 3
End With
Cells(4, 1).Select
Msg = "The simulation predicts there will be a period of no DI " + _
```

```
            "being available. The results of the simulation will be " + _
            " incorrect. The simulation will therefore be terminated." + _
            " Please change input data on worksheet of the DI storage tank. "
            Style = vbOKOnly
            Title = "A critical Message"
            Response = MsgBox(Msg, Style, Title)
            End                         'Terminate simulation.
        End If
        If iTen = 10 Then
            'Write to sheet for graph of Water Volume in DI Tank:
            'Only every 10th Datapoint is written to the sheet.
            Sheets("Sheet 17 - DI Volume in Tank").Select
            Cells((LCount / 10) + 3, LDayDataCounter) = sDICalcVolumeTank
            iTen = 0
        End If
        '
        If iI = 1200 Then
            iI = 0
            sProgress = sProgress + sIncrement
            Call Progress_Indicator(sProgress)
        End If
        iI = iI + 1
        '
        iTen = iTen + 1
        LCount = LCount + 1
    Loop
    LDayDataCounter = LDayDataCounter + 1
Loop
'
'Write DI Water Demand from Loop to sheet:
Sheets("Sheet 16 - Water Demand DI").Select
LDayDataCounter = 1
Do While LDayDataCounter < iDayData + 1
    iTen = 1
    iI = 0
    LCount = 1
    Do While LCount < 86400 + 1
        If iTen = 10 Then
            iTen = 0
            Cells(LCount / 10 + 3, LDayDataCounter) _
            = sDIWaterConsumptionFromLoop(LCount, LDayDataCounter)
        End If
        '
        If iI = 1200 Then
            iI = 0
            sProgress = sProgress + sIncrement
            Call Progress_Indicator(sProgress)
        End If
        iI = iI + 1
        '
        iTen = iTen + 1
        LCount = LCount + 1
    Loop
    LDayDataCounter = LDayDataCounter + 1
Loop
'
'Write DI Water Demand from Generation to Tank to sheet:
Sheets("Sheet 18 - DI Gen. to Tank").Select
LDayDataCounter = 1
Do While LDayDataCounter < iDayData + 1
    iTen = 1
    iI = 0
    LCount = 1
    Do While LCount < 86400 + 1
        If iTen = 10 Then
            iTen = 0
            Cells(LCount / 10 + 3, LDayDataCounter) _
            = sDIWaterToTank(LCount, LDayDataCounter)
        End If
```

```
        '
        If iI = 1200 Then
           iI = 0
           sProgress = sProgress + sIncrement
           Call Progress_Indicator(sProgress)
        End If
        iI = iI + 1
        '
        iTen = iTen + 1
        LCount = LCount + 1
    Loop
    LDayDataCounter = LDayDataCounter + 1
Loop
'
'Write Water Generation Data to sheet "Water Demand DI":
Sheets("Sheet 16 - Water Demand DI").Select
LDayDataCounter = 1
Do While LDayDataCounter < iDayData + 1
    iTen = 0
    iI = 0
    LCount = 1
    Do While LCount < 86400 + 1
        If iTen = 10 Then
           iTen = 0
           Cells(LCount / 10 + 3, LDayDataCounter) = _
           sDIWaterConsumptionFromLoop(LCount, LDayDataCounter)
        End If
        '
        If iI = 1200 Then
           iI = 0
           sProgress = sProgress + sIncrement
           Call Progress_Indicator(sProgress)
        End If
        iI = iI + 1
        '
        iTen = iTen + 1
        LCount = LCount + 1
    Loop
    LDayDataCounter = LDayDataCounter + 1
Loop
'
'Write min. water Volume in tank to sheet:
Sheets("Sheet 17 - DI Volume in Tank").Select
LCount = 1
iI = 0
Do While LCount < 8640 + 1
    Cells(LCount + 3, 11) = sDIMinVolumeTank
    '
    If iI = 1200 Then
       iI = 0
       sProgress = sProgress + sIncrement
       Call Progress_Indicator(sProgress)
    End If
    iI = iI + 1
    '
    LCount = LCount + 1
Loop
'
'Write Water Demand WFI for all simulated days into one Column for overall graph:
Sheets("Sheet 16 - Water Demand DI").Select
LCount = 1
iNCount = 0
iI = 0
Do While LCount < 8640 + 1
    If iNCount = 2 Then
       Cells(Int(LCount / 2) + 3, 14) = Cells(LCount + 3, 1)
       Cells(Int(8643 / 2) + Int(LCount / 2), 14) = Cells(LCount + 3, 2)
       Cells(CSng(2) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 3)
       Cells(CSng(3) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 4)
```

```
        Cells(CSng(4) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 5)
        Cells(CSng(5) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 6)
        Cells(CSng(6) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 7)
        iNCount = 0
    End If
    '
    If iI = 150 Then
        iI = 0
        sProgress = sProgress + sIncrement
        Call Progress_Indicator(sProgress)
    End If
    iI = iI + 1
    '
    iNCount = iNCount + 1
    LCount = LCount + 1
Loop
'
'Write Water Demand WFI for all simulated days into one Column for overall graph:
Sheets("Sheet 18 - DI Gen. to Tank").Select
LCount = 1
iNCount = 0
iI = 0
Do While LCount < 8640 + 1
    If iNCount = 2 Then
        Cells(Int(LCount / 2) + 3, 14) = Cells(LCount + 3, 1)
        Cells(Int(8643 / 2) + Int(LCount / 2), 14) = Cells(LCount + 3, 2)
        Cells(CSng(2) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 3)
        Cells(CSng(3) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 4)
        Cells(CSng(4) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 5)
        Cells(CSng(5) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 6)
        Cells(CSng(6) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 7)
        iNCount = 0
    End If
    '
    If iI = 150 Then
        iI = 0
        sProgress = sProgress + sIncrement
        Call Progress_Indicator(sProgress)
    End If
    iI = iI + 1
    '
    iNCount = iNCount + 1
    LCount = LCount + 1
Loop
'
'Write Water Demand WFI for all simulated days into one Column for overall graph:
Sheets("Sheet 17 - DI Volume in Tank").Select
LCount = 1
iNCount = 0
iI = 0
Do While LCount < 8640 + 1
    If iNCount = 2 Then
        Cells(Int(LCount / 2) + 3, 14) = Cells(LCount + 3, 1)
        Cells(Int(8643 / 2) + Int(LCount / 2), 14) = Cells(LCount + 3, 2)
        Cells(CSng(2) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 3)
        Cells(CSng(3) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 4)
        Cells(CSng(4) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 5)
        Cells(CSng(5) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 6)
        Cells(CSng(6) * Int(8643 / 2) + Int(LCount / 2) - 2, 14) = Cells(LCount + 3, 7)
        iNCount = 0
    End If
    '
    If iI = 150 Then
        iI = 0
        sProgress = sProgress + sIncrement
        Call Progress_Indicator(sProgress)
    End If
    iI = iI + 1
    '
```

```
        iNCount = iNCount + 1
        LCount = LCount + 1
    Loop
    '
    End Sub
'
'
'
Sub Analysis_of_Data_WFI_Day_1()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine does the analysis of the WFI data for Day 1.   @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterFromGen = WFI from Generation Plant [m3/s].
    ' sWFIVolumeTank = Volume of WFI storage tank [m3].
    ' sWFIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sWFIStartVolumeTank = Start volume in WFI storage tank [m3].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' iNCount = Calculation of WFI generation utilisation [-].
    ' sWFIVolume(Array) = Volume in WFI storage tank [-].
    ' sValue = Calculation of WFI generation utilisation [-].
    ' sTemp = Temporary value [-].
    ' iLocationOfPercentile = For calculation of percentile.
    ' iNCount = Counter for writing to sheets [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sMin = Minimum value.
    ' sMax = Maximum value.
    ' sAverage = Average value.
    ' sWFISum = Overall WFI consumption from loop excluding tank filling [m3].
    '
    Dim sWFIWaterFromGen As Single, sWFIVolumeTank As Single
    Dim sWFIStartVolumeTank As Single, sWFIMinVolumeTank As Single
    Dim sMin As Single, sMinTemp As Single, sMax As Single, sTemp As Single
    Dim sWFIVolume(8650) As Single, sWFILoopConsumption(8650) As Single
    Dim sAverage As Single, sValue As Single, sWFISum As Single
    Dim iNCount As Integer, iRowNumber As Integer, iCount As Integer
    Dim iLocationOfPercentile As Integer, sWFIFromGen(8650) As Single
    '
    'Read in Data:
    Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
    sWFIWaterFromGen = Cells(4, 6)                    '[m3/h].
    sWFIVolumeTank = Cells(7, 6)                      '[m3].
    sWFIMinVolumeTank = Cells(11, 6)                  '[m3].
    sWFIStartVolumeTank = Cells(9, 6)                 '[m3].
    'End Read in Data.
    '
    Sheets("Sheet 9 - Report Page WFI Day 1").Select
    Cells(4, 7) = sWFIWaterFromGen
    Cells(5, 7) = sWFIVolumeTank
    Cells(7, 7) = sWFIMinVolumeTank
    Cells(6, 7) = sWFIStartVolumeTank
    '
    'Read in Data for Volume in WFI Tank [m3]:
    Sheets("Sheet 21 - WFI Volume in Tank").Select
    iCount = 1
    Do While iCount < 8640 + 1
        sWFIVolume(iCount) = Cells(iCount + 3, 1)
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Calculate the minimum Volume in WFI Tank [m3]:
    sMin = sWFIVolume(1)
```

```
iCount = 2
Do While iCount < 8640 + 1
   sTemp = sWFIVolume(iCount)
   If sTemp < sMin Then
      sMin = sTemp
   End If
   iCount = iCount + 1
Loop
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Range("Q4") = sMin
'
'Find the maximum Volume in WFI Tank [m3]:
sMax = sWFIVolume(1)
iCount = 2
Do While iCount < 8640 + 1
   sTemp = sWFIVolume(iCount)
   If sTemp > sMax Then
      sMax = sTemp
   End If
   iCount = iCount + 1
Loop
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Range("Q5") = sMax
'
'Calculate the range of the Volume in the storage tank [m3]:
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Range("Q6") = sMax - sMin
'
'Calculate the average Volume in the WFI Storage Tank [m3]:
Sheets("Sheet 21 - WFI Volume in Tank").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
   sAverage = sAverage + Cells(iCount + 3, 1)
   iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Range("Q8") = sAverage
'
'Calculate the average water flow from the WFI generation plant [m3/h]:
Sheets("Sheet 20 - WFI Gen. to Tank").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
   sAverage = sAverage + Cells(iCount + 3, 1)
   iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Range("Q9") = sAverage
'
'Calculate the average offtake from the WFI loop [m3/h]:
Sheets("Sheet 19 - Water Demand WFI").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
   sAverage = sAverage + Cells(iCount + 3, 1)
   iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Range("Q10") = sAverage
'
'Read in Data for Volume in WFI Tank [m3]:
Sheets("Sheet 21 - WFI Volume in Tank").Select
iCount = 1
Do While iCount < 8640 + 1
   sWFIVolume(iCount) = Cells(iCount + 3, 1)
```

```
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Sort algorithm for sWFIVolume(iCount):
    iCount = 2
    Do While iCount < 8640 + 1
        sTemp = sWFIVolume(iCount)
        For iRowNumber = iCount - 1 To 1 Step -1
            If (sWFIVolume(iRowNumber) <= sTemp) Then
                GoTo 10
            End If
            sWFIVolume(iRowNumber + 1) = sWFIVolume(iRowNumber)
        Next iRowNumber
        iRowNumber = 0
10      sWFIVolume(iRowNumber + 1) = sTemp
    iCount = iCount + 1
    Loop
    '
    'Calculate the 90% percentile WFI Water Volume in Tank [m3]:
    iLocationOfPercentile = Int(90 / 100 * 8640)
    iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
    Sheets("Sheet 9 - Report Page WFI Day 1").Select
    Range("Q12") = sWFIVolume(iLocationOfPercentile)
    '
    'Calculate the 60% percentile WFI Water Volume in Tank [m3]:
    iLocationOfPercentile = Int(60 / 100 * 8640)
    iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
    Sheets("Sheet 9 - Report Page WFI Day 1").Select
    Range("Q13") = sWFIVolume(iLocationOfPercentile)
    '
    'Calculate the overall WFI Water Consumption from Loop excluding the tank [m3]:
    Sheets("Sheet 19 - Water Demand WFI").Select
    sWFISum = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sWFISum = sWFISum + Cells(iCount, 1) / 3600 * 10
        iCount = iCount + 1
    Loop
    Sheets("Sheet 9 - Report Page WFI Day 1").Select
    Range("Q15") = sWFISum
    '
    'Calculate the overall WFI Water to Storage Tank [m3]:
    Sheets("Sheet 20 - WFI Gen. to Tank").Select
    sWFISum = 0
    iCount = 4
    Do While iCount < 8340 + 1
        sWFISum = sWFISum + Cells(iCount, 1) / 3600 * 10
        iCount = iCount + 1
    Loop
    Sheets("Sheet 9 - Report Page WFI Day 1").Select
    Range("Q16") = sWFISum
    '
    'Calculate the WFI Generation utilisation [-]:
    Sheets("Sheet 20 - WFI Gen. to Tank").Select
    iNCount = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sValue = Cells(iCount, 1)
        If sValue <> 0 Then
            iNCount = iNCount + 1
        End If
        iCount = iCount + 1
    Loop
    Sheets("Sheet 9 - Report Page WFI Day 1").Select
    Range("Q18") = iNCount / 8640
    Range("B5").Select
    '
End Sub
```

```
'
'
'
Sub Analysis_of_Data_DI_Day_1()
    '
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '@ This subroutine does the analysis of the DI data for Day 1.  @
   '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   '
   'INPUTS
   ' sWDIaterFromGen = DI from Generation Plant [m3/s].
   ' sDIVolumeTank = Volume of DI storage tank [m3].
   ' sDIMinVolumeTank = Min. allowable volume of tank [m3].
   ' sDIStartVolumeTank = Start volume in DI storage tank [m3].
   ' sDIVolume() = Volume in DI storage tank [m3].
   ' sDIFlow() = DI flow to storage tank [m3/s].
   '
   'INTERMEDIATE VARIABLES
   ' iRowNumber = Number of rows as per Input Table [-].
   ' iCount = Index for DoWhile Loop [-].
   ' iNCount = Calculation of DI generation utilisation [-].
   ' sValue = Calculation of WFI generation utilisation [-].
   ' sTemp = Temporary value [various].
   ' iLocationOfPercentile = For calculation of percentile [m3].
   ' iNCount = Counter for writing to sheets [-].
   '
   'CALCULATED VALUES/OUTPUTS
   ' sMin = Minimum value [various].
   ' sMax = Maximum value [various].
   ' sAverage = Average value [various].
   ' sDISum = Overall WFI consumption from loop excluding tank filling [m3].
   '
   Dim sDIWaterFromGen As Single, sDIVolumeTank As Single
   Dim sDIMinVolumeTank As Single, sDIStartVolumeTank As Single, sAverage As Single
   Dim iRowNumber As Integer, iCount As Integer, iNCount As Integer
   Dim sDIVolume(8640) As Single, sDIFlow(8640) As Single, sDISum As Single
   Dim sDILoopConsumption(8640) As Single, sDIFromGen(8650) As Single
   Dim sMin As Single, sMax As Single, sValue As Single, sTemp As Single
   Dim iLocationOfPercentile As Integer
   '
   'Read in Data:
   Sheets("Sheet 4 - DI Gen. & Tank Data").Select
   sDIWaterFromGen = Cells(4, 6)                  '[m3/h].
   sDIVolumeTank = Cells(7, 6)                    '[m3].
   sDIMinVolumeTank = Cells(11, 6)                '[m3].
   sDIStartVolumeTank = Cells(9, 6)               '[m3].
   'End Read in Data.
   '
   Sheets("Sheet 6 - Report Page DI Day 1").Select
   '
   Cells(4, 7) = sDIWaterFromGen
   Cells(5, 7) = sDIVolumeTank
   Cells(6, 7) = sDIStartVolumeTank
   Cells(7, 7) = sDIMinVolumeTank
   '
   'Read in Data for Volume in DI Tank [m3]:
   Sheets("Sheet 17 - DI Volume in Tank").Select
   iCount = 1
   Do While iCount < 8640 + 1
      sDIVolume(iCount) = Cells(iCount + 3, 1)
      iCount = iCount + 1
   Loop
   'End Read in Data.
   '
   'Find the minimum Volume in DI Tank [m3]:
   sTemp = sDIVolume(1)
   iCount = 1
   Do While iCount < 8640 + 1
      sMin = sDIVolume(iCount)
```

```
    If sMin < sTemp Then
        sTemp = sMin
    End If
    iCount = iCount + 1
Loop
Sheets("Sheet 6 - Report Page DI Day 1").Select
Range("Q4") = sMin
'
'Find the maximum Volume in DI Tank [m3]:
sTemp = sDIVolume(1)
iCount = 1
Do While iCount < 8640 + 1
    sMax = sDIVolume(iCount)
    If sMax > sTemp Then
        sTemp = sMax
    End If
    iCount = iCount + 1
Loop
Sheets("Sheet 6 - Report Page DI Day 1").Select
Range("Q5") = sMax
'
'Calculate the range of the Volume in the storage tank [m3]:
Sheets("Sheet 6 - Report Page DI Day 1").Select
Range("Q6") = sMax - sMin
'
'Calculate the average Volume in DI Tank [m3]:
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + sDIVolume(iCount)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("Sheet 6 - Report Page DI Day 1").Select
Range("Q8") = sAverage
'
'Calculate the average water flow from the DI generation plant [m3/h]:
Sheets("Sheet 18 - DI Gen. to Tank").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, 1)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640 * 3600
Sheets("Sheet 6 - Report Page DI Day 1").Select
Range("Q10") = sAverage
'
'Calculate the average offtake from the DI loop [m3/h]:
Sheets("Sheet 16 - Water Demand DI").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, 1)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640 * 3600
Sheets("Sheet 6 - Report Page DI Day 1").Select
Range("Q9") = sAverage
'
'Sort algorithm for vDIVolume(iCount):
iCount = 2
Do While iCount < 8640 + 1
    sTemp = sDIVolume(iCount)
    For iRowNumber = iCount - 1 To 1 Step -1
        If (sDIVolume(iRowNumber) <= sTemp) Then
            GoTo 10
        End If
        sDIVolume(iRowNumber + 1) = sDIVolume(iRowNumber)
```

```
        Next iRowNumber
        iRowNumber = 0
10      sDIVolume(iRowNumber + 1) = sTemp
    iCount = iCount + 1
    Loop
    '
    'Calculate the 90% percentile DI Water Volume in Tank [m3]:
    iLocationOfPercentile = Int(90 / 100 * 8640)
    iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
    Sheets("Sheet 6 - Report Page DI Day 1").Select
    Range("Q12") = sDIVolume(iLocationOfPercentile)
    '
    'Calculate the 60% percentile DI Water Volume in Tank [m3]:
    iLocationOfPercentile = Int(60 / 100 * 8640)
    iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
    Sheets("Sheet 6 - Report Page DI Day 1").Select
    Range("Q13") = sDIVolume(iLocationOfPercentile)
    '
    'Read in Data for DI Water Loop Consumption [m3/h]:
    Sheets("Sheet 16 - Water Demand DI").Select
    iCount = 1
    Do While iCount < 8640 + 1
        sDILoopConsumption(iCount) = Cells(iCount + 3, 1)
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Sort algorithm for sDILoopConsumption(iCount) [m3/s]:
    iCount = 2
    Do While iCount < 8640 + 1
        sTemp = sDILoopConsumption(iCount)
        For iRowNumber = iCount - 1 To 1 Step -1
            If (sDILoopConsumption(iRowNumber) <= sTemp) Then
                GoTo 20
            End If
            sDILoopConsumption(iRowNumber + 1) = sDILoopConsumption(iRowNumber)
        Next iRowNumber
        iRowNumber = 0
20      sDILoopConsumption(iRowNumber + 1) = sTemp
    iCount = iCount + 1
    Loop
    '
    'Read in Data for DI Water from Generation Plant [m3/h]:
    Sheets("Sheet 18 - DI Gen. to Tank").Select
    iCount = 1
    Do While iCount < 8640 + 1
        sDIFromGen(iCount) = Cells(iCount + 3, 1)
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    'Sort algorithm for sDILoopConsumption(iCount) [m3/h]:
    iCount = 2
    Do While iCount < 8640 + 1
        sTemp = sDIFromGen(iCount)
        For iRowNumber = iCount - 1 To 1 Step -1
            If (sDIFromGen(iRowNumber) <= sTemp) Then
                GoTo 30
            End If
            sDIFromGen(iRowNumber + 1) = sDIFromGen(iRowNumber)
        Next iRowNumber
        iRowNumber = 0
30      sDIFromGen(iRowNumber + 1) = sTemp
    iCount = iCount + 1
    Loop
    '
    'Calculate the overall DI Water Consumption from Loop excluding the tank [m3]:
    Sheets("Sheet 16 - Water Demand DI").Select
    sDISum = 0
```

```
    iCount = 4
    Do While iCount < 8640 + 1
        sDISum = sDISum + Cells(iCount, 1)
        iCount = iCount + 1
    Loop
    Sheets("Sheet 6 - Report Page DI Day 1").Select
    Range("Q15") = sDISum * 10
    '
    'Calculate the overall DI Water to Storage Tank [m3]:
    Sheets("Sheet 18 - DI Gen. to Tank").Select
    sDISum = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sDISum = sDISum + Cells(iCount, 1)
        iCount = iCount + 1
    Loop
    Sheets("Sheet 6 - Report Page DI Day 1").Select
    Range("Q16") = sDISum * 10
    '
    'Calculate the DI Generation utilisation [-]:
    Sheets("Sheet 18 - DI Gen. to Tank").Select
    iNCount = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sValue = Cells(iCount, 1)
        If sValue <> 0 Then
            iNCount = iNCount + 1
        End If
        iCount = iCount + 1
    Loop
    Sheets("Sheet 6 - Report Page DI Day 1").Select
    Range("Q18") = iNCount / 8640
    Range("B5").Select
    '
End Sub
'
'
'
Sub Analysis_WFI_Other_Days()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine does the analysis of all data for of the WFI data for the day as    @
    '@ selected by the user.                                                               @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterFromGen = WFI from Generation Plant [m3/s].
    ' sWFIVolumeTank = Volume of WFI storage tank [m3].
    ' sWFIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sWFIStartVolumeTank = Start volume in WFI storage tank [m3].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' sWFIVolume() = Volume in WFI storage tank [-].
    ' sValue = Calculation of WFI generation utilisation [-].
    ' sTemp = Temporary value [-].
    ' iLocationOfPercentile = For calculation of percentile.
    '
    'CALCULATED VALUES/OUTPUTS
    ' sMin = Minimum value.
    ' sMedian = Median value.
    ' sMax = Maximum value.
    ' sAverage = Average value.
    ' sWFISum = Overall WFI consumption from loop excluding tank filling [m3].
    '
    Dim sDaySelected As String, Alpha As String, Col As String
    Dim sMin As Single, sValue As Single
    Dim sMax As Single, s90Volume As Single
```

```
Dim sAverage As Single, sTemp As Single
Dim sWFIVolumeSim(8640) As Single
Dim sWFISum As Single, sWFIWaterFromGen As Single
Dim sWFIVolumeTank As Single, sWFIMinVolumeTank As Single, sWFIStartVolumeTank As Single
Dim iCount As Integer, iRowNumber As Integer, iLocationOfPercentile As Integer
Dim iDaySelected As Integer, iTemp As Integer
'
Application.ScreenUpdating = False
'
Sheets("10 - Report Page WFI Other Days").Select
iDaySelected = Cells(11, 4)
sDaySelected = Cells(11, 4)
'
Select Case iDaySelected
Case 1
   Alpha = "A"
Case 2
   Alpha = "B"
Case 3
   Alpha = "C"
Case 4
   Alpha = "D"
Case 5
   Alpha = "E"
Case 6
   Alpha = "F"
Case 7
   Alpha = "G"
End Select
'
Sheets("10 - Report Page WFI Other Days").Select
Range("A1:O1").Select
ActiveCell.FormulaR1C1 = "WFI Results for Day " + sDaySelected
'
ActiveSheet.ChartObjects("Chart 3").Activate
ActiveChart.ChartTitle.Select
Selection.Characters.Text = "WFI Water Volume in Storage Tank Day " _
+ sDaySelected + " [m3] "
Selection.AutoScaleFont = False
With Selection.Characters(Start:=1, Length:=100).Font
   .Name = "Arial"
   .FontStyle = "Bold"
   .Size = 16.5
   .Strikethrough = False
   .Superscript = False
   .Subscript = False
   .OutlineFont = False
   .Shadow = False
   .Underline = xlUnderlineStyleNone
   .ColorIndex = xlAutomatic
End With
Selection.AutoScaleFont = False
'
ActiveChart.PlotArea.Select
ActiveChart.SetSourceData Source:=Sheets("Sheet 21 - WFI Volume in Tank").Range( _
Alpha + "4:" + Alpha + "8430"), PlotBy:=xlColumns
ActiveChart.SeriesCollection.NewSeries
'
ActiveWindow.Visible = False
ActiveSheet.ChartObjects("Chart 4").Activate
ActiveChart.ChartTitle.Select
Selection.Characters.Text = _
   "WFI Water Generation to Storage Tank - Day " + sDaySelected + " [m3/s] "
Selection.AutoScaleFont = False
With Selection.Characters(Start:=1, Length:=100).Font
   .Name = "Arial"
   .FontStyle = "Bold"
   .Size = 16.5
   .Strikethrough = False
```

```
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Selection.AutoScaleFont = False
'
ActiveChart.SetSourceData Source:=Sheets("Sheet 20 - WFI Gen. to Tank").Range( _
    Alpha + "4:" + Alpha + "8640"), PlotBy:=xlColumns
ActiveSheet.ChartObjects("Chart 2").Activate
ActiveChart.ChartTitle.Select
Selection.Characters.Text = _
    "WFI Water Demand from WFI Distribution System - Day " + sDaySelected + " [m3] "
Selection.AutoScaleFont = False
With Selection.Characters(Start:=1, Length:=100).Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 15.75
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Selection.AutoScaleFont = False
'
ActiveChart.SetSourceData Source:=Sheets("Sheet 19 - Water Demand WFI").Range(Alpha _
+ "4:" + Alpha + "8430"), PlotBy:=xlColumns
'
Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
sWFIWaterFromGen = Cells(4, 6)                  '[m3/h].
sWFIVolumeTank = Cells(7, 6)                    '[m3].
sWFIMinVolumeTank = Cells(11, 6)                '[m3].
sWFIStartVolumeTank = Cells(9, 6)               '[m3].
'
Sheets("10 - Report Page WFI Other Days").Select
Range("G4") = sWFIWaterFromGen
Range("G5") = sWFIVolumeTank
Range("G6") = sWFIMinVolumeTank
Range("G7") = sWFIStartVolumeTank
'
'Calculate the average and minimum Volume in the WFI Storage Tank [m3]:
Sheets("Sheet 21 - WFI Volume in Tank").Select
sAverage = 0
sMin = sWFIVolumeTank
iCount = 1
Do While iCount < 8640 + 1
    sTemp = Cells(iCount + 3, iDaySelected)
    If sTemp < sMin Then
        sMin = sTemp
    End If
    sAverage = sAverage + Cells(iCount + 3, iDaySelected)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8640
Sheets("10 - Report Page WFI Other Days").Select
Range("Q4") = sMin
Range("Q8") = sAverage
'
'Find the maximum Volume in WFI Tank [m3]:
Sheets("Sheet 21 - WFI Volume in Tank").Select
sMax = Cells(1 + 3, iDaySelected)
iCount = 1
Do While iCount < 8640 + 1
    sTemp = Cells(iCount + 3, iDaySelected)
```

```
    If sTemp > sMax Then
      sMax = sTemp
    End If
    iCount = iCount + 1
  Loop
  Sheets("10 - Report Page WFI Other Days").Select
  Range("Q5") = sMax
  '
  'Calculate the range of the Volume in the storage tank [m3]:
  Sheets("10 - Report Page WFI Other Days").Select
  Range("Q6") = sMax - sMin
  '
  'Calculate the average water flow from the WFI generation plant [m3/h]:
  Sheets("Sheet 20 - WFI Gen. to Tank").Select
  sAverage = 0
  iCount = 1
  Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, iDaySelected)
    iCount = iCount + 1
  Loop
  sAverage = sAverage / 8640
  Sheets("10 - Report Page WFI Other Days").Select
  Range("Q9") = sAverage
  '
  'Calculate the average offtake from the WFI loop [m3/h]:
  Sheets("Sheet 19 - Water Demand WFI").Select
  sAverage = 0
  iCount = 1
  Do While iCount < 8640 + 1
    sAverage = sAverage + Cells(iCount + 3, iDaySelected)
    iCount = iCount + 1
  Loop
  sAverage = sAverage / 8640
  Sheets("10 - Report Page WFI Other Days").Select
  Range("Q10") = sAverage
  '
  'Calculate the 90% percentile WFI Water Volume in Tank [m3]:
  Sheets("Sheet 21 - WFI Volume in Tank").Select
  iCount = 1
  Do While iCount < 8640 + 1
    sWFIVolumeSim(iCount) = Cells(iCount + 3, iDaySelected)
    iCount = iCount + 1
  Loop
  iCount = 2
  Do While iCount < 8640 + 1
    sTemp = sWFIVolumeSim(iCount)
    For iRowNumber = iCount - 1 To 1 Step -1
      If (sWFIVolumeSim(iRowNumber) <= sTemp) Then
        GoTo 30
      End If
      sWFIVolumeSim(iRowNumber + 1) = sWFIVolumeSim(iRowNumber)
    Next iRowNumber
    iRowNumber = 0
30      sWFIVolumeSim(iRowNumber + 1) = sTemp
    iCount = iCount + 1
  Loop
  iLocationOfPercentile = Int(90 / 100 * 8640)
  iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
  Sheets("10 - Report Page WFI Other Days").Select
  Range("Q12") = sWFIVolumeSim(iLocationOfPercentile)
  '
  'Calculate the 60% percentile WFI Water Volume in Tank [m3]:
  iLocationOfPercentile = Int(60 / 100 * 8640)
  iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
  Sheets("10 - Report Page WFI Other Days").Select
  Range("Q13") = sWFIVolumeSim(iLocationOfPercentile)
  '
  'Calculate the WFI Generation utilisation [-]:
  Sheets("Sheet 20 - WFI Gen. to Tank").Select
```

```
    iTemp = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sValue = Cells(iCount + 3, iDaySelected)
        If sValue <> 0 Then
            iTemp = iTemp + 1
        End If
        iCount = iCount + 1
    Loop
    Sheets("10 - Report Page WFI Other Days").Select
    Range("Q18") = iTemp / 8640
    '
    'Calculate the overall WFI Water Consumption from Loop excluding the tank [m3]:
    Sheets("Sheet 19 - Water Demand WFI").Select
    sWFISum = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sWFISum = sWFISum + Cells(iCount, iDaySelected) / 3600 * 10
        iCount = iCount + 1
    Loop
    Sheets("10 - Report Page WFI Other Days").Select
    Range("Q15") = sWFISum
    '
    'Calculate the overall WFI Water to Storage Tank [m3]:
    Sheets("Sheet 20 - WFI Gen. to Tank").Select
    sWFISum = 0
    iCount = 4
    Do While iCount < 8340 + 1
        sWFISum = sWFISum + Cells(iCount, iDaySelected) / 3600 * 10
        iCount = iCount + 1
    Loop
    Sheets("10 - Report Page WFI Other Days").Select
    Range("Q16") = sWFISum
    '
    Application.ScreenUpdating = True
    Sheets("10 - Report Page WFI Other Days").Select
    Range("B5").Select
    '
End Sub
'
'
'
Sub Analysis_DI_Other_Days()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine does the analysis of all data for of the DI data for the day as   @
    '@ selected by the user.                                                              @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIWaterFromGen = DI from Generation Plant [m3/s].
    ' sDIVolumeTank = Volume of DI storage tank [m3].
    ' sDIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sDIStartVolumeTank = Start volume in DI storage tank [m3].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
    ' iCount = Index for DoWhile Loop [-].
    ' iNCount = Calculation of DI generation utilisation [-].
    ' sDIVolume() = Volume in DI storage tank [-].
    ' sDIFlow() = DI flow to storage tank [-].
    ' sMin1, sMin2 = Calculation of minimum value [  ].
    ' sValue = Calculation of DI generation utilisation [-].
    ' sTemp = Temporary value [-].
    ' iLocationOfPercentile = For calculation of percentile.
    '
    'CALCULATED VALUES/OUTPUTS
    ' sMin = Minimum value.
    ' sMedian = Median value.
```

```
' sMax = Maximum value.
' sAverage = Average value.
' sDISum = Overall DI consumption [m3].
' s90Volume = 90% percentile of tank Volume [m3].
' sWFISum = Overall WFI consumption from loop excluding tank filling [m3].
'
Dim sDaySelected As String, Alpha As String
Dim sMin As Single, sMin1 As Single, sMin2 As Single, sValue As Single
Dim sMax As Single, sDIVolumeSim(8640) As Single
Dim sAverage As Single, s90Volume As Single, sTemp As Single
Dim sDISum As Single, sDISum1 As Single, sDISum2 As Single, sDIWaterFromGen As Single
Dim sDIVolumeTank As Single, sDIMinVolumeTank As Single, sDIStartVolumeTank As Single
Dim iCount As Integer, iNCount As Integer, iRowNumber As Integer, iDaySelected As Integer
Dim iLocationOfPercentile As Integer, iTemp As Integer
'
Application.ScreenUpdating = False
'
Sheets("7 - Report Page DI Other Days").Select
sDaySelected = Cells(10, 4)
iDaySelected = Cells(10, 4)
'
Select Case sDaySelected
Case 1
   Alpha = "A"
Case 2
   Alpha = "B"
Case 3
   Alpha = "C"
Case 4
   Alpha = "D"
Case 5
   Alpha = "E"
Case 6
   Alpha = "F"
Case 7
   Alpha = "G"
End Select
'
Sheets("7 - Report Page DI Other Days").Select
Range("A1:O1").Select
ActiveCell.FormulaR1C1 = "DI Results for Day " + sDaySelected
'
ActiveSheet.ChartObjects("Chart 3").Activate
ActiveChart.ChartTitle.Select
Selection.Characters.Text = "DI Water Volume in Storage Tank Day " _
+ sDaySelected + " [m3] "
Selection.AutoScaleFont = False
With Selection.Characters(Start:=1, Length:=100).Font
   .Name = "Arial"
   .FontStyle = "Bold"
   .Size = 16.5
   .Strikethrough = False
   .Superscript = False
   .Subscript = False
   .OutlineFont = False
   .Shadow = False
   .Underline = xlUnderlineStyleNone
   .ColorIndex = xlAutomatic
End With
Selection.AutoScaleFont = False
'
ActiveChart.PlotArea.Select
ActiveChart.SetSourceData Source:=Sheets("Sheet 17 - DI Volume in Tank").Range(Alpha _
+ "4:" + Alpha + "8430"), PlotBy:=xlColumns
ActiveChart.SeriesCollection.NewSeries
'
ActiveWindow.Visible = False
ActiveSheet.ChartObjects("Chart 4").Activate
ActiveChart.ChartTitle.Select
```

```
Selection.Characters.Text = _
    "DI Water Generation to Storage Tank -  Day " + sDaySelected + " [m3/s] "
Selection.AutoScaleFont = False
With Selection.Characters(Start:=1, Length:=100).Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 16.5
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Selection.AutoScaleFont = False
ActiveChart.SetSourceData Source:=Sheets("Sheet 18 - DI Gen. to Tank").Range( _
    Alpha + "4:" + Alpha + "8640"), PlotBy:=xlColumns
ActiveSheet.ChartObjects("Chart 2").Activate
ActiveChart.ChartTitle.Select
Selection.Characters.Text = _
    "DI Water Demand from Distribution System - Day " + sDaySelected + " [m3] "
Selection.AutoScaleFont = False
With Selection.Characters(Start:=1, Length:=100).Font
    .Name = "Arial"
    .FontStyle = "Bold"
    .Size = 15.75
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Selection.AutoScaleFont = False
'
ActiveChart.SetSourceData Source:=Sheets("Sheet 16 - Water Demand DI").Range(Alpha _
+ "4:" + Alpha + "8430"), PlotBy:=xlColumns
'
Sheets("Sheet 4 - DI Gen. & Tank Data").Select
sDIWaterFromGen = Cells(4, 6)                    '[m3/h].
sDIVolumeTank = Cells(7, 6)                      '[m3].
sDIMinVolumeTank = Cells(11, 6)                  '[m3].
sDIStartVolumeTank = Cells(9, 6)                 '[m3].
'
Sheets("7 - Report Page DI Other Days").Select
Range("G4") = sDIWaterFromGen
Range("G5") = sDIVolumeTank
Range("G6") = sDIStartVolumeTank
Range("G7") = sDIMinVolumeTank
'
'Calculate the average and minimum Volume in the DI Storage Tank [m3]:
Sheets("Sheet 17 - DI Volume in Tank").Select
sAverage = 0
sMin = sDIVolumeTank
iCount = 1
Do While iCount < 8640 + 1
    sTemp = Cells(iCount + 3, iDaySelected)
    If sMin1 < sTemp Then
        sMin = sTemp
    End If
    sAverage = sAverage + Cells(iCount + 3, iDaySelected)
    iCount = iCount + 1
Loop
sAverage = sAverage / 8643
Sheets("7 - Report Page DI Other Days").Select
Range("Q4") = sMin
Range("Q8") = sAverage
```

```
'
'Find the maximum Volume in DI Tank [m3]:
Sheets("Sheet 17 - DI Volume in Tank").Select
sTemp = Cells(1 + 3, iDaySelected)
iCount = 1
Do While iCount < 8640 + 1
   sMax = Cells(iCount + 3, iDaySelected)
   If sTemp > sMax Then
      sMax = sTemp
   End If
   iCount = iCount + 1
Loop
Sheets("7 - Report Page DI Other Days").Select
Range("Q5") = sMax
'
'Calculate the range of the Volume in the storage tank [m3]:
Sheets("7 - Report Page DI Other Days").Select
Range("Q6") = sMax - sMin
'
'Calculate average DI Water Loop Consumption m3/h]:
Sheets("Sheet 16 - Water Demand DI").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
   sAverage = sAverage + Cells(iCount + 3, iDaySelected)
   iCount = iCount + 1
Loop
sAverage = sAverage / 8640 * 3600
Sheets("7 - Report Page DI Other Days").Select
Range("Q9") = sAverage
'
'Calculate average flow from DI Generation plant [m3/h]:
Sheets("Sheet 18 - DI Gen. to Tank").Select
sAverage = 0
iCount = 1
Do While iCount < 8640 + 1
   sAverage = sAverage + Cells(iCount + 3, iDaySelected) / 3600 * 10
   iCount = iCount + 1
Loop
sAverage = sAverage / 8640 * 3600
Sheets("7 - Report Page DI Other Days").Select
Range("Q10") = sAverage
'
'Calculate the 90% percentile DI Water Volume in Tank [m3]:
Sheets("Sheet 17 - DI Volume in Tank").Select
iCount = 1
Do While iCount < 8640 + 1
   sDIVolumeSim(iCount) = Cells(iCount + 3, iDaySelected) / 3600 * 10
   iCount = iCount + 1
Loop
iCount = 2
Do While iCount < 8640 + 1
   sTemp = sDIVolumeSim(iCount)
   For iRowNumber = iCount - 1 To 1 Step -1
      If (sDIVolumeSim(iRowNumber) <= sTemp) Then
         GoTo 30
      End If
      sDIVolumeSim(iRowNumber + 1) = sDIVolumeSim(iRowNumber)
   Next iRowNumber
   iRowNumber = 0
30    sDIVolumeSim(iRowNumber + 1) = sTemp
   iCount = iCount + 1
Loop
iLocationOfPercentile = Int(90 / 100 * 8640)
iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
Sheets("7 - Report Page DI Other Days").Select
Range("Q12") = sDIVolumeSim(iLocationOfPercentile)
'
'Calculate the 60% percentile DI Water Volume in Tank [m3]:
```

```
    iLocationOfPercentile = Int(60 / 100 * 8640)
    iLocationOfPercentile = Application.Max(iLocationOfPercentile, 1)
    Sheets("7 - Report Page DI Other Days").Select
    Range("Q13") = sDIVolumeSim(iLocationOfPercentile)
    '
    'Calculate the DI Generation utilisation [-]:
    Sheets("Sheet 18 - DI Gen. to Tank").Select
    iTemp = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sValue = Cells(iCount + 3, 1)
        If sValue <> 0 Then
            iTemp = iTemp + 1
        End If
        iCount = iCount + 1
    Loop
    Sheets("7 - Report Page DI Other Days").Select
    Range("Q18") = iTemp / 8640
    '
    'Calculate the overall DI Water Consumption from Loop excluding the tank [m3]:
    Sheets("Sheet 16 - Water Demand DI").Select
    sDISum = 0
    iCount = 4
    Do While iCount < 8640 + 1
        sDISum = sDISum + Cells(iCount, iDaySelected)
        iCount = iCount + 1
    Loop
    Sheets("7 - Report Page DI Other Days").Select
    Range("Q15") = sDISum * 10
    '
    'Calculate the overall DI Water to Storage Tank [m3]:
    Sheets("Sheet 18 - DI Gen. to Tank").Select
    sDISum = 0
    iCount = 4
    Do While iCount < 8340 + 1
        sDISum = sDISum + Cells(iCount, iDaySelected)
        iCount = iCount + 1
    Loop
    Sheets("7 - Report Page DI Other Days").Select
    Range("Q16") = sDISum * 10
    '
    Application.ScreenUpdating = True
    Sheets("7 - Report Page DI Other Days").Select
    Range("B5").Select
    '
    End Sub
    '
    '
    '
Private Sub Analysis_of_Data_WFI_and_DI_all_Days()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine analyses the data for the WFI & DI systems.  @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIWaterFromGen = WFI water from generation [m3/h].
    ' sDIWaterFromGen = DI water from generation [m3/h].
    ' sWFIVolumeTank = Volume of WFI storage tank [m3].
    ' sDIVolumeTank = Volume of DI storage tank [m3].
    ' sWFIMinVolumeTank = Min. allowable volume of WFI tank [m3].
    ' sDIMinVolumeTank = Min. allowable volume of DI tank [m3].
    ' sWFIStartVolumeTank = Start volume in WFI storage tank [m3].
    ' sDIStartVolumeTank = Start volume in DI storage tank [m3].
    ' iNumberOfSimulatedDays = Days to be simulated [-].
    ' bOnlyOneLoop = Only the WFI loop will be simulated [-].
    '
    'INTERMEDIATE VARIABLES
    ' iRowNumber = Number of rows as per Input Table [-].
```

```
' iCount = Index for DoWhile Loop [-].
' iNCount = Calculation of WFI generation utilisation [-].
' sWFIVolume() = Volume in WFI storage tank [-].
' sValue = Calculation of WFI generation utilisation [-].
' sTemp = Temporary value [-].
' iLocationOfPercentile = For calculation of percentile.
' LCount = Index for DoWhile Loop [-].
' LRowNumber = Index for DoWhile Loop [-].
' iNCount = Index for DoWhile Loop [-].
' iNCount1 = Index for DoWhile Loop [-].
'
'CALCULATED sALUES/OUTPUTS
' sWFIVolumeSim = Calculate the 90% percentile WFI Water Volume in Tank [m3].
' sDIVolumeSim = Calculate the 90% percentile DI Water Volume in Tank [m3].
' sMin = Minimum value.
' sMax = Maximum value.
' sAverage = Average value.
' s90Volume = 90% percentile of tank Volume [m3].
' sWFISum = Overall WFI consumption from loop excluding tank filling [m3].
' LLocationOfPercentile = Location of percentile [-].
' LUtilcount = Calculates the utilisation of equipment [-].
'
Dim sWFIWaterFromGen As Single, sWFIVolumeTank As Single, sWFIMinVolumeTank As Single
Dim sWFIStartVolumeTank As Single, sDIWaterFromGen As Single, sDIVolumeTank As Single
Dim sDIMinVolumeTank As Single, sDIStartVolumeTank As Single
Dim sMin As Single, sAverage As Single, sValue As Single, sTemp As Single
Dim iCount As Integer, iLocationOfPercentile As Integer
Dim LCount As Long, LRowNumber As Long, LLocationOfPercentile As Long, LUtilcount As Long
Dim iNCount As Integer, iNCount1 As Integer, iNumberOfSimulatedDays As Integer
Dim sWFIVolumeSim() As Single, sWFILoopConsumptionSim() As Single
Dim sDIVolumeSim() As Single, sDILoopConsumptionSim() As Single
Dim bOnlyWFILoop As Boolean
'
ReDim sWFIVolumeSim(61000)
ReDim sDIVolumeSim(61000)
'
Application.ScreenUpdating = False
'
Sheets("Frontpage - WFI Loop Input Data").Select
bOnlyWFILoop = Cells(19, 2)
iNumberOfSimulatedDays = Cells(14, 2)
'
If bOnlyWFILoop = False Then   'Simulate DI and WFI System simultaneously.
   'First Analysis for the WFI system:
   Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
   sWFIWaterFromGen = Cells(4, 6)                '[m3/h].
   sWFIVolumeTank = Cells(7, 6)                  '[m3].
   sWFIMinVolumeTank = Cells(11, 6)              '[m3].
   sWFIStartVolumeTank = Cells(9, 6)             '[m3].
   '
   Sheets("11 - Report Page WFI All Days").Select
   Range("G4") = sWFIWaterFromGen
   Range("G5") = sWFIVolumeTank
   Range("G6") = sWFIMinVolumeTank
   Range("G7") = sWFIStartVolumeTank
   '
   'Calculate the average and minimum Volume in the WFI Storage Tank [m3]:
   Sheets("Sheet 21 - WFI Volume in Tank").Select
   sAverage = 0
   sMin = sWFIVolumeTank
   iNCount = 1
   Do While iNCount < iNumberOfSimulatedDays + 1
      iCount = 1
      Do While iCount < 8640 + 1
         sTemp = Cells(iCount + 3, iNCount)
         If sTemp < sMin Then
            sMin = sTemp
         End If
         sAverage = sAverage + Cells(iCount + 3, iNCount)
```

```
            iCount = iCount + 1
        Loop
        iNCount = iNCount + 1
    Loop
    sAverage = sAverage / (CSng(8640) * CSng(iNumberOfSimulatedDays))
    Sheets("11 - Report Page WFI All Days").Select
    Range("O4") = sMin
    Range("O5") = sAverage
    '
    'Calculate the 90% percentile WFI Water Volume in Tank [m3]:
    Sheets("Sheet 21 - WFI Volume in Tank").Select
    iCount = 1
    iNCount = 1
    Do While iNCount < iNumberOfSimulatedDays + 1
        Do While iCount < 8640 + 1
            sWFIVolumeSim(iCount + 8640 * (iNCount - 1)) = Cells(iCount + 3, iNCount)
            iCount = iCount + 1
        Loop
        iNCount = iNCount + 1
    Loop
    '
    'Sort algorithm for sWFIVolume(iCount):
    LCount = 2
    Do While LCount < CLng(8640) * iNumberOfSimulatedDays
        sTemp = sWFIVolumeSim(LCount)
        For LRowNumber = LCount - 1 To 1 Step -1
            If (sWFIVolumeSim(LRowNumber) <= sTemp) Then
                GoTo 10
            End If
            sWFIVolumeSim(LRowNumber + 1) = sWFIVolumeSim(LRowNumber)
        Next LRowNumber
        LRowNumber = 0
10      sWFIVolumeSim(LRowNumber + 1) = sTemp
        LCount = LCount + 1
    Loop
    LLocationOfPercentile = Int(90 / 100 * CLng(8640) * iNumberOfSimulatedDays)
    LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
    Sheets("11 - Report Page WFI All Days").Select
    Range("O6") = sWFIVolumeSim(LLocationOfPercentile)
    '
    'Calculate the WFI Generation utilisation [-]:
    Sheets("Sheet 20 - WFI Gen. to Tank").Select
    iNCount = 1
    LUtilcount = 0
    Do While iNCount < iNumberOfSimulatedDays + 1
        iCount = 4
        Do While iCount < 8640 + 1
            sValue = Cells(iCount, 1)
            If sValue <> 0 Then
                LUtilcount = LUtilcount + 1
            End If
            iCount = iCount + 1
        Loop
        iNCount = iNCount + 1
    Loop
    Sheets("11 - Report Page WFI All Days").Select
    Range("O7") = LUtilcount / (CLng(8640) * iNumberOfSimulatedDays)
    Range("A5").Select
    '
    'Second Analysis for the DI system:
    Sheets("Sheet 4 - DI Gen. & Tank Data").Select
    sDIWaterFromGen = Cells(4, 6)              '[m3/h].
    sDIVolumeTank = Cells(7, 6)                '[m3].
    sDIMinVolumeTank = Cells(11, 6)            '[m3].
    sDIStartVolumeTank = Cells(9, 6)           '[m3].
    '
    Sheets("8 - Report Page DI All Days").Select
    Range("G4") = sDIWaterFromGen
    Range("G5") = sDIVolumeTank
```

```
Range("G6") = sDIMinVolumeTank
Range("G7") = sDIStartVolumeTank
'
'Calculate the average and minimum Volume in the DI Storage Tank [m3]:
'Read in Data for Volume in DI Tank:
Sheets("Sheet 17 - DI Volume in Tank").Select
sAverage = 0
sMin = sDIVolumeTank
iNCount = 1
Do While iNCount < iNumberOfSimulatedDays + 1
   iCount = 1
   Do While iCount < 8640 + 1
      sTemp = Cells(iCount + 3, iNCount)
      If sTemp < sMin Then
         sMin = sTemp
      End If
      sAverage = sAverage + Cells(iCount + 3, 1)
      iCount = iCount + 1
   Loop
   iNCount = iNCount + 1
Loop
sAverage = sAverage / (CSng(8643) * CSng(iNumberOfSimulatedDays))
Sheets("8 - Report Page DI All Days").Select
Range("O4") = sMin
Range("O5") = sAverage
'
'Calculate the 90% percentile DI Water Volume in Tank [m3]:
Sheets("Sheet 17 - DI Volume in Tank").Select
iCount = 1
iNCount = 1
Do While iNCount < iNumberOfSimulatedDays + 1
   Do While iCount < 8640 + 1
      sDIVolumeSim(iCount + 8640 * (iNCount - 1)) = Cells(iCount + 3, iNCount)
      iCount = iCount + 1
   Loop
   iNCount = iNCount + 1
Loop
'Sort algorithm for sDIVolumeSim(iCount):
LCount = 2
Do While LCount < CLng(8640) * iNumberOfSimulatedDays
   sTemp = sDIVolumeSim(iCount)
   For LRowNumber = LCount - 1 To 1 Step -1
      If (sDIVolumeSim(LRowNumber) <= sTemp) Then
         GoTo 20
      End If
      sDIVolumeSim(LRowNumber + 1) = sDIVolumeSim(LRowNumber)
   Next LRowNumber
   LRowNumber = 0
20    sDIVolumeSim(LRowNumber + 1) = sTemp
   LCount = LCount + 1
Loop
LLocationOfPercentile = Int(90 / 100 * CLng(8640) * iNumberOfSimulatedDays)
LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
Sheets("8 - Report Page DI All Days").Select
Range("O6") = sDIVolumeSim(LLocationOfPercentile)
'
'Calculate the DI Generation utilisation [-]:
Sheets("Sheet 18 - DI Gen. to Tank").Select
iNCount = 1
LUtilcount = 0
Do While iNCount < iNumberOfSimulatedDays + 1
   iCount = 4
   Do While iCount < 8640 + 1
      sValue = Cells(iCount, 1)
      If sValue <> 0 Then
         LUtilcount = LUtilcount + 1
      End If
      iCount = iCount + 1
   Loop
```

```
        iNCount = iNCount + 1
      Loop
      Sheets("8 - Report Page DI All Days").Select
      Range("O7") = LUtilcount / (CLng(8640) * iNumberOfSimulatedDays)
      Range("A5").Select
      'End analysis DI and WFI loop.
      '
    Else                    'Simulate the WFI System only.
      'Read in Data:
      'First Analysis for WFI:
      Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
      sWFIWaterFromGen = Cells(4, 6)              '[m3/h].
      sWFIVolumeTank = Cells(7, 6)                '[m3].
      sWFIMinVolumeTank = Cells(11, 6)            '[m3].
      sWFIStartVolumeTank = Cells(9, 6)           '[m3].
      '
      Sheets("11 - Report Page WFI All Days").Select
      Range("G4") = sWFIWaterFromGen
      Range("G5") = sWFIVolumeTank
      Range("G6") = sWFIMinVolumeTank
      Range("G7") = sWFIStartVolumeTank
      '
      'Calculate the average and minimum Volume in the WFI Storage Tank [m3]:
      Sheets("Sheet 21 - WFI Volume in Tank").Select
      sAverage = 0
      sMin = sWFIVolumeTank
      iNCount = 1
      Do While iNCount < iNumberOfSimulatedDays + 1
        iCount = 1
        Do While iCount < 8640 + 1
          sTemp = Cells(iCount + 3, iNCount)
          If sTemp < sMin Then
            sMin = sTemp
          End If
          sAverage = sAverage + Cells(iCount + 3, iNCount)
          iCount = iCount + 1
        Loop
        iNCount = iNCount + 1
      Loop
      sAverage = sAverage / (CSng(8640) * CSng(iNumberOfSimulatedDays))
      Sheets("11 - Report Page WFI All Days").Select
      Range("O4") = sMin
      Range("O5") = sAverage
      '
      'Calculate the 90% percentile WFI Water Volume in Tank [m3]:
      Sheets("Sheet 21 - WFI Volume in Tank").Select
      iCount = 1
      iNCount = 1
      Do While iNCount < iNumberOfSimulatedDays + 1
        Do While iCount < 8640 + 1
          sWFIVolumeSim(iCount + 8640 * (iNCount - 1)) = Cells(iCount + 3, iNCount)
          iCount = iCount + 1
        Loop
        iNCount = iNCount + 1
      Loop
      '
      'Sort algorithm for sWFIVolume(iCount):
      LCount = 2
      Do While LCount < CLng(8640) * iNumberOfSimulatedDays
        sTemp = sWFIVolumeSim(LCount)
        For LRowNumber = LCount - 1 To 1 Step -1
          If (sWFIVolumeSim(LRowNumber) <= sTemp) Then
            GoTo 30
          End If
          sWFIVolumeSim(LRowNumber + 1) = sWFIVolumeSim(LRowNumber)
        Next LRowNumber
        LRowNumber = 0
30      sWFIVolumeSim(LRowNumber + 1) = sTemp
        LCount = LCount + 1
```

```
        Loop
        LLocationOfPercentile = Int(90 / 100 * CLng(8640) * iNumberOfSimulatedDays)
        LLocationOfPercentile = Application.Max(LLocationOfPercentile, 1)
        Sheets("11 - Report Page WFI All Days").Select
        Range("O6") = sWFIVolumeSim(LLocationOfPercentile)
        '
        'Calculate the WFI Generation utilisation [-]:
        Sheets("Sheet 20 - WFI Gen. to Tank").Select
        iNCount = 1
        LUtilcount = 0
        Do While iNCount < iNumberOfSimulatedDays + 1
            iCount = 4
            Do While iCount < 8640 + 1
                sValue = Cells(iCount, 1)
                If sValue <> 0 Then
                    LUtilcount = LUtilcount + 1
                End If
                iCount = iCount + 1
            Loop
            iNCount = iNCount + 1
        Loop
        Sheets("11 - Report Page WFI All Days").Select
        Range("O7") = LUtilcount / (CLng(8640) * iNumberOfSimulatedDays)
        Range("A5").Select
    End If
    '
End Sub
'
'
'
Sub Time_WFI_Low()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This Subroutine calculates how often the volume in the              @
    '@ WFI water storage tank is below a volume over the time horizon.     @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIVolumeNotToBeBelow = Volume in Tank shall not be below this Volume [m3].
    ' sWFIVolume = Simulated Volume of tank [m3].
    '
    'INTERMEDIATE VARIABLES
    ' iCount = Index for Do While-Loop [-].
    ' iRow = Index for Do While-Loop [-].
    ' LNumberOfVolumesBelow = Predicted number of Volumes below the user input [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sTimeBelowWFI = Estimated time that Volume in tank is below user input [-].
    '
    Dim sWFIVolumeNotToBeBelow As Single, sWFIVolume As Single
    Dim sTimeBelowWFI As Single
    Dim iCount As Integer, iRow As Integer, LNumberOfVolumesWFIBelow As Integer
    '
    Application.ScreenUpdating = False
    '
    'Read in Data:
    Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
    sWFIVolumeNotToBeBelow = Cells(22, 6)
    '
    Sheets("Sheet 21 - WFI Volume in Tank").Select
    iRow = 1
    iCount = 1
    Do While iCount < iDayData + 1
        Do While iRow < 8640 + 1
            sWFIVolume = Cells(iRow + 3, iCount)
            If sWFIVolume < sWFIVolumeNotToBeBelow Then
                LNumberOfVolumesWFIBelow = LNumberOfVolumesWFIBelow + 1
            End If
            iRow = iRow + 1
```

```
        Loop
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    sTimeBelowWFI = LNumberOfVolumesWFIBelow / 6    '6 as data is every 10 second only.
    Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
    Cells(27, 6) = sTimeBelowWFI
    '
    Application.ScreenUpdating = True
    '
End Sub
'
'
'
Sub Time_DI_Low()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This Subroutine calculates how often the volume in the                @
    '@ DI water storage tank is below a volume over the time horizon.       @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sDIVolumeNotToBeBelow = Volume in Tank shall not be below this Volume [m3].
    ' sDIVolume = Simulated Volume of tank [m3].
    '
    'INTERMEDIATE VARIABLES
    ' iCount = Index for Do While-Loop [-].
    ' iRow = Index for Do While-Loop [-].
    ' LNumberOfVolumesBelow = Predicted number of Volumes below the user input [-].
    '
    'CALCULATED VALUES/OUTPUTS
    ' sTimeBelowDI = Estimated time that Volume in tank is below user input [-].
    '
    Dim sDIVolumeNotToBeBelow As Single, sDIVolume As Single
    Dim sTimeBelowDI As Single
    Dim iCount As Integer, iRow As Integer, LNumberOfVolumesDIBelow As Integer
    '
    Application.ScreenUpdating = False
    '
    'Read in Data:
    Sheets("Sheet 4 - DI Gen. & Tank Data").Select
    sDIVolumeNotToBeBelow = Cells(22, 6)
    '
    Sheets("Sheet 17 - DI Volume in Tank").Select
    iRow = 1
    iCount = 1
    Do While iCount < iDayData + 1
        Do While iRow < 8640 + 1
            sDIVolume = Cells(iRow + 3, iCount)
            If sDIVolume < sDIVolumeNotToBeBelow Then
                LNumberOfVolumesDIBelow = LNumberOfVolumesDIBelow + 1
            End If
            iRow = iRow + 1
        Loop
        iCount = iCount + 1
    Loop
    'End Read in Data.
    '
    sTimeBelowDI = LNumberOfVolumesDIBelow / 6    '6 as data is every 10 second only.
    Sheets("Sheet 4 - DI Gen. & Tank Data").Select
    Cells(27, 6) = sTimeBelowDI
    '
    Application.ScreenUpdating = True
    '
End Sub
'
'
'
```

```
Sub Delete_All_Input_Data()
    '
    '@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine deletes all input data.          @
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Dim Msg, Style, Title, Response
    Msg = "This action will delete all data on this Event Table. Do you wish" + _
    " to continue? Press 'Yes' to continue or 'No' to cancel this action."
    Style = vbYesNo + vbDefaultButton2
    Title = "A critical Message"
    Response = MsgBox(Msg, Style, Title)
    If Response = vbNo Then
        End
    End If
    Range("G5:IN204").Select
    Selection.ClearContents
    Range("A2").Select
    '
End Sub
'
'
'
Sub Progress_Indicator(Progress)
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine displays the process indicator .  @
    '@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    SimulationProgress.RedProgressLabel.Width = Progress
    DoEvents
    '
End Sub
'
'
'
Sub Cell_Select_A4()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine moves the active cell to A4.   @
    '@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Sheets("Sheet 16 - Water Demand DI").Select
    Range("A1").Select
    Sheets("Sheet 19 - Water Demand WFI").Select
    Range("A1").Select
    Sheets("Sheet 17 - DI Volume in Tank").Select
    Range("A1").Select
    Sheets("Sheet 21 - WFI Volume in Tank").Select
    Range("A1").Select
    Sheets("Sheet 20 - WFI Gen. to Tank").Select
    Range("A1").Select
    Sheets("Sheet 18 - DI Gen. to Tank").Select
    Range("A1").Select
    Sheets("Sheet 6 - Report Page DI Day 1").Select
    Range("A9").Select
    Sheets("8 - Report Page DI All Days").Select
    Range("A9").Select
    Sheets("Sheet 9 - Report Page WFI Day 1").Select
    Range("A9").Select
    Sheets("11 - Report Page WFI All Days").Select
    Range("A4").Select
    Sheets("Frontpage - WFI Loop Input Data").Select
    Range("A4").Select
    Sheets("7 - Report Page DI Other Days").Select
    Range("A9").Select
    Sheets("8 - Report Page DI All Days").Select
    Range("A9").Select
    Sheets("10 - Report Page WFI Other Days").Select
```

```
    Range("A9").Select
    Sheets("11 - Report Page WFI All Days").Select
    Range("A9").Select
    '
End Sub
'
'
'
Sub Delete_Output_Data()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This subroutine deletes all output data from the spreadsheets.     @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Application.ScreenUpdating = False
    Sheets("Sheet 16 - Water Demand DI").Select
    Rows("6:65536").Select
    Selection.ClearContents
    Range("A1").Select
    Sheets("Sheet 17 - DI Volume in Tank").Select
    Rows("6:65536").Select
    Selection.ClearContents
    Range("A1").Select
    Sheets("Sheet 18 - DI Gen. to Tank").Select
    Rows("6:65536").Select
    Selection.ClearContents
    Range("A1").Select
    Sheets("Sheet 19 - Water Demand WFI").Select
    Rows("6:65536").Select
    Selection.ClearContents
    Range("A1").Select
    Sheets("Sheet 20 - WFI Gen. to Tank").Select
    Rows("6:65536").Select
    Selection.ClearContents
    Range("A1").Select
    Sheets("Sheet 21 - WFI Volume in Tank").Select
    Rows("6:65536").Select
    Selection.ClearContents
    Range("A1").Select
    Sheets("Frontpage - WFI Loop Input Data").Select
    Range("A5").Select
    Application.ScreenUpdating = True
    '
End Sub
'
'
'
Sub Min_Allowable_Volume_And_Pump_Flowrate()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This routine writes the appropriate "minimum allowable Tank Volume" values   @
    '@ and WFI,DI Pump Flowrates to all appropriate sheets.                          @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    'INPUTS
    ' sWFIMinVolumeTank = Global variable [-].
    ' sDIMinVolumeTank = Min. allowable volume of tank [m3].
    ' sWFIPump = WFI pump flowrate [m3/h].
    ' sDIPump = DI pump flowrate [m3/h].
    ' iNumberofRepeats = Global variable [-].
    '
    'INTERMEDIATE VARIABLES
    ' LRepeats = Index for DoWhile Loop [-].
    ' LRow = Index for DoWhile Loop [-].
    '
    Dim LRepeats As Long, LRow As Long
    Dim sDIMinVolumeTank As Single, sDIPump As Single, sWFIPump As Single
    '
    'Read data in:
```

```
    Sheets("Sheet 4 - DI Gen. & Tank Data").Select
    sDIMinVolumeTank = Cells(11, 6)                 '[m3].
    sDIPump = Cells(14, 6)                          '[m3/h].
    '
    Sheets("Sheet 5 - WFI Gen. & Tank Data").Select
    sWFIPump = Cells(17, 6)                          '[m3/h].
    sWFIMinVolumeTank = Cells(11, 6)                 '[m3].
    '
    'Write minimum volume and Pump Flowrate to the WFI & DI Repeat Sheets:
    LRow = 1
    Do While LRow < 8640
       '
       Sheets("Sheet 19 - Water Demand WFI").Select
       Cells(LRow + 3, 16) = sWFIPump
       Cells(LRow + 3, 17) = sWFIMinVolumeTank
       '
       LRow = LRow + 1
    Loop
    '
    If bOnlyWFILoop = False Then
       LRow = 1
       Do While LRow < 8640
          '
          If bOnlyWFILoop = False Then
             '
             Sheets("Sheet 16 - Water Demand DI").Select
             Cells(LRow + 3, 16) = sDIPump
             Cells(LRow + 3, 17) = sDIMinVolumeTank
             '
          End If
       Loop
    End If
End Sub
'
'
'
Sub Remove_Old_Data()
    '
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '@ This routine removes all old data still residing on the spreadsheet   @
    '@ before starting a new calculation.                                    @
    '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    '
    Sheets("Sheet 16 - Water Demand DI").Select
    Range("A4:IV65536").Select
    Selection.ClearContents
    Range("A4").Select
    '
    Sheets("Sheet 19 - Water Demand WFI").Select
    Range("A4:IV65536").Select
    Selection.ClearContents
    Range("A4").Select
    '
    Sheets("Sheet 17 - DI Volume in Tank").Select
    Range("A4:IV65536").Select
    Selection.ClearContents
    Range("A4").Select
    '
    Sheets("Sheet 21 - WFI Volume in Tank").Select
    Range("A4:IV65536").Select
    Selection.ClearContents
    Range("A4").Select
    '
    Sheets("Sheet 20 - WFI Gen. to Tank").Select
    Range("A4:IV65536").Select
    Selection.ClearContents
    Range("A4").Select
    '
    Sheets("Sheet 18 - DI Gen. to Tank").Select
```

```
Range("A4:IV65536").Select
Selection.ClearContents
Range("A4").Select
'
Sheets("Sheet 6 - Report Page DI Day 1").Select
Range("A2").Select
Selection.ClearContents
Range("A8").Select
Selection.ClearContents
Range("G4:G7").Select
Selection.ClearContents
Range("G9").Select
Selection.ClearContents
Range("Q4:Q22").Select
Selection.ClearContents
Range("A2").Select
'
Sheets("8 - Report Page DI All Days").Select
Range("A2").Select
Selection.ClearContents
Range("G4:G7").Select
Selection.ClearContents
Range("O4:O8").Select
Selection.ClearContents
Range("A2").Select
'
Sheets("Sheet 9 - Report Page WFI Day 1").Select
Range("A2").Select
Selection.ClearContents
Range("A8").Select
Selection.ClearContents
Range("G4:G7").Select
Selection.ClearContents
Range("G10").Select
Selection.ClearContents
Range("Q4:Q20").Select
Selection.ClearContents
Range("A2").Select
'
Sheets("11 - Report Page WFI All Days").Select
Range("A2").Select
Selection.ClearContents
Range("G4:G7").Select
Selection.ClearContents
Range("O4:O8").Select
Selection.ClearContents
Range("A2").Select
'
Sheets("Frontpage - WFI Loop Input Data").Select
Range("B42").Select
Selection.ClearContents
Range("A4").Select
'
Sheets("7 - Report Page DI Other Days").Select
Range("A2").Select
Selection.ClearContents
Range("G4:G7").Select
Selection.ClearContents
Range("Q4:Q20").Select
Selection.ClearContents
Range("A2").Select
'
Sheets("10 - Report Page WFI Other Days").Select
Range("G4:G8").Select
Selection.ClearContents
Range("Q4:Q20").Select
Selection.ClearContents
Range("A2").Select
'
```

```
    sProgress = sProgress + 1 * sIncrement
    Call Progress_Indicator(sProgress)
    '
End Sub
'
'
'
'
'
Function Pump_Analysis(Col As Integer, sPump As Single)
    '
    Dim i As Integer
    Dim sWFIWaterDemand(8650) As Single
    Dim sExceedPump As Single
    '
    i = 1
    sExceedPump = 0
    '
    Do While i < 8640 + 1
        '
        sWFIWaterDemand(i) = Cells(i + 3, Col)       'Row, Column
        '
        If sWFIWaterDemand(i) > sPump Then
            sExceedPump = sExceedPump + 1
        End If
        i = i + 1
    Loop
    '
    sExceedPump = sExceedPump * 10 / 60     '[min]
    '
    Pump_Analysis = sExceedPump
    '
End Function
'
'
'
'                                   END OF PROGRAM - FUZZY PURIFIED WATER SIMULATION
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

Appendix 4 Difficulties with Excel as a Programming Platform

At the outset of this research, it was assumed that the in-built functions of Excel such as the tools for statistical analysis and the statistical distributions would greatly reduce the programming effort and potential errors while programming a simulation model within Excel. But the literature review revealed many papers, which reported on the dangers of uncritical use of the following in-built Excel functions:

1. The random number generator (RNG) is of unknown quality [168].
2. The statistical distributions are inaccurate [169].
3. The statistical analysis tools are incorrect [170].
4. The graphical capabilities are poor [171].

Potentially any one of these four deficiencies may compromise the use of Excel as the simulation platform for a stochastic simulation. The latter two could compromise the use of Excel for a fuzzy logic simulation. In the following these four alleged deficiencies and the workarounds are described in more detail.

Random Number Generator

A RNG or in other words a suitable uniform distribution is a requirement for a stochastic simulation [2, 41, 168]. Unfortunately, the RNG as supplied within Excel is not a suitable generator, because Microsoft incorrectly states that this generator is the Wichmann-Hill (WH) random generator [168]. As a result, the RNG included in Excel is a generator of unknown performance in terms of repeatability and randomness rendering it unusable for a stochastic simulation according to McCullough [168].

In view of this, the in-built Excel RNG was replaced with a VBA version of the WH RNG [172] as programmed by Steele and Douglas [173] (see program of the stochastic simulation included in Appendix 2).

Stochastic Functions

Further to the uniform (RNG) distribution, other in-built probabilistic distributions of Excel are numerically inaccurate [169, 174]. These distributions include the Binominal, Poisson, inverse standard Normal, inverse Beta, inverse Student's t and the inverse F [169]. The software developed uses the cumulative Normal and Beta only, both of which are, according to Knüsel [174], correct. Therefore, in summary, the cumulative Normal, Beta and the replaced version of the uniform (RNG) distributions are safe to use in the context of this work.

Statistical Analysis Functions

Pottel [170] and Siminoff [175] from their studies into the accuracy of the in-built Excel functions conclude that many of the statistical functions included in the Excel spreadsheet are not suitable for serious statistical analysis and should therefore not be used. For instance the percentiles computation in Excel is, according to Pottel [170], incorrect. Therefore, code for the computation of the percentile had to be written (see programs included in Appendix 2 & 3).

This dissertation adopts the percentile definition from Mendenhall and Sincich [91]:

> "To find the p^{th} percentile, calculate the quantity $i=p(n+1)/100$ and round to the nearest integer. The measurement with this rank, denoted y_n is the p^{th} percentile."

Furthermore the large numbers of data points generated by the simulations exceed the number of inputs the relevant point estimators included in Excel can manipulate, requiring that code for the computation of the average, minimum and maximum had to be written.

Graphical Output

Excel is equipped with graphical capabilities, which are very useful for displaying the outputs of the various simulations. The graphical capabilities of Excel are, however, limited [171]. In particular, these limitations demand a reduction in resolution of the graphical displays.

In order to display a graph in Excel, the data must first be stored on the spreadsheet. The maximum row number of Excel is 65,536 rows, while the integration of the storage tank mass balance gives 86,400 data points per simulated day. In other words, Excel cannot store all the data points of a simulated day on one row on the spreadsheet. With some additional VBA programming effort it is possible to store the data points over two adjacent rows. For three reasons this was not done. First, the volume in a WFI storage tank changes slowly and therefore a one second resolution is not necessary. Second, the program would become more complicated should the data be stored on more than one row. Therefore only every 10^{th} data point is stored on the spreadsheet, reducing the graphical and analytical resolution of the one day simulation to 10 seconds. And third, the Excel graphs can only display 32,000 data points.

Appendix 5 Distributions used in the Stochastic Model

In this appendix the three stochastic distributions, which are used in the proposed stochastic models of the DI/WFI system, namely the uniform, Normal and Beta distribution are briefly introduced. Many other stochastic distributions are described in the literature and may be included in the proposed stochastic model of a HPW if desirable. Two references containing many more statistical distributions are Bury [45] and Evans et al. [46].

Probability density functions (PDF) provide the well-known statistical description of the behaviour of a random variable. A PDF of a continuous random variable X is a function with the following three parameters:

1. $f(X) \geq 0$ (Probability shall be greater or equal to zero)
2. $\int_{-\infty}^{+\infty} f(X) = 1$ (Area under PDF is 1.)
3. $P(a \leq X \leq b) = \int_{a}^{b} f(x)dx$ (The area under the PDF between two points a and b is the probability that the random variable X lies between a and b).

(App. 5.1)

A cumulative probability function (CDF) is defined as the integral of the PDF for each X as below:

$$F(X) = \int_0^X f(X)dX \qquad \text{(App. 5.2)}$$

The PDF of the continuous **uniform** distribution is defined as [91]:

$$U(x \mid a,b) = \begin{cases} \dfrac{1}{b-a}, & \text{if} \quad x \in [a,b] \\ 0, & \text{otherwise} \end{cases} \qquad \text{(App. 5.3)}$$

The uniform distribution specifies that all values in the interval (a, b) are equally likely as shown in Figure App. 5.1.

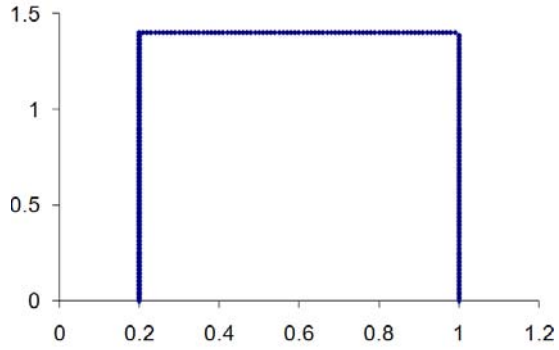


Figure App. 5.1 PDF of the uniform distribution

The CDF of the uniform distribution is defined in a closed form as:

$$F(X \mid a,b) = \frac{X-a}{b-a} \qquad \text{(App. 5.4)}$$

The PDF of the **Normal** or **Gaussian** distribution [91] for various standard deviations σ and mean μ is shown in Figure App. 5.2. This distribution is a continuous, bell-shape, symmetric distribution. It is one of the most important distributions in statistics as a surprising number of phenomena can be described by it [91]. Measurement data, for instance, is often assumed to be Normal [28]. The Normal distribution is defined by:

$$f(X) = \frac{1}{\sqrt{2\pi\pi}} \exp\left(\frac{-(X-\mu)^2}{2\sigma^2}\right) \quad -\infty < X < \infty \qquad \text{(App. 5.5)}$$

with:

standard deviation $\sigma = \sqrt{\frac{\sum (X-\mu)^2}{N}}$ and

mean: $\mu = \frac{\sum X}{N}$

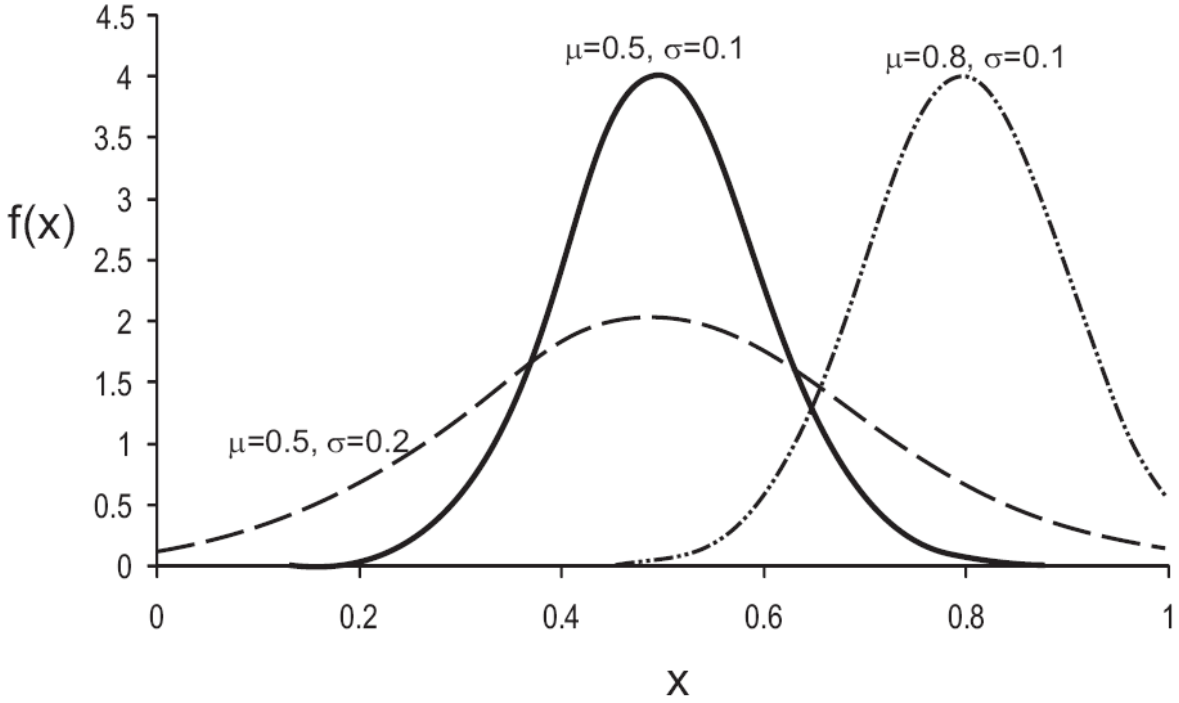


Figure App. 5.2 PDF of various Normal distributions

The CDF of the Normal distribution has no closed form solutions; only approximations exist.

The **Beta** distribution [91] is a finitely bounded distribution and has two shape parameters α and β. The PDF of the Beta distribution is given by [91]:

$$f(X) = \begin{cases} \dfrac{X^{\alpha-1}(1-X)^{\beta-1}}{B(\alpha,\beta)} & \text{if } 0 \le X \le 1; \alpha > 0; \beta > 0 \\ 0 & \text{elsewhere} \end{cases} \qquad \text{(App. 5.6)}$$

with

$$B(\alpha,\beta) = \int_0^1 X^{\alpha-1}(1-X)^{\beta-1}\,dX = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

with

$$\Gamma(\alpha) = \int_0^{\infty} \mathrm{X}^{\alpha\text{-}1} e^{-\mathrm{X}} d\mathrm{X}$$

Four plots of the Beta PDF distribution with various α and β are given in Figure App. 5.3.

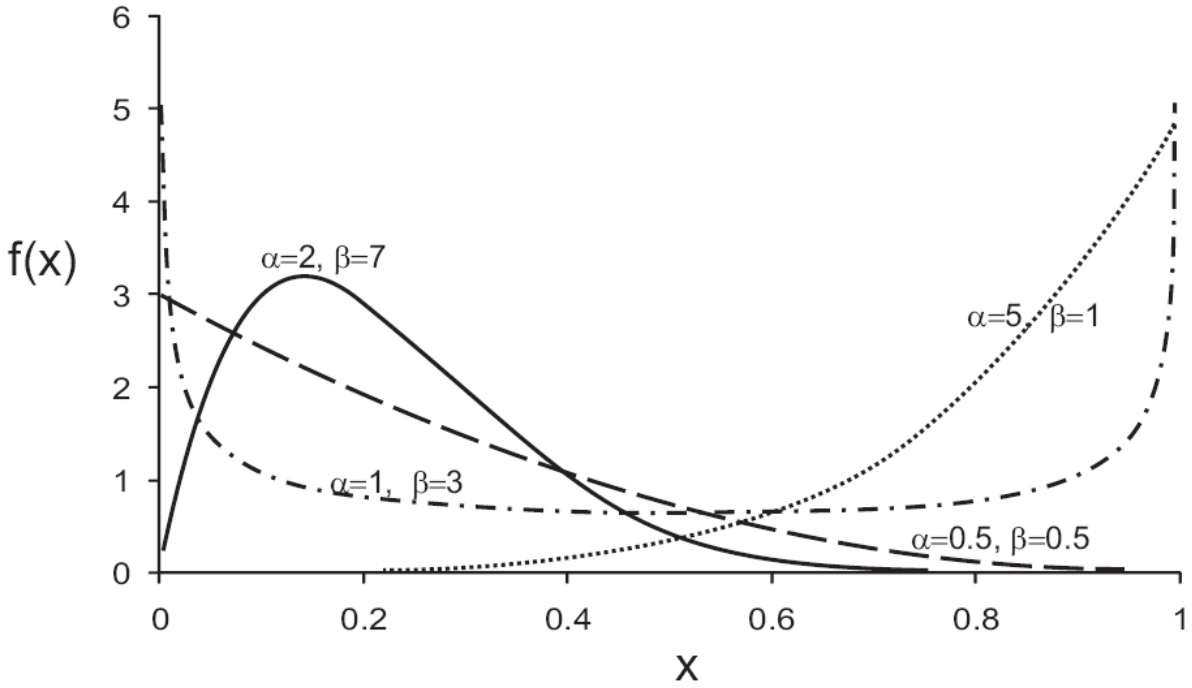


Figure App. 5.3 PDF of various Beta distributions

The CDF of the Beta distribution has no closed form solutions for all α and β; only approximations exist.

Appendix 6 Input Data Sets for the Deterministic Simulations

This appendix contains the input data used for the various deterministic DI/WFI simulations.

WFI Water Offtake Input Data

No.	Tag No.	Description of Offtake	Water Offtake [m^3/hr]	1 Max. Uncertainty [%]	1 Description of Uncertainty	1 Start Time (Tap Open) [24 hour clock, hours:min]	1 End Time (Tap Closed) [24 hour clock, hours:min]	1 Max.Time Uncertainty Time Open [min]	1 Description of Time Uncertainty	2 Start Time (Tap Open) [24 hour clock, hours:min]	2 End Time (Tap Closed) [24 hour clock, hours:min]	2 Max.Time Uncertainty Time Open [min]	2 Description of Time Uncertainty	3 Start Time (Tap Open) [24 hour clock, hours:min]	3 End Time (Tap Closed) [24 hour clock, hours:min]	3 Max.Time Uncertainty Time Open [min]	3 Description of Time Uncertainty	4 Start Time (Tap Open) [24 hour clock, hours:min]	4 End Time (Tap Closed) [24 hour clock, hours:min]
1		**ORIGINAL PROCESS DATA**																	
2		**1. BULK PROCESSING SUITE**																	
3	T-100	Pre-rinse	1.5			2:00	2:05			10:00	10:35			18:10	18:15				
4	T-100	Fill	1			2:00	2:06			10:30	10:36			18:10	18:15				
5	T-101	Pre-rinse	1.5			3:10	3:17			11:25	11:45			19:20	19:40				
6	T-101	Fill	1			3:15	3:25			11:15	11:25			19:30	19:40				
7	T-102	Pre-rinse	1.5			1:20	1:30			9:30	9:40			17:10	17:20				
8	T-102	Fill	1			3:15	3:25			10:15	10:20			18:15	18:20				
9	T-103	Pre-rinse	1.5			1:00	1:10			9:00	9:10			19:30	19:40				
10	T-103	Fill	1			6:00	6:30												
11	BT-100	Buffer Tank pre-rinse	1.5			1:15	1:20			9:00	9:05			17:30	17:35				
12	BT-100	Buffer Tank fill	1.5			2:00	2:10			10:00	10:10			19:00	19:10				
13	BT-101	Buffer Tank pre-rinse	1.5			2:30	3:00			10:30	11:00			19:30	20:00				
14	BT-101	Buffer Tank fill	1.5			3:00	3:30			11:00	11:30			20:00	20:30				
15	BT-102	Buffer Tank pre-rinse	1.5			21:00	21:05												
16	BT-102	Buffer Tank fill	1.5			21:15	21:20												
17	BT-103	Buffer Tank pre-rinse	1.5			2:00	2:05												
18	BT-103	Buffer Tank fill	1.5			0:30	0:45												
19	N/A	Line Rinsing	1			0:30	0:45			6:00	6:15			7:20	7:35			8:40	8:55
20																			
21		**2. FILLING SUITE**																	
22	PW-100	Parts Wash pre-rinse	2.5			0:30	0:40			11:00	11:10			16:00	16:10				
23	PW-100	Fill Parts Washer	5			1:30	2:10			12:30	12:50			17:30	17:55				
24	PW-101	Parts Wash pre-rinse	2.5			3:15	3:30			12:45	12:55			16:30	16:40				
25	PW-101	Fill Parts Washer	2.5			5:30	6:40			13:00	13:10			17:00	17:10				
26	PW-102	Parts Wash pre-rinse	2.5			7:30	7:40			11:15	11:25			16:15	16:25				
27	PW-102	Fill Parts Washer	2.5			8:00	8:10			13:00	13:10			17:00	17:10				
28	ST-100	Stopper Washer & Steriliser pre-rinse	2			4:00	4:10			12:00	12:10			18:30	18:40				
29	ST-100	Stopper Processor	3			5:00	6:00			13:00	14:00			19:00	20:00				
30	ST-101	Stopper Washer & Steriliser pre-rinse	2			6:00	6:10			14:00	14:10			20:00	20:10				
31	ST-101	Stopper Processor	3			5:00	6:00			15:00	16:00			21:00	22:00				
32	ST-102	Stopper Washer & Steriliser pre-rinse	2			6:00	6:10			14:00	14:10			22:00	22:10				
33	ST-102	Stopper Processor	3			7:00	8:10			15:30	16:30			23:00	23:59				
34	ST-103	Stopper Washer & Steriliser pre-rinse	2			0:35	0:40			2:00	2:10			11:00	11:10			13:00	13:40
35	ST-103	Stopper Processor	2			1:15	1:25			2:30	2:50			12:00	12:10			13:40	13:50
36	ST-104	Stopper Washer & Steriliser pre-rinse	2			2:15	2:25			3:30	3:40			13:00	13:10			14:30	14:50
37	ST-104	Stopper Processor	2			3:00	3:25			4:40	4:50			13:30	13:40			15:00	15:20
38	VF-100	Vial Rinser - Fill	0.5			0:15	0:25			1:00	1:30			2:00	2:30				
39	VF-101	Vial Rinser - Fill	0.5			6:00	6:35			7:00	7:35			8:00	8:30				
40	VF-102	Vial Rinser - Fill	0.5			10:00	10:35			11:00	11:35			12:00	12:30				
41	VF-103	Vial Rinser - Fill	0.5			20:00	20:35			21:00	21:35			22:00	22:30				
42	PW-100	Parts washer Sanitization	2.5			22:00	22:05												
43	PW-101	Parts washer Sanitization	2.5			21:00	21:05												
44	PW-102	Parts washer Sanitization	2.5			22:30	22:35												
45	ST-100	Stopper Processor Sanitation	2.5			20:00	20:05												
46	ST-101	Stopper Processor Sanitation	2.5			21:00	21:05												
47	ST-102	Stopper Processor Sanitation	2.5			22:00	22:05												
48	ST-103	Stopper Processor Sanitation	2.5			2:00	2:05												
49	ST-104	Stopper Processor Sanitation	2.5			5:00	5:05												
50	VF-100	Vial rinser Sanitisation	2.5			6:30	6:35			7:30	7:40								
51	VF-101	Vial rinser Sanitisation	2.5			4:45	4:50			5:45	5:55								
52	VF-102	Vial rinser Sanitisation	2.5			6:00	6:10			7:00	7:10								
53	VF-103	Vial rinser Sanitisation	2.5			3:00	3:10												
54																			
55		**2. WASHING SUITE**																	
56	PW-100	Parts Washer Pre Rinse	2.5			2:30	2:35			5:30	5:35			6:40	6:45			9:00	9:05
57	PW-101	Parts Washer Pre Rinse	5			4:00	5:15			12:30	12:45			16:00	16:15				
58	PW-102	Parts Washer Pre Rinse	2.5			5:00	5:30			6:30	7:00			10:30	11:00			13:00	13:30
59	PW-100	Parts Washer Sanitization	2.5			22:30	22:35												
60	PW-101	Parts Washer Sanitization	2.5			23:30	23:35												
61	PW-102	Parts Washer Sanitization	2.5			4:00	4:05												
62																			
63		**4. CIP SUITE**																	
64	T-100	T-100	5			2:00	2:30			8:15	8:30			16:15	16:30				
65	T-101	T-101	5			2:15	2:45			10:15	10:30			18:15	18:30				
66	T-102	T-102	5			4:15	4:30			12:15	12:30			20:15	20:30				
67	T-103	T-103 RESCHEDULE from 6:00	2.5			7:00	7:40			17:00	17:15								
68	VF-100	Vial Filler	2.5			23:00	23:40												
69	VF-101	Vial Filler	5			13:00	13:25												
70	VF-102	Vial Filler	5			17:00	17:25												
71	VF-103	Vial Filler	5			19:00	19:25												
72	LYO-100	Lyophyliser	2.5			5:00	5:30												
73	LYO-101	Lyophyliser	2.5			11:00	11:30												
74	LYO-102	Lyophyliser	2.5			21:00	21:30												
75	LYO-103	Lyophyliser	2.5			23:30	23:59												
76	LYO-104	Lyophyliser	2.5			23:30	23:59												
77	VFC-100	Vial Filler CIP skid	1			23:30	23:40												
78	VFC-101	Vial Filler CIP skid	1			23:00	23:10												
79	VFC-102	Vial Filler CIP skid	1			22:00	22:10												
80	VFC-103	Vial Filler CIP skid	1			21:30	21:40												
81	LYC-100	Lyophyliser CIP skid	1			21:00	21:10												
82	LYC-101	Lyophyliser CIP skid	1			20:00	20:10												
83	LYC-102	Lyophyliser CIP skid	1			3:00	3:05												
84	LYC-103	Lyophyliser CIP skid	1			4:00	4:05												
85																			
86																			
87		**FIRST INCREASE IN PRODUCTION**																	
88		**BY MAXIMISING USE OF EQUIPMENT**																	
89																			
90		**1. BULK PROCESSING SUITE**																	
91	T-100	Fill	1			2:00	2:06			10:30	10:36			18:10	18:15				
92	T-101	Fill	1			3:15	3:25			11:15	11:25			19:30	19:40				
93	T-102	Fill	1			3:15	3:25			10:15	10:20			18:15	18:20				
94	T-103	Fill	1			6:00	6:30												
95	BT-100	Buffer Tank fill	1.5			2:00	2:10			10:00	10:10			19:00	19:10				
96	BT-101	Buffer Tank fill	1.5			3:00	3:30			11:00	11:30			20:00	20:30				
97	BT-102	Buffer Tank fill	1.5			21:15	21:20												
98	BT-103	Buffer Tank fill	1.5			0:30	0:45												
99																			
100		**2. FILLING SUITE**																	
101	ST-100	Stopper Washer & Steriliser pre-rinse	0.5			4:00	4:10			12:00	12:10			18:30	18:40				
102	ST-100	Stopper Processor	2			5:00	6:00			13:00	14:00			19:00	20:00				
103	ST-101	Stopper Washer & Steriliser pre-rinse	0.5			6:00	6:10			14:00	14:10			20:00	20:10				
104	ST-101	Stopper Processor	2			5:00	6:00			15:00	16:00			21:00	22:00				
105	ST-102	Stopper Washer & Steriliser pre-rinse	0.5			6:00	6:10			14:00	14:10			22:00	22:10				
106	ST-102	Stopper Processor	2			7:00	8:10			15:30	16:30			23:00	23:59				
107	ST-103	Stopper Washer & Steriliser pre-rinse	0.5			0:35	0:40			2:00	2:10			11:00	11:10			13:00	13:40
108	ST-103	Stopper Processor	0.5			1:15	1:25			2:30	2:40			12:00	12:10			13:40	13:50
109	ST-104	Stopper Washer & Steriliser pre-rinse	0.5			2:15	2:25			3:30	3:40			13:00	13:10			14:30	14:50
110	ST-104	Stopper Processor	0.5			3:00	3:25			4:30	5:10			13:30	13:40			15:00	15:10
111	VF-100	Vial Rinser - Fill	0.5			0:15	0:30			1:30	2:00			3:00	3:30				
112	VF-101	Vial Rinser - Fill	0.5			6:00	6:35			7:00	7:30			8:00	8:30				
113	VF-102	Vial Rinser - Fill	0.5			10:00	10:35			11:00	11:35			12:00	12:30				
114	VF-103	Vial Rinser - Fill	0.5			20:00	20:35			21:00	21:35			22:00	22:30				
115																			

116																				
117		**SECOND INCREASE IN PRODUCTION**																		
118		**ADD ANOTHER FILLING SUITE**																		
119		**STAGGER BY 2 HOURS**																		
120																				
121		**2. FILLING SUITE**																		
122	PW-100a	Parts Wash pre-rinse	2.5			2:40	2:50			13:00	13:10			18:00	18:10					
123	PW-100a	Fill Parts Washer	5			3:30	4:10			14:30	14:50			19:30	19:55					
124	PW-101a	Parts Wash pre-rinse	2.5			5:15	5:30			14:45	14:55			18:30	18:40					
125	PW-101a	Fill Parts Washer	2.5			7:30	7:40			15:00	15:10			19:00	19:10					
126	PW-102a	Parts Wash pre-rinse	2.5			9:30	9:40			13:15	13:25			18:15	18:25					
127	PW-102a	Fill Parts Washer	2.5			10:00	10:10			15:00	15:10			19:00	19:10					
128	ST-100a	Stopper Washer & Steriliser pre-rinse	2			6:00	6:10			14:00	14:10			20:30	20:40					
129	ST-100a	Stopper Processor	3			7:00	8:00			15:00	16:00			21:00	22:00					
130	ST-101a	Stopper Washer & Steriliser pre-rinse	2			8:00	8:10			16:00	16:10			22:00	22:10					
131	ST-101a	Stopper Processor	3			7:00	8:00			17:00	18:00			23:00	23:25					
132	ST-102a	Stopper Washer & Steriliser pre-rinse	2			8:00	8:10			16:00	16:10									
133	ST-102a	Stopper Processor	3			9:00	10:10			17:30	18:30									
134	ST-103a	Stopper Washer & Steriliser pre-rinse	2			2:35	2:40			4:00	4:10			13:00	13:10			15:00	15:40	
135	ST-103a	Stopper Processor	2			3:15	3:25			4:30	4:40			14:00	14:10			15:40	15:50	
136	ST-104a	Stopper Washer & Steriliser pre-rinse	2			4:15	4:25			5:30	5:40			15:00	15:10			16:30	16:50	
137	ST-104a	Stopper Processor	2			5:00	5:25			6:30	7:10			15:30	15:40			16:50	17:00	
138	VF-100a	Vial Rinser - Fill	0.5			2:15	2:25			3:00	3:30			4:00	4:30					
139	VF-101a	Vial Rinser - Fill	0.5			8:00	8:35			9:00	9:35			10:00	10:30					
140	VF-102a	Vial Rinser - Fill	0.5			12:00	12:35			13:00	13:35			14:00	14:30					
141	VF-103a	Vial Rinser - Fill	0.5			22:00	22:35			23:00	23:35									
142																				
143	PW-101a	Parts washer Sanitization	2.5			23:00	23:05													
144																				
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22	2.5			23:00	23:05													
146	ST-101a	Stopper Processor Sanitation	2.5			23:00	23:05													
147																				
148	ST-103a	Stopper Processor Sanitation	2.5			4:00	4:05													
149	ST-104a	Stopper Processor Sanitation	2.5			7:00	7:05													
150	VF-100a	Vial rinser Sanitisation	2.5			8:30	8:35			9:30	9:40									
151	VF-101a	Vial rinser Sanitisation	2.5			6:45	6:50			7:45	7:55									
152	VF-102a	Vial rinser Sanitisation	2.5			8:00	8:10			9:00	9:10									
153	VF-103a	Vial rinser Sanitisation	2.5			5:00	5:10													
154																				
155																				
156		The second Filling Suite requires additional Bulk																		
157		Processing, Washing & CIP from the existing																		
158		Suites as follows: 2 hours staggered																		
159																				
160		**1. BULK PROCESSING SUITE**																		
161	T-100	Pre-rinse	1.5			4:00	4:05			12:00	12:35			20:10	20:15					
162	T-100	Fill	1			4:00	4:06			12:30	12:36			20:10	20:15					
163	T-101	Pre-rinse	1.5			5:10	5:17			13:25	13:45			21:20	21:40					
164	T-101	Fill	1			5:15	5:25			13:15	13:25			21:30	21:40					
165	T-102	Pre-rinse	1.5			3:20	3:30			11:30	11:40			19:10	19:20					
166	T-102	Fill	1			5:15	5:25			12:15	12:20			20:15	21:20					
167	T-103	Pre-rinse	1.5			3:00	3:10			11:00	11:10			21:30	21:40					
168	T-103	Fill	1			8:00	8:30													
169	BT-100	Buffer Tank pre-rinse	1.5			3:15	3:20			11:00	11:05			19:30	19:35					
170	BT-100	Buffer Tank fill	1.5			4:00	4:10			12:00	12:10			21:00	21:10					
171	BT-101	Buffer Tank pre-rinse	1.5			4:30	5:00			12:30	13:00			21:30	22:00					
172	BT-101	Buffer Tank fill	1.5			5:00	5:30			13:00	13:30			22:00	22:30					
173	BT-102	Buffer Tank pre-rinse	1.5			23:00	23:05													
174	BT-102	Buffer Tank fill	1.5			23:15	23:20													
175	BT-103	Buffer Tank pre-rinse	1.5			4:00	4:05													
176	BT-103	Buffer Tank fill	1.5			2:30	2:45													
177	N/A	Line Rinsing	1			2:30	2:45			8:00	8:15			9:20	9:35			10:40	10:55	
178																				
179																				
180		**2. WASHING SUITE**																		
181	PW-100	Parts Washer Pre Rinse	2.5			4:30	4:35			7:30	7:35			8:40	8:45			11:00	11:05	
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00	5			6:30	7:45			14:30	14:45			18:00	18:15					
183	PW-102	Parts Washer Pre Rinse	2.5			7:00	7:30			8:30	9:00			10:30	12:00			14:00	15:30	
184	PW-100	Parts Washer Sanitization																		
185	PW-101	Parts Washer Sanitization																		
186	PW-102	Parts Washer Sanitization	2.5			6:00	6:05													
187																				
188		**4. CIP SUITE**																		
189	T-100	T-100	5			4:00	4:30			10:15	10:30			18:15	18:30					
190	T-101	T-101	5			4:15	4:45			12:15	12:30			20:15	20:30					
191	T-102	T-102 RESCHEDULE from 6:15	5			8:00	8:15			14:15	14:30			22:15	22:30					
192	T-103	T-103	2.5			8:00	8:40			19:00	19:15									
193	VF-100	Vial Filler																		
194	VF-101	Vial Filler	5			15:00	15:25													
195	VF-102	Vial Filler	5			19:00	19:25													
196	VF-103	Vial Filler RESCHEDULE from 21:00	5			20:00	20:25													
197	LYO-100	Lyophyliser RESCHEDULE from 7:00	2.5			8:00	8:30													
198	LYO-101	Lyophyliser	2.5			13:00	13:30													
199	LYO-102	Lyophyliser	2.5			23:00	23:30													
200	LYO-103	Lyophyliser																		
201	LYO-104	Lyophyliser																		
202	VFC-100	Vial Filler CIP skid																		
203	VFC-101	Vial Filler CIP skid																		
204	VFC-102	Vial Filler CIP skid																		
205	VFC-103	Vial Filler CIP skid	1			23:30	23:40													
206	LYC-100	Lyophyliser CIP skid	1			23:00	23:10													
207	LYC-101	Lyophyliser CIP skid	1			22:00	22:10													
208	LYC-102	Lyophyliser CIP skid	1			5:00	5:05													
209	LYC-103	Lyophyliser CIP skid	1			6:00	6:05													

211																			
212		**THIRD INCREASE IN PRODUCTION**																	
213		**BY MAXIMISING USE OF EQUIPMENT**																	
214																			
215		**1. BULK PROCESSING SUITE**																	
216	T-100	Fill	1			4:00	4:06			12:30	12:36			20:10	20:15				
217	T-101	Fill	1			5:15	5:25			13:15	13:25			21:30	21:40				
218	T-102	Fill	1			5:15	5:25			12:15	12:20			20:15	21:20				
219	T-103	Fill	1			8:00	8:30												
220	BT-100	Buffer Tank fill	1.5			4:00	4:10			12:00	12:10			21:00	21:10				
221	BT-101	Buffer Tank fill	1.5			5:00	5:30			13:00	13:30			22:00	22:30				
222	BT-102	Buffer Tank fill	1.5			23:15	23:20												
223	BT-103	Buffer Tank fill	1.5			2:30	2:45												
224																			
225		**2. FILLING SUITE**																	
226	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5			6:00	6:10			14:00	14:10			20:30	20:40				
227	ST-100a	Stopper Processor	2			7:00	8:00			15:00	16:00			21:00	22:00				
228	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5			8:00	8:10			16:00	16:10			22:00	22:10				
229	ST-101a	Stopper Processor	2			7:00	8:00			17:00	18:00			23:00	23:25				
230	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5			8:00	8:10			16:00	16:10								
231	ST-102a	Stopper Processor	2			9:00	10:10			17:30	18:30								
232	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5			2:35	2:40			4:00	4:10			13:00	13:10			15:00	15:40
233	ST-103a	Stopper Processor	0.5			3:15	3:25			4:30	4:40			14:00	14:10			15:40	15:50
234	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5			4:15	4:25			5:30	5:40			15:00	15:10			16:30	16:50
235	ST-104a	Stopper Processor	0.5			5:00	5:25			6:30	7:10			15:30	15:40			16:50	17:00
236	VF-100a	Vial Rinser - Fill	0.5			2:15	2:25			3:00	3:30			4:00	4:30				
237	VF-101a	Vial Rinser - Fill	0.5			8:00	8:35			9:00	9:35			10:00	10:30				
238	VF-102a	Vial Rinser - Fill	0.5			12:00	12:35			13:00	13:35			14:00	14:30				
239	VF-103a	Vial Rinser - Fill	0.5			22:00	22:35			23:00	23:35								
240																			
241																			
242																			
243																			
244		**FOURTH INCREASE IN PRODUCTION**																	
245		**ADD THIRD FILLING SUITE**																	
246		**STAGGER BY 2 HOURS**																	
247																			
248		**2. FILLING SUITE**																	
249	PW-100a	Parts Wash pre-rinse	2.5			4:40	4:50			15:00	15:10			20:00	20:10				
250	PW-100a	Fill Parts Washer	5			5:30	6:10			16:30	16:50			21:30	21:55				
251	PW-101a	Parts Wash pre-rinse	2.5			7:15	7:30			16:45	16:55			20:30	22:40				
252	PW-101a	Fill Parts Washer	2.5			9:30	9:40			17:00	17:10			21:00	23:10				
253	PW-102a	Parts Wash pre-rinse	2.5			11:30	11:40			15:15	17:25			20:15	20:25				
254	PW-102a	Fill Parts Washer	2.5			12:00	12:10			17:00	17:10			21:00	21:10				
255	ST-100a	Stopper Washer & Steriliser pre-rinse	2			8:00	8:10			16:00	18:10			22:30	22:40				
256	ST-100a	Stopper Processor	3			9:00	10:00			17:00	18:00								
257	ST-101a	Stopper Washer & Steriliser pre-rinse	2			11:00	11:10			18:00	18:10								
258	ST-101a	Stopper Processor	3			9:00	10:00			19:00	20:00								
259	ST-102a	Stopper Washer & Steriliser pre-rinse	2			10:00	10:10			18:00	18:10								
260	ST-102a	Stopper Processor	3			11:00	12:10			19:30	20:30								
261	ST-103a	Stopper Washer & Steriliser pre-rinse	2			5:35	5:40			6:50	7:05			15:00	15:10			17:00	17:40
262	ST-103a	Stopper Processor	2			5:15	5:25			6:30	6:40			16:00	16:10			17:40	17:50
263	ST-104a	Stopper Washer & Steriliser pre-rinse	2			6:15	6:25			7:30	7:40			17:00	17:10			18:30	18:50
264	ST-104a	Stopper Processor	2			7:00	7:25			8:30	9:10			17:30	17:40			18:50	19:00
265	VF-100a	Vial Rinser - Fill	0.5			4:15	4:25			5:00	5:30			6:00	6:30				
266	VF-101a	Vial Rinser - Fill	0.5			10:00	10:35			11:00	11:35			12:00	12:30				
267	VF-102a	Vial Rinser - Fill	0.5			14:00	14:35			15:00	15:35			16:00	16:30				
268																			
269																			
270	PW-101a	Parts washer Sanitization	2.5			1:00	1:05												
271																			
272	ST-100a	Stopper Processor Sanitation	2.5			1:00	1:05												
273	ST-101a	Stopper Processor Sanitation	2.5			0:30	0:35												
274																			
275	ST-103a	Stopper Processor Sanitation	2.5			6:00	6:05												
276	ST-104a	Stopper Processor Sanitation	2.5			9:00	9:05												
277	VF-100a	Vial rinser Sanitisation	2.5			10:30	10:35			11:30	11:40								
278	VF-101a	Vial rinser Sanitisation	2.5			8:45	8:50			9:45	9:55								
279	VF-102a	Vial rinser Sanitisation	2.5			10:00	10:10			11:00	11:10								
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00	2.5			8:00	8:10												
281																			
282																			
283																			
284																			
285																			
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																	
287		**1. BULK PROCESSING SUITE**																	
288	T-100	Pre-rinse	1.5			6:00	6:05			14:00	14:35			22:10	22:15				
289	T-100	Fill	1			6:00	6:06			14:30	14:36			22:10	22:15				
290	T-101	Pre-rinse	1.5			7:10	7:17			15:25	15:45			23:20	23:40				
291	T-101	Fill	1			7:15	7:25			15:15	15:25			23:30	23:40				
292	T-102	Pre-rinse	1.5			5:20	5:30			13:30	13:40			21:10	21:20				
293	T-102	Fill	1			7:15	7:25			14:15	14:20			22:15	22:20				
294	T-103	Pre-rinse	1.5			5:00	5:10			13:00	13:10			23:30	23:40				
295	T-103	Fill	1			11:00	11:30												
296	BT-100	Buffer Tank pre-rinse	1.5			5:15	5:20			13:00	13:05			21:30	21:35				
297	BT-100	Buffer Tank fill	1.5			6:00	6:10			14:00	14:10			23:00	23:10				
298	BT-101	Buffer Tank pre-rinse	1.5			6:30	7:00			14:30	15:00								
299	BT-101	Buffer Tank fill	1.5			7:00	7:30			15:00	15:30								
300	BT-102	Buffer Tank pre-rinse																	
301	BT-102	Buffer Tank fill																	
302	BT-103	Buffer Tank pre-rinse	1.5			6:00	6:05												
303	BT-103	Buffer Tank fill	1.5			4:30	4:45												
304	N/A	Line Rinsing	1			4:30	4:45			10:00	10:15			11:20	11:35			12:40	12:55
305																			
306																			
307																			
308																			
309		**2. WASHING SUITE**																	
310	PW-100	Parts Washer Pre Rinse	2.5			6:30	6:35			9:30	9:35			10:40	10:45			13:00	13:05
311	PW-101	Parts Washer Pre Rinse	5			8:00	9:15			16:30	16:45			20:00	20:15				
312	PW-102	Parts Washer Pre Rinse	2.5			9:00	9:30			10:30	11:00			14:30	15:00			16:00	16:30
313	PW-100	Parts Washer Sanitization																	
314	PW-101	Parts Washer Sanitization																	
315	PW-102	Parts Washer Sanitization	2.5			10:00	10:05												
316																			
317		**4. CIP SUITE**																	
318	T-100	T-100	5			6:00	6:30			12:15	12:30			20:15	20:30				
319	T-101	T-101	5			6:15	6:45			16:15	16:30			22:15	22:30				
320	T-102	T-102	5			1:15	1:30			8:15	8:30			16:15	16:30				
321	T-103	T-103	2.5			10:00	10:40			21:00	21:15								
322	VF-100	Vial Filler	2.5			3:00	3:40												
323	VF-101	Vial Filler	5			17:00	17:25												
324	VF-102	Vial Filler RESCHEDULE from 21:00	5			20:00	20:25												
325	VF-103	Vial Filler	5			23:00	23:25												
326	LYO-100	Lyophyliser	2.5			9:00	9:30												
327	LYO-101	Lyophyliser	2.5			15:00	15:30												
328	LYO-102	Lyophyliser	2.5			1:30	2:00												
329	LYO-103	Lyophyliser	2.5			2:30	2:59												
330	LYO-104	Lyophyliser	2.5			2:30	2:59												
331	VFC-100	Vial Filler CIP skid	1			2:30	2:40												
332	VFC-101	Vial Filler CIP skid	1			2:00	2:10												
333	VFC-102	Vial Filler CIP skid	1			3:00	3:10												
334	VFC-103	Vial Filler CIP skid	1			1:30	1:40												
335	LYC-100	Lyophyliser CIP skid	1			1:20	1:30												
336	LYC-101	Lyophyliser CIP skid	1			1:15	1:25												
337	LYC-102	Lyophyliser CIP skid	1			7:00	7:05												
338	LYC-103	Lyophyliser CIP skid	1			8:00	8:05												

342		**FIFTH INCREASE IN PRODUCTION**																		
343		**BY MAXIMISING USE OF EQUIPMENT**																		
344																				
345		**1. BULK PROCESSING SUITE**																		
346	T-100	Fill	1			6:00	6:06			14:30	14:36			22:10	22:15					
347	T-101	Fill	1			7:15	7:25			15:15	15:25			23:30	23:40					
348	T-102	Fill	1			5:15	5:25			12:15	12:20			20:15	21:20					
349	T-103	Fill	1			7:15	7:25			14:15	14:20			22:15	22:20					
350	BT-100	Buffer Tank fill	1.5			6:00	6:10			14:00	14:10			23:00	23:10					
351	BT-101	Buffer Tank fill	1.5			7:00	7:30			15:00	15:30									
352	BT-102	Buffer Tank fill																		
353	BT-103	Buffer Tank fill	1.5			6:00	6:05													
354																				
355		**2. FILLING SUITE**																		
356	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5			8:00	8:10			16:00	18:10			22:30	22:40					
357	ST-100a	Stopper Processor	2			9:00	10:00			17:00	18:00									
358	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5			11:00	11:10			18:00	18:10									
359	ST-101a	Stopper Processor	2			9:00	10:00			19:00	20:00									
360	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5			10:00	10:10			18:00	18:10									
361	ST-102a	Stopper Processor	2			11:00	12:10			19:30	20:30									
362	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5			5:35	5:40			6:50	7:05			15:00	15:10			17:00	17:40	
363	ST-103a	Stopper Processor	0.5			5:15	5:25			6:30	6:40			16:00	16:10			17:40	17:50	
364	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5			6:15	6:25			7:30	7:40			17:00	17:10			18:30	18:50	
365	ST-104a	Stopper Processor	0.5			7:00	7:25			8:30	9:10			17:30	17:40			18:50	19:00	
366	VF-100a	Vial Rinser - Fill	0.5			4:15	4:25			5:00	5:30			6:00	6:30					
367	VF-101a	Vial Rinser - Fill	0.5			10:00	10:35			11:00	11:35			12:00	12:30					
368	VF-102a	Vial Rinser - Fill	0.5			14:00	14:35			15:00	15:35			16:00	16:30					
369																				
370																				
371		**SIXTH INCREASE IN PRODUCTION**																		
372		**ADD FOURTH FILLING SUITE**																		
373		**RESCHEDULE BY 2 HOURS (STAGGERED)**																		
374																				
375		**2. FILLING SUITE**																		
376	PW-100a	Parts Wash pre-rinse	2.5			6:40	6:50			17:00	17:10			22:00	22:10					
377	PW-100a	Fill Parts Washer	5			7:30	8:10			18:30	18:50									
378	PW-101a	Parts Wash pre-rinse	2.5			9:15	9:30			18:45	18:55			22:30	23:55					
379	PW-101a	Fill Parts Washer	2.5			11:30	11:40			19:00	19:10			23:00	23:55					
380	PW-102a	Parts Wash pre-rinse	2.5			13:30	13:40			19:15	19:25			22:15	23:55					
381	PW-102a	Fill Parts Washer	2.5			14:00	14:10			19:00	19:10			23:00	23:55					
382	ST-100a	Stopper Washer & Steriliser pre-rinse	2			10:00	10:10			18:00	20:10									
383	ST-100a	Stopper Processor	3			11:00	12:00			19:00	20:00									
384	ST-101a	Stopper Washer & Steriliser pre-rinse	2			13:00	13:10			20:00	20:10									
385	ST-101a	Stopper Processor	3			11:00	12:00			21:00	22:00									
386	ST-102a	Stopper Washer & Steriliser pre-rinse	2			12:00	12:10			20:00	20:10									
387	ST-102a	Stopper Processor	3			13:00	14:10			21:30	22:30									
388	ST-103a	Stopper Washer & Steriliser pre-rinse	2			7:35	7:40			8:50	9:05			17:00	17:10			19:00	19:40	
389	ST-103a	Stopper Processor	2			7:15	7:25			8:30	8:40			18:00	18:10			19:40	19:50	
390	ST-104a	Stopper Washer & Steriliser pre-rinse	2			8:15	8:25			9:30	9:40			19:00	19:10			20:30	20:50	
391	ST-104a	Stopper Processor	2			9:00	9:25			10:30	11:10			19:30	19:40			20:50	21:00	
392	VF-100a	Vial Rinser - Fill	0.5			6:15	6:25			7:00	7:30			8:00	8:30					
393	VF-101a	Vial Rinser - Fill	0.5			12:00	12:35			13:00	13:35			14:00	14:30					
394	VF-102a	Vial Rinser - Fill	0.5			16:00	16:35			17:00	17:35			18:00	18:30					
395																				
396																				
397	PW-101a	Parts washer Sanitization	2.5			3:00	3:05													
398																				
399	ST-100a	Stopper Processor Sanitation	2.5			3:00	3:05													
400	ST-101a	Stopper Processor Sanitation	2.5			2:30	2:35													
401																				
402	ST-103a	Stopper Processor Sanitation	2.5			8:00	8:05													
403	ST-104a	Stopper Processor Sanitation	2.5			11:00	11:05													
404	VF-100a	Vial rinser Sanitisation	2.5			12:30	12:35			13:30	15:40									
405	VF-101a	Vial rinser Sanitisation	2.5			10:45	10:50			11:45	11:55									
406	VF-102a	Vial rinser Sanitisation	2.5			12:00	12:10			13:00	13:10									
407	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00	2.5																	
408																				
409																				
410																				
411		**SEVENTH INCREASE IN PRODUCTION**																		
412		**BY MAXIMISING USE OF EQUIPMENT**																		
413																				
414		**1. BULK PROCESSING SUITE**																		
415	T-100	Fill	1			7:00	8:06			16:30	16:36									
416	T-101	Fill	1			9:15	9:25			17:15	17:25									
417	T-102	Fill	1			7:15	7:25			14:15	14:20			22:15	23:20					
418	T-103	Fill	1			9:15	9:25			16:15	16:20									
419	BT-100	Buffer Tank fill	1.5			8:00	8:10			16:00	16:10									
420	BT-101	Buffer Tank fill	1.5			9:00	9:30			17:00	17:30									
421	BT-102	Buffer Tank fill																		
422	BT-103	Buffer Tank fill	1.5			8:00	8:05													
423																				
424		**2. FILLING SUITE**																		
425	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5			10:00	10:10			18:00	18:10									
426	ST-100a	Stopper Processor	2			11:00	12:00			19:00	20:00									
427	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5			13:00	13:10			20:00	20:10									
428	ST-101a	Stopper Processor	2			11:00	12:00			21:00	22:00									
429	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5			12:00	12:10			20:00	20:10									
430	ST-102a	Stopper Processor	2			12:00	14:10			21:30	22:30									
431	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5			7:35	7:40			8:50	9:05			17:00	17:10			19:00	19:40	
432	ST-103a	Stopper Processor	0.5			7:15	7:25			8:30	8:40			18:00	18:10			19:40	19:50	
433	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5			8:15	8:25			9:30	9:40			19:00	19:10			20:30	20:50	
434	ST-104a	Stopper Processor	0.5			9:00	9:25			10:30	11:10			19:30	19:40			20:50	21:00	
435	VF-100a	Vial Rinser - Fill	0.5			6:15	6:25			7:00	7:30			8:00	8:30					
436	VF-101a	Vial Rinser - Fill	0.5			12:00	12:35			13:00	13:35			14:00	14:30					
437	VF-102a	Vial Rinser - Fill	0.5			16:00	16:35			17:00	17:35			18:00	18:30					

WFI Water Offtake Input Data							WFI Water Offtake Input Data								WFI Water Offtake Input Data								er Offtake Input Data	
1			5				6				7				8				9				10	
No.	Tag No.	Description of Offtake	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]
1		**ORIGINAL PROCESS DATA**																						
2		**1. BULK PROCESSING SUITE**																						
3	T-100	Pre-rinse																						
4	T-100	Fill																						
5	T-101	Pre-rinse																						
6	T-101	Fill																						
7	T-102	Pre-rinse																						
8	T-102	Fill																						
9	T-103	Pre-rinse																						
10	T-103	Fill																						
11	BT-100	Buffer Tank pre-rinse																						
12	BT-100	Buffer Tank fill																						
13	BT-101	Buffer Tank pre-rinse																						
14	BT-101	Buffer Tank fill																						
15	BT-102	Buffer Tank pre-rinse																						
16	BT-102	Buffer Tank fill																						
17	BT-103	Buffer Tank pre-rinse																						
18	BT-103	Buffer Tank fill																						
19	N/A	Line Rinsing	9:45	10:00			11:00	11:15			15:00	15:15			16:15	16:30			17:20	17:35			18:30	18:45
20																								
21		**2. FILLING SUITE**																						
22	PW-100	Parts Wash pre-rinse																						
23	PW-100	Fill Parts Washer																						
24	PW-101	Parts Wash pre-rinse																						
25	PW-101	Fill Parts Washer																						
26	PW-102	Parts Wash pre-rinse																						
27	PW-102	Fill Parts Washer																						
28	ST-100	Stopper Washer & Steriliser pre-rinse																						
29	ST-100	Stopper Processor																						
30	ST-101	Stopper Washer & Steriliser pre-rinse																						
31	ST-101	Stopper Processor																						
32	ST-102	Stopper Washer & Steriliser pre-rinse																						
33	ST-102	Stopper Processor																						
34	ST-103	Stopper Washer & Steriliser pre-rinse	18:00	18:10			19:30	19:40																
35	ST-103	Stopper Processor	19:00	19:10			20:30	20:40																
36	ST-104	Stopper Washer & Steriliser pre-rinse	20:00	20:10			21:30	21:40			23:00	23:10												
37	ST-104	Stopper Processor	20:30	20:40			22:00	22:10			23:20	23:30												
38	VF-100	Vial Rinser - Fill																						
39	VF-101	Vial Rinser - Fill																						
40	VF-102	Vial Rinser - Fill																						
41	VF-103	Vial Rinser - Fill																						
42	PW-100	Parts washer Sanitization																						
43	PW-101	Parts washer Sanitization																						
44	PW-102	Parts washer Sanitization																						
45	ST-100	Stopper Processor Sanitation																						
46	ST-101	Stopper Processor Sanitation																						
47	ST-102	Stopper Processor Sanitation																						
48	ST-103	Stopper Processor Sanitation																						
49	ST-104	Stopper Processor Sanitation																						
50	VF-100	Vial rinser Sanitisation																						
51	VF-101	Vial rinser Sanitisation																						
52	VF-102	Vial rinser Sanitisation																						
53	VF-103	Vial rinser Sanitisation																						
54																								
55		**2. WASHING SUITE**																						
56	PW-100	Parts Washer Pre Rinse	10:30	10:35			15:30	15:35			16:45	16:50			17:50	17:55			19:00	19:05				
57	PW-101	Parts Washer Pre Rinse																						
58	PW-102	Parts Washer Pre Rinse	21:30	22:00			23:30	23:55																
59	PW-100	Parts Washer Sanitization																						
60	PW-101	Parts Washer Sanitization																						
61	PW-102	Parts Washer Sanitization																						
62																								
63		**4. CIP SUITE**																						
64	T-100	T-100																						
65	T-101	T-101																						
66	T-102	T-102																						
67	T-103	T-103 RESCHEDULE from 6:00																						
68	VF-100	Vial Filler																						
69	VF-101	Vial Filler																						
70	VF-102	Vial Filler																						
71	VF-103	Vial Filler																						
72	LYO-100	Lyophyliser																						
73	LYO-101	Lyophyliser																						
74	LYO-102	Lyophyliser																						
75	LYO-103	Lyophyliser																						
76	LYO-104	Lyophyliser																						
77	VFC-100	Vial Filler CIP skid																						
78	VFC-101	Vial Filler CIP skid																						
79	VFC-102	Vial Filler CIP skid																						
80	VFC-103	Vial Filler CIP skid																						
81	LYC-100	Lyophyliser CIP skid																						
82	LYC-101	Lyophyliser CIP skid																						
83	LYC-102	Lyophyliser CIP skid																						
84	LYC-103	Lyophyliser CIP skid																						
85																								
86																								
87		**FIRST INCREASE IN PRODUCTION**																						
88		**BY MAXIMISING USE OF EQUIPMENT**																						
89																								
90		**1. BULK PROCESSING SUITE**																						
91	T-100	Fill																						
92	T-101	Fill																						
93	T-102	Fill																						
94	T-103	Fill																						
95	BT-100	Buffer Tank fill																						
96	BT-101	Buffer Tank fill																						
97	BT-102	Buffer Tank fill																						
98	BT-103	Buffer Tank fill																						
99																								
100		**2. FILLING SUITE**																						
101	ST-100	Stopper Washer & Steriliser pre-rinse																						
102	ST-100	Stopper Processor																						
103	ST-101	Stopper Washer & Steriliser pre-rinse																						
104	ST-101	Stopper Processor																						
105	ST-102	Stopper Washer & Steriliser pre-rinse																						
106	ST-102	Stopper Processor																						
107	ST-103	Stopper Washer & Steriliser pre-rinse	18:00	18:10			19:30	19:40																
108	ST-103	Stopper Processor	19:00	19:10			20:30	20:40																
109	ST-104	Stopper Washer & Steriliser pre-rinse	20:00	20:10			21:30	21:40			22:50	23:00												
110	ST-104	Stopper Processor	20:30	20:40			22:00	22:10			23:15	23:45												

116																								
117		**SECOND INCREASE IN PRODUCTION**																						
118		**ADD ANOTHER FILLING SUITE**																						
119		**STAGGER BY 2 HOURS**																						
120																								
121		**2. FILLING SUITE**																						
122	PW-100a	Parts Wash pre-rinse																						
123	PW-100a	Fill Parts Washer																						
124	PW-101a	Parts Wash pre-rinse																						
125	PW-101a	Fill Parts Washer																						
126	PW-102a	Parts Wash pre-rinse																						
127	PW-102a	Fill Parts Washer																						
128	ST-100a	Stopper Washer & Steriliser pre-rinse																						
129	ST-100a	Stopper Processor																						
130	ST-101a	Stopper Washer & Steriliser pre-rinse																						
131	ST-101a	Stopper Processor																						
132	ST-102a	Stopper Washer & Steriliser pre-rinse																						
133	ST-102a	Stopper Processor																						
134	ST-103a	Stopper Washer & Steriliser pre-rinse	20:00	20:10			21:30	21:40																
135	ST-103a	Stopper Processor	21:00	21:10			22:30	22:40																
136	ST-104a	Stopper Washer & Steriliser pre-rinse	22:00	22:10																				
137	ST-104a	Stopper Processor	22:30	22:40																				
138	VF-100a	Vial Rinser - Fill																						
139	VF-101a	Vial Rinser - Fill																						
140	VF-102a	Vial Rinser - Fill																						
141	VF-103a	Vial Rinser - Fill																						
142																								
143	PW-101a	Parts washer Sanitization																						
144																								
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22:00																						
146	ST-101a	Stopper Processor Sanitation																						
147																								
148	ST-103a	Stopper Processor Sanitation																						
149	ST-104a	Stopper Processor Sanitation																						
150	VF-100a	Vial rinser Sanitisation																						
151	VF-101a	Vial rinser Sanitisation																						
152	VF-102a	Vial rinser Sanitisation																						
153	VF-103a	Vial rinser Sanitisation																						
154																								
155																								
156		The second Filling Suite requires additional Bulk																						
157		Processing, Washing & CIP from the existing																						
158		Suites as follows: 2 hours staggered																						
159																								
160		**1. BULK PROCESSING SUITE**																						
161	T-100	Pre-rinse																						
162	T-100	Fill																						
163	T-101	Pre-rinse																						
164	T-101	Fill																						
165	T-102	Pre-rinse																						
166	T-102	Fill																						
167	T-103	Pre-rinse																						
168	T-103	Fill																						
169	BT-100	Buffer Tank pre-rinse																						
170	BT-100	Buffer Tank fill																						
171	BT-101	Buffer Tank pre-rinse																						
172	BT-101	Buffer Tank fill																						
173	BT-102	Buffer Tank pre-rinse																						
174	BT-102	Buffer Tank fill																						
175	BT-103	Buffer Tank pre-rinse																						
176	BT-103	Buffer Tank fill																						
177	N/A	Line Rinsing	11:45	12:00			13:00	13:15			17:00	17:15			18:15	18:30			19:20	19:35			20:30	20:45
178																								
179																								
180		**2. WASHING SUITE**																						
181	PW-100	Parts Washer Pre Rinse	12:30	12:35			17:30	17:35			18:45	18:50			19:50	19:55			21:00	21:05				
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00																						
183	PW-102	Parts Washer Pre Rinse																						
184	PW-100	Parts Washer Sanitization																						
185	PW-101	Parts Washer Sanitization																						
186	PW-102	Parts Washer Sanitization																						
187																								
188		**4. CIP SUITE**																						
189	T-100	T-100																						
190	T-101	T-101																						
191	T-102	T-102 RESCHEDULE from 6:15																						
192	T-103	T-103																						
193	VF-100	Vial Filler																						
194	VF-101	Vial Filler																						
195	VF-102	Vial Filler																						
196	VF-103	Vial Filler RESCHEDULE from 21:00																						
197	LYO-100	Lyophyliser RESCHEDULE from 7:00																						
198	LYO-101	Lyophyliser																						
199	LYO-102	Lyophyliser																						
200	LYO-103	Lyophyliser																						
201	LYO-104	Lyophyliser																						
202	VFC-100	Vial Filler CIP skid																						
203	VFC-101	Vial Filler CIP skid																						
204	VFC-102	Vial Filler CIP skid																						
205	VFC-103	Vial Filler CIP skid																						
206	LYC-100	Lyophyliser CIP skid																						
207	LYC-101	Lyophyliser CIP skid																						
208	LYC-102	Lyophyliser CIP skid																						
209	LYC-103	Lyophyliser CIP skid																						

211																									
212		**THIRD INCREASE IN PRODUCTION**																							
213		**BY MAXIMISING USE OF EQUIPMENT**																							
214																									
215		**1. BULK PROCESSING SUITE**																							
216	T-100	Fill																							
217	T-101	Fill																							
218	T-102	Fill																							
219	T-103	Fill																							
220	BT-100	Buffer Tank fill																							
221	BT-101	Buffer Tank fill																							
222	BT-102	Buffer Tank fill																							
223	BT-103	Buffer Tank fill																							
224																									
225		**2. FILLING SUITE**																							
226	ST-100a	Stopper Washer & Steriliser pre-rinse																							
227	ST-100a	Stopper Processor																							
228	ST-101a	Stopper Washer & Steriliser pre-rinse																							
229	ST-101a	Stopper Processor																							
230	ST-102a	Stopper Washer & Steriliser pre-rinse																							
231	ST-102a	Stopper Processor																							
232	ST-103a	Stopper Washer & Steriliser pre-rinse	20:00	20:10			21:30	21:40																	
233	ST-103a	Stopper Processor	21:00	21:10			22:30	22:40																	
234	ST-104a	Stopper Washer & Steriliser pre-rinse	22:00	22:10																					
235	ST-104a	Stopper Processor	22:30	22:40																					
236	VF-100a	Vial Rinser - Fill																							
237	VF-101a	Vial Rinser - Fill																							
238	VF-102a	Vial Rinser - Fill																							
239	VF-103a	Vial Rinser - Fill																							
240																									
241																									
242																									
243																									
244		**FOURTH INCREASE IN PRODUCTION**																							
245		**ADD THIRD FILLING SUITE**																							
246		**STAGGER BY 2 HOURS**																							
247																									
248		**2. FILLING SUITE**																							
249	PW-100a	Parts Wash pre-rinse																							
250	PW-100a	Fill Parts Washer																							
251	PW-101a	Parts Wash pre-rinse																							
252	PW-101a	Fill Parts Washer																							
253	PW-102a	Parts Wash pre-rinse																							
254	PW-102a	Fill Parts Washer																							
255	ST-100a	Stopper Washer & Steriliser pre-rinse																							
256	ST-100a	Stopper Processor																							
257	ST-101a	Stopper Washer & Steriliser pre-rinse																							
258	ST-101a	Stopper Processor																							
259	ST-102a	Stopper Washer & Steriliser pre-rinse																							
260	ST-102a	Stopper Processor																							
261	ST-103a	Stopper Washer & Steriliser pre-rinse	22:00	22:10																					
262	ST-103a	Stopper Processor	23:00	23:10																					
263	ST-104a	Stopper Washer & Steriliser pre-rinse																							
264	ST-104a	Stopper Processor																							
265	VF-100a	Vial Rinser - Fill																							
266	VF-101a	Vial Rinser - Fill																							
267	VF-102a	Vial Rinser - Fill																							
268																									
269																									
270	PW-101a	Parts washer Sanitization																							
271																									
272	ST-100a	Stopper Processor Sanitation																							
273	ST-101a	Stopper Processor Sanitation																							
274																									
275	ST-103a	Stopper Processor Sanitation																							
276	ST-104a	Stopper Processor Sanitation																							
277	VF-100a	Vial rinser Sanitisation																							
278	VF-101a	Vial rinser Sanitisation																							
279	VF-102a	Vial rinser Sanitisation																							
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																							
281																									
282																									
283																									
284																									
285																									
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																							
287		**1. BULK PROCESSING SUITE**																							
288	T-100	Pre-rinse																							
289	T-100	Fill																							
290	T-101	Pre-rinse																							
291	T-101	Fill																							
292	T-102	Pre-rinse																							
293	T-102	Fill																							
294	T-103	Pre-rinse																							
295	T-103	Fill																							
296	BT-100	Buffer Tank pre-rinse																							
297	BT-100	Buffer Tank fill																							
298	BT-101	Buffer Tank pre-rinse																							
299	BT-101	Buffer Tank fill																							
300	BT-102	Buffer Tank pre-rinse																							
301	BT-102	Buffer Tank fill																							
302	BT-103	Buffer Tank pre-rinse																							
303	BT-103	Buffer Tank fill																							
304	N/A	Line Rinsing	13:45	14:00			15:00	15:15			19:00	19:15			20:15	20:30			21:20	21:35			22:30	22:45	
305																									
306																									
307																									
308																									
309		**2. WASHING SUITE**																							
310	PW-100	Parts Washer Pre Rinse	14:30	14:35			19:30	19:35			20:45	20:50			21:50	21:55			23:00	23:05					
311	PW-101	Parts Washer Pre Rinse																							
312	PW-102	Parts Washer Pre Rinse																							
313	PW-100	Parts Washer Sanitization																							
314	PW-101	Parts Washer Sanitization																							
315	PW-102	Parts Washer Sanitization																							
316																									
317		**4. CIP SUITE**																							
318	T-100	T-100																							
319	T-101	T-101																							
320	T-102	T-102																							
321	T-103	T-103																							
322	VF-100	Vial Filler																							
323	VF-101	Vial Filler																							
324	VF-102	Vial Filler RESCHEDULE from 21:00																							
325	VF-103	Vial Filler																							
326	LYO-100	Lyophyliser																							
327	LYO-101	Lyophyliser																							
328	LYO-102	Lyophyliser																							
329	LYO-103	Lyophyliser																							
330	LYO-104	Lyophyliser																							
331	VFC-100	Vial Filler CIP skid																							
332	VFC-101	Vial Filler CIP skid																							
333	VFC-102	Vial Filler CIP skid																							
334	VFC-103	Vial Filler CIP skid																							
335	LYC-100	Lyophyliser CIP skid																							
336	LYC-101	Lyophyliser CIP skid																							

340																				
341																				
342		**FIFTH INCREASE IN PRODUCTION**																		
343		**BY MAXIMISING USE OF EQUIPMENT**																		
344																				
345		**1. BULK PROCESSING SUITE**																		
346	T-100	Fill																		
347	T-101	Fill																		
348	T-102	Fill																		
349	T-103	Fill																		
350	BT-100	Buffer Tank fill																		
351	BT-101	Buffer Tank fill																		
352	BT-102	Buffer Tank fill																		
353	BT-103	Buffer Tank fill																		
354																				
355		**2. FILLING SUITE**																		
356	ST-100a	Stopper Washer & Steriliser pre-rinse																		
357	ST-100a	Stopper Processor																		
358	ST-101a	Stopper Washer & Steriliser pre-rinse																		
359	ST-101a	Stopper Processor																		
360	ST-102a	Stopper Washer & Steriliser pre-rinse																		
361	ST-102a	Stopper Processor																		
362	ST-103a	Stopper Washer & Steriliser pre-rinse	22:00	22:10																
363	ST-103a	Stopper Processor	23:00	23:10																
364	ST-104a	Stopper Washer & Steriliser pre-rinse																		
365	ST-104a	Stopper Processor																		
366	VF-100a	Vial Rinser - Fill																		
367	VF-101a	Vial Rinser - Fill																		
368	VF-102a	Vial Rinser - Fill																		
369																				
370																				
371		**SIXTH INCREASE IN PRODUCTION**																		
372		**ADD FOURTH FILLING SUITE**																		
373		**RESCHEDULE BY 2 HOURS (STAGGERED)**																		
374																				
375		**2. FILLING SUITE**																		
376	PW-100a	Parts Wash pre-rinse																		
377	PW-100a	Fill Parts Washer																		
378	PW-101a	Parts Wash pre-rinse																		
379	PW-101a	Fill Parts Washer																		
380	PW-102a	Parts Wash pre-rinse																		
381	PW-102a	Fill Parts Washer																		
382	ST-100a	Stopper Washer & Steriliser pre-rinse																		
383	ST-100a	Stopper Processor																		
384	ST-101a	Stopper Washer & Steriliser pre-rinse																		
385	ST-101a	Stopper Processor																		
386	ST-102a	Stopper Washer & Steriliser pre-rinse																		
387	ST-102a	Stopper Processor																		
388	ST-103a	Stopper Washer & Steriliser pre-rinse																		
389	ST-103a	Stopper Processor																		
390	ST-104a	Stopper Washer & Steriliser pre-rinse																		
391	ST-104a	Stopper Processor																		
392	VF-100a	Vial Rinser - Fill																		
393	VF-101a	Vial Rinser - Fill																		
394	VF-102a	Vial Rinser - Fill																		
395																				
396																				
397	PW-101a	Parts washer Sanitization																		
398																				
399	ST-100a	Stopper Processor Sanitation																		
400	ST-101a	Stopper Processor Sanitation																		
401																				
402	ST-103a	Stopper Processor Sanitation																		
403	ST-104a	Stopper Processor Sanitation																		
404	VF-100a	Vial rinser Sanitisation																		
405	VF-101a	Vial rinser Sanitisation																		
406	VF-102a	Vial rinser Sanitisation																		
407	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																		
408																				
409																				
410																				
411		**SEVENTH INCREASE IN PRODUCTION**																		
412		**BY MAXIMISING USE OF EQUIPMENT**																		
413																				
414		**1. BULK PROCESSING SUITE**																		
415	T-100	Fill																		
416	T-101	Fill																		
417	T-102	Fill																		
418	T-103	Fill																		
419	BT-100	Buffer Tank fill																		
420	BT-101	Buffer Tank fill																		
421	BT-102	Buffer Tank fill																		
422	BT-103	Buffer Tank fill																		
423																				
424		**2. FILLING SUITE**																		
425	ST-100a	Stopper Washer & Steriliser pre-rinse																		
426	ST-100a	Stopper Processor																		
427	ST-101a	Stopper Washer & Steriliser pre-rinse																		
428	ST-101a	Stopper Processor																		
429	ST-102a	Stopper Washer & Steriliser pre-rinse																		
430	ST-102a	Stopper Processor																		
431	ST-103a	Stopper Washer & Steriliser pre-rinse																		
432	ST-103a	Stopper Processor																		
433	ST-104a	Stopper Washer & Steriliser pre-rinse																		
434	ST-104a	Stopper Processor																		
435	VF-100a	Vial Rinser - Fill																		
436	VF-101a	Vial Rinser - Fill																		
437	VF-102a	Vial Rinser - Fill																		

WFI Water Offtake Input Data							WFI Water Offtake Input Data							
1			11				12				13			
No.	Tag No.	Description of Offtake	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty
1		ORIGINAL PROCESS DATA												
2		1. BULK PROCESSING SUITE												
3	T-100	Pre-rinse												
4	T-100	Fill												
5	T-101	Pre-rinse												
6	T-101	Fill												
7	T-102	Pre-rinse												
8	T-102	Fill												
9	T-103	Pre-rinse												
10	T-103	Fill												
11	BT-100	Buffer Tank pre-rinse												
12	BT-100	Buffer Tank fill												
13	BT-101	Buffer Tank pre-rinse												
14	BT-101	Buffer Tank fill												
15	BT-102	Buffer Tank pre-rinse												
16	BT-102	Buffer Tank fill												
17	BT-103	Buffer Tank pre-rinse												
18	BT-103	Buffer Tank fill												
19	N/A	Line Rinsing	19:40	19:55			22:00	22:15						
20														
21		2. FILLING SUITE												
22	PW-100	Parts Wash pre-rinse												
23	PW-100	Fill Parts Washer												
24	PW-101	Parts Wash pre-rinse												
25	PW-101	Fill Parts Washer												
26	PW-102	Parts Wash pre-rinse												
27	PW-102	Fill Parts Washer												
28	ST-100	Stopper Washer & Steriliser pre-rinse												
29	ST-100	Stopper Processor												
30	ST-101	Stopper Washer & Steriliser pre-rinse												
31	ST-101	Stopper Processor												
32	ST-102	Stopper Washer & Steriliser pre-rinse												
33	ST-102	Stopper Processor												
34	ST-103	Stopper Washer & Steriliser pre-rinse												
35	ST-103	Stopper Processor												
36	ST-104	Stopper Washer & Steriliser pre-rinse												
37	ST-104	Stopper Processor												
38	VF-100	Vial Rinser - Fill												
39	VF-101	Vial Rinser - Fill												
40	VF-102	Vial Rinser - Fill												
41	VF-103	Vial Rinser - Fill												
42	PW-100	Parts washer Sanitization												
43	PW-101	Parts washer Sanitization												
44	PW-102	Parts washer Sanitization												
45	ST-100	Stopper Processor Sanitation												
46	ST-101	Stopper Processor Sanitation												
47	ST-102	Stopper Processor Sanitation												
48	ST-103	Stopper Processor Sanitation												
49	ST-104	Stopper Processor Sanitation												
50	VF-100	Vial rinser Sanitisation												
51	VF-101	Vial rinser Sanitisation												
52	VF-102	Vial rinser Sanitisation												
53	VF-103	Vial rinser Sanitisation												
54														
55		2. WASHING SUITE												
56	PW-100	Parts Washer Pre Rinse												
57	PW-101	Parts Washer Pre Rinse												
58	PW-102	Parts Washer Pre Rinse												
59	PW-100	Parts Washer Sanitization												
60	PW-101	Parts Washer Sanitization												
61	PW-102	Parts Washer Sanitization												
62														
63		4. CIP SUITE												
64	T-100	T-100												
65	T-101	T-101												
66	T-102	T-102												
67	T-103	T-103 RESCHEDULE from 6:00												
68	VF-100	Vial Filler												
69	VF-101	Vial Filler												
70	VF-102	Vial Filler												
71	VF-103	Vial Filler												
72	LYO-100	Lyophyliser												
73	LYO-101	Lyophyliser												
74	LYO-102	Lyophyliser												
75	LYO-103	Lyophyliser												

116													
117		**SECOND INCREASE IN PRODUCTION**											
118		**ADD ANOTHER FILLING SUITE**											
119		**STAGGER BY 2 HOURS**											
120													
121		**2. FILLING SUITE**											
122	PW-100a	Parts Wash pre-rinse											
123	PW-100a	Fill Parts Washer											
124	PW-101a	Parts Wash pre-rinse											
125	PW-101a	Fill Parts Washer											
126	PW-102a	Parts Wash pre-rinse											
127	PW-102a	Fill Parts Washer											
128	ST-100a	Stopper Washer & Steriliser pre-rinse											
129	ST-100a	Stopper Processor											
130	ST-101a	Stopper Washer & Steriliser pre-rinse											
131	ST-101a	Stopper Processor											
132	ST-102a	Stopper Washer & Steriliser pre-rinse											
133	ST-102a	Stopper Processor											
134	ST-103a	Stopper Washer & Steriliser pre-rinse											
135	ST-103a	Stopper Processor											
136	ST-104a	Stopper Washer & Steriliser pre-rinse											
137	ST-104a	Stopper Processor											
138	VF-100a	Vial Rinser - Fill											
139	VF-101a	Vial Rinser - Fill											
140	VF-102a	Vial Rinser - Fill											
141	VF-103a	Vial Rinser - Fill											
142													
143	PW-101a	Parts washer Sanitization											
144													
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22:00											
146	ST-101a	Stopper Processor Sanitation											
147													
148	ST-103a	Stopper Processor Sanitation											
149	ST-104a	Stopper Processor Sanitation											
150	VF-100a	Vial rinser Sanitisation											
151	VF-101a	Vial rinser Sanitisation											
152	VF-102a	Vial rinser Sanitisation											
153	VF-103a	Vial rinser Sanitisation											
154													
155													
156		The second Filling Suite requires additional Bulk											
157		Processing, Washing & CIP from the existing											
158		Suites as follows: 2 hours staggered											
159													
160		**1. BULK PROCESSING SUITE**											
161	T-100	Pre-rinse											
162	T-100	Fill											
163	T-101	Pre-rinse											
164	T-101	Fill											
165	T-102	Pre-rinse											
166	T-102	Fill											
167	T-103	Pre-rinse											
168	T-103	Fill											
169	BT-100	Buffer Tank pre-rinse											
170	BT-100	Buffer Tank fill											
171	BT-101	Buffer Tank pre-rinse											
172	BT-101	Buffer Tank fill											
173	BT-102	Buffer Tank pre-rinse											
174	BT-102	Buffer Tank fill											
175	BT-103	Buffer Tank pre-rinse											
176	BT-103	Buffer Tank fill											
177	N/A	Line Rinsing	21:40	21:55									
178													
179													
180		**2. WASHING SUITE**											
181	PW-100	Parts Washer Pre Rinse											
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00											
183	PW-102	Parts Washer Pre Rinse											
184	PW-100	Parts Washer Sanitization											
185	PW-101	Parts Washer Sanitization											
186	PW-102	Parts Washer Sanitization											
187													
188		**4. CIP SUITE**											
189	T-100	T-100											
190	T-101	T-101											
191	T-102	T-102 RESCHEDULE from 6:15											
192	T-103	T-103											

Appendix 7 Input Data Set for the Stochastic Simulations

This appendix contains the input data sets used for the various stochastic DI/WFI simulations.

WFI Water Offtake Input Data										WFI Water Offtake Input Data								ter Offtake Input Data			
1										2				3				4			
No.	Tag No.	Description of Offtake	Water Offtake [m^3/hr]	Max. Uncertainty [%]	Description of Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Max.Time Uncertainty Time Open [min]	Description of Time Uncertainty
1		**ORIGINAL PROCESS DATA**																			
2		**1. BULK PROCESSING SUITE**																			
3	T-100	Pre-rinse	1.5	1	u	2:00	2:05	0:30	u	10:00	10:35	0:20	u	18:10	18:15	0:20	u				
4	T-100	Fill	1	1	u	2:00	2:06	0:20	u	10:30	10:36	0:20	n	18:10	18:15	0:20	n				
5	T-101	Pre-rinse	1.5	1	u	3:10	3:17	0:30	u	11:25	11:45	0:30	n	19:20	19:40	0:30	n				
6	T-101	Fill	1	1	u	3:15	3:25	0:30	u	11:15	11:25	0:30	u	19:30	19:40	0:30	u				
7	T-102	Pre-rinse	1.5	1	u	1:20	1:30	0:20	u	9:30	9:40	0:20	n	17:10	17:20	0:20	n				
8	T-102	Fill	1	1	u	3:15	3:25	0:20	u	10:15	10:20	0:20	n	18:15	18:20	0:20	n				
9	T-103	Pre-rinse	1.5	1	u	1:00	1:10	0:20	u	9:00	9:10	0:20	n	19:30	19:40	0:20	n				
10	T-103	Fill	1	1	u	6:00	6:30	0:20	u												
11	BT-100	Buffer Tank pre-rinse	1.5	1	u	1:15	1:20	0:20	u	9:00	9:05	0:20	u	17:30	17:35	0:20	u				
12	BT-100	Buffer Tank fill	1.5	1	u	2:00	2:10	0:20	u	10:00	10:10	0:20	u	19:00	19:10	0:20	u				
13	BT-101	Buffer Tank pre-rinse	1.5	1	u	2:30	3:00	0:20	u	10:30	11:00	0:20	n	19:30	20:00	0:20	u				
14	BT-101	Buffer Tank fill	1.5	1	u	3:00	3:30	0:20	u	11:00	11:30	0:20	n	20:00	20:30	0:20	n				
15	BT-102	Buffer Tank pre-rinse	1.5	1	u	21:00	21:05	0:20	u												
16	BT-102	Buffer Tank fill	1.5	1	u	21:15	21:20	0:20	u												
17	BT-103	Buffer Tank pre-rinse	1.5	1	u	2:00	2:05	0:20	n												
18	BT-103	Buffer Tank fill	1.5	1	u	0:30	0:45	0:20	n												
19	N/A	Line Rinsing	1	10	u	0:30	0:45	0:20	u	6:00	6:15	0:20	u	7:20	7:35	0:20	u	8:40	8:55	0:20	u
20																					
21		**2. FILLING SUITE**																			
22	PW-100	Parts Wash pre-rinse	2.5	10	u	0:30	0:40	0:20	u	11:00	11:10	0:20	u	16:00	16:10	0:20	u				
23	PW-100	Fill Parts Washer	5	10	u	1:30	2:10	0:30	u	12:30	12:50	0:30	u	17:30	17:55	0:30	u				
24	PW-101	Parts Wash pre-rinse	2.5	10	u	3:15	3:30	0:30	u	12:45	12:55	0:30	u	16:30	16:40	0:30	u				
25	PW-101	Fill Parts Washer	2.5	10	u	5:30	6:40	0:30	u	13:00	13:10	0:30	u	17:00	17:10	0:30	u				
26	PW-102	Parts Wash pre-rinse	2.5	10	u	7:30	7:40	0:30	u	11:15	11:25	0:30	u	16:15	16:25	0:30	u				
27	PW-102	Fill Parts Washer	2.5	10	u	8:00	8:10	0:30	u	13:00	13:10	0:30	u	17:00	17:10	0:30	u				
28	ST-100	Stopper Washer & Steriliser pre-rinse	2	1	u	4:00	4:10	0:30	u	12:00	12:10	0:30	u	18:30	18:40	0:30	u				
29	ST-100	Stopper Processor	3	1	n	5:00	6:00	0:30	n	13:00	14:00	0:30	n	19:00	20:00	0:30	n				
30	ST-101	Stopper Washer & Steriliser pre-rinse	2	1	n	6:00	6:10	0:30	n	14:00	14:10	0:30	n	20:00	20:10	0:30	n				
31	ST-101	Stopper Processor	3	1	n	5:00	6:00	0:30	n	15:00	16:00	0:30	n	21:00	22:00	0:30	n				
32	ST-102	Stopper Washer & Steriliser pre-rinse	2	1	n	6:00	6:10	0:30	n	14:00	14:10	0:30	n	22:00	22:10	0:30	n				
33	ST-102	Stopper Processor	3	1	n	7:00	8:10	0:30	n	15:30	16:30	0:30	n	23:00	23:59	0:30	n				
34	ST-103	Stopper Washer & Steriliser pre-rinse	2	1	n	0:35	0:40	0:30	n	2:00	2:10	0:30	n	11:00	11:10	0:30	n	13:00	13:40	0:30	n
35	ST-103	Stopper Processor	2	1	n	1:15	1:25	0:30	n	2:30	2:50	0:30	n	12:00	12:10	0:30	n	13:40	13:50	0:30	n
36	ST-104	Stopper Washer & Steriliser pre-rinse	2	1	n	2:15	2:25	0:30	n	3:30	3:40	0:30	n	13:00	13:10	0:30	n	14:30	14:50	0:30	n
37	ST-104	Stopper Processor	2	1	n	3:00	3:25	0:30	n	4:40	4:50	0:30	n	13:30	13:40	0:30	n	15:00	15:20	0:30	n
38	VF-100	Vial Rinser - Fill	0.5	1	n	0:15	0:25	0:10	n	1:00	1:30	0:10	n	2:00	2:30	0:10	n				
39	VF-101	Vial Rinser - Fill	0.5	1	n	6:00	6:35	0:10	n	7:00	7:35	0:10	n	8:00	8:30	0:10	n				
40	VF-102	Vial Rinser - Fill	0.5	1	n	10:00	10:35	0:10	n	11:00	11:35	0:10	n	12:00	12:30	0:10	n				
41	VF-103	Vial Rinser - Fill	0.5	1	n	20:00	20:35	0:10	n	21:00	21:35	0:10	n	22:00	22:30	0:10	n				
42	PW-100	Parts washer Sanitization	2.5	1	u	22:00	22:05	0:20	u												
43	PW-101	Parts washer Sanitization	2.5	1	u	21:00	21:05	0:20	u												
44	PW-102	Parts washer Sanitization	2.5	1	u	22:30	22:35	0:20	u												
45	ST-100	Stopper Processor Sanitation	2.5	1	u	20:00	20:05	0:20	u												
46	ST-101	Stopper Processor Sanitation	2.5	1	u	21:00	21:05	0:20	u												
47	ST-102	Stopper Processor Sanitation	2.5	1	u	22:00	22:05	0:20	u												
48	ST-103	Stopper Processor Sanitation	2.5	1	u	2:00	2:05	0:20	u												
49	ST-104	Stopper Processor Sanitation	2.5	1	u	5:00	5:05	0:20	u												
50	VF-100	Vial rinser Sanitisation	2.5	1	u	6:30	6:35	0:20	u	7:30	7:40	0:20	u								
51	VF-101	Vial rinser Sanitisation	2.5	1	u	4:45	4:50	0:20	u	5:45	5:55	0:20	u								
52	VF-102	Vial rinser Sanitisation	2.5	1	u	6:00	6:10	0:20	u	7:00	7:10	0:20	u								
53	VF-103	Vial rinser Sanitisation	2.5	1	u	3:00	3:10	0:20	u												
54																					
55		**2. WASHING SUITE**																			
56	PW-100	Parts Washer Pre Rinse	2.5	1	n	2:30	2:35	0:20	n	5:30	5:35	0:20	n	6:40	6:45	0:20	n	9:00	9:05	0:20	n
57	PW-101	Parts Washer Pre Rinse	5	1	n	4:00	5:15	0:20	n	12:30	12:45	0:20	n	16:00	16:15	0:20	n				
58	PW-102	Parts Washer Pre Rinse	2.5	1	n	5:00	5:30	0:20	n	6:30	7:00	0:20	n	10:30	11:00	0:20	n	13:00	13:30	0:20	n
59	PW-100	Parts Washer Sanitization	2.5	1	n	22:30	22:35	0:20	n												
60	PW-101	Parts Washer Sanitization	2.5	1	u	23:30	23:35	0:20	u												
61	PW-102	Parts Washer Sanitization	2.5	1	u	4:00	4:05	0:20	u												
62																					
63		**4. CIP SUITE**																			
64	T-100	T-100	5	1	u	2:00	2:30	1:00	u	8:15	8:30	1:00	u	16:15	16:30	1:00	u				
65	T-101	T-101	5	1	u	2:15	2:45	1:00	u	10:15	10:30	1:00	u	18:15	18:30	1:00	u				
66	T-102	T-102	5	1	u	4:15	4:30	1:00	u	12:15	12:30	1:00	u	20:15	20:30	1:00	u				
67	T-103	T-103 RESCHEDULE from 6:00	2.5	1	u	7:00	7:40	1:00	u	17:00	17:15	1:00	u								
68	VF-100	Vial Filler	2.5	1	u	23:00	23:40	1:00	u												
69	VF-101	Vial Filler	5	1	u	13:00	13:25	1:00	u												
70	VF-102	Vial Filler	5	1	u	17:00	17:25	1:00	u												
71	VF-103	Vial Filler	5	1	u	19:00	19:25	1:00	u												
72	LYO-100	Lyophyliser	2.5	1	u	5:00	5:30	1:00	u												
73	LYO-101	Lyophyliser	2.5	1	u	11:00	11:30	1:00	u												
74	LYO-102	Lyophyliser	2.5	1	u	21:00	21:30	1:00	u												
75	LYO-103	Lyophyliser	2.5	1	u	23:30	23:59	1:00	u												
76	LYO-104	Lyophyliser	2.5	1	u	23:30	23:59	1:00	u												
77	VFC-100	Vial Filler CIP skid	1	1	u	23:30	23:40	1:00	u												
78	VFC-101	Vial Filler CIP skid	1	1	u	23:00	23:10	1:00	u												
79	VFC-102	Vial Filler CIP skid	1	1	u	22:00	22:10	1:00	u												
80	VFC-103	Vial Filler CIP skid	1	1	u	21:30	21:40	1:00	u												
81	LYC-100	Lyophyliser CIP skid	1	1	u	21:00	21:10	1:00	u												
82	LYC-101	Lyophyliser CIP skid	1	1	u	20:00	20:10	1:00	u												
83	LYC-102	Lyophyliser CIP skid	1	1	u	3:00	3:05	1:00	u												
84	LYC-103	Lyophyliser CIP skid	1	1	u	4:00	4:05	1:00	u												
85																					
86																					
87		**FIRST INCREASE IN PRODUCTION**																			
88		**BY MAXIMISING USE OF EQUIPMENT**																			
89																					
90		**1. BULK PROCESSING SUITE**																			
91	T-100	Fill	1	1	u	2:00	2:06	0:20	u	10:30	10:36	0:20	n	18:10	18:15	0:20	n				
92	T-101	Fill	1	1	u	3:15	3:25	0:30	u	11:15	11:25	0:30	u	19:30	19:40	0:30	u				
93	T-102	Fill	1	1	u	3:15	3:25	0:20	u	10:15	10:20	0:20	n	18:15	18:20	0:20	n				
94	T-103	Fill	1	1	u	6:00	6:30	0:20	u												
95	BT-100	Buffer Tank fill	1.5	1	u	2:00	2:10	0:20	u	10:00	10:10	0:20	u	19:00	19:10	0:20	u				
96	BT-101	Buffer Tank fill	1.5	1	u	3:00	3:30	0:20	u	11:00	11:30	0:20	n	20:00	20:30	0:20	n				
97	BT-102	Buffer Tank fill	1.5	1	u	21:15	21:20	0:20	u												
98	BT-103	Buffer Tank fill	1.5	1	u	0:30	0:45	0:20	n												
99																					
100		**2. FILLING SUITE**																			
101	ST-100	Stopper Washer & Steriliser pre-rinse	0.5	1	u	4:00	4:10	0:30	u	12:00	12:10	0:30	u	18:30	18:40	0:30	u				
102	ST-100	Stopper Processor	2	1	n	5:00	6:00	0:30	n	13:00	14:00	0:30	n	19:00	20:00	0:30	n				
103	ST-101	Stopper Washer & Steriliser pre-rinse	0.5	1	n	6:00	6:10	0:30	n	14:00	14:10	0:30	n	20:00	20:10	0:30	n				
104	ST-101	Stopper Processor	2	1	n	5:00	6:00	0:30	n	15:00	16:00	0:30	n	21:00	22:00	0:30	n				
105	ST-102	Stopper Washer & Steriliser pre-rinse	0.5	1	n	6:00	6:10	0:30	n	14:00	14:10	0:30	n	22:00	22:10	0:30	n				
106	ST-102	Stopper Processor	2	1	n	7:00	8:10	0:30	n	15:30	16:30	0:30	n	23:00	23:59	0:30	n				
107	ST-103	Stopper Washer & Steriliser pre-rinse	0.5	1	n	0:35	0:40	0:30	n	2:00	2:10	0:30	n	11:00	11:10	0:30	n	13:00	13:40	0:30	n
108	ST-103	Stopper Processor	0.5	1	n	1:15	1:25	0:30	n	2:30	2:40	0:30	n	12:00	12:10	0:30	n	13:40	13:50	0:30	n
109	ST-104	Stopper Washer & Steriliser pre-rinse	0.5	1	n	2:15	2:25	0:30	n	3:30	3:40	0:30	n	13:00	13:10	0:30	n	14:30	14:50	0:30	n
110	ST-104	Stopper Processor	0.5	1	n	3:00	3:25	0:30	n	4:30	5:10	0:30	n	13:30	13:40	0:30	n	15:00	15:10	0:30	n
111	VF-100	Vial Rinser - Fill	0.5	1	n	0:15	0:30	0:10	n	1:30	2:00	0:10	n	3:00	3:30	0:10	n				
112	VF-101	Vial Rinser - Fill	0.5	1	n	6:00	6:35	0:10	n	7:00	7:30	0:10	n	8:00	8:30	0:10	n				
113	VF-102	Vial Rinser - Fill	0.5	1	n	10:00	10:35	0:10	n	11:00	11:35	0:10	n	12:00	12:30	0:10	n				
114	VF-103	Vial Rinser - Fill	0.5	1	n	20:00	20:35	0:10	n	21:00	21:35	0:10	n	22:00	22:30	0:10	n				
115																					

116																					
117		**SECOND INCREASE IN PRODUCTION**																			
118		**ADD ANOTHER FILLING SUITE**																			
119		**RESCHEDULE BY 2 HOURS (STAGGERED)**																			
120																					
121		**2. FILLING SUITE**																			
122	PW-100a	Parts Wash pre-rinse	2.5	10	u	2:40	2:50	0:30	u	13:00	13:10	0:30	u	18:00	18:10	0:30	u				
123	PW-100a	Fill Parts Washer	5	10	u	3:30	4:10	0:30	u	14:30	14:50	0:30	u	19:30	19:55	0:30	u				
124	PW-101a	Parts Wash pre-rinse	2.5	10	u	5:15	5:30	0:30	u	14:45	14:55	0:30	u	18:30	18:40	0:30	u				
125	PW-101a	Fill Parts Washer	2.5	10	u	7:30	7:40	0:30	u	15:00	15:10	0:30	u	19:00	19:10	0:30	u				
126	PW-102a	Parts Wash pre-rinse	2.5	10	u	9:30	9:40	0:30	u	13:15	13:25	0:30	u	18:15	18:25	0:30	u				
127	PW-102a	Fill Parts Washer	2.5	10	u	10:00	10:10	0:30	u	15:00	15:10	0:30	u	19:00	19:10	0:30	u				
128	ST-100a	Stopper Washer & Steriliser pre-rinse	2	1	u	6:00	6:10	0:30	u	14:00	14:10	0:30	u	20:30	20:40	0:30	u				
129	ST-100a	Stopper Processor	3	1	n	7:00	8:00	0:30	n	15:00	16:00	0:30	n	21:00	22:00	0:30	n				
130	ST-101a	Stopper Washer & Steriliser pre-rinse	2	1	n	8:00	8:10	0:30	n	16:00	16:10	0:30	n	22:00	22:10	0:30	n				
131	ST-101a	Stopper Processor	3	1	n	7:00	8:00	0:30	n	17:00	18:00	0:30	n	23:00	23:25	0:30	n				
132	ST-102a	Stopper Washer & Steriliser pre-rinse	2	1	n	8:00	8:10	0:30	n	16:00	16:10	0:30	n								
133	ST-102a	Stopper Processor	3	1	n	9:00	10:10	0:30	n	17:30	18:30	0:30	n								
134	ST-103a	Stopper Washer & Steriliser pre-rinse	2	1	n	2:35	2:40	0:30	n	4:00	4:10	0:30	n	13:00	13:10	0:30	n	15:00	15:40	0:30	n
135	ST-103a	Stopper Processor	2	1	n	3:15	3:25	0:30	n	4:30	4:40	0:30	n	14:00	14:10	0:30	n	15:40	15:50	0:30	n
136	ST-104a	Stopper Washer & Steriliser pre-rinse	2	1	n	4:15	4:25	0:30	n	5:30	5:40	0:30	n	15:00	15:10	0:30	n	16:30	16:50	0:30	n
137	ST-104a	Stopper Processor	2	1	n	5:00	5:25	0:30	n	6:30	7:10	0:30	n	15:30	15:40	0:30	n	16:50	17:00	0:30	n
138	VF-100a	Vial Rinser - Fill	0.5	1	n	2:15	2:25	0:10	n	3:00	3:30	0:10	n	4:00	4:30	0:10	n				
139	VF-101a	Vial Rinser - Fill	0.5	1	n	8:00	8:35	0:10	n	9:00	9:35	0:10	n	10:00	10:30	0:10	n				
140	VF-102a	Vial Rinser - Fill	0.5	1	n	12:00	12:35	0:10	n	13:00	13:35	0:10	n	14:00	14:30	0:10	n				
141	VF-103a	Vial Rinser - Fill	0.5	1	n	22:00	22:35	0:10	n	23:00	23:35	0:10	n								
142																					
143	PW-101a	Parts washer Sanitization	2.5	1	u	23:00	23:05	0:20	u												
144																					
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22	2.5	1	u	23:00	23:05	0:20	u												
146	ST-101a	Stopper Processor Sanitation	2.5	1	u	23:00	23:05	0:20	u												
147																					
148	ST-103a	Stopper Processor Sanitation	2.5	1	u	4:00	4:05	0:20	u												
149	ST-104a	Stopper Processor Sanitation	2.5	1	u	7:00	7:05	0:20	u												
150	VF-100a	Vial rinser Sanitisation	2.5	1	u	8:30	8:35	0:20	u	9:30	9:40	0:20	u								
151	VF-101a	Vial rinser Sanitisation	2.5	1	u	6:45	6:50	0:20	u	7:45	7:55	0:20	u								
152	VF-102a	Vial rinser Sanitisation	2.5	1	u	8:00	8:10	0:20	u	9:00	9:10	0:20	u								
153	VF-103a	Vial rinser Sanitisation	2.5	1	u	5:00	5:10	0:20	u												
154																					
155																					
156		The second Filling Suite requires additional Bulk																			
157		Processing, Washing & CIP from the existing																			
158		Suites as follows: 2 hours staggered																			
159																					
160		**1. BULK PROCESSING SUITE**																			
161	T-100	Pre-rinse	1.5	1	u	4:00	4:05	0:20	u	12:00	12:35	0:20	u	20:10	20:15	0:20	u				
162	T-100	Fill	1	1	u	4:00	4:06	0:20	u	12:30	12:36	0:20	n	20:10	20:15	0:20	n				
163	T-101	Pre-rinse	1.5	1	u	5:10	5:17	0:30	u	13:25	13:45	0:30	n	21:20	21:40	0:30	n				
164	T-101	Fill	1	1	u	5:15	5:25	0:30	u	13:15	13:25	0:30	u	21:30	21:40	0:30	u				
165	T-102	Pre-rinse	1.5	1	u	3:20	3:30	0:20	u	11:30	11:40	0:20	n	19:10	19:20	0:20	n				
166	T-102	Fill	1	1	u	5:15	5:25	0:20	u	12:15	12:20	0:20	n	20:15	21:20	0:20	n				
167	T-103	Pre-rinse	1.5	1	u	3:00	3:10	0:20	u	11:00	11:10	0:20	n	21:30	21:40	0:20	n				
168	T-103	Fill	1	1	u	8:00	8:30	0:20	u												
169	BT-100	Buffer Tank pre-rinse	1.5	1	u	3:15	3:20	0:20	u	11:00	11:05	0:20	u	19:30	19:35	0:20	u				
170	BT-100	Buffer Tank fill	1.5	1	u	4:00	4:10	0:20	u	12:00	12:10	0:20	u	21:00	21:10	0:20	u				
171	BT-101	Buffer Tank pre-rinse	1.5	1	u	4:30	5:00	0:20	u	12:30	13:00	0:20	n	21:30	22:00	0:20	u				
172	BT-101	Buffer Tank fill	1.5	1	u	5:00	5:30	0:20	u	13:00	13:30	0:20	n	22:00	22:30	0:20	n				
173	BT-102	Buffer Tank pre-rinse	1.5	1	u	23:00	23:05	0:20	u												
174	BT-102	Buffer Tank fill	1.5	1	u	23:15	23:20	0:20	u												
175	BT-103	Buffer Tank pre-rinse	1.5	1	u	4:00	4:05	0:20	n												
176	BT-103	Buffer Tank fill	1.5	1	u	2:30	2:45	0:20	n												
177	N/A	Line Rinsing	1	10	u	2:30	2:45	0:20	u	8:00	8:15	0:20	u	9:20	9:35	0:20	u	10:40	10:55	0:20	u
178																					
179																					
180		**2. WASHING SUITE**																			
181	PW-100	Parts Washer Pre Rinse	2.5	1	n	4:30	4:35	0:20	n	7:30	7:35	0:20	n	8:40	8:45	0:20	n	11:00	11:05	0:20	n
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00	5	1	n	6:30	7:45	0:20	n	14:30	14:45	0:20	n	18:00	18:15	0:20	n				
183	PW-102	Parts Washer Pre Rinse	2.5	1	n	7:00	7:30	0:20	n	8:30	9:00	0:20	n	10:30	12:00	0:20	n	14:00	15:30	0:20	n
184	PW-100	Parts Washer Sanitization																			
185	PW-101	Parts Washer Sanitization																			
186	PW-102	Parts Washer Sanitization	2.5	1	u	6:00	6:05	0:20	u												
187																					
188		**4. CIP SUITE**																			
189	T-100	T-100	5	1	u	4:00	4:30	1:00	u	10:15	10:30	1:00	u	18:15	18:30	1:00	u				
190	T-101	T-101	5	1	u	4:15	4:45	1:00	u	12:15	12:30	1:00	u	20:15	20:30	1:00	u				
191	T-102	T-102 RESCHEDULE from 6:15	5	1	u	8:00	8:15	1:00	u	14:15	14:30	1:00	u	22:15	22:30	1:00	u				
192	T-103	T-103	2.5	1	u	8:00	8:40	1:00	u	19:00	19:15	1:00	u								
193	VF-100	Vial Filler																			
194	VF-101	Vial Filler	5	1	u	15:00	15:25	1:00	u												
195	VF-102	Vial Filler	5	1	u	19:00	19:25	1:00	u												
196	VF-103	Vial Filler RESCHEDULE from 21:00	5	1	u	20:00	20:25	1:00	u												
197	LYO-100	Lyophyliser RESCHEDULE from 7:00	2.5	1	u	8:00	8:30	1:00	u												
198	LYO-101	Lyophyliser	2.5	1	u	13:00	13:30	1:00	u												
199	LYO-102	Lyophyliser	2.5	1	u	23:00	23:30	1:00	u												
200	LYO-103	Lyophyliser																			
201	LYO-104	Lyophyliser																			
202	VFC-100	Vial Filler CIP skid																			
203	VFC-101	Vial Filler CIP skid																			
204	VFC-102	Vial Filler CIP skid																			
205	VFC-103	Vial Filler CIP skid	1	1	u	23:30	23:40	1:00	u												
206	LYC-100	Lyophyliser CIP skid	1	1	u	23:00	23:10	1:00	u												
207	LYC-101	Lyophyliser CIP skid	1	1	u	22:00	22:10	1:00	u												
208	LYC-102	Lyophyliser CIP skid	1	1	u	5:00	5:05	1:00	u												
209	LYC-103	Lyophyliser CIP skid	1	1	u	6:00	6:05	1:00	u												
210																					
211																					
212		**THIRD INCREASE IN PRODUCTION**																			
213		**BY MAXIMISING USE OF EQUIPMENT**																			
214																					
215		**1. BULK PROCESSING SUITE**																			
216	T-100	Fill	1	1	u	4:00	4:06	0:20	u	12:30	12:36	0:20	n	20:10	20:15	0:20	n				
217	T-101	Fill	1	1	u	5:15	5:25	0:30	u	13:15	13:25	0:30	u	21:30	21:40	0:30	u				
218	T-102	Fill	1	1	u	5:15	5:25	0:20	u	12:15	12:20	0:20	n	20:15	21:20	0:20	n				
219	T-103	Fill	1	1	u	8:00	8:30	0:20	u												
220	BT-100	Buffer Tank fill	1.5	1	u	4:00	4:10	0:20	u	12:00	12:10	0:20	u	21:00	21:10	0:20	u				
221	BT-101	Buffer Tank fill	1.5	1	u	5:00	5:30	0:20	u	13:00	13:30	0:20	n	22:00	22:30	0:20	n				
222	BT-102	Buffer Tank fill	1.5	1	u	23:15	23:20	0:20	u												
223	BT-103	Buffer Tank fill	1.5	1	u	2:30	2:45	0:20	n												
224																					
225		**2. FILLING SUITE**																			
226	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5	1	u	6:00	6:10	0:30	u	14:00	14:10	0:30	u	20:30	20:40	0:30	u				
227	ST-100a	Stopper Processor	2	1	n	7:00	8:00	0:30	n	15:00	16:00	0:30	n	21:00	22:00	0:30	n				
228	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	8:00	8:10	0:30	n	16:00	16:10	0:30	n	22:00	22:10	0:30	n				
229	ST-101a	Stopper Processor	2	1	n	7:00	8:00	0:30	n	17:00	18:00	0:30	n	23:00	23:25	0:30	n				
230	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	8:00	8:10	0:30	n	16:00	16:10	0:30	n								
231	ST-102a	Stopper Processor	2	1	n	9:00	10:10	0:30	n	17:30	18:30	0:30	n								
232	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	2:35	2:40	0:30	n	4:00	4:10	0:30	n	13:00	13:10	0:30	n	15:00	15:40	0:30	n
233	ST-103a	Stopper Processor	0.5	1	n	3:15	3:25	0:30	n	4:30	4:40	0:30	n	14:00	14:10	0:30	n	15:40	15:50	0:30	n
234	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	4:15	4:25	0:30	n	5:30	5:40	0:30	n	15:00	15:10	0:30	n	16:30	16:50	0:30	n
235	ST-104a	Stopper Processor	0.5	1	n	5:00	5:25	0:30	n	6:30	7:10	0:30	n	15:30	15:40	0:30	n	16:50	17:00	0:30	n
236	VF-100a	Vial Rinser - Fill	0.5	1	n	2:15	2:25	0:10	n	3:00	3:30	0:10	n	4:00	4:30	0:10	n				
237	VF-101a	Vial Rinser - Fill	0.5	1	n	8:00	8:35	0:10	n	9:00	9:35	0:10	n	10:00	10:30	0:10	n				
238	VF-102a	Vial Rinser - Fill	0.5	1	n	12:00	12:35	0:10	n	13:00	13:35	0:10	n	14:00	14:30	0:10	n				
239	VF-103a	Vial Rinser - Fill	0.5	1	n	22:00	22:35	0:10	n	23:00	23:35	0:10	n								

244		**FOURTH INCREASE IN PRODUCTION**																			
245		**ADD THIRD FILLING SUITE**																			
246		**RESCHEDULE BY 2 HOURS (STAGGERED)**																			
247																					
248		**2. FILLING SUITE**																			
249	PW-100a	Parts Wash pre-rinse	2.5	10	u	4:40	4:50	0:30	u	15:00	15:10	0:30	u	20:00	20:10	0:30	u				
250	PW-100a	Fill Parts Washer	5	10	u	5:30	6:10	0:30	u	16:30	16:50	0:30	u	21:30	21:55	0:30	u				
251	PW-101a	Parts Wash pre-rinse	2.5	10	u	7:15	7:30	0:30	u	16:45	16:55	0:30	u	20:30	22:40	0:30	u				
252	PW-101a	Fill Parts Washer	2.5	10	u	9:30	9:40	0:30	u	17:00	17:10	0:30	u	21:00	23:10	0:30	u				
253	PW-102a	Parts Wash pre-rinse	2.5	10	u	11:30	11:40	0:30	u	15:15	17:25	0:30	u	20:15	20:25	0:30	u				
254	PW-102a	Fill Parts Washer	2.5	10	u	12:00	12:10	0:30	u	17:00	17:10	0:30	u	21:00	21:10	0:30	u				
255	ST-100a	Stopper Washer & Steriliser pre-rinse	2	1	u	8:00	8:10	0:30	u	16:00	18:10	0:30	u	22:30	22:40	0:30	u				
256	ST-100a	Stopper Processor	3	1	n	9:00	10:00	0:30	n	17:00	18:00	0:30	n								
257	ST-101a	Stopper Washer & Steriliser pre-rinse	2	1	n	11:00	11:10	0:30	n	18:00	18:10	0:30	n								
258	ST-101a	Stopper Processor	3	1	n	9:00	10:00	0:30	n	19:00	20:00	0:30	n								
259	ST-102a	Stopper Washer & Steriliser pre-rinse	2	1	n	10:00	10:10	0:30	n	18:00	18:10	0:30	n								
260	ST-102a	Stopper Processor	3	1	n	11:00	12:10	0:30	n	19:30	20:30	0:30	n								
261	ST-103a	Stopper Washer & Steriliser pre-rinse	2	1	n	5:35	5:40	0:30	n	6:50	7:05	0:30	n	15:00	15:10	0:30	n	17:00	17:40	0:30	n
262	ST-103a	Stopper Processor	2	1	n	5:15	5:25	0:30	n	6:30	6:40	0:30	n	16:00	16:10	0:30	n	17:40	17:50	0:30	n
263	ST-104a	Stopper Washer & Steriliser pre-rinse	2	1	n	6:15	6:25	0:30	n	7:30	7:40	0:30	n	17:00	17:10	0:30	n	18:30	18:50	0:30	n
264	ST-104a	Stopper Processor	2	1	n	7:00	7:25	0:30	n	8:30	9:10	0:30	n	17:30	17:40	0:30	n	18:50	19:00	0:30	n
265	VF-100a	Vial Rinser - Fill	0.5	1	n	4:15	4:25	0:10	n	5:00	5:30	0:10	n	6:00	6:30	0:10	n				
266	VF-101a	Vial Rinser - Fill	0.5	1	n	10:00	10:35	0:10	n	11:00	11:35	0:10	n	12:00	12:30	0:10	n				
267	VF-102a	Vial Rinser - Fill	0.5	1	n	14:00	14:35	0:10	n	15:00	15:35	0:10	n	16:00	16:30	0:10	n				
268																					
269																					
270	PW-101a	Parts washer Sanitization	2.5	1	u	1:00	1:05	0:20	u												
271																					
272	ST-100a	Stopper Processor Sanitation	2.5	1	u	1:00	1:05	0:20	u												
273	ST-101a	Stopper Processor Sanitation	2.5	1	u	0:30	0:35	0:20	u												
274																					
275	ST-103a	Stopper Processor Sanitation	2.5	1	u	6:00	6:05	0:20	u												
276	ST-104a	Stopper Processor Sanitation	2.5	1	u	9:00	9:05	0:20	u												
277	VF-100a	Vial rinser Sanitisation	2.5	1	u	10:30	10:35	0:20	u	11:30	11:40	0:20	u								
278	VF-101a	Vial rinser Sanitisation	2.5	1	u	8:45	8:50	0:20	u	9:45	9:55	0:20	u								
279	VF-102a	Vial rinser Sanitisation	2.5	1	u	10:00	10:10	0:20	u	11:00	11:10	0:20	u								
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00	2.5	1	u	8:00	8:10	0:20	u												
281																					
282																					
283																					
284																					
285																					
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																			
287		**1. BULK PROCESSING SUITE**																			
288	T-100	Pre-rinse	1.5	1	u	6:00	6:05	0:20	u	14:00	14:35	0:20	u	22:10	22:15	0:20	u				
289	T-100	Fill	1	1	u	6:00	6:06	0:20	u	14:30	14:36	0:20	n	22:10	22:15	0:20	n				
290	T-101	Pre-rinse	1.5	1	u	7:10	7:17	0:30	u	15:25	15:45	0:30	n	23:20	23:40	0:30	n				
291	T-101	Fill	1	1	u	7:15	7:25	0:30	u	15:15	15:25	0:30	u	23:30	23:40	0:30	u				
292	T-102	Pre-rinse	1.5	1	u	5:20	5:30	0:20	u	13:30	13:40	0:20	n	21:10	21:20	0:20	n				
293	T-102	Fill	1	1	u	7:15	7:25	0:20	u	14:15	14:20	0:20	n	22:15	22:20	0:20	n				
294	T-103	Pre-rinse	1.5	1	u	5:00	5:10	0:20	u	13:00	13:10	0:20	n	23:30	23:40	0:20	n				
295	T-103	Fill	1	1	u	11:00	11:30	0:20	u												
296	BT-100	Buffer Tank pre-rinse	1.5	1	u	5:15	5:20	0:20	u	13:00	13:05	0:20	u	21:30	21:35	0:20	u				
297	BT-100	Buffer Tank fill	1.5	1	u	6:00	6:10	0:20	u	14:00	14:10	0:20	u	23:00	23:10	0:20	u				
298	BT-101	Buffer Tank pre-rinse	1.5	1	u	6:30	7:00	0:20	u	14:30	15:00	0:20	n								
299	BT-101	Buffer Tank fill	1.5	1	u	7:00	7:30	0:20	u	15:00	15:30	0:20	n								
300	BT-102	Buffer Tank pre-rinse																			
301	BT-102	Buffer Tank fill																			
302	BT-103	Buffer Tank pre-rinse	1.5	1	u	6:00	6:05	0:20	n												
303	BT-103	Buffer Tank fill	1.5	1	u	4:30	4:45	0:20	n												
304	N/A	Line Rinsing	1	10	u	4:30	4:45	0:20	u	10:00	10:15	0:20	u	11:20	11:35	0:20	u	12:40	12:55	0:20	u
305																					
306																					
307																					
308																					
309		**2. WASHING SUITE**																			
310	PW-100	Parts Washer Pre Rinse	2.5	1	n	6:30	6:35	0:20	n	9:30	9:35	0:20	n	10:40	10:45	0:20	n	13:00	13:05	0:20	n
311	PW-101	Parts Washer Pre Rinse	5	1	n	8:00	9:15	0:20	n	16:30	16:45	0:20	n	20:00	20:15	0:20	n				
312	PW-102	Parts Washer Pre Rinse	2.5	1	n	9:00	9:30	0:20	n	10:30	11:00	0:20	n	14:30	15:00	0:20	n	16:00	16:30	0:20	n
313	PW-100	Parts Washer Sanitization																			
314	PW-101	Parts Washer Sanitization																			
315	PW-102	Parts Washer Sanitization	2.5	1	u	10:00	10:05	0:20	u												
316																					
317		**4. CIP SUITE**																			
318	T-100	T-100	5	1	u	6:00	6:30	1:00	u	12:15	12:30	1:00	u	20:15	20:30	1:00	u				
319	T-101	T-101	5	1	u	6:15	6:45	1:00	u	16:15	16:30	1:00	u	22:15	22:30	1:00	u				
320	T-102	T-102	5	1	u	1:15	1:30	1:00	u	8:15	8:30	1:00	u	16:15	16:30	1:00	u				
321	T-103	T-103	2.5	1	u	10:00	10:40	1:00	u	21:00	21:15	1:00	u								
322	VF-100	Vial Filler	2.5	1	u	3:00	3:40	1:00	u												
323	VF-101	Vial Filler	5	1	u	17:00	17:25	1:00	u												
324	VF-102	Vial Filler RESCHEDULE from 21:00	5	1	u	20:00	20:25	1:00	u												
325	VF-103	Vial Filler	5	1	u	23:00	23:25	1:00	u												
326	LYO-100	Lyophyliser	2.5	1	u	9:00	9:30	1:00	u												
327	LYO-101	Lyophyliser	2.5	1	u	15:00	15:30	1:00	u												
328	LYO-102	Lyophyliser	2.5	1	u	1:30	2:00	1:00	u												
329	LYO-103	Lyophyliser	2.5	1	u	2:30	2:59	1:00	u												
330	LYO-104	Lyophyliser	2.5	1	u	2:30	2:59	1:00	u												
331	VFC-100	Vial Filler CIP skid	1	1	u	2:30	2:40	1:00	u												
332	VFC-101	Vial Filler CIP skid	1	1	u	2:00	2:10	1:00	u												
333	VFC-102	Vial Filler CIP skid	1	1	u	3:00	3:10	1:00	u												
334	VFC-103	Vial Filler CIP skid	1	1	u	1:30	1:40	1:00	u												
335	LYC-100	Lyophyliser CIP skid	1	1	u	1:20	1:30	1:00	u												
336	LYC-101	Lyophyliser CIP skid	1	1	u	1:15	1:25	1:00	u												
337	LYC-102	Lyophyliser CIP skid	1	1	u	7:00	7:05	1:00	u												
338	LYC-103	Lyophyliser CIP skid	1	1	u	8:00	8:05	1:00	u												

342		**FIFTH INCREASE IN PRODUCTION**																			
343		**BY MAXIMISING USE OF EQUIPMENT**																			
344																					
345		**1. BULK PROCESSING SUITE**																			
346	T-100	Fill	1	1	u	6:00	6:06	0:20	u	14:30	14:36	0:20	n	22:10	22:15	0:20	n				
347	T-101	Fill	1	1	u	7:15	7:25	0:30	u	15:15	15:25	0:30	u	23:30	23:40	0:30	u				
348	T-102	Fill	1	1	u	5:15	5:25	0:20	u	12:15	12:20	0:20	n	20:15	21:20	0:20	n				
349	T-103	Fill	1	1	u	7:15	7:25	0:20	u	14:15	14:20	0:20	n	22:15	22:20	0:20	n				
350	BT-100	Buffer Tank fill	1.5	1	u	6:00	6:10	0:20	u	14:00	14:10	0:20	u	23:00	23:10	0:20	u				
351	BT-101	Buffer Tank fill	1.5	1	u	7:00	7:30	0:20	u	15:00	15:30	0:20	n								
352	BT-102	Buffer Tank fill																			
353	BT-103	Buffer Tank fill	1.5	1	u	6:00	6:05	0:20	n												
354																					
355		**2. FILLING SUITE**																			
356	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5	1	u	8:00	8:10	0:30	u	16:00	18:10	0:30	u	22:30	22:40	0:30	u				
357	ST-100a	Stopper Processor	2	1	n	9:00	10:00	0:30	n	17:00	18:00	0:30	n								
358	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	11:00	11:10	0:30	n	18:00	18:10	0:30	n								
359	ST-101a	Stopper Processor	2	1	n	9:00	10:00	0:30	n	19:00	20:00	0:30	n								
360	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	10:00	10:10	0:30	n	18:00	18:10	0:30	n								
361	ST-102a	Stopper Processor	2	1	n	11:00	12:10	0:30	n	19:30	20:30	0:30	n								
362	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	5:35	5:40	0:30	n	6:50	7:05	0:30	n	15:00	15:10	0:30	n	17:00	17:40	0:30	n
363	ST-103a	Stopper Processor	0.5	1	n	5:15	5:25	0:30	n	6:30	6:40	0:30	n	16:00	16:10	0:30	n	17:40	17:50	0:30	n
364	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	6:15	6:25	0:30	n	7:30	7:40	0:30	n	17:00	17:10	0:30	n	18:30	18:50	0:30	n
365	ST-104a	Stopper Processor	0.5	1	n	7:00	7:25	0:30	n	8:30	9:10	0:30	n	17:30	17:40	0:30	n	18:50	19:00	0:30	n
366	VF-100a	Vial Rinser - Fill	0.5	1	n	4:15	4:25	0:10	n	5:00	5:30	0:10	n	6:00	6:30	0:10	n				
367	VF-101a	Vial Rinser - Fill	0.5	1	n	10:00	10:35	0:10	n	11:00	11:35	0:10	n	12:00	12:30	0:10	n				
368	VF-102a	Vial Rinser - Fill	0.5	1	n	14:00	14:35	0:10	n	15:00	15:35	0:10	n	16:00	16:30	0:10	n				
369			0.5																		
370																					
371		**SIXTH INCREASE IN PRODUCTION**																			
372		**ADD FOURTH FILLING SUITE**																			
373		**RESCHEDULE BY 2 HOURS (STAGGERED)**																			
374																					
375		**2. FILLING SUITE**																			
376	PW-100a	Parts Wash pre-rinse	2.5	10	u	6:40	6:50	0:30	u	17:00	17:10	0:30	u	22:00	22:10	0:30	u				
377	PW-100a	Fill Parts Washer	5	10	u	7:30	8:10	0:30	u	18:30	18:50	0:30	u								
378	PW-101a	Parts Wash pre-rinse	2.5	10	u	9:15	9:30	0:30	u	18:45	18:55	0:30	u	22:30	23:55	0:30	u				
379	PW-101a	Fill Parts Washer	2.5	10	u	11:30	11:40	0:30	u	19:00	19:10	0:30	u	23:00	23:55	0:30	u				
380	PW-102a	Parts Wash pre-rinse	2.5	10	u	13:30	13:40	0:30	u	19:15	19:25	0:30	u	22:15	23:55	0:30	u				
381	PW-102a	Fill Parts Washer	2.5	10	u	14:00	14:10	0:30	u	19:00	19:10	0:30	u	23:00	23:55	0:30	u				
382	ST-100a	Stopper Washer & Steriliser pre-rinse	2	1	u	10:00	10:10	0:30	u	18:00	20:10	0:30	u								
383	ST-100a	Stopper Processor	3	1	n	11:00	12:00	0:30	n	19:00	20:00	0:30	n								
384	ST-101a	Stopper Washer & Steriliser pre-rinse	2	1	n	13:00	13:10	0:30	n	20:00	20:10	0:30	n								
385	ST-101a	Stopper Processor	3	1	n	11:00	12:00	0:30	n	21:00	22:00	0:30	n								
386	ST-102a	Stopper Washer & Steriliser pre-rinse	2	1	n	12:00	12:10	0:30	n	20:00	20:10	0:30	n								
387	ST-102a	Stopper Processor	3	1	n	13:00	14:10	0:30	n	21:30	22:30	0:30	n								
388	ST-103a	Stopper Washer & Steriliser pre-rinse	2	1	n	7:35	7:40	0:30	n	8:50	9:05	0:30	n	17:00	17:10	0:30	n	19:00	19:40	0:30	n
389	ST-103a	Stopper Processor	2	1	n	7:15	7:25	0:30	n	8:30	8:40	0:30	n	18:00	18:10	0:30	n	19:40	19:50	0:30	n
390	ST-104a	Stopper Washer & Steriliser pre-rinse	2	1	n	8:15	8:25	0:30	n	9:30	9:40	0:30	n	19:00	19:10	0:30	n	20:30	20:50	0:30	n
391	ST-104a	Stopper Processor	2	1	n	9:00	9:25	0:30	n	10:30	11:10	0:30	n	19:30	19:40	0:30	n	20:50	21:00	0:30	n
392	VF-100a	Vial Rinser - Fill	0.5	1	n	6:15	6:25	0:10	n	7:00	7:30	0:10	n	8:00	8:30	0:10	n				
393	VF-101a	Vial Rinser - Fill	0.5	1	n	12:00	12:35	0:10	n	13:00	13:35	0:10	n	14:00	14:30	0:10	n				
394	VF-102a	Vial Rinser - Fill	0.5	1	n	16:00	16:35	0:10	n	17:00	17:35	0:10	n	18:00	18:30	0:10	n				
395																					
396																					
397	PW-101a	Parts washer Sanitization	2.5	1	u	3:00	3:05	0:20	u												
398																					
399	ST-100a	Stopper Processor Sanitation	2.5	1	u	3:00	3:05	0:20	u												
400	ST-101a	Stopper Processor Sanitation	2.5	1	u	2:30	2:35	0:20	u												
401																					
402	ST-103a	Stopper Processor Sanitation	2.5	1	u	8:00	8:05	0:20	u												
403	ST-104a	Stopper Processor Sanitation	2.5	1	u	11:00	11:05	0:20	u												
404	VF-100a	Vial rinser Sanitisation	2.5	1	u	12:30	12:35	0:20	u	13:30	15:40	0:20	u								
405	VF-101a	Vial rinser Sanitisation	2.5	1	u	10:45	10:50	0:20	u	11:45	11:55	0:20	u								
406	VF-102a	Vial rinser Sanitisation	2.5	1	u	12:00	12:10	0:20	u	13:00	13:10	0:20	u								
407	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00	2.5	1																	
408																					
409																					
410																					
411		**SEVENTH INCREASE IN PRODUCTION**																			
412		**BY MAXIMISING USE OF EQUIPMENT**																			
413																					
414		**1. BULK PROCESSING SUITE**																			
415	T-100	Fill	1	1	u	7:00	8:06	0:20	u	16:30	16:36	0:20	n								
416	T-101	Fill	1	1	u	9:15	9:25	0:30	u	17:15	17:25	0:30	u								
417	T-102	Fill	1	1	u	7:15	7:25	0:20	u	14:15	14:20	0:20	n	22:15	23:20	0:20	n				
418	T-103	Fill	1	1	u	9:15	9:25	0:20	u	16:15	16:20	0:20	n								
419	BT-100	Buffer Tank fill	1.5	1	u	8:00	8:10	0:20	u	16:00	16:10	0:20	u								
420	BT-101	Buffer Tank fill	1.5	1	u	9:00	9:30	0:20	u	17:00	17:30	0:20	n								
421	BT-102	Buffer Tank fill																			
422	BT-103	Buffer Tank fill	1.5	1	u	8:00	8:05	0:20	n												
423																					
424		**2. FILLING SUITE**																			
425	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5	1	u	10:00	10:10	0:30	u	18:00	18:10	0:30	u								
426	ST-100a	Stopper Processor	2	1	n	11:00	12:00	0:30	n	19:00	20:00	0:30	n								
427	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	13:00	13:10	0:30	n	20:00	20:10	0:30	n								
428	ST-101a	Stopper Processor	2	1	n	11:00	12:00	0:30	n	21:00	22:00	0:30	n								
429	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	12:00	12:10	0:30	n	20:00	20:10	0:30	n								
430	ST-102a	Stopper Processor	2	1	n	12:00	14:10	0:30	n	21:30	22:30	0:30	n								
431	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	7:35	7:40	0:30	n	8:50	9:05	0:30	n	17:00	17:10	0:30	n	19:00	19:40	0:30	n
432	ST-103a	Stopper Processor	0.5	1	n	7:15	7:25	0:30	n	8:30	8:40	0:30	n	18:00	18:10	0:30	n	19:40	19:50	0:30	n
433	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5	1	n	8:15	8:25	0:30	n	9:30	9:40	0:30	n	19:00	19:10	0:30	n	20:30	20:50	0:30	n
434	ST-104a	Stopper Processor	0.5	1	n	9:00	9:25	0:30	n	10:30	11:10	0:30	n	19:30	19:40	0:30	n	20:50	21:00	0:30	n
435	VF-100a	Vial Rinser - Fill	0.5	1	n	6:15	6:25	0:10	n	7:00	7:30	0:10	n	8:00	8:30	0:10	n				
436	VF-101a	Vial Rinser - Fill	0.5	1	n	12:00	12:35	0:10	n	13:00	13:35	0:10	n	14:00	14:30	0:10	n				
437	VF-102a	Vial Rinser - Fill	0.5	1	n	16:00	16:35	0:10	n	17:00	17:35	0:10	n	18:00	18:30	0:10	n				

WFI Water Offtake Input Data

			5				6				7				8			
No.	**Tag No.**	**Description of Offtake**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**
1		**ORIGINAL PROCESS DATA**																
2		**1. BULK PROCESSING SUITE**																
3	T-100	Pre-rinse																
4	T-100	Fill																
5	T-101	Pre-rinse																
6	T-101	Fill																
7	T-102	Pre-rinse																
8	T-102	Fill																
9	T-103	Pre-rinse																
10	T-103	Fill																
11	BT-100	Buffer Tank pre-rinse																
12	BT-100	Buffer Tank fill																
13	BT-101	Buffer Tank pre-rinse																
14	BT-101	Buffer Tank fill																
15	BT-102	Buffer Tank pre-rinse																
16	BT-102	Buffer Tank fill																
17	BT-103	Buffer Tank pre-rinse																
18	BT-103	Buffer Tank fill																
19	N/A	Line Rinsing	9:45	10:00	0:20	u	11:00	11:15	0:20	u	15:00	15:15	0:20	u	16:15	16:30	0:20	u
20																		
21		**2. FILLING SUITE**																
22	PW-100	Parts Wash pre-rinse																
23	PW-100	Fill Parts Washer																
24	PW-101	Parts Wash pre-rinse																
25	PW-101	Fill Parts Washer																
26	PW-102	Parts Wash pre-rinse																
27	PW-102	Fill Parts Washer																
28	ST-100	Stopper Washer & Steriliser pre-rinse																
29	ST-100	Stopper Processor																
30	ST-101	Stopper Washer & Steriliser pre-rinse																
31	ST-101	Stopper Processor																
32	ST-102	Stopper Washer & Steriliser pre-rinse																
33	ST-102	Stopper Processor																
34	ST-103	Stopper Washer & Steriliser pre-rinse	18:00	18:10	0:30	n	19:30	19:40	0:30	n								
35	ST-103	Stopper Processor	19:00	19:10	0:30	n	20:30	20:40	0:30	n								
36	ST-104	Stopper Washer & Steriliser pre-rinse	20:00	20:10	0:30	n	21:30	21:40	0:30	n	23:00	23:10	0:30	n				
37	ST-104	Stopper Processor	20:30	20:40	0:30	n	22:00	22:10	0:30	n	23:20	23:30	0:30	n				
38	VF-100	Vial Rinser - Fill																
39	VF-101	Vial Rinser - Fill																
40	VF-102	Vial Rinser - Fill																
41	VF-103	Vial Rinser - Fill																
42	PW-100	Parts washer Sanitization																
43	PW-101	Parts washer Sanitization																
44	PW-102	Parts washer Sanitization																
45	ST-100	Stopper Processor Sanitation																
46	ST-101	Stopper Processor Sanitation																
47	ST-102	Stopper Processor Sanitation																
48	ST-103	Stopper Processor Sanitation																
49	ST-104	Stopper Processor Sanitation																
50	VF-100	Vial rinser Sanitisation																
51	VF-101	Vial rinser Sanitisation																
52	VF-102	Vial rinser Sanitisation																
53	VF-103	Vial rinser Sanitisation																
54																		
55		**2. WASHING SUITE**																
56	PW-100	Parts Washer Pre Rinse	10:30	10:35	0:20	n	15:30	15:35	0:20	n	16:45	16:50	0:20	n	17:50	17:55	0:20	n
57	PW-101	Parts Washer Pre Rinse																
58	PW-102	Parts Washer Pre Rinse	21:30	22:00	0:20	n	23:30	23:55	0:20	n								
59	PW-100	Parts Washer Sanitization																
60	PW-101	Parts Washer Sanitization																
61	PW-102	Parts Washer Sanitization																
62																		
63		**4. CIP SUITE**																
64	T-100	T-100																
65	T-101	T-101																
66	T-102	T-102																
67	T-103	T-103 RESCHEDULE from 6:00																
68	VF-100	Vial Filler																
69	VF-101	Vial Filler																
70	VF-102	Vial Filler																
71	VF-103	Vial Filler																
72	LYO-100	Lyophyliser																
73	LYO-101	Lyophyliser																
74	LYO-102	Lyophyliser																
75	LYO-103	Lyophyliser																
76	LYO-104	Lyophyliser																
77	VFC-100	Vial Filler CIP skid																
78	VFC-101	Vial Filler CIP skid																
79	VFC-102	Vial Filler CIP skid																
80	VFC-103	Vial Filler CIP skid																
81	LYC-100	Lyophyliser CIP skid																
82	LYC-101	Lyophyliser CIP skid																
83	LYC-102	Lyophyliser CIP skid																
84	LYC-103	Lyophyliser CIP skid																
85																		
86																		
87		**FIRST INCREASE IN PRODUCTION**																
88		**BY MAXIMISING USE OF EQUIPMENT**																
89																		
90		**1. BULK PROCESSING SUITE**																
91	T-100	Fill																
92	T-101	Fill																
93	T-102	Fill																
94	T-103	Fill																
95	BT-100	Buffer Tank fill																
96	BT-101	Buffer Tank fill																
97	BT-102	Buffer Tank fill																
98	BT-103	Buffer Tank fill																
99																		
100		**2. FILLING SUITE**																
101	ST-100	Stopper Washer & Steriliser pre-rinse																
102	ST-100	Stopper Processor																
103	ST-101	Stopper Washer & Steriliser pre-rinse																
104	ST-101	Stopper Processor																
105	ST-102	Stopper Washer & Steriliser pre-rinse																
106	ST-102	Stopper Processor																
107	ST-103	Stopper Washer & Steriliser pre-rinse	18:00	18:10	0:30	n	19:30	19:40	0:30	n								
108	ST-103	Stopper Processor	19:00	19:10	0:30	n	20:30	20:40	0:30	n								
109	ST-104	Stopper Washer & Steriliser pre-rinse	20:00	20:10	0:30	n	21:30	21:40	0:30	n	22:50	23:00	0:30	n				
110	ST-104	Stopper Processor	20:30	20:40	0:30	n	22:00	22:10	0:30	n	23:15	23:45	0:30	n				

117		**SECOND INCREASE IN PRODUCTION**																
118		**ADD ANOTHER FILLING SUITE**																
119		**RESCHEDULE BY 2 HOURS (STAGGERED)**																
120																		
121		**2. FILLING SUITE**																
122	PW-100a	Parts Wash pre-rinse																
123	PW-100a	Fill Parts Washer																
124	PW-101a	Parts Wash pre-rinse																
125	PW-101a	Fill Parts Washer																
126	PW-102a	Parts Wash pre-rinse																
127	PW-102a	Fill Parts Washer																
128	ST-100a	Stopper Washer & Steriliser pre-rinse																
129	ST-100a	Stopper Processor																
130	ST-101a	Stopper Washer & Steriliser pre-rinse																
131	ST-101a	Stopper Processor																
132	ST-102a	Stopper Washer & Steriliser pre-rinse																
133	ST-102a	Stopper Processor																
134	ST-103a	Stopper Washer & Steriliser pre-rinse	20:00	20:10	0:30	n	21:30	21:40	0:30	n								
135	ST-103a	Stopper Processor	21:00	21:10	0:30	n	22:30	22:40	0:30	n								
136	ST-104a	Stopper Washer & Steriliser pre-rinse	22:00	22:10	0:30	n												
137	ST-104a	Stopper Processor	22:30	22:40	0:30	n												
138	VF-100a	Vial Rinser - Fill																
139	VF-101a	Vial Rinser - Fill																
140	VF-102a	Vial Rinser - Fill																
141	VF-103a	Vial Rinser - Fill																
142																		
143	PW-101a	Parts washer Sanitization																
144																		
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22:00																
146	ST-101a	Stopper Processor Sanitation																
147																		
148	ST-103a	Stopper Processor Sanitation																
149	ST-104a	Stopper Processor Sanitation																
150	VF-100a	Vial rinser Sanitisation																
151	VF-101a	Vial rinser Sanitisation																
152	VF-102a	Vial rinser Sanitisation																
153	VF-103a	Vial rinser Sanitisation																
154																		
155																		
156		The second Filling Suite requires additional Bulk																
157		Processing, Washing & CIP from the existing																
158		Suites as follows: 2 hours staggered																
159																		
160		**1. BULK PROCESSING SUITE**																
161	T-100	Pre-rinse																
162	T-100	Fill																
163	T-101	Pre-rinse																
164	T-101	Fill																
165	T-102	Pre-rinse																
166	T-102	Fill																
167	T-103	Pre-rinse																
168	T-103	Fill																
169	BT-100	Buffer Tank pre-rinse																
170	BT-100	Buffer Tank fill																
171	BT-101	Buffer Tank pre-rinse																
172	BT-101	Buffer Tank fill																
173	BT-102	Buffer Tank pre-rinse																
174	BT-102	Buffer Tank fill																
175	BT-103	Buffer Tank pre-rinse																
176	BT-103	Buffer Tank fill																
177	N/A	Line Rinsing	11:45	12:00	0:20	u	13:00	13:15	0:20	u	17:00	17:15	0:20	u	18:15	18:30	0:20	u
178																		
179																		
180		**2. WASHING SUITE**																
181	PW-100	Parts Washer Pre Rinse	12:30	12:35	0:20	n	17:30	17:35	0:20	n	18:45	18:50	0:20	n	19:50	19:55	0:20	n
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00																
183	PW-102	Parts Washer Pre Rinse																
184	PW-100	Parts Washer Sanitization																
185	PW-101	Parts Washer Sanitization																
186	PW-102	Parts Washer Sanitization																
187																		
188		**4. CIP SUITE**																
189	T-100	T-100																
190	T-101	T-101																
191	T-102	T-102 RESCHEDULE from 6:15																
192	T-103	T-103																
193	VF-100	Vial Filler																
194	VF-101	Vial Filler																
195	VF-102	Vial Filler																
196	VF-103	Vial Filler RESCHEDULE from 21:00																
197	LYO-100	Lyophyliser RESCHEDULE from 7:00																
198	LYO-101	Lyophyliser																
199	LYO-102	Lyophyliser																
200	LYO-103	Lyophyliser																
201	LYO-104	Lyophyliser																
202	VFC-100	Vial Filler CIP skid																
203	VFC-101	Vial Filler CIP skid																
204	VFC-102	Vial Filler CIP skid																
205	VFC-103	Vial Filler CIP skid																
206	LYC-100	Lyophyliser CIP skid																
207	LYC-101	Lyophyliser CIP skid																
208	LYC-102	Lyophyliser CIP skid																
209	LYC-103	Lyophyliser CIP skid																
210																		
211																		
212		**THIRD INCREASE IN PRODUCTION**																
213		**BY MAXIMISING USE OF EQUIPMENT**																
214																		
215		**1. BULK PROCESSING SUITE**																
216	T-100	Fill																
217	T-101	Fill																
218	T-102	Fill																
219	T-103	Fill																
220	BT-100	Buffer Tank fill																
221	BT-101	Buffer Tank fill																
222	BT-102	Buffer Tank fill																
223	BT-103	Buffer Tank fill																
224																		
225		**2. FILLING SUITE**																
226	ST-100a	Stopper Washer & Steriliser pre-rinse																
227	ST-100a	Stopper Processor																
228	ST-101a	Stopper Washer & Steriliser pre-rinse																
229	ST-101a	Stopper Processor																
230	ST-102a	Stopper Washer & Steriliser pre-rinse																
231	ST-102a	Stopper Processor																
232	ST-103a	Stopper Washer & Steriliser pre-rinse	20:00	20:10	0:30	n	21:30	21:40	0:30	n								
233	ST-103a	Stopper Processor	21:00	21:10	0:30	n	22:30	22:40	0:30	n								
234	ST-104a	Stopper Washer & Steriliser pre-rinse	22:00	22:10	0:30	n												
235	ST-104a	Stopper Processor	22:30	22:40	0:30	n												
236	VF-100a	Vial Rinser - Fill																

244		**FOURTH INCREASE IN PRODUCTION**																
245		**ADD THIRD FILLING SUITE**																
246		**RESCHEDULE BY 2 HOURS (STAGGERED)**																
247																		
248		**2. FILLING SUITE**																
249	PW-100a	Parts Wash pre-rinse																
250	PW-100a	Fill Parts Washer																
251	PW-101a	Parts Wash pre-rinse																
252	PW-101a	Fill Parts Washer																
253	PW-102a	Parts Wash pre-rinse																
254	PW-102a	Fill Parts Washer																
255	ST-100a	Stopper Washer & Steriliser pre-rinse																
256	ST-100a	Stopper Processor																
257	ST-101a	Stopper Washer & Steriliser pre-rinse																
258	ST-101a	Stopper Processor																
259	ST-102a	Stopper Washer & Steriliser pre-rinse																
260	ST-102a	Stopper Processor																
261	ST-103a	Stopper Washer & Steriliser pre-rinse	22:00	22:10	0:30	n												
262	ST-103a	Stopper Processor	23:00	23:10	0:30	n												
263	ST-104a	Stopper Washer & Steriliser pre-rinse																
264	ST-104a	Stopper Processor																
265	VF-100a	Vial Rinser - Fill																
266	VF-101a	Vial Rinser - Fill																
267	VF-102a	Vial Rinser - Fill																
268																		
269																		
270	PW-101a	Parts washer Sanitization																
271																		
272	ST-100a	Stopper Processor Sanitation																
273	ST-101a	Stopper Processor Sanitation																
274																		
275	ST-103a	Stopper Processor Sanitation																
276	ST-104a	Stopper Processor Sanitation																
277	VF-100a	Vial rinser Sanitisation																
278	VF-101a	Vial rinser Sanitisation																
279	VF-102a	Vial rinser Sanitisation																
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																
281																		
282																		
283																		
284																		
285																		
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																
287		**1. BULK PROCESSING SUITE**																
288	T-100	Pre-rinse																
289	T-100	Fill																
290	T-101	Pre-rinse																
291	T-101	Fill																
292	T-102	Pre-rinse																
293	T-102	Fill																
294	T-103	Pre-rinse																
295	T-103	Fill																
296	BT-100	Buffer Tank pre-rinse																
297	BT-100	Buffer Tank fill																
298	BT-101	Buffer Tank pre-rinse																
299	BT-101	Buffer Tank fill																
300	BT-102	Buffer Tank pre-rinse																
301	BT-102	Buffer Tank fill																
302	BT-103	Buffer Tank pre-rinse																
303	BT-103	Buffer Tank fill																
304	N/A	Line Rinsing	13:45	14:00	0:20	u	15:00	15:15	0:20	u	19:00	19:15	0:20	u	20:15	20:30	0:20	u
305																		
306																		
307																		
308																		
309		**2. WASHING SUITE**																
310	PW-100	Parts Washer Pre Rinse	14:30	14:35	0:20	n	19:30	19:35	0:20	n	20:45	20:50	0:20	n	21:50	21:55	0:20	n
311	PW-101	Parts Washer Pre Rinse																
312	PW-102	Parts Washer Pre Rinse																
313	PW-100	Parts Washer Sanitization																
314	PW-101	Parts Washer Sanitization																
315	PW-102	Parts Washer Sanitization																
316																		
317		**4. CIP SUITE**																

342		**FIFTH INCREASE IN PRODUCTION**											
343		**BY MAXIMISING USE OF EQUIPMENT**											
344													
345		**1. BULK PROCESSING SUITE**											
346	T-100	Fill											
347	T-101	Fill											
348	T-102	Fill											
349	T-103	Fill											
350	BT-100	Buffer Tank fill											
351	BT-101	Buffer Tank fill											
352	BT-102	Buffer Tank fill											
353	BT-103	Buffer Tank fill											
354													
355		**2. FILLING SUITE**											
356	ST-100a	Stopper Washer & Steriliser pre-rinse											
357	ST-100a	Stopper Processor											
358	ST-101a	Stopper Washer & Steriliser pre-rinse											
359	ST-101a	Stopper Processor											
360	ST-102a	Stopper Washer & Steriliser pre-rinse											
361	ST-102a	Stopper Processor											
362	ST-103a	Stopper Washer & Steriliser pre-rinse	22:00	22:10	0:30	n							
363	ST-103a	Stopper Processor	23:00	23:10	0:30	n							
364	ST-104a	Stopper Washer & Steriliser pre-rinse											
365	ST-104a	Stopper Processor											
366	VF-100a	Vial Rinser - Fill											
367	VF-101a	Vial Rinser - Fill											
368	VF-102a	Vial Rinser - Fill											
369													
370													
371		**SIXTH INCREASE IN PRODUCTION**											
372		**ADD FOURTH FILLING SUITE**											
373		**RESCHEDULE BY 2 HOURS (STAGGERED)**											
374													
375		**2. FILLING SUITE**											
376	PW-100a	Parts Wash pre-rinse											
377	PW-100a	Fill Parts Washer											
378	PW-101a	Parts Wash pre-rinse											
379	PW-101a	Fill Parts Washer											
380	PW-102a	Parts Wash pre-rinse											
381	PW-102a	Fill Parts Washer											
382	ST-100a	Stopper Washer & Steriliser pre-rinse											
383	ST-100a	Stopper Processor											
384	ST-101a	Stopper Washer & Steriliser pre-rinse											
385	ST-101a	Stopper Processor											
386	ST-102a	Stopper Washer & Steriliser pre-rinse											
387	ST-102a	Stopper Processor											
388	ST-103a	Stopper Washer & Steriliser pre-rinse											
389	ST-103a	Stopper Processor											
390	ST-104a	Stopper Washer & Steriliser pre-rinse											
391	ST-104a	Stopper Processor											
392	VF-100a	Vial Rinser - Fill											
393	VF-101a	Vial Rinser - Fill											
394	VF-102a	Vial Rinser - Fill											
395													
396													
397	PW-101a	Parts washer Sanitization											
398													
399	ST-100a	Stopper Processor Sanitation											
400	ST-101a	Stopper Processor Sanitation											
401													
402	ST-103a	Stopper Processor Sanitation											
403	ST-104a	Stopper Processor Sanitation											
404	VF-100a	Vial rinser Sanitisation											
405	VF-101a	Vial rinser Sanitisation											
406	VF-102a	Vial rinser Sanitisation											
407	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00											
408													
409													
410													
411		**SEVENTH INCREASE IN PRODUCTION**											
412		**BY MAXIMISING USE OF EQUIPMENT**											
413													
414		**1. BULK PROCESSING SUITE**											
415	T-100	Fill											
416	T-101	Fill											
417	T-102	Fill											
418	T-103	Fill											
419	BT-100	Buffer Tank fill											
420	BT-101	Buffer Tank fill											

WFI Water Offtake In

			9				10				11				12			
No.	**Tag No.**	**Description of Offtake**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**	**Start Time (Tap Open) [24 hour clock, hours:min]**	**End Time (Tap Closed) [24 hour clock, hours:min]**	**Max.Time Uncertainty Time Open [min]**	**Description of Time Uncertainty**
1		**ORIGINAL PROCESS DATA**																
2		**1. BULK PROCESSING SUITE**																
3	T-100	Pre-rinse																
4	T-100	Fill																
5	T-101	Pre-rinse																
6	T-101	Fill																
7	T-102	Pre-rinse																
8	T-102	Fill																
9	T-103	Pre-rinse																
10	T-103	Fill																
11	BT-100	Buffer Tank pre-rinse																
12	BT-100	Buffer Tank fill																
13	BT-101	Buffer Tank pre-rinse																
14	BT-101	Buffer Tank fill																
15	BT-102	Buffer Tank pre-rinse																
16	BT-102	Buffer Tank fill																
17	BT-103	Buffer Tank pre-rinse																
18	BT-103	Buffer Tank fill																
19	N/A	Line Rinsing	17:20	17:35	0:20	u	18:30	18:45	0:20	u	19:40	19:55	0:20	u	22:00	22:15	0:20	u
20																		
21		**2. FILLING SUITE**																
22	PW-100	Parts Wash pre-rinse																
23	PW-100	Fill Parts Washer																
24	PW-101	Parts Wash pre-rinse																
25	PW-101	Fill Parts Washer																
26	PW-102	Parts Wash pre-rinse																
27	PW-102	Fill Parts Washer																
28	ST-100	Stopper Washer & Steriliser pre-rinse																
29	ST-100	Stopper Processor																
30	ST-101	Stopper Washer & Steriliser pre-rinse																
31	ST-101	Stopper Processor																
32	ST-102	Stopper Washer & Steriliser pre-rinse																
33	ST-102	Stopper Processor																
34	ST-103	Stopper Washer & Steriliser pre-rinse																
35	ST-103	Stopper Processor																
36	ST-104	Stopper Washer & Steriliser pre-rinse																
37	ST-104	Stopper Processor																
38	VF-100	Vial Rinser - Fill																
39	VF-101	Vial Rinser - Fill																
40	VF-102	Vial Rinser - Fill																
41	VF-103	Vial Rinser - Fill																
42	PW-100	Parts washer Sanitization																
43	PW-101	Parts washer Sanitization																
44	PW-102	Parts washer Sanitization																
45	ST-100	Stopper Processor Sanitation																
46	ST-101	Stopper Processor Sanitation																
47	ST-102	Stopper Processor Sanitation																
48	ST-103	Stopper Processor Sanitation																
49	ST-104	Stopper Processor Sanitation																
50	VF-100	Vial rinser Sanitisation																
51	VF-101	Vial rinser Sanitisation																
52	VF-102	Vial rinser Sanitisation																
53	VF-103	Vial rinser Sanitisation																
54																		
55		**2. WASHING SUITE**																
56	PW-100	Parts Washer Pre Rinse	19:00	19:05	0:20	n												
57	PW-101	Parts Washer Pre Rinse																
58	PW-102	Parts Washer Pre Rinse																
59	PW-100	Parts Washer Sanitization																
60	PW-101	Parts Washer Sanitization																
61	PW-102	Parts Washer Sanitization																
62																		
63		**4. CIP SUITE**																
64	T-100	T-100																
65	T-101	T-101																
66	T-102	T-102																
67	T-103	T-103 RESCHEDULE from 6:00																
68	VF-100	Vial Filler																
69	VF-101	Vial Filler																
70	VF-102	Vial Filler																
71	VF-103	Vial Filler																
72	LYO-100	Lyophyliser																
73	LYO-101	Lyophyliser																
74	LYO-102	Lyophyliser																
75	LYO-103	Lyophyliser																
76	LYO-104	Lyophyliser																
77	VFC-100	Vial Filler CIP skid																
78	VFC-101	Vial Filler CIP skid																
79	VFC-102	Vial Filler CIP skid																
80	VFC-103	Vial Filler CIP skid																
81	LYC-100	Lyophyliser CIP skid																

117		**SECOND INCREASE IN PRODUCTION**																
118		**ADD ANOTHER FILLING SUITE**																
119		**RESCHEDULE BY 2 HOURS (STAGGERED)**																
120																		
121		**2. FILLING SUITE**																
122	PW-100a	Parts Wash pre-rinse																
123	PW-100a	Fill Parts Washer																
124	PW-101a	Parts Wash pre-rinse																
125	PW-101a	Fill Parts Washer																
126	PW-102a	Parts Wash pre-rinse																
127	PW-102a	Fill Parts Washer																
128	ST-100a	Stopper Washer & Steriliser pre-rinse																
129	ST-100a	Stopper Processor																
130	ST-101a	Stopper Washer & Steriliser pre-rinse																
131	ST-101a	Stopper Processor																
132	ST-102a	Stopper Washer & Steriliser pre-rinse																
133	ST-102a	Stopper Processor																
134	ST-103a	Stopper Washer & Steriliser pre-rinse																
135	ST-103a	Stopper Processor																
136	ST-104a	Stopper Washer & Steriliser pre-rinse																
137	ST-104a	Stopper Processor																
138	VF-100a	Vial Rinser - Fill																
139	VF-101a	Vial Rinser - Fill																
140	VF-102a	Vial Rinser - Fill																
141	VF-103a	Vial Rinser - Fill																
142																		
143	PW-101a	Parts washer Sanitization																
144																		
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22:00																
146	ST-101a	Stopper Processor Sanitation																
147																		
148	ST-103a	Stopper Processor Sanitation																
149	ST-104a	Stopper Processor Sanitation																
150	VF-100a	Vial rinser Sanitisation																
151	VF-101a	Vial rinser Sanitisation																
152	VF-102a	Vial rinser Sanitisation																
153	VF-103a	Vial rinser Sanitisation																
154																		
155																		
156		The second Filling Suite requires additional Bulk																
157		Processing, Washing & CIP from the existing																
158		Suites as follows: 2 hours staggered																
159																		
160		**1. BULK PROCESSING SUITE**																
161	T-100	Pre-rinse																
162	T-100	Fill																
163	T-101	Pre-rinse																
164	T-101	Fill																
165	T-102	Pre-rinse																
166	T-102	Fill																
167	T-103	Pre-rinse																
168	T-103	Fill																
169	BT-100	Buffer Tank pre-rinse																
170	BT-100	Buffer Tank fill																
171	BT-101	Buffer Tank pre-rinse																
172	BT-101	Buffer Tank fill																
173	BT-102	Buffer Tank pre-rinse																
174	BT-102	Buffer Tank fill																
175	BT-103	Buffer Tank pre-rinse																
176	BT-103	Buffer Tank fill																
177	N/A	Line Rinsing	19:20	19:35	0:20	u	20:30	20:45	0:20	u	21:40	21:55	0:20	u				
178																		
179																		
180		**2. WASHING SUITE**																
181	PW-100	Parts Washer Pre Rinse	21:00	21:05	0:20	n												
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00																
183	PW-102	Parts Washer Pre Rinse																
184	PW-100	Parts Washer Sanitization																
185	PW-101	Parts Washer Sanitization																
186	PW-102	Parts Washer Sanitization																
187																		
188		**4. CIP SUITE**																
189	T-100	T-100																
190	T-101	T-101																
191	T-102	T-102 RESCHEDULE from 6:15																
192	T-103	T-103																
193	VF-100	Vial Filler																
194	VF-101	Vial Filler																
195	VF-102	Vial Filler																
196	VF-103	Vial Filler RESCHEDULE from 21:00																
197	LYO-100	Lyophyliser RESCHEDULE from 7:00																
198	LYO-101	Lyophyliser																

244		**FOURTH INCREASE IN PRODUCTION**																
245		**ADD THIRD FILLING SUITE**																
246		**RESCHEDULE BY 2 HOURS (STAGGERED)**																
247																		
248		**2. FILLING SUITE**																
249	PW-100a	Parts Wash pre-rinse																
250	PW-100a	Fill Parts Washer																
251	PW-101a	Parts Wash pre-rinse																
252	PW-101a	Fill Parts Washer																
253	PW-102a	Parts Wash pre-rinse																
254	PW-102a	Fill Parts Washer																
255	ST-100a	Stopper Washer & Steriliser pre-rinse																
256	ST-100a	Stopper Processor																
257	ST-101a	Stopper Washer & Steriliser pre-rinse																
258	ST-101a	Stopper Processor																
259	ST-102a	Stopper Washer & Steriliser pre-rinse																
260	ST-102a	Stopper Processor																
261	ST-103a	Stopper Washer & Steriliser pre-rinse																
262	ST-103a	Stopper Processor																
263	ST-104a	Stopper Washer & Steriliser pre-rinse																
264	ST-104a	Stopper Processor																
265	VF-100a	Vial Rinser - Fill																
266	VF-101a	Vial Rinser - Fill																
267	VF-102a	Vial Rinser - Fill																
268																		
269																		
270	PW-101a	Parts washer Sanitization																
271																		
272	ST-100a	Stopper Processor Sanitation																
273	ST-101a	Stopper Processor Sanitation																
274																		
275	ST-103a	Stopper Processor Sanitation																
276	ST-104a	Stopper Processor Sanitation																
277	VF-100a	Vial rinser Sanitisation																
278	VF-101a	Vial rinser Sanitisation																
279	VF-102a	Vial rinser Sanitisation																
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																
281																		
282																		
283																		
284																		
285																		
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																
287		**1. BULK PROCESSING SUITE**																
288	T-100	Pre-rinse																
289	T-100	Fill																
290	T-101	Pre-rinse																
291	T-101	Fill																
292	T-102	Pre-rinse																
293	T-102	Fill																
294	T-103	Pre-rinse																
295	T-103	Fill																
296	BT-100	Buffer Tank pre-rinse																
297	BT-100	Buffer Tank fill																
298	BT-101	Buffer Tank pre-rinse																
299	BT-101	Buffer Tank fill																
300	BT-102	Buffer Tank pre-rinse																
301	BT-102	Buffer Tank fill																
302	BT-103	Buffer Tank pre-rinse																
303	BT-103	Buffer Tank fill																
304	N/A	Line Rinsing	21:20	21:35	0:20	u	22:30	22:45	0:20	u								
305																		
306																		
307																		
308																		
309		**2. WASHING SUITE**																
310	PW-100	Parts Washer Pre Rinse	23:00	23:05	0:20	n												
311	PW-101	Parts Washer Pre Rinse																
312	PW-102	Parts Washer Pre Rinse																
313	PW-100	Parts Washer Sanitization																
314	PW-101	Parts Washer Sanitization																
315	PW-102	Parts Washer Sanitization																
316																		
317		**4. CIP SUITE**																
318	T-100	T-100																
319	T-101	T-101																
320	T-102	T-102																
321	T-103	T-103																
322	VF-100	Vial Filler																

Appendix 8 Input Data Set for the Fuzzy Logic Simulations

This appendix contains the input data sets used for the various fuzzy logic DI/WFI simulations.

			WFI Water Offtake Input Data								WFI Water Offtake Input Data 2								WFI Water Offtake Input Data 3							
No.	Tag No.	Description of Offtake	Water Offtake [m^3/hr]	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority
1		**ORIGINAL PROCESS DATA**																								
2		**1. BULK PROCESSING SUITE**																								
3	T-100	Pre-rinse	1.5	2:00	2:05	0:00	0:02	0:08	0:10	2	10:00	10:35	0:00	0:02	0:08	0:10	2	18:10	18:15	0:00	0:02	0:08	0:10	2
4	T-100	Fill	1	2:00	2:06	0:00	0:01	0:04	0:05	3	10:30	10:36	0:00	0:01	0:04	0:05	3	18:10	18:15	0:00	0:01	0:04	0:05	3
5	T-101	Pre-rinse	1.5	3:10	3:17	0:00	0:02	0:08	0:10	2	11:25	11:45	0:00	0:02	0:08	0:10	2	19:20	19:40	0:00	0:02	0:08	0:10	2
6	T-101	Fill	1	3:15	3:25	0:00	0:01	0:04	0:05	3	11:15	11:25	0:00	0:01	0:04	0:05	3	19:30	19:40	0:00	0:01	0:04	0:05	3
7	T-102	Pre-rinse	1.5	1:20	1:30	0:00	0:02	0:08	0:10	2	9:30	9:40	0:00	0:02	0:08	0:10	2	17:10	17:20	0:00	0:02	0:08	0:10	2
8	T-102	Fill	1	3:15	3:25	0:00	0:01	0:04	0:05	3	10:15	10:20	0:00	0:01	0:04	0:05	3	18:15	18:20	0:00	0:01	0:04	0:05	3
9	T-103	Pre-rinse	1.5	1:00	1:10	0:00	0:02	0:08	0:10	2	9:00	9:10	0:00	0:02	0:08	0:10	2	19:30	19:40	0:00	0:02	0:08	0:10	2
10	T-103	Fill	1	6:00	6:30	0:00	0:01	0:04	0:05	3			0:00	0:01	0:04	0:05	3			0:00	0:01	0:04	0:05	3
11	BT-100	Buffer Tank pre-rinse	1.5	1:15	1:20	0:00	0:02	0:08	0:10	2	9:00	9:05	0:00	0:02	0:08	0:10	2	17:30	17:35	0:00	0:02	0:08	0:10	2
12	BT-100	Buffer Tank fill	1.5	2:00	2:10	0:00	0:01	0:04	0:05	2	10:00	10:10	0:00	0:01	0:04	0:05	2	19:00	19:10	0:00	0:01	0:04	0:05	2
13	BT-101	Buffer Tank pre-rinse	1.5	2:30	3:00	0:00	0:02	0:08	0:10	2	10:30	11:00	0:00	0:02	0:08	0:10	2	19:30	20:00	0:00	0:02	0:08	0:10	2
14	BT-101	Buffer Tank fill	1.5	3:00	3:30	0:00	0:01	0:04	0:05	2	11:00	11:30	0:00	0:01	0:04	0:05	2	20:00	20:30	0:00	0:01	0:04	0:05	2
15	BT-102	Buffer Tank pre-rinse	1.5	21:00	21:05	0:00	0:02	0:08	0:10	2														
16	BT-102	Buffer Tank fill	1.5	21:15	21:20	0:00	0:01	0:04	0:05	2														
17	BT-103	Buffer Tank pre-rinse	1.5	2:00	2:05	0:00	0:02	0:08	0:10	2														
18	BT-103	Buffer Tank fill	1.5	0:30	0:45	0:00	0:01	0:04	0:05	2														
19	N/A	Line Rinsing	1	0:30	0:45	0:00	0:02	0:08	0:10	2	6:00	6:15	0:00	0:02	0:08	0:10	2	7:20	7:35	0:00	0:02	0:08	0:10	2
20																								
21		**2. FILLING SUITE**																						
22	PW-100	Parts Wash pre-rinse	1	0:30	0:40	0:00	0:01	0:04	0:05	2	11:00	11:10	0:00	0:01	0:04	0:05	2	16:00	16:10	0:00	0:01	0:04	0:05	2
23	PW-100	Fill Parts Washer	1	1:30	2:10	0:00	0:02	0:08	0:10	2	12:30	12:50	0:00	0:02	0:08	0:10	2	17:30	17:55	0:00	0:02	0:08	0:10	2
24	PW-101	Parts Wash pre-rinse	1	3:15	3:30	0:00	0:01	0:04	0:05	2	12:45	12:55	0:00	0:01	0:04	0:05	2	16:30	16:40	0:00	0:01	0:04	0:05	2
25	PW-101	Fill Parts Washer	1	5:30	6:40	0:00	0:02	0:08	0:10	2	13:00	13:10	0:00	0:02	0:08	0:10	2	17:00	17:10	0:00	0:02	0:08	0:10	2
26	PW-102	Parts Wash pre-rinse	1	7:30	7:40	0:00	0:01	0:04	0:05	2	11:15	11:25	0:00	0:01	0:04	0:05	2	16:15	16:25	0:00	0:01	0:04	0:05	2
27	PW-102	Fill Parts Washer	1	8:00	8:10	0:00	0:02	0:08	0:10	2	13:00	13:10	0:00	0:02	0:08	0:10	2	17:00	17:10	0:00	0:02	0:08	0:10	2
28	ST-100	Stopper Washer & Steriliser pre-rinse	2	4:00	4:10	0:00	0:01	0:04	0:05	2	12:00	12:10	0:00	0:01	0:04	0:05	2	18:30	18:40	0:00	0:01	0:04	0:05	2
29	ST-100	Stopper Processor	3	5:00	6:00	0:00	0:02	0:08	0:10	2	13:00	14:00	0:00	0:02	0:08	0:10	2	19:00	20:00	0:00	0:02	0:08	0:10	2
30	ST-101	Stopper Washer & Steriliser pre-rinse	2	6:00	6:10	0:00	0:01	0:04	0:05	2	14:00	14:10	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2
31	ST-101	Stopper Processor	3	6:00	6:00	0:00	0:02	0:08	0:10	2	15:00	16:00	0:00	0:02	0:08	0:10	2	21:00	22:00	0:00	0:02	0:08	0:10	2
32	ST-102	Stopper Washer & Steriliser pre-rinse	2	6:00	6:10	0:00	0:01	0:04	0:05	2	14:00	14:10	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2
33	ST-102	Stopper Processor	3	7:00	8:10	0:00	0:02	0:08	0:10	2	15:30	16:30	0:00	0:02	0:08	0:10	2	23:00	23:59	0:00	0:02	0:08	0:10	2
34	ST-103	Stopper Washer & Steriliser pre-rinse	2	0:35	0:40	0:00	0:01	0:04	0:05	2	2:00	2:10	0:00	0:01	0:04	0:05	2	11:00	11:10	0:00	0:01	0:04	0:05	2
35	ST-103	Stopper Processor	2	1:15	1:25	0:00	0:02	0:08	0:10	2	2:30	2:50	0:00	0:02	0:08	0:10	2	12:00	12:10	0:00	0:02	0:08	0:10	2
36	ST-104	Stopper Washer & Steriliser pre-rinse	2	2:15	2:25	0:00	0:01	0:04	0:05	2	3:30	3:40	0:00	0:01	0:04	0:05	2	13:00	13:10	0:00	0:01	0:04	0:05	2
37	ST-104	Stopper Processor	2	3:00	3:25	0:00	0:02	0:08	0:10	2	4:40	4:50	0:00	0:02	0:08	0:10	2	13:30	13:40	0:00	0:02	0:08	0:10	2
38	VF-100	Vial Rinser - Fill	0.5	0:15	0:25	0:00	0:01	0:04	0:05	3	1:00	1:30	0:00	0:01	0:04	0:05	3	2:00	2:30	0:00	0:01	0:04	0:05	3
39	VF-101	Vial Rinser - Fill	0.5	6:00	6:35	0:00	0:02	0:08	0:10	3	7:00	7:35	0:00	0:02	0:08	0:10	3	8:00	8:30	0:00	0:02	0:08	0:10	3
40	VF-102	Vial Rinser - Fill	0.5	10:00	10:35	0:00	0:01	0:04	0:05	3	11:00	11:35	0:00	0:01	0:04	0:05	3	12:00	12:30	0:00	0:01	0:04	0:05	3
41	VF-103	Vial Rinser - Fill	0.5	20:00	20:35	0:00	0:02	0:08	0:10	3	21:00	21:35	0:00	0:02	0:08	0:10	3	22:00	22:30	0:00	0:02	0:08	0:10	3
42	PW-100	Parts washer Sanitization	2.5	22:00	22:05	0:00	0:01	0:04	0:05	2														
43	PW-101	Parts washer Sanitization	2.5	21:00	21:05	0:00	0:02	0:08	0:10	2														
44	PW-102	Parts washer Sanitization	2.5	22:30	22:35	0:00	0:01	0:04	0:05	2														
45	ST-100	Stopper Processor Sanitation	2.5	20:00	20:05	0:00	0:02	0:08	0:10	3														
46	ST-101	Stopper Processor Sanitation	2.5	21:00	21:05	0:00	0:01	0:04	0:05	3														
47	ST-102	Stopper Processor Sanitation	2.5	22:00	22:05	0:00	0:02	0:08	0:10	3														
48	ST-103	Stopper Processor Sanitation	2.5	2:00	2:05	0:00	0:01	0:04	0:05	3														
49	ST-104	Stopper Processor Sanitation	2.5	5:00	5:05	0:00	0:02	0:08	0:10	3														
50	VF-100	Vial rinser Sanitisation	2.5	6:30	6:35	0:00	0:01	0:04	0:05	3	7:30	7:40	0:00	0:01	0:04	0:05	3							
51	VF-101	Vial rinser Sanitisation	2.5	4:45	4:50	0:00	0:02	0:08	0:10	3	5:45	5:55	0:00	0:02	0:08	0:10	3							
52	VF-102	Vial rinser Sanitisation	2.5	6:00	6:10	0:00	0:01	0:04	0:05	3	7:00	7:10	0:00	0:01	0:04	0:05	3							
53	VF-103	Vial rinser Sanitisation	2.5	3:00	3:10	0:00	0:02	0:08	0:10	3														
54																								
55		**2. WASHING SUITE**																						
56	PW-100	Parts Washer Pre Rinse	2.5	2:30	2:35	0:00	0:01	0:04	0:05	1	5:30	5:35	0:00	0:01	0:04	0:05	1	6:40	6:45	0:00	0:01	0:04	0:05	1
57	PW-101	Parts Washer Pre Rinse	3	4:00	5:15	0:00	0:02	0:08	0:10	1	12:30	12:45	0:00	0:02	0:08	0:10	1	16:00	16:15	0:00	0:02	0:08	0:10	1
58	PW-102	Parts Washer Pre Rinse	2.5	5:00	5:30	0:00	0:01	0:04	0:05	1	6:30	7:00	0:00	0:01	0:04	0:05	1	10:30	11:00	0:00	0:01	0:04	0:05	1
59	PW-100	Parts Washer Sanitization	2.5	22:30	22:35	0:00	0:02	0:08	0:10	1														
60	PW-101	Parts Washer Sanitization	2.5	23:30	23:35	0:00	0:01	0:04	0:05	1														
61	PW-102	Parts Washer Sanitization	2.5	4:00	4:05	0:00	0:02	0:08	0:10	1														
62																								
63		**4. CIP SUITE**																						
64	T-100	T-100	3	2:00	2:30	0:00	0:01	0:04	0:05	2	8:15	8:30	0:00	0:01	0:04	0:05	2	16:15	16:30	0:00	0:01	0:04	0:05	2
65	T-101	T-101	3	2:15	2:45	0:00	0:02	0:08	0:10	2	10:15	10:30	0:00	0:02	0:08	0:10	2	18:15	18:30	0:00	0:02	0:08	0:10	2
66	T-102	T-102	3	4:15	4:30	0:00	0:01	0:04	0:05	2	12:15	12:30	0:00	0:01	0:04	0:05	2	20:15	20:30	0:00	0:01	0:04	0:05	2
67	T-103	T-103 RESCHEDULE from 6:00	2.5	7:00	7:40	0:00	0:02	0:08	0:10	2	17:00	17:15	0:00	0:02	0:08	0:10	2							
68	VF-100	Vial Filler	2.5	23:00	23:40	0:00	0:01	0:04	0:05	3														
69	VF-101	Vial Filler	3	13:00	13:25	0:00	0:02	0:08	0:10	3														
70	VF-102	Vial Filler	3	17:00	17:25	0:00	0:01	0:04	0:05	3														
71	VF-103	Vial Filler	3	19:00	19:25	0:00	0:02	0:08	0:10	3														
72	LYO-100	Lyophyliser	2.5	5:00	5:30	0:00	0:01	0:04	0:05	2														
73	LYO-101	Lyophyliser	2.5	11:00	11:30	0:00	0:02	0:08	0:10	2														
74	LYO-102	Lyophyliser	2.5	21:00	21:30	0:00	0:01	0:04	0:05	2														
75	LYO-103	Lyophyliser	2.5	23:30	23:59	0:00	0:02	0:08	0:10	2														
76	LYO-104	Lyophyliser	2.5	23:30	23:59	0:00	0:01	0:04	0:05	2														
77	VFC-100	Vial Filler CIP skid	1	23:30	23:40	0:00	0:02	0:08	0:10	3														
78	VFC-101	Vial Filler CIP skid	1	23:00	23:10	0:00	0:01	0:04	0:05	3														
79	VFC-102	Vial Filler CIP skid	1	22:00	22:10	0:00	0:02	0:08	0:10	3														
80	VFC-103	Vial Filler CIP skid	1	21:30	21:40	0:00	0:01	0:04	0:05	3														
81	LYC-100	Lyophyliser CIP skid	1	21:00	21:10	0:00	0:02	0:08	0:10	2														
82	LYC-101	Lyophyliser CIP skid	1	20:00	20:10	0:00	0:01	0:04	0:05	2														
83	LYC-102	Lyophyliser CIP skid	1	3:00	3:05	0:00	0:02	0:08	0:10	2														
84	LYC-103	Lyophyliser CIP skid	1	4:00	4:05	0:00	0:01	0:04	0:05	2														
85																								
86																								
87		**FIRST INCREASE IN PRODUCTION**																						
88		**BY MAXIMISING USE OF EQUIPMENT**																						
89																								
90		**1. BULK PROCESSING SUITE**																						
91	T-100	Fill	1	2:00	2:06	0:00	0:02	0:08	0:10	3	10:30	10:36	0:00	0:02	0:08	0:10	3	18:10	18:15	0:00	0:02	0:08	0:10	3
92	T-101	Fill	1	3:15	3:25	0:00	0:01	0:04	0:05	3	11:15	11:25	0:00	0:01	0:04	0:05	3	19:30	19:40	0:00	0:01	0:04	0:05	3
93	T-102	Fill	1	3:15	3:25	0:00	0:02	0:08	0:10	3	10:15	10:20	0:00	0:02	0:08	0:10	3	18:15	18:20	0:00	0:02	0:08	0:10	3
94	T-103	Fill	1	6:00	6:30	0:00	0:01	0:04	0:05	3														
95	BT-100	Buffer Tank fill	1.5	2:00	2:10	0:00	0:02	0:08	0:10	3	10:00	10:10	0:00	0:02	0:08	0:10	3	19:00	19:10	0:00	0:02	0:08	0:10	3
96	BT-101	Buffer Tank fill	1.5	3:00	3:30	0:00	0:01	0:04	0:05	3	11:00	11:30	0:00	0:01	0:04	0:05	3	20:00	20:30	0:00	0:01	0:04	0:05	3
97	BT-102	Buffer Tank fill	1.5	21:15	21:20	0:00	0:02	0:08	0:10	3														
98	BT-103	Buffer Tank fill	1.5	0:30	0:45	0:00	0:01	0:04	0:05	3														
99																								
100		**2. FILLING SUITE**																						
101	ST-100	Stopper Washer & Steriliser pre-rinse	0.5	4:00	4:10	0:00	0:02	0:08	0:10	2	12:00	12:10	0:00	0:02	0:08	0:10	2	18:30	18:40	0:00	0:02	0:08	0:10	2
102	ST-100	Stopper Processor	2	5:00	6:00	0:00	0:01	0:04	0:05	2	13:00	14:00	0:00	0:01	0:04	0:05	2	19:00	20:00	0:00	0:01	0:04	0:05	2
103	ST-101	Stopper Washer & Steriliser pre-rinse	0.5	6:00	6:10	0:00	0:02	0:08	0:10	2	14:00	14:10	0:00	0:02	0:08	0:10	2	20:00	20:10	0:00	0:02	0:08	0:10	2
104	ST-101	Stopper Processor	2	5:00	6:00	0:00	0:01	0:04	0:05	2	15:00	16:00	0:00	0:01	0:04	0:05	2	21:00	22:00	0:00	0:01	0:04	0:05	2
105	ST-102	Stopper Washer & Steriliser pre-rinse	0.5	6:00	6:10	0:00	0:02	0:08	0:10	2	14:00	14:10	0:00	0:02	0:08	0:10	2	22:00	22:10	0:00	0:02	0:08	0:10	2
106	ST-102	Stopper Processor	2	7:00	8:10	0:00	0:01	0:04	0:05	2	15:30	16:30	0:00	0:01	0:04	0:05	2	23:00	23:59	0:00	0:01	0:04	0:05	2
107	ST-103	Stopper Washer & Steriliser pre-rinse	0.5	0:35	0:40	0:00	0:02	0:08	0:10	2	2:00	2:10	0:00	0:02	0:08	0:10	2	11:00	11:10	0:00	0:02	0:08	0:10	2
108	ST-103	Stopper Processor	0.5	1:15	1:25	0:00	0:01	0:04	0:05	2	2:30	2:40	0:00	0:01	0:04	0:05	2	12:00	12:10	0:00	0:01	0:04	0:05	2
109	ST-104	Stopper Washer & Steriliser pre-rinse	0.5	2:15	2:25	0:00	0:02	0:08	0:10	2	3:30	3:40	0:00	0:02	0:08	0:10	2	13:00	13:10	0:00	0:02	0:08	0:10	2
110	ST-104	Stopper Processor	0.5	3:00	3:25	0:00	0:01	0:04	0:05	2	4:30	5:10	0:00	0:01	0:04	0:05	2	13:30	13:40	0:00	0:01	0:04	0:05	2
111	VF-100	Vial Rinser - Fill	0.5	0:15	0:30	0:00	0:02	0:08	0:10	3	1:30	2:00	0:00	0:02	0:08	0:10	3	3:00	3:30	0:00	0:02	0:08	0:10	3
112	VF-101	Vial Rinser - Fill	0.5	6:00	6:35	0:00	0:01	0:04	0:05	3	7:00	7:30	0:00	0:01	0:04	0:05	3	8:00	8:30	0:00	0:01	0:04	0:05	3
113	VF-102	Vial Rinser - Fill	0.5	10:00	10:35	0:00	0:02	0:08	0:10	3	11:00	11:35	0:00	0:02	0:08	0:10	3	12:00	12:30	0:00	0:02	0:08	0:10	3
114	VF-103	Vial Rinser - Fill	0.5	20:00	20:35	0:00	0:01	0:04	0:05	3	21:00	21:35	0:00	0:01	0:04	0:05	3	22:00	22:30	0:00	0:01	0:04	0:05	3
115																								

117		**SECOND INCREASE IN PRODUCTION**																					
118		**ADD ANOTHER FILLING SUITE**																					
119		**RESCHEDULE BY 2 HOURS (STAGGERED)**																					
120																							
121		**2. FILLING SUITE**																					
122	PW-100a	Parts Wash pre-rinse	1	2:40	2:50	0:00	0:01	0:04	0:05	2	13:00	13:10	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2
123	PW-100a	Fill Parts Washer	1	3:30	4:10	0:00	0:02	0:08	0:10	2	14:30	14:50	0:00	0:02	0:08	0:10	2	19:30	19:55	0:00	0:02	0:08	0:10	2
124	PW-101a	Parts Wash pre-rinse	1	5:15	5:30	0:00	0:01	0:04	0:05	2	14:45	14:55	0:00	0:01	0:04	0:05	2	18:30	18:40	0:00	0:01	0:04	0:05	2
125	PW-101a	Fill Parts Washer	1	7:30	7:40	0:00	0:02	0:08	0:10	2	15:00	15:10	0:00	0:02	0:08	0:10	2	19:00	19:10	0:00	0:02	0:08	0:10	2
126	PW-102a	Parts Wash pre-rinse	1	9:30	9:40	0:00	0:01	0:04	0:05	2	13:15	13:25	0:00	0:01	0:04	0:05	2	18:15	18:25	0:00	0:01	0:04	0:05	2
127	PW-102a	Fill Parts Washer	1	10:00	10:10	0:00	0:02	0:08	0:10	2	15:00	15:10	0:00	0:02	0:08	0:10	2	19:00	19:10	0:00	0:02	0:08	0:10	2
128	ST-100a	Stopper Washer & Steriliser pre-rinse	2	6:00	6:10	0:00	0:01	0:04	0:05	2	14:00	14:10	0:00	0:01	0:04	0:05	2	20:30	20:40	0:00	0:01	0:04	0:05	2
129	ST-100a	Stopper Processor	3	7:00	8:00	0:00	0:02	0:08	0:10	2	15:00	16:00	0:00	0:02	0:08	0:10	2	21:00	22:00	0:00	0:02	0:08	0:10	2
130	ST-101a	Stopper Washer & Steriliser pre-rinse	2	8:00	8:10	0:00	0:01	0:04	0:05	2	16:00	16:10	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2
131	ST-101a	Stopper Processor	3	7:00	8:00	0:00	0:02	0:08	0:10	2	17:00	18:00	0:00	0:02	0:08	0:10	2	23:00	23:25	0:00	0:02	0:08	0:10	2
132	ST-102a	Stopper Washer & Steriliser pre-rinse	2	8:00	8:10	0:00	0:01	0:04	0:05	2	16:00	16:10	0:00	0:01	0:04	0:05	2							
133	ST-102a	Stopper Processor	3	9:00	10:10	0:00	0:02	0:08	0:10	2	17:30	18:30	0:00	0:02	0:08	0:10	2							
134	ST-103a	Stopper Washer & Steriliser pre-rinse	2	2:35	2:40	0:00	0:01	0:04	0:05	2	4:00	4:10	0:00	0:01	0:04	0:05	2	13:00	13:10	0:00	0:01	0:04	0:05	2
135	ST-103a	Stopper Processor	2	3:15	3:25	0:00	0:02	0:08	0:10	2	4:30	4:40	0:00	0:02	0:08	0:10	2	14:00	14:10	0:00	0:02	0:08	0:10	2
136	ST-104a	Stopper Washer & Steriliser pre-rinse	2	4:15	4:25	0:00	0:01	0:04	0:05	2	5:30	5:40	0:00	0:01	0:04	0:05	2	15:00	15:10	0:00	0:01	0:04	0:05	2
137	ST-104a	Stopper Processor	2	5:00	5:25	0:00	0:02	0:08	0:10	2	6:30	7:10	0:00	0:02	0:08	0:10	2	15:30	15:40	0:00	0:02	0:08	0:10	2
138	VF-100a	Vial Rinser - Fill	0.5	2:15	2:25	0:00	0:01	0:04	0:05	3	3:00	3:30	0:00	0:01	0:04	0:05	3	4:00	4:30	0:00	0:01	0:04	0:05	3
139	VF-101a	Vial Rinser - Fill	0.5	8:00	8:35	0:00	0:02	0:08	0:10	3	9:00	9:35	0:00	0:02	0:08	0:10	3	10:00	10:30	0:00	0:02	0:08	0:10	3
140	VF-102a	Vial Rinser - Fill	0.5	12:00	12:35	0:00	0:01	0:04	0:05	3	13:00	13:35	0:00	0:01	0:04	0:05	3	14:00	14:30	0:00	0:01	0:04	0:05	3
141	VF-103a	Vial Rinser - Fill	0.5	22:00	22:35	0:00	0:02	0:08	0:10	3	23:00	23:35	0:00	0:02	0:08	0:10	3			0:00	0:02	0:08	0:10	
142																								
143	PW-101a	Parts washer Sanitization	2.5	23:00	23:05	0:00	0:02	0:08	0:10	2														
144						0:00	0:01	0:04	0:05															
145	ST-100a	Stopper Processor Sanitation RESCHEDULE fro	2.5	23:00	23:05	0:00	0:02	0:08	0:10	3														
146	ST-101a	Stopper Processor Sanitation	2.5	23:00	23:05	0:00	0:01	0:04	0:05	3														
147																								
148	ST-103a	Stopper Processor Sanitation	2.5	4:00	4:05	0:00	0:01	0:04	0:05	3														
149	ST-104a	Stopper Processor Sanitation	2.5	7:00	7:05	0:00	0:02	0:08	0:10	3														
150	VF-100a	Vial rinser Sanitisation	2.5	8:30	8:35	0:00	0:01	0:04	0:05	3	9:30	9:40	0:00	0:01	0:04	0:05	3							
151	VF-101a	Vial rinser Sanitisation	2.5	6:45	6:50	0:00	0:02	0:08	0:10	3	7:45	7:55	0:00	0:02	0:08	0:10	3							
152	VF-102a	Vial rinser Sanitisation	2.5	8:00	8:10	0:00	0:01	0:04	0:05	3	9:00	9:10	0:00	0:01	0:04	0:05	3							
153	VF-103a	Vial rinser Sanitisation	2.5	5:00	5:10	0:00	0:02	0:08	0:10	3														
154																								
155																								
156		The second Filling Suite requires additional Bulk																						
157		Processing, Washing & CIP from the existing																						
158		Suites as follows: 2 hours staggered																						
159																								
160		**1. BULK PROCESSING SUITE**																						
161	T-100	Pre-rinse	1.5	4:00	4:05	0:00	0:02	0:08	0:10	2	12:00	12:35	0:00	0:02	0:08	0:10	2	20:10	20:15	0:00	0:02	0:08	0:10	2
162	T-100	Fill	1	4:00	4:06	0:00	0:01	0:04	0:05	3	12:30	12:36	0:00	0:01	0:04	0:05	3	20:10	20:15	0:00	0:01	0:04	0:05	3
163	T-101	Pre-rinse	1.5	5:10	5:17	0:00	0:02	0:08	0:10	2	13:25	13:45	0:00	0:02	0:08	0:10	2	21:20	21:40	0:00	0:02	0:08	0:10	2
164	T-101	Fill	1	5:15	5:25	0:00	0:01	0:04	0:05	3	13:15	13:25	0:00	0:01	0:04	0:05	3	21:30	21:40	0:00	0:01	0:04	0:05	3
165	T-102	Pre-rinse	1.5	3:20	3:30	0:00	0:02	0:08	0:10	2	11:30	11:40	0:00	0:02	0:08	0:10	2	19:10	19:20	0:00	0:02	0:08	0:10	2
166	T-102	Fill	1	5:15	5:25	0:00	0:01	0:04	0:05	3	12:15	12:20	0:00	0:01	0:04	0:05	3	20:15	21:20	0:00	0:01	0:04	0:05	3
167	T-103	Pre-rinse	1.5	3:00	3:10	0:00	0:02	0:08	0:10	2	11:00	11:10	0:00	0:02	0:08	0:10	2	21:30	21:40	0:00	0:02	0:08	0:10	2
168	T-103	Fill	1	8:00	8:30	0:00	0:01	0:04	0:05	3														
169	BT-100	Buffer Tank pre-rinse	1.5	3:15	3:20	0:00	0:02	0:08	0:10	2	11:00	11:05	0:00	0:02	0:08	0:10	2	19:30	19:35	0:00	0:02	0:08	0:10	2
170	BT-100	Buffer Tank fill	1.5	4:00	4:10	0:00	0:01	0:04	0:05	2	12:00	12:10	0:00	0:01	0:04	0:05	2	21:00	21:10	0:00	0:01	0:04	0:05	2
171	BT-101	Buffer Tank pre-rinse	1.5	4:30	5:00	0:00	0:02	0:08	0:10	2	12:30	13:00	0:00	0:02	0:08	0:10	2	21:30	22:00	0:00	0:02	0:08	0:10	2
172	BT-101	Buffer Tank fill	1.5	5:00	5:30	0:00	0:01	0:04	0:05	2	13:00	13:30	0:00	0:01	0:04	0:05	2	22:00	22:30	0:00	0:01	0:04	0:05	2
173	BT-102	Buffer Tank pre-rinse	1.5	23:00	23:05	0:00	0:02	0:08	0:10	2														
174	BT-102	Buffer Tank fill	1.5	23:15	23:20	0:00	0:01	0:04	0:05	2														
175	BT-103	Buffer Tank pre-rinse	1.5	4:00	4:05	0:00	0:02	0:08	0:10	2														
176	BT-103	Buffer Tank fill	1.5	2:30	2:45	0:00	0:01	0:04	0:05	2														
177	N/A	Line Rinsing	1	2:30	2:45	0:00	0:02	0:08	0:10	1	8:00	8:15	0:00	0:02	0:08	0:10	1	9:20	9:35	0:00	0:02	0:08	0:10	1
178																								
179																								
180		**2. WASHING SUITE**																						
181	PW-100	Parts Washer Pre Rinse	2.5	4:30	4:35	0:00	0:02	0:08	0:10	1	7:30	7:35	0:00	0:02	0:08	0:10	1	8:40	8:45	0:00	0:02	0:08	0:10	1
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:0	3	6:30	7:45	0:00	0:01	0:04	0:05	1	14:30	14:45	0:00	0:01	0:04	0:05	1	18:00	18:15	0:00	0:01	0:04	0:05	1
183	PW-102	Parts Washer Pre Rinse	2.5	7:00	7:30	0:00	0:02	0:08	0:10	1	8:30	9:00	0:00	0:02	0:08	0:10	1	10:30	12:00	0:00	0:02	0:08	0:10	1
184	PW-100	Parts Washer Sanitization				0:00	0:01	0:04	0:05															
185	PW-101	Parts Washer Sanitization				0:00	0:02	0:08	0:10															
186	PW-102	Parts Washer Sanitization	2.5	6:00	6:05	0:00	0:01	0:04	0:05	1														
187																								
188		**4. CIP SUITE**																						
189	T-100	T-100	3	4:00	4:30	0:00	0:02	0:08	0:10	2	10:15	10:30	0:00	0:02	0:08	0:10	2	18:15	18:30	0:00	0:02	0:08	0:10	2
190	T-101	T-101	3	4:15	4:45	0:00	0:01	0:04	0:05	2	12:15	12:30	0:00	0:01	0:04	0:05	2	20:15	20:30	0:00	0:01	0:04	0:05	2
191	T-102	T-102 RESCHEDULE from 6:15	3	8:00	8:15	0:00	0:02	0:08	0:10	2	14:15	14:30	0:00	0:02	0:08	0:10	2	22:15	22:30	0:00	0:02	0:08	0:10	2
192	T-103	T-103	2.5	8:00	8:40	0:00	0:01	0:04	0:05	2	19:00	19:15	0:00	0:01	0:04	0:05	2							
193	VF-100	Vial Filler				0:00	0:02	0:08	0:10															
194	VF-101	Vial Filler	3	15:00	15:25	0:00	0:01	0:04	0:05	3														
195	VF-102	Vial Filler	3	19:00	19:25	0:00	0:02	0:08	0:10	3														
196	VF-103	Vial Filler RESCHEDULE from 21:00	3	20:00	20:25	0:00	0:01	0:04	0:05	3														
197	LYO-100	Lyophyliser RESCHEDULE from 7:00	2.5	8:00	8:30	0:00	0:02	0:08	0:10	3														
198	LYO-101	Lyophyliser	2.5	13:00	13:30	0:00	0:01	0:04	0:05	2														
199	LYO-102	Lyophyliser	2.5	23:00	23:30	0:00	0:02	0:08	0:10	3														
200																								
201																								
202																								
203																								
204																								
205	VFC-103	Vial Filler CIP skid	1	23:30	23:40	0:00	0:02	0:08	0:10	2														
206	LYC-100	Lyophyliser CIP skid	1	23:00	23:10	0:00	0:01	0:04	0:05	2														
207	LYC-101	Lyophyliser CIP skid	1	22:00	22:10	0:00	0:02	0:08	0:10	2														
208	LYC-102	Lyophyliser CIP skid	1	5:00	5:05	0:00	0:01	0:04	0:05	2														
209	LYC-103	Lyophyliser CIP skid	1	6:00	6:05	0:00	0:02	0:08	0:10	2														
210																								
211																								
212		**THIRD INCREASE IN PRODUCTION**																						
213		**BY MAXIMISING USE OF EQUIPMENT**																						
214																								
215		**1. BULK PROCESSING SUITE**																						
216	T-100	Fill	1	4:00	4:06	0:00	0:01	0:04	0:05	3	12:30	12:36	0:00	0:01	0:04	0:05	3	20:10	20:15	0:00	0:01	0:04	0:05	3
217	T-101	Fill	1	5:15	5:25	0:00	0:02	0:08	0:10	2	13:15	13:25	0:00	0:02	0:08	0:10	2	21:30	21:40	0:00	0:02	0:08	0:10	2
218	T-102	Fill	1	5:15	5:25	0:00	0:01	0:04	0:05	3	12:15	12:20	0:00	0:01	0:04	0:05	3	20:15	21:20	0:00	0:01	0:04	0:05	3
219	T-103	Fill	1	8:00	8:30	0:00	0:02	0:08	0:10	2														
220	BT-100	Buffer Tank fill	1.5	4:00	4:10	0:00	0:01	0:04	0:05	2	12:00	12:10	0:00	0:01	0:04	0:05	2	21:00	21:10	0:00	0:01	0:04	0:05	2
221	BT-101	Buffer Tank fill	1.5	5:00	5:30	0:00	0:02	0:08	0:10	2	13:00	13:30	0:00	0:02	0:08	0:10	2	22:00	22:30	0:00	0:02	0:08	0:10	2
222	BT-102	Buffer Tank fill	1.5	23:15	23:20	0:00	0:01	0:04	0:05	2														
223	BT-103	Buffer Tank fill	1.5	2:30	2:45	0:00	0:02	0:08	0:10	2														
224																								
225		**2. FILLING SUITE**																						
226	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5	6:00	6:10	0:00	0:01	0:04	0:05	2	14:00	14:10	0:00	0:01	0:04	0:05	2	20:30	20:40	0:00	0:01	0:04	0:05	2
227	ST-100a	Stopper Processor	2	7:00	8:00	0:00	0:02	0:08	0:10	2	15:00	16:00	0:00	0:02	0:08	0:10	2	21:00	22:00	0:00	0:02	0:08	0:10	2
228	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5	8:00	8:10	0:00	0:01	0:04	0:05	2	16:00	16:10	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2
229	ST-101a	Stopper Processor	2	7:00	8:00	0:00	0:02	0:08	0:10	2	17:00	18:00	0:00	0:02	0:08	0:10	2	23:00	23:25	0:00	0:02	0:08	0:10	2
230	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5	8:00	8:10	0:00	0:01	0:04	0:05	2	16:00	16:10	0:00	0:01	0:04	0:05	2			0:00	0:01	0:04	0:05	
231	ST-102a	Stopper Processor	2	9:00	10:10	0:00	0:02	0:08	0:10	2	17:30	18:30	0:00	0:02	0:08	0:10	2			0:00	0:02	0:08	0:10	
232	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5	2:35	2:40	0:00	0:01	0:04	0:05	2	4:00	4:10	0:00	0:01	0:04	0:05	2	13:00	13:10	0:00	0:01	0:04	0:05	2
233	ST-103a	Stopper Processor	0.5	3:15	3:25	0:00	0:02	0:08	0:10	2	4:30	4:40	0:00	0:02	0:08	0:10	2	14:00	14:10	0:00	0:02	0:08	0:10	2
234	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5	4:15	4:25	0:00	0:01	0:04	0:05	2	5:30	5:40	0:00	0:01	0:04	0:05	2	15:00	15:10	0:00	0:01	0:04	0:05	2
235	ST-104a	Stopper Processor	0.5	5:00	5:25	0:00	0:02	0:08	0:10	2	6:30	7:10	0:00	0:02	0:08	0:10	2	15:30	15:40	0:00	0:02	0:08	0:10	2
236	VF-100a	Vial Rinser - Fill	0.5	2:15	2:25	0:00	0:01	0:04	0:05	3	3:00	3:30	0:00	0:01	0:04	0:05	3	4:00	4:30	0:00	0:01	0:04	0:05	3
237	VF-101a	Vial Rinser - Fill	0.5	8:00	8:35	0:00	0:02	0:08	0:10	3	9:00	9:35	0:00	0:02	0:08	0:10	3	10:00	10:30	0:00	0:02	0:08	0:10	3
238	VF-102a	Vial Rinser - Fill	0.5	12:00	12:35	0:00	0:01	0:04	0:05	3	13:00	13:35	0:00	0:01	0:04	0:05	3	14:00	14:30	0:00	0:01	0:04	0:05	3
239	VF-103a	Vial Rinser - Fill	0.5	22:00	22:35	0:00	0:02	0:08	0:10	3	23:00	23:35	0:00	0:02	0:08	0:10	3							

244		**FOURTH INCREASE IN PRODUCTION**																						
245		**ADD THIRD FILLING SUITE**																						
246		**RESCHEDULE BY 2 HOURS (STAGGERED)**																						
247																								
248		**2. FILLING SUITE**																						
249	PW-100a	Parts Wash pre-rinse	2.5	4:40	4:50	0:00	0:01	0:04	0:05	2	15:00	15:10	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2
250	PW-100a	Fill Parts Washer	5	5:30	6:10	0:00	0:02	0:08	0:10	2	16:30	16:50	0:00	0:02	0:08	0:10	2	21:30	21:55	0:00	0:02	0:08	0:10	2
251	PW-101a	Parts Wash pre-rinse	2.5	7:15	7:30	0:00	0:01	0:04	0:05	2	16:45	16:55	0:00	0:01	0:04	0:05	2	20:30	22:40	0:00	0:01	0:04	0:05	2
252	PW-101a	Fill Parts Washer	2.5	9:30	9:40	0:00	0:02	0:08	0:10	2	17:00	17:10	0:00	0:02	0:08	0:10	2	21:00	23:10	0:00	0:02	0:08	0:10	2
253	PW-102a	Parts Wash pre-rinse	2.5	11:30	11:40	0:00	0:01	0:04	0:05	2	15:15	17:25	0:00	0:01	0:04	0:05	2	20:15	20:25	0:00	0:01	0:04	0:05	2
254	PW-102a	Fill Parts Washer	2.5	12:00	12:10	0:00	0:02	0:08	0:10	2	17:00	17:10	0:00	0:02	0:08	0:10	2	21:00	21:10	0:00	0:02	0:08	0:10	2
255	ST-100a	Stopper Washer & Steriliser pre-rinse	2	8:00	8:10	0:00	0:01	0:04	0:05	2	16:00	18:10	0:00	0:01	0:04	0:05	2	22:30	22:40	0:00	0:01	0:04	0:05	2
256	ST-100a	Stopper Processor	3	9:00	10:00	0:00	0:02	0:08	0:10	2	17:00	18:00	0:00	0:02	0:08	0:10	2							
257	ST-101a	Stopper Washer & Steriliser pre-rinse	2	11:00	11:10	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2							
258	ST-101a	Stopper Processor	3	9:00	10:00	0:00	0:02	0:08	0:10	2	19:00	20:00	0:00	0:02	0:08	0:10	2							
259	ST-102a	Stopper Washer & Steriliser pre-rinse	2	10:00	10:10	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2							
260	ST-102a	Stopper Processor	3	11:00	12:10	0:00	0:02	0:08	0:10	2	19:30	20:30	0:00	0:02	0:08	0:10	2							
261	ST-103a	Stopper Washer & Steriliser pre-rinse	2	5:35	5:40	0:00	0:01	0:04	0:05	2	6:00	6:10	0:00	0:01	0:04	0:05	2	15:00	15:10	0:00	0:01	0:04	0:05	2
262	ST-103a	Stopper Processor	2	5:15	5:25	0:00	0:02	0:08	0:10	2	6:30	6:40	0:00	0:02	0:08	0:10	2	16:00	16:10	0:00	0:02	0:08	0:10	2
263	ST-104a	Stopper Washer & Steriliser pre-rinse	2	6:15	6:25	0:00	0:01	0:04	0:05	2	7:30	7:40	0:00	0:01	0:04	0:05	2	17:00	17:10	0:00	0:01	0:04	0:05	2
264	ST-104a	Stopper Processor	2	7:00	7:25	0:00	0:02	0:08	0:10	2	8:30	9:10	0:00	0:02	0:08	0:10	2	17:30	17:40	0:00	0:02	0:08	0:10	2
265	VF-100a	Vial Rinser - Fill	0.5	4:15	4:25	0:00	0:01	0:04	0:05	3	5:00	5:30	0:00	0:01	0:04	0:05	3	6:00	6:30	0:00	0:01	0:04	0:05	3
266	VF-101a	Vial Rinser - Fill	0.5	10:00	10:35	0:00	0:02	0:08	0:10	3	11:00	11:35	0:00	0:02	0:08	0:10	3	12:00	12:30	0:00	0:02	0:08	0:10	3
267	VF-102a	Vial Rinser - Fill	0.5	14:00	14:35	0:00	0:01	0:04	0:05	3	15:00	15:35	0:00	0:01	0:04	0:05	3	16:00	16:30	0:00	0:01	0:04	0:05	3
268																								
269																								
270	PW-101a	Parts washer Sanitization	2.5	1:00	1:05	0:00	0:02	0:08	0:10	2														
271																								
272	ST-100a	Stopper Processor Sanitation	2.5	1:00	1:05	0:00	0:02	0:08	0:10	3														
273	ST-101a	Stopper Processor Sanitation	2.5	0:30	0:35	0:00	0:01	0:04	0:05	3														
274																								
275	ST-103a	Stopper Processor Sanitation	2.5	6:00	6:05	0:00	0:01	0:04	0:05	3														
276	ST-104a	Stopper Processor Sanitation	2.5	9:00	9:05	0:00	0:02	0:08	0:10	3														
277	VF-100a	Vial rinser Sanitisation	2.5	10:30	10:35	0:00	0:01	0:04	0:05	3	11:30	11:40	0:00	0:01	0:04	0:05	3							
278	VF-101a	Vial rinser Sanitisation	2.5	8:45	8:50	0:00	0:02	0:08	0:10	3	9:45	9:55	0:00	0:02	0:08	0:10	3							
279	VF-102a	Vial rinser Sanitisation	2.5	10:00	10:10	0:00	0:01	0:04	0:05	3	11:00	11:10	0:00	0:01	0:04	0:05	3							
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00	2.5	8:00	8:10	0:00	0:02	0:08	0:10	3														
281																								
282																								
283																								
284																								
285																								
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																						
287		**1. BULK PROCESSING SUITE**																						
288	T-100	Pre-rinse	1.5	6:00	6:05	0:00	0:02	0:08	0:10	2	14:00	14:35	0:00	0:02	0:08	0:10	2	22:10	22:15	0:00	0:02	0:08	0:10	2
289	T-100	Fill	1	6:00	6:06	0:00	0:01	0:04	0:05	3	14:30	14:36	0:00	0:01	0:04	0:05	3	22:10	22:15	0:00	0:01	0:04	0:05	3
290	T-101	Pre-rinse	1.5	7:10	7:17	0:00	0:02	0:08	0:10	2	15:25	15:45	0:00	0:02	0:08	0:10	2	23:20	23:40	0:00	0:02	0:08	0:10	2
291	T-101	Fill	1	7:15	7:25	0:00	0:01	0:04	0:05	3	15:15	15:25	0:00	0:01	0:04	0:05	3	23:30	23:40	0:00	0:01	0:04	0:05	3
292	T-102	Pre-rinse	1.5	5:20	5:30	0:00	0:02	0:08	0:10	2	13:30	13:40	0:00	0:02	0:08	0:10	2	21:10	21:20	0:00	0:02	0:08	0:10	2
293	T-102	Fill	1	7:15	7:25	0:00	0:01	0:04	0:05	3	14:15	14:20	0:00	0:01	0:04	0:05	3	22:15	22:20	0:00	0:01	0:04	0:05	3
294	T-103	Pre-rinse	1.5	5:00	5:10	0:00	0:02	0:08	0:10	2	13:00	13:10	0:00	0:02	0:08	0:10	2	23:30	23:40	0:00	0:02	0:08	0:10	2
295	T-103	Fill	1	11:00	11:30	0:00	0:01	0:04	0:05	3														
296	BT-100	Buffer Tank pre-rinse	1.5	5:15	5:20	0:00	0:02	0:08	0:10	2	13:00	13:05	0:00	0:02	0:08	0:10	2	21:30	21:35	0:00	0:02	0:08	0:10	2
297	BT-100	Buffer Tank fill	1.5	6:00	6:10	0:00	0:01	0:04	0:05	2	14:00	14:10	0:00	0:01	0:04	0:05	2	23:00	23:10	0:00	0:01	0:04	0:05	2
298	BT-101	Buffer Tank pre-rinse	1.5	6:30	7:00	0:00	0:02	0:08	0:10	2	14:30	15:00	0:00	0:02	0:08	0:10	2							
299	BT-101	Buffer Tank fill	1.5	7:00	7:30	0:00	0:01	0:04	0:05	2	15:00	15:30	0:00	0:01	0:04	0:05	2							
300	BT-102	Buffer Tank pre-rinse																						
301	BT-102	Buffer Tank fill																						
302	BT-103	Buffer Tank pre-rinse	1.5	6:00	6:05	0:00	0:02	0:08	0:10	2			0:00	0:02	0:08	0:10	2							
303	BT-103	Buffer Tank fill	1.5	4:30	4:45	0:00	0:01	0:04	0:05	2			0:00	0:01	0:04	0:05	2							
304	N/A	Line Rinsing	1	4:30	4:45	0:00	0:02	0:08	0:10	2	10:00	10:15	0:00	0:02	0:08	0:10	2	11:20	7:35	0:00	0:02	0:08	0:10	2
305																								
306																								
307																								
308																								
309		**2. WASHING SUITE**																						
310	PW-100	Parts Washer Pre Rinse	2.5	6:30	6:35	0:00	0:01	0:04	0:05	1	9:30	9:35	0:00	0:01	0:04	0:05	1	10:40	10:45	0:00	0:01	0:04	0:05	1
311	PW-101	Parts Washer Pre Rinse	5	8:00	9:15	0:00	0:02	0:08	0:10	1	16:30	16:45	0:00	0:02	0:08	0:10	1	20:00	20:15	0:00	0:02	0:08	0:10	1
312	PW-102	Parts Washer Pre Rinse	2.5	9:00	9:30	0:00	0:01	0:04	0:05	1	10:30	11:00	0:00	0:01	0:04	0:05	1	14:30	15:00	0:00	0:01	0:04	0:05	1
313	PW-100	Parts Washer Sanitization																						
314	PW-101	Parts Washer Sanitization																						
315	PW-102	Parts Washer Sanitization	2.5	10:00	10:10	0:00	0:02	0:08	0:10	1														
316																								
317		**4. CIP SUITE**																						
318	T-100	T-100	5	6:00	6:30	0:00	0:01	0:04	0:05	2	12:15	12:30	0:00	0:01	0:04	0:05	2	20:15	20:30	0:00	0:01	0:04	0:05	2
319	T-101	T-101	5	6:15	6:45	0:00	0:02	0:08	0:10	2	16:15	16:30	0:00	0:02	0:08	0:10	2	22:15	22:30	0:00	0:02	0:08	0:10	2
320	T-102	T-102	5	1:15	1:30	0:00	0:01	0:04	0:05	2	16:15	16:30	0:00	0:01	0:04	0:05	2	1:15	1:30	0:00	0:01	0:04	0:05	2
321	T-103	T-103	2.5	10:00	10:40	0:00	0:02	0:08	0:10	2	21:00	21:15	0:00	0:02	0:08	0:10	2							
322	VF-100	Vial Filler	2.5	3:00	3:40	0:00	0:01	0:04	0:05	2														
323	VF-101	Vial Filler	5	17:00	17:25	0:00	0:02	0:08	0:10	3														
324	VF-102	Vial Filler RESCHEDULE from 21:00	5	20:00	20:25	0:00	0:01	0:04	0:05	3														
325	VF-103	Vial Filler	5	23:00	23:25	0:00	0:02	0:08	0:10	3														
326	LYO-100	Lyophyliser	2.5	9:00	9:30	0:00	0:01	0:04	0:05	2														
327	LYO-101	Lyophyliser	2.5	15:00	15:30	0:00	0:02	0:08	0:10	2														
328	LYO-102	Lyophyliser	2.5	1:30	2:00	0:00	0:01	0:04	0:05	2														
329	LYO-103	Lyophyliser	2.5	2:30	2:59	0:00	0:02	0:08	0:10	2														
330	LYO-104	Lyophyliser	2.5	2:30	2:59	0:00	0:01	0:04	0:05	2														
331	VFC-100	Vial Filler CIP skid	1	2:30	2:40	0:00	0:02	0:08	0:10	3														
332	VFC-101	Vial Filler CIP skid	1	2:00	2:10	0:00	0:01	0:04	0:05	3														
333	VFC-102	Vial Filler CIP skid	1	3:00	3:10	0:00	0:02	0:08	0:10	3														
334	VFC-103	Vial Filler CIP skid	1	1:30	1:40	0:00	0:01	0:04	0:05	3														
335	LYC-100	Lyophyliser CIP skid	1	1:20	1:30	0:00	0:02	0:08	0:10	2														
336	LYC-101	Lyophyliser CIP skid	1	1:15	1:25	0:00	0:01	0:04	0:05	2														
337	LYC-102	Lyophyliser CIP skid	1	7:00	7:05	0:00	0:02	0:08	0:10	2														
338	LYC-103	Lyophyliser CIP skid	1	8:00	8:05	0:00	0:01	0:04	0:05	2														

342		**FIFTH INCREASE IN PRODUCTION**																					
343		**BY MAXIMISING USE OF EQUIPMENT**																					
344																							
345		**1. BULK PROCESSING SUITE**																					
346	T-100	Fill	1	6:00	6:06	0:00	0:01	0:04	0:05	3	14:30	14:36	0:00	0:01	0:04	0:05	3	22:10	22:15	0:00	0:01	0:04	0:05	3
347	T-101	Fill	1	7:15	7:25	0:00	0:02	0:08	0:10	3	15:15	15:25	0:00	0:02	0:08	0:10	3	23:30	23:40	0:00	0:02	0:08	0:10	3
348	T-102	Fill	1	5:15	5:25	0:00	0:01	0:04	0:05	3	12:15	12:20	0:00	0:01	0:04	0:05	3	20:15	21:20	0:00	0:01	0:04	0:05	3
349	T-103	Fill	1	7:15	7:25	0:00	0:02	0:08	0:10	3	14:15	14:20	0:00	0:02	0:08	0:10	3	22:15	22:20	0:00	0:02	0:08	0:10	3
350	BT-100	Buffer Tank fill	1.5	6:00	6:10	0:00	0:01	0:04	0:05	3	14:00	14:10	0:00	0:01	0:04	0:05	3	23:00	23:10	0:00	0:01	0:04	0:05	3
351	BT-101	Buffer Tank fill	1.5	7:00	7:30	0:00	0:02	0:08	0:10	3	15:00	15:30	0:00	0:02	0:08	0:10	3							
352	BT-102	Buffer Tank fill																						
353	BT-103	Buffer Tank fill	1.5	6:00	6:05	0:00	0:02	0:08	0:10	3														
354																								
355		**2. FILLING SUITE**																						
356	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5	8:00	8:10	0:00	0:01	0:04	0:05	2	16:00	18:10	0:00	0:01	0:04	0:05	2	22:30	22:40	0:00	0:01	0:04	0:05	2
357	ST-100a	Stopper Processor	2	9:00	10:00	0:00	0:02	0:08	0:10	2	17:00	18:00	0:00	0:02	0:08	0:10	2							
358	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5	11:00	11:10	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2							
359	ST-101a	Stopper Processor	2	9:00	10:00	0:00	0:02	0:08	0:10	2	19:00	20:00	0:00	0:02	0:08	0:10	2							
360	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5	10:00	10:10	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2							
361	ST-102a	Stopper Processor	2	11:00	12:10	0:00	0:02	0:08	0:10	2	19:30	20:30	0:00	0:02	0:08	0:10	2							
362	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5	5:35	5:40	0:00	0:01	0:04	0:05	2	6:50	7:05	0:00	0:01	0:04	0:05	2	15:00	15:10	0:00	0:01	0:04	0:05	2
363	ST-103a	Stopper Processor	0.5	5:15	5:25	0:00	0:02	0:08	0:10	2	6:30	6:40	0:00	0:02	0:08	0:10	2	16:00	16:10	0:00	0:02	0:08	0:10	2
364	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5	6:15	6:25	0:00	0:01	0:04	0:05	2	7:30	7:40	0:00	0:01	0:04	0:05	2	17:00	17:10	0:00	0:01	0:04	0:05	2
365	ST-104a	Stopper Processor	0.5	7:00	7:25	0:00	0:01	0:04	0:05	2	8:30	9:10	0:00	0:01	0:04	0:05	2	17:30	17:40	0:00	0:01	0:04	0:05	2
366	VF-100a	Vial Rinser - Fill	0.5	4:15	4:25	0:00	0:02	0:08	0:10	2	5:00	5:30	0:00	0:02	0:08	0:10	2	6:00	6:30	0:00	0:02	0:08	0:10	2
367	VF-101a	Vial Rinser - Fill	0.5	10:00	10:35	0:00	0:01	0:04	0:05	2	11:00	11:35	0:00	0:01	0:04	0:05	2	12:00	12:30	0:00	0:01	0:04	0:05	2
368	VF-102a	Vial Rinser - Fill	0.5	14:00	14:35	0:00	0:02	0:08	0:10	2	15:00	15:35	0:00	0:02	0:08	0:10	2	16:00	16:30	0:00	0:02	0:08	0:10	2
369																								
370																								
371		**SIXTH INCREASE IN PRODUCTION**																						
372		**ADD FOURTH FILLING SUITE**																						
373		**RESCHEDULE BY 2 HOURS (STAGGERED)**																						
374																								
375		**2. FILLING SUITE**																						
376	PW-100a	Parts Wash pre-rinse	2.5	6:40	6:50	0:00	0:01	0:04	0:05	2	17:00	17:10	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2
377	PW-100a	Fill Parts Washer	5	7:30	8:10	0:00	0:02	0:08	0:10	2	18:30	18:50	0:00	0:02	0:08	0:10	2							
378	PW-101a	Parts Wash pre-rinse	2.5	9:15	9:30	0:00	0:01	0:04	0:05	2	18:45	18:55	0:00	0:01	0:04	0:05	2	22:30	23:55	0:00	0:01	0:04	0:05	2
379	PW-101a	Fill Parts Washer	2.5	11:30	11:40	0:00	0:02	0:08	0:10	2	19:00	19:10	0:00	0:02	0:08	0:10	2	23:00	23:55	0:00	0:02	0:08	0:10	2
380	PW-102a	Parts Wash pre-rinse	2.5	13:30	13:40	0:00	0:01	0:04	0:05	2	19:15	19:25	0:00	0:01	0:04	0:05	2	22:15	23:55	0:00	0:01	0:04	0:05	2
381	PW-102a	Fill Parts Washer	2.5	14:00	14:10	0:00	0:02	0:08	0:10	2	19:00	19:10	0:00	0:02	0:08	0:10	2	23:00	23:55	0:00	0:02	0:08	0:10	2
382	ST-100a	Stopper Washer & Steriliser pre-rinse	2	10:00	10:10	0:00	0:01	0:04	0:05	2	18:00	20:10	0:00	0:01	0:04	0:05	2							
383	ST-100a	Stopper Processor	3	11:00	12:00	0:00	0:02	0:08	0:10	2	19:00	20:00	0:00	0:02	0:08	0:10	2							
384	ST-101a	Stopper Washer & Steriliser pre-rinse	2	13:00	13:10	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2							
385	ST-101a	Stopper Processor	3	11:00	12:00	0:00	0:01	0:04	0:05	2	21:00	22:00	0:00	0:01	0:04	0:05	2							
386	ST-102a	Stopper Washer & Steriliser pre-rinse	2	12:00	12:10	0:00	0:02	0:08	0:10	2	20:00	20:10	0:00	0:02	0:08	0:10	2							
387	ST-102a	Stopper Processor	3	13:00	14:10	0:00	0:01	0:04	0:05	2	21:30	22:30	0:00	0:01	0:04	0:05	2							
388	ST-103a	Stopper Washer & Steriliser pre-rinse	2	7:35	7:40	0:00	0:02	0:08	0:10	2	8:50	9:05	0:00	0:02	0:08	0:10	2	17:00	17:10	0:00	0:02	0:08	0:10	2
389	ST-103a	Stopper Processor	2	7:15	7:25	0:00	0:01	0:04	0:05	2	8:30	8:40	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2
390	ST-104a	Stopper Washer & Steriliser pre-rinse	2	8:15	8:25	0:00	0:02	0:08	0:10	2	9:30	9:40	0:00	0:02	0:08	0:10	2	19:00	19:10	0:00	0:02	0:08	0:10	2
391	ST-104a	Stopper Processor	2	9:00	9:25	0:00	0:01	0:04	0:05	2	10:30	11:10	0:00	0:01	0:04	0:05	2	19:30	19:40	0:00	0:01	0:04	0:05	2
392	VF-100a	Vial Rinser - Fill	0.5	6:15	6:25	0:00	0:02	0:08	0:10	3	7:00	7:30	0:00	0:02	0:08	0:10	3	8:00	8:30	0:00	0:02	0:08	0:10	3
393	VF-101a	Vial Rinser - Fill	0.5	12:00	12:35	0:00	0:01	0:04	0:05	3	13:00	13:35	0:00	0:01	0:04	0:05	3	14:00	14:30	0:00	0:01	0:04	0:05	3
394	VF-102a	Vial Rinser - Fill	0.5	16:00	16:35	0:00	0:02	0:08	0:10	3	17:00	17:35	0:00	0:02	0:08	0:10	3	18:00	18:30	0:00	0:02	0:08	0:10	3
395																								
396																								
397	PW-101a	Parts washer Sanitization	2.5	3:00	3:05	0:00	0:01	0:04	0:05	2														
398																								
399	ST-100a	Stopper Processor Sanitation	2.5	3:00	3:05	0:00	0:01	0:04	0:05	3														
400	ST-101a	Stopper Processor Sanitation	2.5	2:30	2:35	0:00	0:02	0:08	0:10	3														
401																								
402	ST-103a	Stopper Processor Sanitation	2.5	8:00	8:05	0:00	0:02	0:08	0:10	3														
403	ST-104a	Stopper Processor Sanitation	2.5	11:00	11:05	0:00	0:01	0:04	0:05	3														
404	VF-100a	Vial rinser Sanitisation	2.5	12:30	12:35	0:00	0:02	0:08	0:10	3	13:30	15:40	0:00	0:02	0:08	0:10	3							
405	VF-101a	Vial rinser Sanitisation	2.5	10:45	10:50	0:00	0:01	0:04	0:05	3	11:45	11:55	0:00	0:01	0:04	0:05	3							
406	VF-102a	Vial rinser Sanitisation	2.5	12:00	12:10	0:00	0:02	0:08	0:10	3	13:00	13:10	0:00	0:02	0:08	0:10	3							
407	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00	2.5																					
408																								
409																								
410																								
411		**SEVENTH INCREASE IN PRODUCTION**																						
412		**BY MAXIMISING USE OF EQUIPMENT**																						
413																								
414		**1. BULK PROCESSING SUITE**																						
415	T-100	Fill	1	7:00	8:06	0:00	0:01	0:04	0:05	3	16:30	16:36	0:00	0:01	0:04	0:05	3							
416	T-101	Fill	1	9:15	9:25	0:00	0:02	0:08	0:10	3	17:15	17:25	0:00	0:02	0:08	0:10	3							
417	T-102	Fill	1	7:15	7:25	0:00	0:01	0:04	0:05	3	14:15	14:20	0:00	0:01	0:04	0:05	3	22:15	23:20	0:00	0:01	0:04	0:05	3
418	T-103	Fill	1	9:15	9:25	0:00	0:02	0:08	0:10	3	16:15	16:20	0:00	0:02	0:08	0:10	3							
419	BT-100	Buffer Tank fill	1.5	8:00	8:10	0:00	0:01	0:04	0:05	2	16:00	16:10	0:00	0:01	0:04	0:05	2							
420	BT-101	Buffer Tank fill	1.5	9:00	9:30	0:00	0:02	0:08	0:10	2	17:00	17:30	0:00	0:02	0:08	0:10	2							
421	BT-102	Buffer Tank fill																						
422	BT-103	Buffer Tank fill	1.5	8:00	8:05	0:00	0:02	0:08	0:10	2														
423																								
424		**2. FILLING SUITE**																						
425	ST-100a	Stopper Washer & Steriliser pre-rinse	0.5	10:00	10:10	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2							
426	ST-100a	Stopper Processor	2	11:00	12:00	0:00	0:02	0:08	0:10	2	19:00	20:00	0:00	0:02	0:08	0:10	2							
427	ST-101a	Stopper Washer & Steriliser pre-rinse	0.5	13:00	13:10	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2							
428	ST-101a	Stopper Processor	2	11:00	12:00	0:00	0:02	0:08	0:10	2	21:00	22:00	0:00	0:02	0:08	0:10	2							
429	ST-102a	Stopper Washer & Steriliser pre-rinse	0.5	12:00	12:10	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2							
430	ST-102a	Stopper Processor	2	12:00	14:10	0:00	0:02	0:08	0:10	2	21:30	22:30	0:00	0:02	0:08	0:10	2							
431	ST-103a	Stopper Washer & Steriliser pre-rinse	0.5	7:35	7:40	0:00	0:01	0:04	0:05	2	8:50	9:05	0:00	0:01	0:04	0:05	2	17:00	17:10	0:00	0:01	0:04	0:05	2
432	ST-103a	Stopper Processor	0.5	7:15	7:25	0:00	0:02	0:08	0:10	2	8:30	8:40	0:00	0:02	0:08	0:10	2	18:00	18:10	0:00	0:02	0:08	0:10	2
433	ST-104a	Stopper Washer & Steriliser pre-rinse	0.5	8:15	8:25	0:00	0:01	0:04	0:05	2	9:30	9:40	0:00	0:01	0:04	0:05	2	19:00	19:10	0:00	0:01	0:04	0:05	2
434	ST-104a	Stopper Processor	0.5	9:00	9:25	0:00	0:01	0:04	0:05	2	10:30	11:10	0:00	0:01	0:04	0:05	2	19:30	19:40	0:00	0:01	0:04	0:05	2
435	VF-100a	Vial Rinser - Fill	0.5	6:15	6:25	0:00	0:02	0:08	0:10	3	7:00	7:30	0:00	0:02	0:08	0:10	3	8:00	8:30	0:00	0:02	0:08	0:10	3
436	VF-101a	Vial Rinser - Fill	0.5	12:00	12:35	0:00	0:01	0:04	0:05	3	13:00	13:35	0:00	0:01	0:04	0:05	3	14:00	14:30	0:00	0:01	0:04	0:05	3
437	VF-102a	Vial Rinser - Fill	0.5	16:00	16:35	0:00	0:02	0:08	0:10	3	17:00	17:35	0:00	0:02	0:08	0:10	3	18:00	18:30	0:00	0:02	0:08	0:10	3

	WFI Water Offtake Input Data				WFI Water Offtake Input Data													WFI Water Offtake Input Data									
					4													5									6
No.	Tag No.	Description of Offtake	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority
1		**ORIGINAL PROCESS DATA**																									
2		**1. BULK PROCESSING SUITE**																									
3	T-100	Pre-rinse																									
4	T-100	Fill																									
5	T-101	Pre-rinse																									
6	T-101	Fill																									
7	T-102	Pre-rinse																									
8	T-102	Fill																									
9	T-103	Pre-rinse																									
10	T-103	Fill																									
11	BT-100	Buffer Tank pre-rinse																									
12	BT-100	Buffer Tank fill																									
13	BT-101	Buffer Tank pre-rinse																									
14	BT-101	Buffer Tank fill																									
15	BT-102	Buffer Tank pre-rinse																									
16	BT-102	Buffer Tank fill																									
17	BT-103	Buffer Tank pre-rinse																									
18	BT-103	Buffer Tank fill																									
19	N/A	Line Rinsing	8:40	8:55	0:00	0:02	0:08	0:10	2	9:45	10:00	0:00	0:02	0:08	0:10	2	11:00	11:15	0:00	0:02	0:08	0:10	2
20																											
21		**2. FILLING SUITE**																									
22	PW-100	Parts Wash pre-rinse																									
23	PW-100	Fill Parts Washer																									
24	PW-101	Parts Wash pre-rinse																									
25	PW-101	Fill Parts Washer																									
26	PW-102	Parts Wash pre-rinse																									
27	PW-102	Fill Parts Washer																									
28	ST-100	Stopper Washer & Steriliser pre-rinse																									
29	ST-100	Stopper Processor																									
30	ST-101	Stopper Washer & Steriliser pre-rinse																									
31	ST-101	Stopper Processor																									
32	ST-102	Stopper Washer & Steriliser pre-rinse																									
33	ST-102	Stopper Processor																									
34	ST-103	Stopper Washer & Steriliser pre-rinse	13:00	13:40	0:00	0:01	0:04	0:05	2	18:00	18:10	0:00	0:01	0:04	0:05	2	19:30	19:40	0:00	0:01	0:04	0:05	2
35	ST-103	Stopper Processor	13:40	13:50	0:00	0:02	0:08	0:10	2	19:00	19:10	0:00	0:02	0:08	0:10	2	20:30	20:40	0:00	0:02	0:08	0:10	2
36	ST-104	Stopper Washer & Steriliser pre-rinse	14:30	14:50	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2	21:30	21:40	0:00	0:01	0:04	0:05	2
37	ST-104	Stopper Processor	15:00	15:20	0:00	0:02	0:08	0:10	2	20:30	20:40	0:00	0:02	0:08	0:10	2	22:00	22:10	0:00	0:02	0:08	0:10	2
38	VF-100	Vial Rinser - Fill																									
39	VF-101	Vial Rinser - Fill																									
40	VF-102	Vial Rinser - Fill																									
41	VF-103	Vial Rinser - Fill																									
42	PW-100	Parts washer Sanitization																									
43	PW-101	Parts washer Sanitization																									
44	PW-102	Parts washer Sanitization																									
45	ST-100	Stopper Processor Sanitation																									
46	ST-101	Stopper Processor Sanitation																									
47	ST-102	Stopper Processor Sanitation																									
48	ST-103	Stopper Processor Sanitation																									
49	ST-104	Stopper Processor Sanitation																									
50	VF-100	Vial rinser Sanitisation																									
51	VF-101	Vial rinser Sanitisation																									
52	VF-102	Vial rinser Sanitisation																									
53	VF-103	Vial rinser Sanitisation																									
54																											
55		**2. WASHING SUITE**																									
56	PW-100	Parts Washer Pre Rinse	9:00	9:05	0:00	0:01	0:04	0:05	1	10:30	10:35	0:00	0:01	0:04	0:05	1	15:30	15:35	0:00	0:01	0:04	0:05	1
57	PW-101	Parts Washer Pre Rinse																									
58	PW-102	Parts Washer Pre Rinse	13:00	13:30	0:00	0:01	0:04	0:05	1	21:30	22:00	0:00	0:01	0:04	0:05	1	23:30	23:55	0:00	0:01	0:04	0:05	1
59	PW-100	Parts Washer Sanitization																									
60	PW-101	Parts Washer Sanitization																									
61	PW-102	Parts Washer Sanitization																									
62																											
63		**4. CIP SUITE**																									
64	T-100	T-100																									
65	T-101	T-101																									
66	T-102	T-102																									
67	T-103	T-103 RESCHEDULE from 6:00																									
68	VF-100	Vial Filler																									
69	VF-101	Vial Filler																									
70	VF-102	Vial Filler																									
71	VF-103	Vial Filler																									
72	LYO-100	Lyophyliser																									
73	LYO-101	Lyophyliser																									
74	LYO-102	Lyophyliser																									
75	LYO-103	Lyophyliser																									
76	LYO-104	Lyophyliser																									
77	VFC-100	Vial Filler CIP skid																									
78	VFC-101	Vial Filler CIP skid																									
79	VFC-102	Vial Filler CIP skid																									
80	VFC-103	Vial Filler CIP skid																									
81	LYC-100	Lyophyliser CIP skid																									
82	LYC-101	Lyophyliser CIP skid																									
83	LYC-102	Lyophyliser CIP skid																									
84	LYC-103	Lyophyliser CIP skid																									
85																											
86																											
87		**FIRST INCREASE IN PRODUCTION**																									
88		**BY MAXIMISING USE OF EQUIPMENT**																									
89																											
90		**1. BULK PROCESSING SUITE**																									
91	T-100	Fill																									
92	T-101	Fill																									
93	T-102	Fill																									
94	T-103	Fill																									
95	BT-100	Buffer Tank fill																									
96	BT-101	Buffer Tank fill																									
97	BT-102	Buffer Tank fill																									
98	BT-103	Buffer Tank fill																									
99																											
100		**2. FILLING SUITE**																									
101	ST-100	Stopper Washer & Steriliser pre-rinse																									
102	ST-100	Stopper Processor																									
103	ST-101	Stopper Washer & Steriliser pre-rinse																									
104	ST-101	Stopper Processor																									
105	ST-102	Stopper Washer & Steriliser pre-rinse																									
106	ST-102	Stopper Processor																									
107	ST-103	Stopper Washer & Steriliser pre-rinse	13:00	13:40	0:00	0:02	0:08	0:10	2	18:00	18:10	0:00	0:02	0:08	0:10	2	19:30	19:40	0:00	0:02	0:08	0:10	2
108	ST-103	Stopper Processor	13:40	13:50	0:00	0:01	0:04	0:05	2	19:00	19:10	0:00	0:01	0:04	0:05	2	20:30	20:40	0:00	0:01	0:04	0:05	2
109	ST-104	Stopper Washer & Steriliser pre-rinse	14:30	14:50	0:00	0:02	0:08	0:10	2	20:00	20:10	0:00	0:02	0:08	0:10	2	21:30	21:40	0:00	0:02	0:08	0:10	2
110	ST-104	Stopper Processor	15:00	15:10	0:00	0:01	0:04	0:05	2	20:30	20:40	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2
111	VF-100	Vial Rinser - Fill																									

117		**SECOND INCREASE IN PRODUCTION**																				
118		**ADD ANOTHER FILLING SUITE**																				
119		**RESCHEDULE BY 2 HOURS (STAGGERED)**																				
120																						
121		**2. FILLING SUITE**																				
122	PW-100a	Parts Wash pre-rinse																				
123	PW-100a	Fill Parts Washer																				
124	PW-101a	Parts Wash pre-rinse																				
125	PW-101a	Fill Parts Washer																				
126	PW-102a	Parts Wash pre-rinse																				
127	PW-102a	Fill Parts Washer																				
128	ST-100a	Stopper Washer & Sterilliser pre-rinse																				
129	ST-100a	Stopper Processor																				
130	ST-101a	Stopper Washer & Sterilliser pre-rinse																				
131	ST-101a	Stopper Processor																				
132	ST-102a	Stopper Washer & Sterilliser pre-rinse																				
133	ST-102a	Stopper Processor																				
134	ST-103a	Stopper Washer & Sterilliser pre-rinse	15:00	15:40	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2	21:30	21:40	0:00	0:01	0:04	0:05	2
135	ST-103a	Stopper Processor	15:40	15:50	0:00	0:02	0:08	0:10	2	21:00	21:10	0:00	0:02	0:08	0:10	2	22:30	22:40	0:00	0:02	0:08	0:10	2
136	ST-104a	Stopper Washer & Sterilliser pre-rinse	16:30	16:50	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2			0:00	0:01	0:04	0:05	
137	ST-104a	Stopper Processor	16:50	17:00	0:00	0:02	0:08	0:10	2	22:30	22:40	0:00	0:02	0:08	0:10	2			0:00	0:02	0:08	0:10	
138	VF-100a	Vial Rinser - Fill																				
139	VF-101a	Vial Rinser - Fill																				
140	VF-102a	Vial Rinser - Fill																				
141	VF-103a	Vial Rinser - Fill																				
142																						
143	PW-101a	Parts washer Sanitization																				
144																						
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22:00																				
146	ST-101a	Stopper Processor Sanitation																				
147																						
148	ST-103a	Stopper Processor Sanitation																				
149	ST-104a	Stopper Processor Sanitation																				
150	VF-100a	Vial rinser Sanitisation																				
151	VF-101a	Vial rinser Sanitisation																				
152	VF-102a	Vial rinser Sanitisation																				
153	VF-103a	Vial rinser Sanitisation																				
154																						
155																						
156		The second Filling Suite requires additional Bulk																				
157		Processing, Washing & CIP from the existing																				
158		Suites as follows: 2 hours staggered																				
159																						
160		**1. BULK PROCESSING SUITE**																				
161	T-100	Pre-rinse																				
162	T-100	Fill																				
163	T-101	Pre-rinse																				
164	T-101	Fill																				
165	T-102	Pre-rinse																				
166	T-102	Fill																				
167	T-103	Pre-rinse																				
168	T-103	Fill																				
169	BT-100	Buffer Tank pre-rinse																				
170	BT-100	Buffer Tank fill																				
171	BT-101	Buffer Tank pre-rinse																				
172	BT-101	Buffer Tank fill																				
173	BT-102	Buffer Tank pre-rinse																				
174	BT-102	Buffer Tank fill																				
175	BT-103	Buffer Tank pre-rinse																				
176	BT-103	Buffer Tank fill																				
177	N/A	Line Rinsing	10:40	10:55	0:00	0:02	0:08	0:10	1	11:45	12:00	0:00	0:02	0:08	0:10	1	13:00	13:15	0:00	0:02	0:08	0:10	1
178																						
179																						
180		**2. WASHING SUITE**																				
181	PW-100	Parts Washer Pre Rinse	11:00	11:05	0:00	0:02	0:08	0:10	1	12:30	12:35	0:00	0:02	0:08	0:10	1	17:30	17:35	0:00	0:02	0:08	0:10	1
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00			0:00	0:01	0:04	0:05															
183	PW-102	Parts Washer Pre Rinse	14:00	15:30	0:00	0:02	0:08	0:10	1														
184	PW-100	Parts Washer Sanitization																				
185	PW-101	Parts Washer Sanitization																				
186	PW-102	Parts Washer Sanitization																				
187																						
188		**4. CIP SUITE**																				
189	T-100	T-100																				
190	T-101	T-101																				
191	T-102	T-102 RESCHEDULE from 6:15																				
192	T-103	T-103																				
193	VF-100	Vial Filler																				
194	VF-101	Vial Filler																				
195	VF-102	Vial Filler																				
196	VF-103	Vial Filler RESCHEDULE from 21:00																				
197	LYO-100	Lyophyliser RESCHEDULE from 7:00																				
198	LYO-101	Lyophyliser																				
199	LYO-102	Lyophyliser																				
200																						
201																						
202																						
203																						
204																						
205	VFC-103	Vial Filler CIP skid																				
206	LYC-100	Lyophyliser CIP skid																				
207	LYC-101	Lyophyliser CIP skid																				
208	LYC-102	Lyophyliser CIP skid																				
209	LYC-103	Lyophyliser CIP skid																				
210																						
211																						
212		**THIRD INCREASE IN PRODUCTION**																				
213		**BY MAXIMISING USE OF EQUIPMENT**																				
214																						
215		**1. BULK PROCESSING SUITE**																				
216	T-100	Fill																				
217	T-101	Fill																				
218	T-102	Fill																				
219	T-103	Fill																				
220	BT-100	Buffer Tank fill																				
221	BT-101	Buffer Tank fill																				
222	BT-102	Buffer Tank fill																				
223	BT-103	Buffer Tank fill																				
224																						
225		**2. FILLING SUITE**																				
226	ST-100a	Stopper Washer & Sterilliser pre-rinse																				
227	ST-100a	Stopper Processor																				
228	ST-101a	Stopper Washer & Sterilliser pre-rinse																				
229	ST-101a	Stopper Processor																				
230	ST-102a	Stopper Washer & Sterilliser pre-rinse																				
231	ST-102a	Stopper Processor																				
232	ST-103a	Stopper Washer & Sterilliser pre-rinse	15:00	15:40	0:00	0:01	0:04	0:05	2	20:00	20:10	0:00	0:01	0:04	0:05	2	21:30	21:40	0:00	0:01	0:04	0:05	2
233	ST-103a	Stopper Processor	15:40	15:50	0:00	0:02	0:08	0:10	2	21:00	21:10	0:00	0:02	0:08	0:10	2	22:30	22:40	0:00	0:02	0:08	0:10	2
234	ST-104a	Stopper Washer & Sterilliser pre-rinse	16:30	16:50	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2			0:00	0:01	0:04	0:05	
235	ST-104a	Stopper Processor	16:50	17:00	0:00	0:02	0:08	0:10	2	22:30	22:40	0:00	0:02	0:08	0:10	2			0:00	0:02	0:08	0:10	
236	VF-100a	Vial Rinser - Fill																				
237	VF-101a	Vial Rinser - Fill																				
238	VF-102a	Vial Rinser - Fill																				
239	VF-103a	Vial Rinser - Fill																				
240																						
241																						
242																						

244		**FOURTH INCREASE IN PRODUCTION**																					
245		**ADD THIRD FILLING SUITE**																					
246		**RESCHEDULE BY 2 HOURS (STAGGERED)**																					
247																							
248		**2. FILLING SUITE**																					
249	PW-100a	Parts Wash pre-rinse																					
250	PW-100a	Fill Parts Washer																					
251	PW-101a	Parts Wash pre-rinse																					
252	PW-101a	Fill Parts Washer																					
253	PW-102a	Parts Wash pre-rinse																					
254	PW-102a	Fill Parts Washer																					
255	ST-100a	Stopper Washer & Steriliser pre-rinse																					
256	ST-100a	Stopper Processor																					
257	ST-101a	Stopper Washer & Steriliser pre-rinse																					
258	ST-101a	Stopper Processor																					
259	ST-102a	Stopper Washer & Steriliser pre-rinse																					
260	ST-102a	Stopper Processor																					
261	ST-103a	Stopper Washer & Steriliser pre-rinse	17:00	17:40	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2							
262	ST-103a	Stopper Processor	17:40	17:50	0:00	0:02	0:08	0:10	2	23:00	23:10	0:00	0:02	0:08	0:10	2							
263	ST-104a	Stopper Washer & Steriliser pre-rinse	18:30	18:50	0:00	0:01	0:04	0:05	2														
264	ST-104a	Stopper Processor	18:50	19:00	0:00	0:02	0:08	0:10	2														
265	VF-100a	Vial Rinser - Fill																					
266	VF-101a	Vial Rinser - Fill																					
267	VF-102a	Vial Rinser - Fill																					
268																							
269																							
270	PW-101a	Parts washer Sanitization																					
271																							
272	ST-100a	Stopper Processor Sanitation																					
273	ST-101a	Stopper Processor Sanitation																					
274																							
275	ST-103a	Stopper Processor Sanitation																					
276	ST-104a	Stopper Processor Sanitation																					
277	VF-100a	Vial rinser Sanitisation																					
278	VF-101a	Vial rinser Sanitisation																					
279	VF-102a	Vial rinser Sanitisation																					
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																					
281																							
282																							
283																							
284																							
285																							
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																					
287		**1. BULK PROCESSING SUITE**																					
288	T-100	Pre-rinse																					
289	T-100	Fill																					
290	T-101	Pre-rinse																					
291	T-101	Fill																					
292	T-102	Pre-rinse																					
293	T-102	Fill																					
294	T-103	Pre-rinse																					
295	T-103	Fill																					
296	BT-100	Buffer Tank pre-rinse																					
297	BT-100	Buffer Tank fill																					
298	BT-101	Buffer Tank pre-rinse																					
299	BT-101	Buffer Tank fill																					
300	BT-102	Buffer Tank pre-rinse																					
301	BT-102	Buffer Tank fill																					
302	BT-103	Buffer Tank pre-rinse																					
303	BT-103	Buffer Tank fill																					
304	N/A	Line Rinsing	12:40	12:55	0:00	0:02	0:08	0:10	2	13:45	14:00	0:00	0:02	0:08	0:10	2	15:00	15:15	0:00	0:02	0:08	0:10	2
305																							
306																							
307																							
308																							
309		**2. WASHING SUITE**																					
310	PW-100	Parts Washer Pre Rinse	13:00	13:05	0:00	0:01	0:04	0:05	1	14:30	14:35	0:00	0:01	0:04	0:05	1	19:30	19:35	0:00	0:01	0:04	0:05	1
311	PW-101	Parts Washer Pre Rinse																					
312	PW-102	Parts Washer Pre Rinse	13:00	13:30	0:00	0:01	0:04	0:05	1														
313	PW-100	Parts Washer Sanitization																					
314	PW-101	Parts Washer Sanitization																					
315	PW-102	Parts Washer Sanitization																					
316																							
317		**4. CIP SUITE**																					
318	T-100	T-100																					
319	T-101	T-101																					
320	T-102	T-102																					
321	T-103	T-103																					
322	VF-100	Vial Filler																					
323	VF-101	Vial Filler																					
324	VF-102	Vial Filler RESCHEDULE from 21:00																					
325	VF-103	Vial Filler																					
326	LYO-100	Lyophyliser																					
327	LYO-101	Lyophyliser																					
328	LYO-102	Lyophyliser																					
329	LYO-103	Lyophyliser																					
330	LYO-104	Lyophyliser																					
331	VFC-100	Vial Filler CIP skid																					
332	VFC-101	Vial Filler CIP skid																					
333	VFC-102	Vial Filler CIP skid																					
334	VFC-103	Vial Filler CIP skid																					
335	LYC-100	Lyophyliser CIP skid																					
336	LYC-101	Lyophyliser CIP skid																					
337	LYC-102	Lyophyliser CIP skid																					
338	LYC-103	Lyophyliser CIP skid																					
339																							
340																							

342		**FIFTH INCREASE IN PRODUCTION**																			
343		**BY MAXIMISING USE OF EQUIPMENT**																			
344																					
345		**1. BULK PROCESSING SUITE**																			
346	T-100	Fill																			
347	T-101	Fill																			
348	T-102	Fill																			
349	T-103	Fill																			
350	BT-100	Buffer Tank fill																			
351	BT-101	Buffer Tank fill																			
352	BT-102	Buffer Tank fill																			
353	BT-103	Buffer Tank fill																			
354																					
355		**2. FILLING SUITE**																			
356	ST-100a	Stopper Washer & Steriliser pre-rinse																			
357	ST-100a	Stopper Processor																			
358	ST-101a	Stopper Washer & Steriliser pre-rinse																			
359	ST-101a	Stopper Processor																			
360	ST-102a	Stopper Washer & Steriliser pre-rinse																			
361	ST-102a	Stopper Processor																			
362	ST-103a	Stopper Washer & Steriliser pre-rinse	17:00	17:40	0:00	0:01	0:04	0:05	2	22:00	22:10	0:00	0:01	0:04	0:05	2					
363	ST-103a	Stopper Processor	17:40	17:50	0:00	0:02	0:08	0:10	2	23:00	23:10	0:00	0:02	0:08	0:10	2					
364	ST-104a	Stopper Washer & Steriliser pre-rinse	18:30	18:50	0:00	0:01	0:04	0:05	2												
365	ST-104a	Stopper Processor	18:50	19:00	0:00	0:01	0:04	0:05	2												
366	VF-100a	Vial Rinser - Fill																			
367	VF-101a	Vial Rinser - Fill																			
368	VF-102a	Vial Rinser - Fill																			
369																					
370																					
371		**SIXTH INCREASE IN PRODUCTION**																			
372		**ADD FOURTH FILLING SUITE**																			
373		**RESCHEDULE BY 2 HOURS (STAGGERED)**																			
374																					
375		**2. FILLING SUITE**																			
376	PW-100a	Parts Wash pre-rinse																			
377	PW-100a	Fill Parts Washer																			
378	PW-101a	Parts Wash pre-rinse																			
379	PW-101a	Fill Parts Washer																			
380	PW-102a	Parts Wash pre-rinse																			
381	PW-102a	Fill Parts Washer																			
382	ST-100a	Stopper Washer & Steriliser pre-rinse																			
383	ST-100a	Stopper Processor																			
384	ST-101a	Stopper Washer & Steriliser pre-rinse																			
385	ST-101a	Stopper Processor																			
386	ST-102a	Stopper Washer & Steriliser pre-rinse																			
387	ST-102a	Stopper Processor																			
388	ST-103a	Stopper Washer & Steriliser pre-rinse	19:00	19:40	0:00	0:02	0:08	0:10	2												
389	ST-103a	Stopper Processor	19:40	19:50	0:00	0:01	0:04	0:05	2												
390	ST-104a	Stopper Washer & Steriliser pre-rinse	20:30	20:50	0:00	0:02	0:08	0:10	2												
391	ST-104a	Stopper Processor	20:50	21:00	0:00	0:01	0:04	0:05	2												
392	VF-100a	Vial Rinser - Fill																			
393	VF-101a	Vial Rinser - Fill																			
394	VF-102a	Vial Rinser - Fill																			
395																					
396																					
397	PW-101a	Parts washer Sanitization																			
398																					
399	ST-100a	Stopper Processor Sanitation																			
400	ST-101a	Stopper Processor Sanitation																			
401																					
402	ST-103a	Stopper Processor Sanitation																			
403	ST-104a	Stopper Processor Sanitation																			
404	VF-100a	Vial rinser Sanitisation																			
405	VF-101a	Vial rinser Sanitisation																			
406	VF-102a	Vial rinser Sanitisation																			
407	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																			
408																					
409																					
410																					
411		**SEVENTH INCREASE IN PRODUCTION**																			
412		**BY MAXIMISING USE OF EQUIPMENT**																			
413																					
414		**1. BULK PROCESSING SUITE**																			
415	T-100	Fill																			
416	T-101	Fill																			
417	T-102	Fill																			
418	T-103	Fill																			
419	BT-100	Buffer Tank fill																			
420	BT-101	Buffer Tank fill																			
421	BT-102	Buffer Tank fill																			
422	BT-103	Buffer Tank fill																			
423																					
424		**2. FILLING SUITE**																			
425	ST-100a	Stopper Washer & Steriliser pre-rinse																			
426	ST-100a	Stopper Processor																			
427	ST-101a	Stopper Washer & Steriliser pre-rinse																			
428	ST-101a	Stopper Processor																			
429	ST-102a	Stopper Washer & Steriliser pre-rinse																			
430	ST-102a	Stopper Processor																			
431	ST-103a	Stopper Washer & Steriliser pre-rinse	19:00	19:40	0:00	0:01	0:04	0:05	2												
432	ST-103a	Stopper Processor	19:40	19:50	0:00	0:02	0:08	0:10	2												
433	ST-104a	Stopper Washer & Steriliser pre-rinse	20:30	20:50	0:00	0:01	0:04	0:05	2												
434	ST-104a	Stopper Processor	20:50	21:00	0:00	0:01	0:04	0:05	2												
435	VF-100a	Vial Rinser - Fill																			
436	VF-101a	Vial Rinser - Fill																			
437	VF-102a	Vial Rinser - Fill																			

WFI Water Offtake Input Data																								
					7								8							9				
No.	Tag No.	Description of Offtake	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority
1		**ORIGINAL PROCESS DATA**																						
2		**1. BULK PROCESSING SUITE**																						
3	T-100	Pre-rinse																						
4	T-100	Fill																						
5	T-101	Pre-rinse																						
6	T-101	Fill																						
7	T-102	Pre-rinse																						
8	T-102	Fill																						
9	T-103	Pre-rinse																						
10	T-103	Fill																						
11	BT-100	Buffer Tank pre-rinse																						
12	BT-100	Buffer Tank fill																						
13	BT-101	Buffer Tank pre-rinse																						
14	BT-101	Buffer Tank fill																						
15	BT-102	Buffer Tank pre-rinse																						
16	BT-102	Buffer Tank fill																						
17	BT-103	Buffer Tank pre-rinse																						
18	BT-103	Buffer Tank fill																						
19	N/A	Line Rinsing	15:00	15:15	0:00	0:02	0:08	0:10	2	16:15	16:30	0:00	0:02	0:08	0:10	2	17:20	17:35	0:00	0:02	0:08	0:10	2	
20																								
21		**2. FILLING SUITE**																						
22	PW-100	Parts Wash pre-rinse																						
23	PW-100	Fill Parts Washer																						
24	PW-101	Parts Wash pre-rinse																						
25	PW-101	Fill Parts Washer																						
26	PW-102	Parts Wash pre-rinse																						
27	PW-102	Fill Parts Washer																						
28	ST-100	Stopper Washer & Steriliser pre-rinse																						
29	ST-100	Stopper Processor																						
30	ST-101	Stopper Washer & Steriliser pre-rinse																						
31	ST-101	Stopper Processor																						
32	ST-102	Stopper Washer & Steriliser pre-rinse																						
33	ST-102	Stopper Processor																						
34	ST-103	Stopper Washer & Steriliser pre-rinse																						
35	ST-103	Stopper Processor																						
36	ST-104	Stopper Washer & Steriliser pre-rinse	23:00	23:10	0:00	0:01	0:04	0:05	2															
37	ST-104	Stopper Processor	23:20	23:30	0:00	0:02	0:08	0:10	2															
38	VF-100	Vial Rinser - Fill																						
39	VF-101	Vial Rinser - Fill																						
40	VF-102	Vial Rinser - Fill																						
41	VF-103	Vial Rinser - Fill																						
42	PW-100	Parts washer Sanitization																						
43	PW-101	Parts washer Sanitization																						
44	PW-102	Parts washer Sanitization																						
45	ST-100	Stopper Processor Sanitation																						
46	ST-101	Stopper Processor Sanitation																						
47	ST-102	Stopper Processor Sanitation																						
48	ST-103	Stopper Processor Sanitation																						
49	ST-104	Stopper Processor Sanitation																						
50	VF-100	Vial rinser Sanitisation																						
51	VF-101	Vial rinser Sanitisation																						
52	VF-102	Vial rinser Sanitisation																						
53	VF-103	Vial rinser Sanitisation																						
54																								
55		**2. WASHING SUITE**																						
56	PW-100	Parts Washer Pre Rinse	16:45	16:50	0:00	0:01	0:04	0:05	1	17:50	17:55	0:00	0:01	0:04	0:05	1	19:00	19:05	0:00	0:01	0:04	0:05	1	
57	PW-101	Parts Washer Pre Rinse																						
58	PW-102	Parts Washer Pre Rinse																						
59	PW-100	Parts Washer Sanitization																						
60	PW-101	Parts Washer Sanitization																						
61	PW-102	Parts Washer Sanitization																						
62																								
63		**4. CIP SUITE**																						
64	T-100	T-100																						
65	T-101	T-101																						
66	T-102	T-102																						
67	T-103	T-103 RESCHEDULE from 6:00																						
68	VF-100	Vial Filler																						
69	VF-101	Vial Filler																						
70	VF-102	Vial Filler																						
71	VF-103	Vial Filler																						
72	LYO-100	Lyophyliser																						
73	LYO-101	Lyophyliser																						
74	LYO-102	Lyophyliser																						
75	LYO-103	Lyophyliser																						
76	LYO-104	Lyophyliser																						
77	VFC-100	Vial Filler CIP skid																						
78	VFC-101	Vial Filler CIP skid																						
79	VFC-102	Vial Filler CIP skid																						
80	VFC-103	Vial Filler CIP skid																						
81	LYC-100	Lyophyliser CIP skid																						
82	LYC-101	Lyophyliser CIP skid																						
83	LYC-102	Lyophyliser CIP skid																						
84	LYC-103	Lyophyliser CIP skid																						
85																								
86																								
87		**FIRST INCREASE IN PRODUCTION**																						
88		**BY MAXIMISING USE OF EQUIPMENT**																						
89																								
90		**1. BULK PROCESSING SUITE**																						
91	T-100	Fill																						
92	T-101	Fill																						
93	T-102	Fill																						
94	T-103	Fill																						
95	BT-100	Buffer Tank fill																						
96	BT-101	Buffer Tank fill																						
97	BT-102	Buffer Tank fill																						
98	BT-103	Buffer Tank fill																						
99																								
100		**2. FILLING SUITE**																						
101	ST-100	Stopper Washer & Steriliser pre-rinse																						
102	ST-100	Stopper Processor																						
103	ST-101	Stopper Washer & Steriliser pre-rinse																						
104	ST-101	Stopper Processor																						
105	ST-102	Stopper Washer & Steriliser pre-rinse																						
106	ST-102	Stopper Processor																						
107	ST-103	Stopper Washer & Steriliser pre-rinse																						
108	ST-103	Stopper Processor																						
109	ST-104	Stopper Washer & Steriliser pre-rinse	22:50	23:00	0:00	0:02	0:08	0:10	2															
110	ST-104	Stopper Processor	23:15	23:45	0:00	0:01	0:04	0:05	2															
111	VF-100	Vial Rinser - Fill																						
112	VF-101	Vial Rinser - Fill																						
113	VF-102	Vial Rinser - Fill																						
114	VF-103	Vial Rinser - Fill																						
115																								

117		**SECOND INCREASE IN PRODUCTION**																					
118		**ADD ANOTHER FILLING SUITE**																					
119		**RESCHEDULE BY 2 HOURS (STAGGERED)**																					
120																							
121		**2. FILLING SUITE**																					
122	PW-100a	Parts Wash pre-rinse																					
123	PW-100a	Fill Parts Washer																					
124	PW-101a	Parts Wash pre-rinse																					
125	PW-101a	Fill Parts Washer																					
126	PW-102a	Parts Wash pre-rinse																					
127	PW-102a	Fill Parts Washer																					
128	ST-100a	Stopper Washer & Steriliser pre-rinse																					
129	ST-100a	Stopper Processor																					
130	ST-101a	Stopper Washer & Steriliser pre-rinse																					
131	ST-101a	Stopper Processor																					
132	ST-102a	Stopper Washer & Steriliser pre-rinse																					
133	ST-102a	Stopper Processor																					
134	ST-103a	Stopper Washer & Steriliser pre-rinse																					
135	ST-103a	Stopper Processor																					
136	ST-104a	Stopper Washer & Steriliser pre-rinse																					
137	ST-104a	Stopper Processor																					
138	VF-100a	Vial Rinser - Fill																					
139	VF-101a	Vial Rinser - Fill																					
140	VF-102a	Vial Rinser - Fill																					
141	VF-103a	Vial Rinser - Fill																					
142																							
143	PW-101a	Parts washer Sanitization																					
144																							
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22:00																					
146	ST-101a	Stopper Processor Sanitation																					
147																							
148	ST-103a	Stopper Processor Sanitation																					
149	ST-104a	Stopper Processor Sanitation																					
150	VF-100a	Vial rinser Sanitisation																					
151	VF-101a	Vial rinser Sanitisation																					
152	VF-102a	Vial rinser Sanitisation																					
153	VF-103a	Vial rinser Sanitisation																					
154																							
155																							
156		The second Filling Suite requires additional Bulk																					
157		Processing, Washing & CIP from the existing																					
158		Suites as follows: 2 hours staggered																					
159																							
160		**1. BULK PROCESSING SUITE**																					
161	T-100	Pre-rinse																					
162	T-100	Fill																					
163	T-101	Pre-rinse																					
164	T-101	Fill																					
165	T-102	Pre-rinse																					
166	T-102	Fill																					
167	T-103	Pre-rinse																					
168	T-103	Fill																					
169	BT-100	Buffer Tank pre-rinse																					
170	BT-100	Buffer Tank fill																					
171	BT-101	Buffer Tank pre-rinse																					
172	BT-101	Buffer Tank fill																					
173	BT-102	Buffer Tank pre-rinse																					
174	BT-102	Buffer Tank fill																					
175	BT-103	Buffer Tank pre-rinse																					
176	BT-103	Buffer Tank fill																					
177	N/A	Line Rinsing	17:00	17:15	0:00	0:02	0:08	0:10	1	18:15	18:30	0:00	0:02	0:08	0:10	1	19:20	19:35	0:00	0:02	0:08	0:10	1
178																							
179																							
180		**2. WASHING SUITE**																					
181	PW-100	Parts Washer Pre Rinse	18:45	18:50	0:00	0:02	0:08	0:10	1	19:50	19:55	0:00	0:02	0:08	0:10	1	21:00	21:05	0:00	0:02	0:08	0:10	1
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00																					
183	PW-102	Parts Washer Pre Rinse																					
184	PW-100	Parts Washer Sanitization																					
185	PW-101	Parts Washer Sanitization																					
186	PW-102	Parts Washer Sanitization																					
187																							
188		**4. CIP SUITE**																					
189	T-100	T-100																					
190	T-101	T-101																					
191	T-102	T-102 RESCHEDULE from 6:15																					
192	T-103	T-103																					
193	VF-100	Vial Filler																					
194	VF-101	Vial Filler																					
195	VF-102	Vial Filler																					
196	VF-103	Vial Filler RESCHEDULE from 21:00																					
197	LYO-100	Lyophyliser RESCHEDULE from 7:00																					
198	LYO-101	Lyophyliser																					
199	LYO-102	Lyophyliser																					
200																							
201																							
202																							
203																							
204																							
205	VFC-103	Vial Filler CIP skid																					
206	LYC-100	Lyophyliser CIP skid																					
207	LYC-101	Lyophyliser CIP skid																					
208	LYC-102	Lyophyliser CIP skid																					
209	LYC-103	Lyophyliser CIP skid																					
210																							
211																							
212		**THIRD INCREASE IN PRODUCTION**																					
213		**BY MAXIMISING USE OF EQUIPMENT**																					
214																							
215		**1. BULK PROCESSING SUITE**																					
216	T-100	Fill																					
217	T-101	Fill																					
218	T-102	Fill																					
219	T-103	Fill																					

244		**FOURTH INCREASE IN PRODUCTION**																				
245		**ADD THIRD FILLING SUITE**																				
246		**RESCHEDULE BY 2 HOURS (STAGGERED)**																				
247																						
248		**2. FILLING SUITE**																				
249	PW-100a	Parts Wash pre-rinse																				
250	PW-100a	Fill Parts Washer																				
251	PW-101a	Parts Wash pre-rinse																				
252	PW-101a	Fill Parts Washer																				
253	PW-102a	Parts Wash pre-rinse																				
254	PW-102a	Fill Parts Washer																				
255	ST-100a	Stopper Washer & Steriliser pre-rinse																				
256	ST-100a	Stopper Processor																				
257	ST-101a	Stopper Washer & Steriliser pre-rinse																				
258	ST-101a	Stopper Processor																				
259	ST-102a	Stopper Washer & Steriliser pre-rinse																				
260	ST-102a	Stopper Processor																				
261	ST-103a	Stopper Washer & Steriliser pre-rinse																				
262	ST-103a	Stopper Processor																				
263	ST-104a	Stopper Washer & Steriliser pre-rinse																				
264	ST-104a	Stopper Processor																				
265	VF-100a	Vial Rinser - Fill																				
266	VF-101a	Vial Rinser - Fill																				
267	VF-102a	Vial Rinser - Fill																				
268																						
269																						
270	PW-101a	Parts washer Sanitization																				
271																						
272	ST-100a	Stopper Processor Sanitation																				
273	ST-101a	Stopper Processor Sanitation																				
274																						
275	ST-103a	Stopper Processor Sanitation																				
276	ST-104a	Stopper Processor Sanitation																				
277	VF-100a	Vial rinser Sanitisation																				
278	VF-101a	Vial rinser Sanitisation																				
279	VF-102a	Vial rinser Sanitisation																				
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																				
281																						
282																						
283																						
284																						
285																						
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																				
287		**1. BULK PROCESSING SUITE**																				
288	T-100	Pre-rinse																				
289	T-100	Fill																				
290	T-101	Pre-rinse																				
291	T-101	Fill																				
292	T-102	Pre-rinse																				
293	T-102	Fill																				
294	T-103	Pre-rinse																				
295	T-103	Fill																				
296	BT-100	Buffer Tank pre-rinse																				
297	BT-100	Buffer Tank fill																				
298	BT-101	Buffer Tank pre-rinse																				
299	BT-101	Buffer Tank fill																				
300	BT-102	Buffer Tank pre-rinse																				
301	BT-102	Buffer Tank fill																				
302	BT-103	Buffer Tank pre-rinse																				
303	BT-103	Buffer Tank fill																				
304	N/A	Line Rinsing	19:00	19:15	0:00	0:02	0:08	0:10	2	20:15	20:30	0:00	0:02	0:08	0:10	2	21:20	21:35	0:00	0:02	0:08	0:10	2
305																						
306																						
307																						
308																						
309		**2. WASHING SUITE**																				
310	PW-100	Parts Washer Pre Rinse	20:45	20:50	0:00	0:01	0:04	0:05	1	21:50	21:55	0:00	0:01	0:04	0:05	1	23:00	23:05	0:00	0:01	0:04	0:05	1
311	PW-101	Parts Washer Pre Rinse																				
312	PW-102	Parts Washer Pre Rinse																				
313	PW-100	Parts Washer Sanitization																				
314	PW-101	Parts Washer Sanitization																				
315	PW-102	Parts Washer Sanitization																				
316																						
317		**4. CIP SUITE**																				
318	T-100	T-100																				
319	T-101	T-101																				
320	T-102	T-102																				

	WFI Water Offtake Input Data				WFI Water Offtake Input Data														WFI Water Offtake Input Data									
					10									11								12						
No.	Tag No.	Description of Offtake	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority
1		ORIGINAL PROCESS DATA																										
2		1. BULK PROCESSING SUITE																										
3	T-100	Pre-rinse																										
4	T-100	Fill																										
5	T-101	Pre-rinse																										
6	T-101	Fill																										
7	T-102	Pre-rinse																										
8	T-102	Fill																										
9	T-103	Pre-rinse																										
10	T-103	Fill																										
11	BT-100	Buffer Tank pre-rinse																										
12	BT-100	Buffer Tank fill																										
13	BT-101	Buffer Tank pre-rinse																										
14	BT-101	Buffer Tank fill																										
15	BT-102	Buffer Tank pre-rinse																										
16	BT-102	Buffer Tank fill																										
17	BT-103	Buffer Tank pre-rinse																										
18	BT-103	Buffer Tank fill																										
19	N/A	Line Rinsing	18:30	18:45	0:00	0:02	0:08	0:10	2	19:40	19:55	0:00	0:02	0:08	0:10	2	22:00	22:15	0:00	0:02	0:08	0:10	2					
20																												
21		2. FILLING SUITE																										
22	PW-100	Parts Wash pre-rinse																										
23	PW-100	Fill Parts Washer																										
24	PW-101	Parts Wash pre-rinse																										
25	PW-101	Fill Parts Washer																										
26	PW-102	Parts Wash pre-rinse																										
27	PW-102	Fill Parts Washer																										
28	ST-100	Stopper Washer & Steriliser pre-rinse																										
29	ST-100	Stopper Processor																										
30	ST-101	Stopper Washer & Steriliser pre-rinse																										
31	ST-101	Stopper Processor																										
32	ST-102	Stopper Washer & Steriliser pre-rinse																										
33	ST-102	Stopper Processor																										
34	ST-103	Stopper Washer & Steriliser pre-rinse																										
35	ST-103	Stopper Processor																										
36	ST-104	Stopper Washer & Steriliser pre-rinse																										
37	ST-104	Stopper Processor																										
38	VF-100	Vial Rinser - Fill																										
39	VF-101	Vial Rinser - Fill																										
40	VF-102	Vial Rinser - Fill																										
41	VF-103	Vial Rinser - Fill																										
42	PW-100	Parts washer Sanitization																										
43	PW-101	Parts washer Sanitization																										
44	PW-102	Parts washer Sanitization																										
45	ST-100	Stopper Processor Sanitation																										
46	ST-101	Stopper Processor Sanitation																										
47	ST-102	Stopper Processor Sanitation																										
48	ST-103	Stopper Processor Sanitation																										
49	ST-104	Stopper Processor Sanitation																										
50	VF-100	Vial rinser Sanitisation																										
51	VF-101	Vial rinser Sanitisation																										
52	VF-102	Vial rinser Sanitisation																										
53	VF-103	Vial rinser Sanitisation																										
54																												
55		2. WASHING SUITE																										
56	PW-100	Parts Washer Pre Rinse			0:00	0:01	0:04	0:05				0:00	0:01	0:04	0:05				0:00	0:01	0:04	0:05						
57	PW-101	Parts Washer Pre Rinse																										
58	PW-102	Parts Washer Pre Rinse																	0:00	0:01	0:04	0:05						
59	PW-100	Parts Washer Sanitization																										
60	PW-101	Parts Washer Sanitization																										
61	PW-102	Parts Washer Sanitization																										
62																												
63		4. CIP SUITE																										
64	T-100	T-100																										
65	T-101	T-101																										
66	T-102	T-102																										
67	T-103	T-103 RESCHEDULE from 6:00																										
68	VF-100	Vial Filler																										
69	VF-101	Vial Filler																										
70	VF-102	Vial Filler																										
71	VF-103	Vial Filler																										
72	LYO-100	Lyophyliser																										
73	LYO-101	Lyophyliser																										
74	LYO-102	Lyophyliser																										
75	LYO-103	Lyophyliser																										
76	LYO-104	Lyophyliser																										
77	VFC-100	Vial Filler CIP skid																										
78	VFC-101	Vial Filler CIP skid																										
79	VFC-102	Vial Filler CIP skid																										
80	VFC-103	Vial Filler CIP skid																										
81	LYC-100	Lyophyliser CIP skid																										
82	LYC-101	Lyophyliser CIP skid																										
83	LYC-102	Lyophyliser CIP skid																										
84	LYC-103	Lyophyliser CIP skid																										
85																												
86																												
87		FIRST INCREASE IN PRODUCTION																										
88		BY MAXIMISING USE OF EQUIPMENT																										
89																												
90		1. BULK PROCESSING SUITE																										
91	T-100	Fill																										
92	T-101	Fill																										
93	T-102	Fill																										
94	T-103	Fill																										
95	BT-100	Buffer Tank fill																										

117		**SECOND INCREASE IN PRODUCTION**																			
118		**ADD ANOTHER FILLING SUITE**																			
119		**RESCHEDULE BY 2 HOURS (STAGGERED)**																			
120																					
121		**2. FILLING SUITE**																			
122	PW-100a	Parts Wash pre-rinse																			
123	PW-100a	Fill Parts Washer																			
124	PW-101a	Parts Wash pre-rinse																			
125	PW-101a	Fill Parts Washer																			
126	PW-102a	Parts Wash pre-rinse																			
127	PW-102a	Fill Parts Washer																			
128	ST-100a	Stopper Washer & Steriliser pre-rinse																			
129	ST-100a	Stopper Processor																			
130	ST-101a	Stopper Washer & Steriliser pre-rinse																			
131	ST-101a	Stopper Processor																			
132	ST-102a	Stopper Washer & Steriliser pre-rinse																			
133	ST-102a	Stopper Processor																			
134	ST-103a	Stopper Washer & Steriliser pre-rinse																			
135	ST-103a	Stopper Processor																			
136	ST-104a	Stopper Washer & Steriliser pre-rinse																			
137	ST-104a	Stopper Processor																			
138	VF-100a	Vial Rinser - Fill																			
139	VF-101a	Vial Rinser - Fill																			
140	VF-102a	Vial Rinser - Fill																			
141	VF-103a	Vial Rinser - Fill																			
142																					
143	PW-101a	Parts washer Sanitization																			
144																					
145	ST-100a	Stopper Processor Sanitation RESCHEDULE from 22:00																			
146	ST-101a	Stopper Processor Sanitation																			
147																					
148	ST-103a	Stopper Processor Sanitation																			
149	ST-104a	Stopper Processor Sanitation																			
150	VF-100a	Vial rinser Sanitisation																			
151	VF-101a	Vial rinser Sanitisation																			
152	VF-102a	Vial rinser Sanitisation																			
153	VF-103a	Vial rinser Sanitisation																			
154																					
155																					
156		The second Filling Suite requires additional Bulk																			
157		Processing, Washing & CIP from the existing																			
158		Suites as follows: 2 hours staggered																			
159																					
160		**1. BULK PROCESSING SUITE**																			
161	T-100	Pre-rinse																			
162	T-100	Fill																			
163	T-101	Pre-rinse																			
164	T-101	Fill																			
165	T-102	Pre-rinse																			
166	T-102	Fill																			
167	T-103	Pre-rinse																			
168	T-103	Fill																			
169	BT-100	Buffer Tank pre-rinse																			
170	BT-100	Buffer Tank fill																			
171	BT-101	Buffer Tank pre-rinse																			
172	BT-101	Buffer Tank fill																			
173	BT-102	Buffer Tank pre-rinse																			
174	BT-102	Buffer Tank fill																			
175	BT-103	Buffer Tank pre-rinse																			
176	BT-103	Buffer Tank fill																			
177	N/A	Line Rinsing	20:30	20:45	0:00	0:02	0:08	0:10	1	21:40	21:55	0:00	0:02	0:08	0:10	1					
178																					
179																					
180		**2. WASHING SUITE**																			
181	PW-100	Parts Washer Pre Rinse																			
182	PW-101	Parts Washer Pre Rinse RESCHEDULE from 6:00																			
183	PW-102	Parts Washer Pre Rinse																			
184	PW-100	Parts Washer Sanitization																			
185	PW-101	Parts Washer Sanitization																			
186	PW-102	Parts Washer Sanitization																			
187																					
188		**4. CIP SUITE**																			
189	T-100	T-100																			
190	T-101	T-101																			
191	T-102	T-102 RESCHEDULE from 6:15																			
192	T-103	T-103																			
193	VF-100	Vial Filler																			
194	VF-101	Vial Filler																			
195	VF-102	Vial Filler																			
196	VF-103	Vial Filler RESCHEDULE from 21:00																			
197	LYO-100	Lyophyliser RESCHEDULE from 7:00																			
198	LYO-101	Lyophyliser																			
199	LYO-102	Lyophyliser																			
200																					
201																					
202																					
203																					
204																					
205	VFC-103	Vial Filler CIP skid																			
206	LYC-100	Lyophyliser CIP skid																			
207	LYC-101	Lyophyliser CIP skid																			
208	LYC-102	Lyophyliser CIP skid																			
209	LYC-103	Lyophyliser CIP skid																			
210																					
211																					
212		**THIRD INCREASE IN PRODUCTION**																			
213		**BY MAXIMISING USE OF EQUIPMENT**																			
214																					
215		**1. BULK PROCESSING SUITE**																			

244		**FOURTH INCREASE IN PRODUCTION**																		
245		**ADD THIRD FILLING SUITE**																		
246		**RESCHEDULE BY 2 HOURS (STAGGERED)**																		
247																				
248		**2. FILLING SUITE**																		
249	PW-100a	Parts Wash pre-rinse																		
250	PW-100a	Fill Parts Washer																		
251	PW-101a	Parts Wash pre-rinse																		
252	PW-101a	Fill Parts Washer																		
253	PW-102a	Parts Wash pre-rinse																		
254	PW-102a	Fill Parts Washer																		
255	ST-100a	Stopper Washer & Steriliser pre-rinse																		
256	ST-100a	Stopper Processor																		
257	ST-101a	Stopper Washer & Steriliser pre-rinse																		
258	ST-101a	Stopper Processor																		
259	ST-102a	Stopper Washer & Steriliser pre-rinse																		
260	ST-102a	Stopper Processor																		
261	ST-103a	Stopper Washer & Steriliser pre-rinse																		
262	ST-103a	Stopper Processor																		
263	ST-104a	Stopper Washer & Steriliser pre-rinse																		
264	ST-104a	Stopper Processor																		
265	VF-100a	Vial Rinser - Fill																		
266	VF-101a	Vial Rinser - Fill																		
267	VF-102a	Vial Rinser - Fill																		
268																				
269																				
270	PW-101a	Parts washer Sanitization																		
271																				
272	ST-100a	Stopper Processor Sanitation																		
273	ST-101a	Stopper Processor Sanitation																		
274																				
275	ST-103a	Stopper Processor Sanitation																		
276	ST-104a	Stopper Processor Sanitation																		
277	VF-100a	Vial rinser Sanitisation																		
278	VF-101a	Vial rinser Sanitisation																		
279	VF-102a	Vial rinser Sanitisation																		
280	VF-103a	Vial rinser Sanitisation RESCHEDULE from 7:00																		
281																				
282																				
283																				
284																				
285																				
286		**EXTRA BULK WASHING AND CIP PROCESSING (4 hours staggered to ORIGINAL ONE)**																		
287		**1. BULK PROCESSING SUITE**																		
288	T-100	Pre-rinse																		
289	T-100	Fill																		
290	T-101	Pre-rinse																		
291	T-101	Fill																		
292	T-102	Pre-rinse																		
293	T-102	Fill																		
294	T-103	Pre-rinse																		
295	T-103	Fill																		
296	BT-100	Buffer Tank pre-rinse																		
297	BT-100	Buffer Tank fill																		
298	BT-101	Buffer Tank pre-rinse																		
299	BT-101	Buffer Tank fill																		
300	BT-102	Buffer Tank pre-rinse																		
301	BT-102	Buffer Tank fill																		
302	BT-103	Buffer Tank pre-rinse																		
303	BT-103	Buffer Tank fill																		
304	N/A	Line Rinsing	22:30	22:45	0:00	0:02	0:08	0:10	2											
305																				
306																				
307																				
308																				
309		**2. WASHING SUITE**																		
310	PW-100	Parts Washer Pre Rinse																		
311	PW-101	Parts Washer Pre Rinse																		
312	PW-102	Parts Washer Pre Rinse																		
313	PW-100	Parts Washer Sanitization																		
314	PW-101	Parts Washer Sanitization																		
315	PW-102	Parts Washer Sanitization																		
316																				
317		**4. CIP SUITE**																		
318	T-100	T-100																		
319	T-101	T-101																		
320	T-102	T-102																		
321	T-103	T-103																		
322	VF-100	Vial Filler																		
323	VF-101	Vial Filler																		
324	VF-102	Vial Filler RESCHEDULE from 21:00																		
325	VF-103	Vial Filler																		
326	LYO-100	Lyophyliser																		
327	LYO-101	Lyophyliser																		
328	LYO-102	Lyophyliser																		
329	LYO-103	Lyophyliser																		
330	LYO-104	Lyophyliser																		
331	VFC-100	Vial Filler CIP skid																		
332	VFC-101	Vial Filler CIP skid																		
333	VFC-102	Vial Filler CIP skid																		
334	VFC-103	Vial Filler CIP skid																		
335	LYC-100	Lyophyliser CIP skid																		
336	LYC-101	Lyophyliser CIP skid																		
337	LYC-102	Lyophyliser CIP skid																		
338	LYC-103	Lyophyliser CIP skid																		
339																				

Extract of Dataset Fuzzy Experiment with "10 Minutes Schedule Uncertainty". The dataset for the "30 Minutes Schedule Uncertainty", "5 Minutes Schedule Uncertainty" and others are similar to the below.

WFI Water Offtake Input Data Fuzzy Fourth Increase 3rd Filling Line												WFI Water Offtake Input Data 2						
No.	Tag No.	Description of Offtake	Water Offtake [m^3/hr]	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time				Priority	Start Time (Tap Open) [24 hour clock, hours:min]	End Time (Tap Closed) [24 hour clock, hours:min]	Fuzzy Trapeziodal Numbers - Operator Setup Time			
1		ORIGINAL PROCESS DATA															
2		1. BULK PROCESSING SUITE															
3	T-100	Pre-rinse	1.5	2:00	2:05	0:00	0:00	0:00	0:00	2	10:00	10:35	0:00	0:00	0:00	0:00
4	T-100	Fill	1	2:00	2:06	0:00	0:02	0:08	0:10	3	10:30	10:36	0:00	0:02	0:08	0:10
5	T-101	Pre-rinse	1.5	3:10	3:17	0:00	0:00	0:00	0:00	2	11:25	11:45	0:00	0:00	0:00	0:00
6	T-101	Fill	1	3:15	3:25	0:00	0:02	0:08	0:10	3	11:15	11:25	0:00	0:02	0:08	0:10
7	T-102	Pre-rinse	1.5	1:20	1:30	0:00	0:00	0:00	0:00	2	9:30	9:40	0:00	0:00	0:00	0:00
8	T-102	Fill	1	3:15	3:25	0:00	0:02	0:08	0:10	3	10:15	10:20	0:00	0:02	0:08	0:10
9	T-103	Pre-rinse	1.5	1:00	1:10	0:00	0:00	0:00	0:00	2	9:00	9:10	0:00	0:00	0:00	0:00
10	T-103	Fill	1	6:00	6:30	0:00	0:02	0:08	0:10	3						
11	BT-100	Buffer Tank pre-rinse	1.5	1:15	1:20	0:00	0:00	0:00	0:00	2	9:00	9:05	0:00	0:00	0:00	0:00
12	BT-100	Buffer Tank fill	1.5	2:00	2:10	0:00	0:02	0:08	0:10	2	10:00	10:10	0:00	0:02	0:08	0:10
13	BT-101	Buffer Tank pre-rinse	1.5	2:30	3:00	0:00	0:00	0:00	0:00	2	10:30	11:00	0:00	0:00	0:00	0:00
14	BT-101	Buffer Tank fill	1.5	3:00	3:30	0:00	0:02	0:08	0:10	2	11:00	11:30	0:00	0:02	0:08	0:10
15	BT-102	Buffer Tank pre-rinse	1.5	21:00	21:05	0:00	0:00	0:00	0:00	2						
16	BT-102	Buffer Tank fill	1.5	21:15	21:20	0:00	0:02	0:08	0:10	2						
17	BT-103	Buffer Tank pre-rinse	1.5	2:00	2:05	0:00	0:00	0:00	0:00	2						
18	BT-103	Buffer Tank fill	1.5	0:30	0:45	0:00	0:02	0:08	0:10	2						
19	N/A	Line Rinsing	1	0:30	0:45	0:00	0:00	0:00	0:00	2	6:00	6:15	0:00	0:00	0:00	0:00
20																
21		2. FILLING SUITE														
22	PW-100	Parts Wash pre-rinse	2.5	0:30	0:40	0:00	0:02	0:08	0:10	2	11:00	11:10	0:00	0:02	0:08	0:10
23	PW-100	Fill Parts Washer	5	1:30	2:10	0:00	0:00	0:00	0:00	2	12:30	12:50	0:00	0:00	0:00	0:00
24	PW-101	Parts Wash pre-rinse	2.5	3:15	3:30	0:00	0:02	0:08	0:10	2	12:45	12:55	0:00	0:02	0:08	0:10
25	PW-101	Fill Parts Washer	2.5	5:30	6:40	0:00	0:00	0:00	0:00	2	13:00	13:10	0:00	0:00	0:00	0:00
26	PW-102	Parts Wash pre-rinse	2.5	7:30	7:40	0:00	0:02	0:08	0:10	2	11:15	11:25	0:00	0:02	0:08	0:10
27	PW-102	Fill Parts Washer	2.5	8:00	8:10	0:00	0:00	0:00	0:00	2	13:00	13:10	0:00	0:00	0:00	0:00
28	ST-100	Stopper Washer & Steriliser pre-rinse	2	4:00	4:10	0:00	0:02	0:08	0:10	2	12:00	12:10	0:00	0:02	0:08	0:10
29	ST-100	Stopper Processor	3	5:00	6:00	0:00	0:00	0:00	0:00	2	13:00	14:00	0:00	0:00	0:00	0:00
30	ST-101	Stopper Washer & Steriliser pre-rinse	2	6:00	6:10	0:00	0:02	0:08	0:10	2	14:00	14:10	0:00	0:02	0:08	0:10
31	ST-101	Stopper Processor	3	5:00	6:00	0:00	0:00	0:00	0:00	2	15:00	16:00	0:00	0:00	0:00	0:00
32	ST-102	Stopper Washer & Steriliser pre-rinse	2	6:00	6:10	0:00	0:02	0:08	0:10	2	14:00	14:10	0:00	0:02	0:08	0:10
33	ST-102	Stopper Processor	3	7:00	8:10	0:00	0:00	0:00	0:00	2	15:30	16:30	0:00	0:00	0:00	0:00
34	ST-103	Stopper Washer & Steriliser pre-rinse	2	0:35	0:40	0:00	0:02	0:08	0:10	2	2:00	2:10	0:00	0:02	0:08	0:10
35	ST-103	Stopper Processor	2	1:15	1:25	0:00	0:00	0:00	0:00	2	2:30	2:50	0:00	0:00	0:00	0:00
36	ST-104	Stopper Washer & Steriliser pre-rinse	2	2:15	2:25	0:00	0:02	0:08	0:10	2	3:30	3:40	0:00	0:02	0:08	0:10
37	ST-104	Stopper Processor	2	3:00	3:25	0:00	0:00	0:00	0:00	2	4:40	4:50	0:00	0:00	0:00	0:00
38	VF-100	Vial Rinser - Fill	0.5	0:15	0:25	0:00	0:02	0:08	0:10	3	1:00	1:30	0:00	0:02	0:08	0:10
39	VF-101	Vial Rinser - Fill	0.5	6:00	6:35	0:00	0:00	0:00	0:00	3	7:00	7:35	0:00	0:00	0:00	0:00
40	VF-102	Vial Rinser - Fill	0.5	10:00	10:35	0:00	0:02	0:08	0:10	3	11:00	11:35	0:00	0:02	0:08	0:10
41	VF-103	Vial Rinser - Fill	0.5	20:00	20:35	0:00	0:00	0:00	0:00	3	21:00	21:35	0:00	0:00	0:00	0:00
42	PW-100	Parts washer Sanitization	2.5	22:00	22:05	0:00	0:02	0:08	0:10	2						

Appendix 9 Publications

This appendix contains the poster publication.

Simulation of DI/WFI Distribution Systems:

exploring deterministic, stochastic and fuzzy logic as methods of uncertainty description with Excel as the modelling platform

Coláiste na hOllscoile Corcaich, Éire
University College Cork, Ireland

Frank Riedewald and Edmond Byrne
Department of Process & Chemical Engineering, University College Cork, Ireland

Introduction

Water For Injection(WFI) and to a lesser degree Deionised Water (DI) are the most expensive utilities in a pharmaceutical plant. WFI is produced by expensive distillation methods. Ongoing quality controls, and regulatory scrutiny further adding to costs.

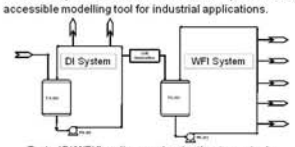
Expansion of WFI plants, in turn are hugely expensive due to validation costs, material of construction etc. Such expansion projects often do not consider that an existing system may be underutilised, as it is difficult to assess if the process extension can be supported by the existing WFI system without the use of simulation programs.

Aim

The aim of this work is to research and identify the best method available to model ***volume and schedule uncertainty*** in a WFI production plant.

A "good model should include uncertainty" [2], however only deterministic simulations have, so far, been applied in practice.

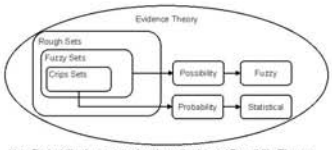
A secondary aim is to examine the suitability of Microsoft Excel as an accessible modelling tool for industrial applications.



Typical DI/WFI flow diagram showing the storage tanks and the distribution pumps.

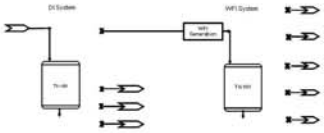
Model Development

The literature reveals service factors, intervals, rough sets, FL and stochastic methods to describe uncertainty. All of these were investigated for their suitability to model uncertainties in WFI systems.



How Probability theory (stochastic methods) and Possibility Theory (FL) fit into Evidence Theory (an overall theory of Uncertainty).

Probability theory (stochastic methods) and as an alternative Possibility Theory (FL) were indentified by the literature review as candidates to describe uncertainty in existing WFI systems.

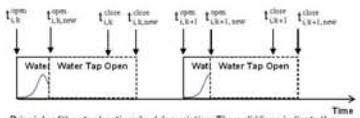


How the models simulate the DI/WFI systems (Note that the piping layout is not modelled).

All models utilise discrete-event simulation [3] to determine the WFI demand profile from the WFI distribution system, and a continuous simulation to calculate the variation of the level in the storage tank.

The **deterministic** model is a subset of the stochastic model and is not described here.

The **stochastic** model describes the variation of the opening and closing times of the WFI off-take valves themselves. Therefore the stochastic model requires historical data or good estimated data, which should be available for expansion projects.

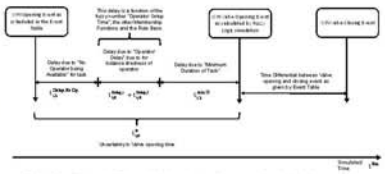


Principle of the stochastic schedule variation: The solid lines indicate the crisp schedule as given in the event table, while the dotted lines indicate the variation of the crisp schedule by stochastic uncertainties.

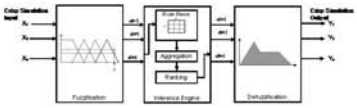
Screenshot of Excel input interface for the stochastic simulation showing the event table (excerpt).

The **Fuzzy Logic (FL)** model simulates schedule uncertainty by modelling operator behaviour. Volume uncertainty is not modelled by the FL model, as it is small compared to schedule uncertainty. In practice, despite the advances of automation, operator interaction with the process is still very common in modern pharmaceutical plants. Therefore the schedule may be affected by human fatigue and other operator influences, which are simulated by the FL model proposed. As a result, the FL model describes schedule uncertainty fundamentally different to the stochastic model.

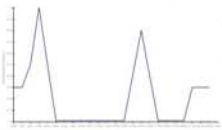
The FL model requires priority rules, number of operators, tiredness level, shift pattern, break pattern, rule bases, aggregation procedures, and defuzzification rules as input parameters.



Principle of how the FL model simulates slippages in schedule by operator fatigue etc.



Principle of the FL simulation showing rule base, aggregation, ranking and defuzzification resulting in a crisp output.



FL Membership Function: Tiredness level of the operator over 24 hours (adopted after Johnsson et al. [4]).

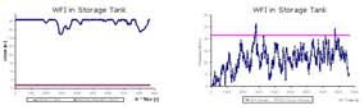
Materials and Methods

All data used is synthetic as an industrial partner could not be identified. Data used is, however, close to an actual WFI system.

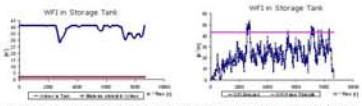
The demand from the production plant in increased by adding more production suites till a delivery limit on the WFI has been reached. This limit turned out to be the distribution pump flowrate.

Results

1. Results of the Stochastic Simulations:

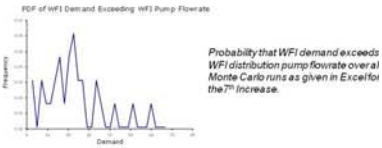


Predicted level in storage tank (left) and WFI demand overtime (right) for the 7th Increase 1st Monte Carlo run.



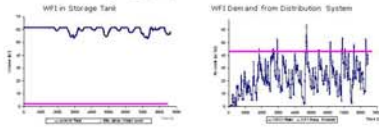
Same as above but 11th Monte Carlo run, note the difference in predicted WFI volume in tank and WFI demand.

Should the WFI demand exceed the WFI Distribution pump flowrate the result will be delays in production, as WFI would not be available for processes located towards the end of the WFI distribution loop as shown on the right hand graphs above.

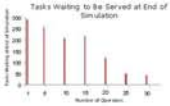


Probability that WFI demand exceeds WFI distribution pump flowrate over a Monte Carlo runs as given in Excel for the 7th Increase.

2. Results of the Fuzzy Logic (FL) Simulations:



FL simulation result of the 7th Increase giving a deterministic result.



Tasks waiting to be served at the end of the simulation timeframe of 24 hours over the number of operators available.

The figure on the left shows the minimum number of operators required, so that all operations are served within a 24 hour time frame for the 1st to the 7th Increase.

Discussion and Conclusions

The Stochastic method is the method of choice to describe uncertainty in WFI systems. It is superior to the FL method on a number of counts:

- The FL model is deterministic and cannot show the day-to-day variation of a real system.
- The FL model is more complex to program (1,200 versus 400 lines of code, more Do-Loops and If-End Blocks, etc. adding to code complexity).
- The FL model requires subjective inputs (i.e. tiredness of operator).
- The FL model cannot improve its accuracy should more information become available. The stochastic model, however, can readily improve its accuracy (Bayesian model).
- The FL model includes the number of operators as an input. A utility system, however, should not determine the number of operators.

In summary, contrary to some suggestions in the literature, FL is neither an alternative [5, 6] nor is it complementary [7] to stochastic methods in the simulation of WFI systems.

Excel can be used to simulate industrial size WFI systems. Model tested a system with over 1,600 WFI valve opening/closing events. However, Excel's graphs are poor (see examples above), the random number generator is not suitable [8] and some of the stochastic distributions and statistical tools may be incorrect [9].

Other Potential Applications of Program

Other utilities in the chemical industry such as heat transfer, steam, etc. may also be assessed/sized using the deterministic or stochastic Excel program.

Bibliography

1. Saraph Prasad V. 2001. Biotech Industry: Simulation and Beyond. In *Proceedings of the 2001 Winter Simulation Conference*, 838-843: Arlington, Virginia, IEEE Computer Society.
2. Pinedo Michael L. 2008. *Scheduling Theory, Algorithms, and Systems*. 3rd ed. New York: Springer.
3. Averill M. Law and David M. Kelton. 1991. *Simulation Modeling and Analysis*. 2nd ed. New York: McGraw-Hill.
4. Anders Johnsson and Jan E. Fröberg. 1975. Work Schedules and Biological Clocks. *Ambio* 4 (1): 45-50.
5. Balasubramanian J. and I. E. Grossmann. 2003. Scheduling optimization under uncertainty - an alternative approach. *Computers & Chemical Engineering* (27): 469-490.
6. Möller Bernd and Michael Beer. 2008. Engineering computation under uncertainty - Capabilities of non-traditional models. *Computers & Structures*. (86): 1024-1041.
7. Zadeh Lotfi A. 1995. Discussion: Probability Theory and Fuzzy Logic Are Complementary Rather Than Competitive. *Technometrics* (37): 271-276.
8. McCullough B. D. 2008. Microsoft Excel's 'Not The Wichmann-Hill' random number generators. *Computational Statistics & Data Analysis* 52 (10): 4587-4593.
9. Yalta A. Talha. 2008. The accuracy of statistical distributions in Microsoft® Excel 2007. *Computational Statistics & Data Analysis* 52 (10): 4579-4586.

Future Work

Validate models on an actual WFI system.

Program Download (Future)

http://www.ucc.ie/processeng (active October 2010)

Contact

frankriedewald@gmail.com Mobile: 00353 0862463 832