SANDIA REPORT
SAND2010-7131
Unlimited Release
December 2010

U.S. Nuclear Regulatory Commission Extremely Low Probability of Rupture Pilot Study:

xLPR Framework Model User's Guide

Patrick D. Mattie, Donald A. Kalinich, and Cedric J. Sallaberry

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000."

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2010-7131
Unlimited Release
December 2010

U.S. Nuclear Regulatory Commission Extremely Low Probability of Rupture Pilot Study:

xLPR Framework Model User's Guide

Patrick D. Mattie and Donald A. Kalinich
Reactor Modeling & Analysis

Cedric J. Sallaberry
Applied Systems Analysis

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0748

Abstract

For the U.S. Nuclear Regulatory Commission (NRC) Extremely Low Probability of Rupture (xLPR) pilot study, Sandia National Laboratories (SNL) was tasked to develop and evaluate a probabilistic framework using a commercial software package for Version 1.0 of the xLPR Code. Version 1.0 of the xLPR code is focused assessing the probability of rupture due to primary water stress corrosion cracking in dissimilar metal welds in pressurizer surge nozzles. Future versions of this framework will expand the capabilities to other cracking mechanisms, and other piping systems for both pressurized water reactors and boiling water reactors. The goal of the pilot study project is to plan the xLPR framework transition from Version 1.0 to Version 2.0; hence the initial Version 1.0 framework and code development will be used to define the requirements for Version 2.0. The software documented in this report has been developed and tested solely for this purpose. This framework and demonstration problem will be used to evaluate the commercial software's capabilities and applicability for use in creating the final version of the xLPR framework.

This report details the design, system requirements, and the steps necessary to use the commercial-code based xLPR framework developed by SNL.

ACKNOWLEDGMENTS

This report documents a major effort as part of a program funded by the United States Nuclear Regulatory Commission (USNRC), Office of Nuclear Regulatory Research. The Sandia team that worked on the xLPR Project would like to thank the xLPR Computational Task Group Lead, David Rudland of USNRC Office of Nuclear Regulatory Research, Division of Engineering. In addition, Aladar Csontos of the USNRC Office of Nuclear Regulatory Research and Craig Harrington of Electric Power Research Institute (EPRI) for their support of this Project. The views herein are those of the authors and do not represent an official position of the USNRC or EPRI.

A number of people at Sandia National Laboratories including, Randall Gauntt, manager Severe Accident Analysis (Org. 6232), Alicia Aragon (Org. 6234), Yvonne McClellan (Org. 6232), and Mona Aragon (Org. 6810) have made significant contributions to production of this report including an internal technical review of this report.

The Sandia team sincerely appreciates our associates in the xLPR Computational Task Group and their generous support during the xLPR pilot study project.

ABBREVIATIONS AND ACRONYMS

ASCII	American Standard Code for Information Interchange
CCDF	complementary cumulative distribution function
COD	crack opening displacement
DLL	dynamically-linked library
DOE	U.S. Department of Energy
DPD	Discrete Probability Distributions
EPRI	Electric Power Research Institute
I/O	Input/Output
NA	Not Applicable
NRC	U.S. Nuclear Regulatory Commission
PDF	portable document format
PND	probability of non-detection
QA	quality assurance
SC	surface crack
SMS	Simple Monte-Carlo sampling
SNL	Sandia National Laboratories
SQUIRT	Seepage Quantification of Upsets in Reactor Tubes
SSE	safe-shutdown earthquake
TWC	through-wall crack
xLPR	Extremely Low Probability of Rupture

CONTENTS

FIGURES

TABLES

1 INTRODUCTION

1.1 General Information

For the U.S. Nuclear Regulatory Commission (NRC) Extremely Low Probability of Rupture (xLPR) pilot study, Sandia National Laboratories (SNL) was tasked to develop and evaluate a probabilistic software framework using a commercial software package for Version 1.0 of the xLPR code.

The xLPR pilot study is focused on the development of a software tool to predict the probability of rupture for a pressurizer surge nozzle dissimilar metal weld. As part of the xLPR pilot study, teams (Computational, Models, Inputs, and Acceptance Criteria) were formed, each with specific long term and short term technical objectives, to address the xLPR issue. In the initial effort, the teams focused on a particular problem -- to determine the feasibility of creating a modular-based computer code for determining the extremely low probability of rupture for a pressurizer surge nozzle dissimilar metal weld. The final outcome of this pilot study will be an understanding of the tools available to solve this problem, limitations associated with these tools, and a firm basis for developing a more robust modular-type code. Further expanding the code based on the results from the pilot study will result in a working modular-type code focused on the generic, primary piping integrity issue (xLPR code).

The objective of the xLPR Framework Model User's Guide is to provide documentation of the capabilities of the xLPR framework using the commercial software as well as to provide simple instructions on how to install, navigate, and run the framework. Since the commercial software itself is graphical and was designed to be self-documenting, its documentation [1,2] should be used in tandem with the framework's User's Guide. The guide will refer to specific parts of the framework model where greater details are provided. However, it should be noted that it is not the intent of this document to provide the technical basis or to be a detailed user's guide for each of the individual software modules that comprise the xLPR model. Each module listed in Section 2.3 contains only a brief description of the module and the parameters required to be passed from the framework through the module interface (e.g., input/output arrays). The modules are documented in a detailed NRC report prepared by the xLPR Models Group [4].

The work described in this report supports NRC's plans for moving forward into Version 2.0 of the xLPR code.

1.2 Authorized Use Permission

This project was funded by the NRC with the collaboration of other national laboratories. The NRC frequently provides codes and technical assistance to domestic organizations for commercial purposes. A brief description of the integration of the multiple software programs that were used to develop the xLPR framework and model is described in this document. This project is very unique because it uses a combination of commercial and SNL developed software/code programs that are integrated to produce the final developed model. GoldSim is commercial software with a specific licensing agreement that provided the solution software for dynamic modeling. The dynamically linked library (DLL) software was developed by SNL from previous NRC and Electric Power Research Institute (EPRI) funded contract work. Funding for this work was provided under NRC Form 173, "xLPR Initial Framework Development", U.S. Department of Energy (DOE) Job Code # N6829, NRC B&R Number 601511127. The statutory, regulatory, and procedural intellectual property policies of the DOE are applicable to the work falling under this work order. Therefore, NRC and DOE requirements for release of portions of this software, codes, or other proprietary information must be strictly adhered to.

SNL, to the extent it is permitted to and asserts copyright therein, grants a royalty-free, nonexclusive, irrevocable worldwide license to the Government to use, reproduce, modify, distribute, prepare derivative works, release, display or disclose the articles, reports, summaries, abstracts, and related documents developed under this Agreement, for any governmental purposes and to have or authorize others to do so.

The commercial software used in this xLPR framework is licensed separately. However, NRC-funded software is software specifically developed for NRC by SNL. NRC shall advise the DOE Patent Counsel with respect to any rights in the software that NRC desires under any particular project, which rights include NRC imposing restrictions on use, and distribution of the software by DOE or SNL. Where NRC has not granted permission to copyright, NRC recognizes that once SNL has delivered to NRC a developed version of a particular code, SNL may exercise the existing right that both SNL and other parties have to further develop, without NRC funds, software codes that are in the public domain and to copyright the new, non-NRC-funded versions of these codes without NRC approval.

All those who create and manage information must understand the intent, purpose, and destination of the information they create in order to manage and protect it properly. Any resources for enabling that understanding include program direction documents, program funding documents between DOE, NRC, SNL, and other laboratories.

1.3 Points of Contact

The points of contact for the xLPR framework are

- Patrick Mattie (SNL) pdmatti@sandia.gov
- David Rudland (NRC) David.Rudland@nrc.gov

1.4 Change Record

1.4.1 Version/Data/Quality Assurance (QA) Designator

- xLPR Model Framework Version 1.0 – QA/NA
- xLPR Model Framework Version 1.01 – QA/NA
- xLPR Model Framework Version 1.02 – QA/NA

1.4.2 Change Log

- **October 13, 2010**
  GSxLPRv1.02_M02.gsm
  GSxLPRv1.02.gsp
  GSxLPRv1.02_M02_Deterministic.gsm
  GSxLPRv1.02_M02_Deterministic.gsp
  Revised controlled versions of xLPR framework model. Used for xLPR Pilot Study Program. Based on development version GSxLPRv1.01_M02.gsm.
  [This version includes a minor correction to the calculation of the probability of non-detection (PND) and reduced the number of file automatically exported from 201 to 52.]

- **October 7, 2010**
  GSxLPRv1.01_M02.gsm
  GSxLPRv1.01.gsp
  GSxLPRv1.01_M02_Deterministic.gsm
  GSxLPRv1.01_M02_Deterministic.gsp
  Revised controlled versions of xLPR framework model. Used for xLPR Pilot Study Program. Based on development version GSxLPRv1.0_M02.gsm.
  [This version includes a minor correction to the calculation of the Discrete Probability Distributions (DPD) Module response weights.]

- **September 30, 2010**
  GSxLPRv1.0_M02.gsm
  GSxLPRv1.0.gsp and GSxLPRv1.0_M02_Deterministic.gsm
  GSxLPRv1.0_M02_Deterministic.gsp
  Initial controlled versions of xLPR framework model. Used for xLPR Pilot Study Program. Based on development version Beta_v2.02_GS10.11_M02.gsm.
  [This is the initial baseline controlled version of the xLPR framework]

2 FRAMEWORK SUMMARY

2.1 Platform

The framework software (e.g., .gsm, .gsp, .EXE and .DLLx) run on personal computers using 32-bit or 64-bit Microsoft Windows operating systems. Windows Vista or Windows 7 (either 32-bit or 64-bit versions) are the recommended operating systems for GoldSim Player©. Windows XP (SP3) and Windows Server 2003 are also supported (although Windows XP 64-bit is not supported). The framework DLLs will run on any platform that will run the GoldSim Player© software.

2.2 Framework Configuration

The following items comprise the xLPR framework

- framework software (GoldSim© or GoldSim Player©)
- framework file (GoldSim file or GoldSim Player file)
- input data (Excel spreadsheet)
- Fortran DLLs
- post-processing executables and their associated input control files

The framework model (when run by the framework software) produces a set of output files. Those output files are in turn used as input by the post-processing executables to produce the post-processed output files. Figure 1 shows the relationship between the items in terms of the GoldSim software and GoldSim Player software.

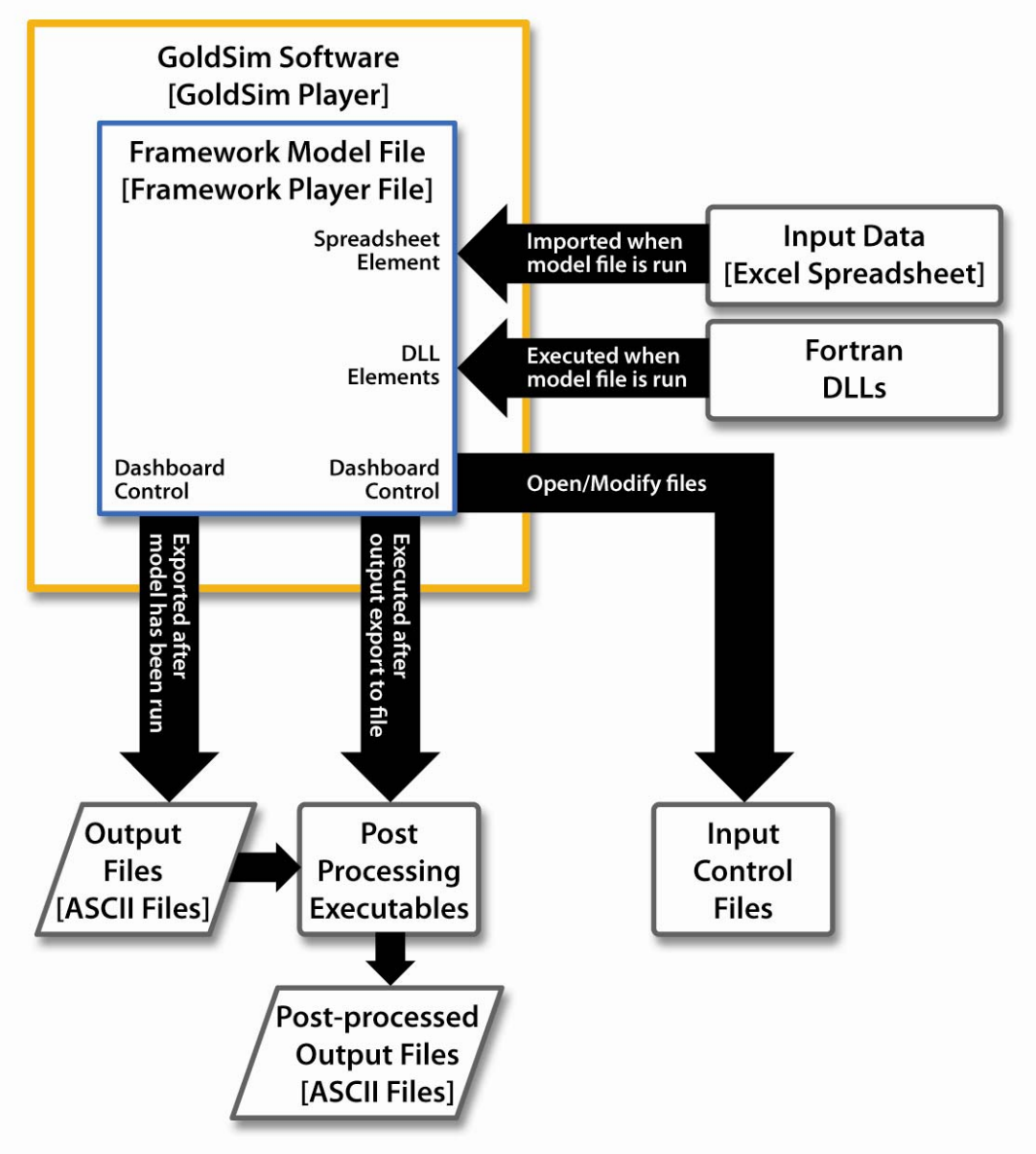


Figure 1. xLPR Framework.

2.2.1 Framework Software

GoldSim© version 10.11, Service Pack 4 is the software in which the framework model is developed and executed. The GoldSim software is documented in the GoldSim User's Guide Volumes 1 and 2 [1]. The GoldSim software is also used to create a GoldSim Player file of the framework module.

The GoldSim Player© software is a special version of the GoldSim software that allows an existing GoldSim model file to be "run" or "viewed" without having to license the GoldSim software. The GoldSim Player software is documented in the GoldSim Player User's Guide [2].

2.2.2 Framework File

The framework model file is a GoldSim file. It contains GoldSim elements. Each element represents a building block of the model. The elements and their interconnections make-up the framework model file portion of the xLPR framework. As depicted in Figure 1 there are specific elements in the framework model file that link it to the Fortran modules and input data. There are also elements that allow results from the model to be exported to American Standard Code for Information Interchange (ASCII) text files.

Chapter 2 of the GoldSim User's Guide provides a general explanation of the GoldSim modeling, elements, and element interconnections. DLL and Spreadsheet elements are described in the GoldSim User's Guide, Chapter 10. The use of Results elements to export data to ASCII files is described in the GoldSim User's Guide, Chapter 8. [1]

The "Development, Analysis, and Evaluation of a Commercial Software Framework for the Study of Extremely Low Probability of Rupture (xLPR) Events at Nuclear Power Plants" report [3] describes the details of the framework model.

The framework player file is a GoldSim Player file that is created from the framework model file using the GoldSim software (see the GoldSim User's Guide, Chapter 9 for details [1]). As seen from Figure 1 the framework player file interacts with the Fortran modules, input data, and output files in the same manner as the framework model file. Dashboard interfaces [1] are created using the developer options available with the licensed GoldSim software and allow a customizable player model interface for use with the GoldSim Player software. The interface allows data input changes, changes to model options (e.g., number of realizations, module options, number of time steps, etc.), and viewing and exporting results. However, the free player software does not allow for changes to the model logic as defined by adding, deleting, or modifying GoldSim model elements.

There are two versions of the framework. One version is configured to run the framework as a probabilistic case (i.e., sampled inputs for multiple realizations), the other is configured to run the framework as a deterministic case (i.e., a single realization with a fixed set of inputs).

The framework files are listed in Table 1.

Table 1. xLPR Framework Files

Framework File	Description
GSxLPRv1.0_M02.gsm	GoldSim software version of the framework configured to run as a probabilistic case.
GSxLPRv1.0_M02.gsp	GoldSim Player software version of the framework configured to run as a probabilistic case.
GSxLPRv1.0_M02_Deterministic.gsm	GoldSim software version of the framework configured to run as a deterministic case.
GSxLPRv1.0_M02_Deterministic.gsp	GoldSim Player software version of the framework configured to run as a deterministic case.

2.2.3 Input Data

The input data for the framework model is contained in an Excel workbook file (BETA_Inputs_AE_09_30_2010.xlsx). A GoldSim "Spreadsheet" element [1] is used to dynamically link to the Excel file. The uncertain parameters and constants are contained in separate worksheets in the Excel workbook.

2.2.4 Fortran DLLs

The Fortran DLLs were compiled using the Intel Fortran compiler (version 11.1.054) using the complier options listed in Table 2. The framework's Fortran DLLs are listed in Table 3. Detailed information on the individual models can be found in the "xLPR v.1.0 Models Report" [4].

Table 2. Fortran DLL Compiler Options

Configuration: Release
Optimization: Maximum Speed plus Higher Level Optimization (/O3)
command line compiler option "/assume:protect-parens"
command line compiler option "/QaxSSE4_2,SSE4_1,SSSE3,SSE3,SSE2,SSE
Runtime Library: Debug Multithreaded (/libs:static /threads /dbglibs)

Table 3. xLPR Framework Fortran DLLs

Fortran Module (DLL)	Description
Coalescense_DLLx_v2.2.dllx	Model used to evaluate whether two adjacent cracks will coalesce.
COD_DLLx_v2.1.dllx	Model used to calculate the crack opening displacement for through-wall cracks (TWC).
crack_init_v2.1.dllx	Model used to calculate the number of cracks initiated, time, and location of the initiated crack.
grower_DLL_v2.1.dllx	Model used to calculate the crack growth.
ISI_DLL_v2.1.dllx	Model used to calculate the PND.
kSurf_DLL_v1.1.dllx	Model used to calculate the stress intensity at the crack surface.
kTWC_DLL_v1.1.dllx	Model used to calculate the stress intensity at a crack's deepest point
load_DLL_v1.1.dllx	Model used to calculate the total axial membrane stress and total bending moment.
SCFail_DLL.v2.1.dllx	Model used to assess the stability of a surface crack (SC) in a pipe subjected to combined tension and bending loading.
SQURT_DLL_V1.1.dllx	Model used to calculate two-phase critical flow rate for water leaking from a TWC.
TWCFail_DLL_v2.1.dllx	Model used to assess the stability of a TWC in a pipe subjected to combined tension and bending loading.

2.2.5 Output

After the framework file has been run, a predetermined set of results can be automatically exported from the framework file to a set of ASCII text files. A listing of those files is provided in Table 4.

Table 4. xLPR Framework Output Files

Time History Output Files	Description
GSxLPRv1_AoT_xxx.txt[0,1]	Crack area over theta (crack length)
GSxLPRv1_CD_xxx.txt[0,1]	Crack Depths (m)
GSxLPRv1_CFO_001.txt[0,1,2]	Probability of Rupture (nominal)
GSxLPRv1_CFOS_001.txt[0,1,2]	Probability of Rupture w/Safe-Shutdown Earthquake (SSE)
GSxLPRv1_CFT_001.txt[0,1,2]	Rupture Time (nominal)
GSxLPRv1_CFTS_001.txt[0,1,2]	Rupture Times w/SSE
GSxLPRv1_CL_001.txt through *_019.txt[0,1]	Crack Lengths (m)
GSxLPRv1_TFO_001.txt [0,1,2]	Probability of first TWC
GSxLPRv1_TFT_001.txt [0,1,2]	Time of first TWC
GSxLPRv1_CO_001.txt through *_019.txt[0,1]	Crack Orientations (rad)
GSxLPRv1_COD_001.txt through *_019.txt[0,1]	Crack Opening Displacements (m)
GSxLPRv1_CT_001.txt through *_019.txt[0,1]	Crack Type (0, -1, -2, or 200)
GSxLPRv1_FLO_001.txt[0,1,2]	First Leak Probability
GSxLPRv1_FLT_001.txt[0,1,2]	First Leak Time (month)
GSxLPRv1_FSA_001.txt[0,1,2]	Fraction of surface area cracked
GSxLPRv1_PND_001.txt through *_019.txt[0,1,2]	PND
GSxLPRv1_SFO_001.txt[0,1,2]	Probability of SC Failure (nominal)
GSxLPRv1_SFOS_001.txt[0,1,2]	Probability of SC Failure w/ SSE
GSxLPRv1_SFT_001.txt[0,1,2]	SC Failure Times (nominal)
GSxLPRv1_SFTS_001.txt[0,1,2]	SC Failure Times w/SSE
GSxLPRv1_SIe_001.txt through *_019.txt[0,1]	Stress Intensity at the Crack Edges (K0)
GSxLPRv1_SIt_001.txt through *_019.txt[0,1]	Stress Intensity at the Deepest Point (K90)
GSxLPRv1_TCOA_001.txt[0,1]	Total Crack Opening Area (m^2)
GSxLPRv1_TCOD_001.txt[0,1]	Total Crack Opening Displacement (m)
GSxLPRv1_TLR_001.txt[0,1,2]	Total Leak Rate (m^3/s)
GSxLPRv1_ToP_001.txt through *_019.txt[0,1]	Ratio of Theta over Pi
GSxLPRv1_TPND_001.txt[0,1,2]	Total PND (product of all individual PND)

[0]indicates files linked to automatically export in v1.0 of the software framework (205 total).
[1]indicates files linked to automatically export in v1.01 of the software framework (207 total).
[2]indicates files linked to automatically export in v1.02 of the software framework (52 total).

2.2.6 Post-Processing

Results from the model file can be post-processed. The post-processing programs and their associated input control files are listed in Table 5. A listing of the input controls for TRANSFORMERS and EXPECTATION is provided in Table 6 and Table 7, respectively.

Output files are generated by TRANSFORMERS_v1.0.exe using the input file names and optional suffix. The output files generated by EXPECTATION_v1.0.exe are described in Table 8.

See Appendices C and D for TRANSFORMERS and EXPECTATION user's guides.

Table 5. Post-Processing Executables and Input Control Files

File	Description
TRANSFORMERS_v1.0.exe	Post Processor used to account for effects from leak rate detection and inspection for cracks
EXPECTATION_v1.0.exe	Post processor used to average any result over the aleatory uncertainty and estimate the mean and selected quantiles
Options.txt	Input control file for TRANSFORMERS
Exp_Options.txt	Input control file for EXPECTATION
inspection.txt	File containing the times (in months) at which inspections will be performed
quantiles.txt	File containing the quantiles of interest
times.txt	File containing the simulation time steps (in months)
variables_list.txt	File containing the names of the exported results files to be post processed by TRANSFORMERS and EXPECTATION

Table 6. Input Controls for TRANSFORMERS (Options.txt)

row	value	Description
1	GSxLPRv1_	output file suffix
2	variables_list.txt	file containing the names of all of the exported variables
3	pnd	prefix for PND files
4	TLR_001.txt	leak rate file name
5	CT	prefix for crack type files
6	times.txt	file containing the simulation time steps (in months)
7	721	number of simulation time steps
8	400	total sample size
9	19	maximum number of cracks considered
10	3	number of comments rows
11	1	number of columns to be ignored
12	1	inspection tag: 1 = inspection, 0 = no inspection
13	inspection.txt	file containing the times (in months) at which inspections are performed – optional: only when inspection tag = 1
14	3	number of inspections – optional: only when inspection tag = 1
15	0	dependency on inspection (0 = independent ; 1 = dependent) – optional: only when inspection tag = 1
16	1	convolution on inspection (0 = minimum ; 1 = product) – optional: only when inspection tag = 1
17	0	leak rate detection: 1 = threshold, 0 = no threshold
18	1.0	leak rate threshold value (m^3/s) – optional: only when leak rate detection tag = 1
19	1	transform tag: 1 = output data transformed, 0 = no transform
20	2	transform option 1: 0:<, 1:=, 2:> – optional: only when transform tag = 1
21	1	transform time option – optional: only when transform tag = 1
22	0.00	threshold value for the transform – optional: only when transform tag = 1
23	0	take max over time or not (1 = max ; 0 = no change)
24	_COR	suffix for output file (“NO” – no suffix – variable file will be overwritten)

Table 7. Input Controls for EXPECTATION (Exp_Options.txt)

row	value	Description
1	GSxLPRv1_	output file suffix
2	variables_list.txt	file containing the names of all of the exported variables
3	quantiles.txt	file listing the quantiles of interest
4	times.txt	file containing the simulation time steps (in months)
5	20	epistemic sample size
6	NO	epistemic uncertainty weight file (NO = unweighted results)
7	N	epistemic weight normalization (“Y” or “N”)
8	721	number of simulation time steps
9	2	number of comments rows
10	1	number of columns to be ignored
11	20	aleatory sample size
12	NO	aleatory uncertainty weight file (NO = unweighted results)
13	N	aleatory weight normalization (“Y” or “N”)
14	3	number of quantiles considered
15	6	log info number (for debugging)
16	NO	suffix (NO = no suffix) (_COR)
17	1	CCDF option (0 = no, 1 = yes)
18	50.00	time (in months) to use for CCDF (optional)
19	30	number of discretization points
20	1	discretization type for the stats (1: linear ; 2: log)

Table 8. EXPECTATION Output Files (Default)

Output Files	Description
GSxLPRv1_V_qqq_ccdf0000yyy.txt	File containing all CCDFs for time steps
GSxLPRv1_V_qqq_exp.txt	File containing the expected value over time
GSxLPRv1_V_qqq_log.txt	Log file used for debugging
GSxLPRv1_V_qqq_stat.txt	File containing the statistics on the expected value
GSxLPRv1_V_qqq_stat0000yyy.txt	File containing the statistics on the CCDFs for time steps
notes: V = variable name qqq = file number yyy = CCDF time (see Table 7, row 12) An output file of each type (e.g., ccdf, exp, log, stat, stat0000yy) is created corresponding to the files listed in variable_list.txt file.	

2.3 Framework Configuration Flow Diagram

The configuration of the xLPR framework is shown in Figure 2. The framework configuration can be divided into two main parts

Initialization
- Data_Source
- Constants
- Uncertain_Parameters
- Uncertainty_Structure
- Time_Zero

Time Loop
- Crack_Initiation
- Crack_Growth
- Coalescense
- Criticality_SC
- Criticality_TWC
- Leakage
- Inspection

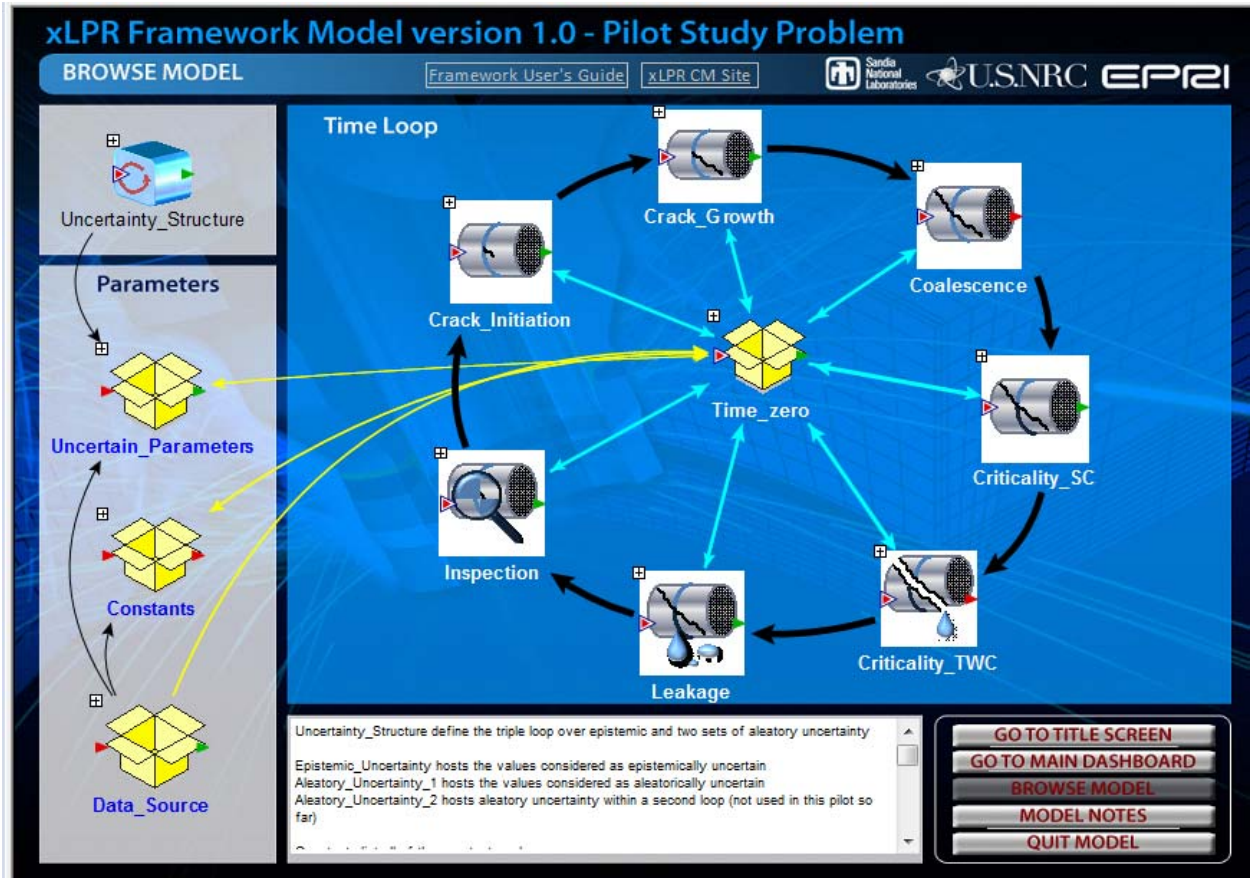


Figure 2. xLPR Framework Configuration Flow Diagram.

In the xLPR framework player file the above diagram can be "browsed". The individual containers can be opened (e.g., see
Figure 3). Also, the influence links (depicted as arrows) and the GoldSim **Affects…** and **Function Of…** features can be used to navigate the framework.

Input data from the input data spreadsheet (see Section 2.2.3) are linked to the framework in the Data_Source container via Spreadsheet elements (see
Figure 3).

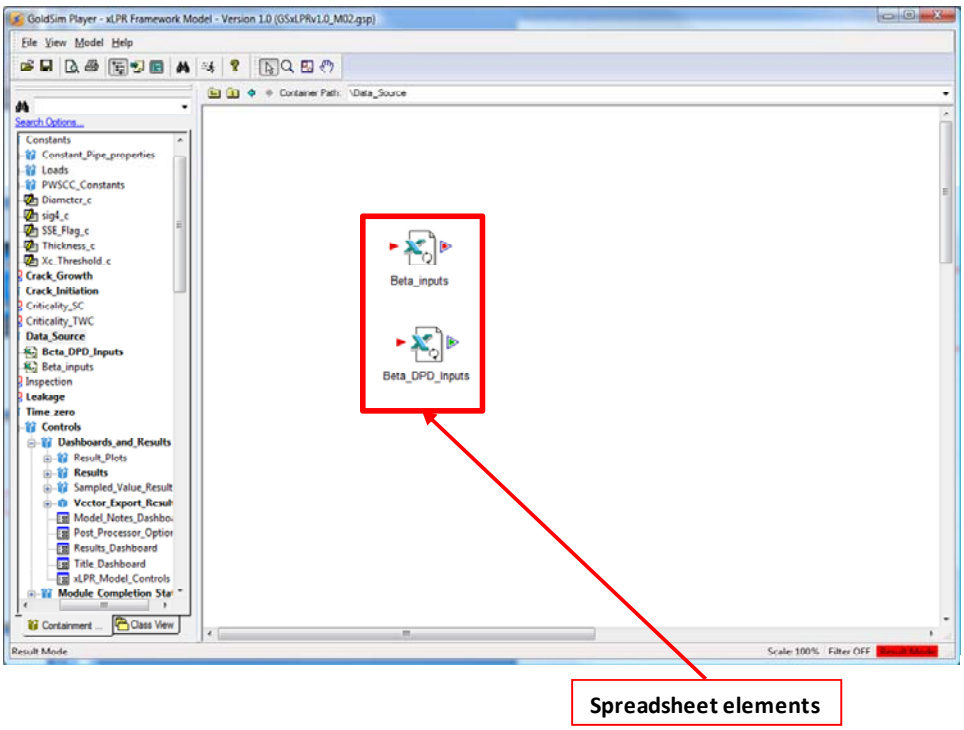


Figure 3. Data_Source Container

Constants are input to Data elements in the Constants container. As seen in Figure 4 the constants are organized using containers.

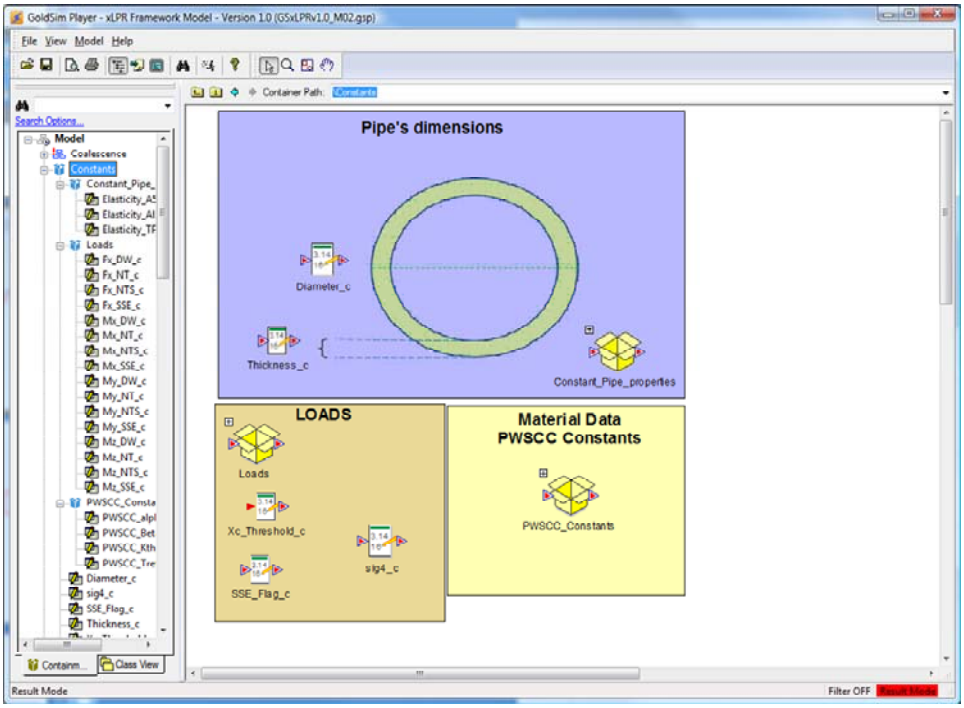


Figure 4. Constants Container

Uncertain parameters are input to Stochastic elements in the Uncertain_Parameters container (see Figure 5). This is also where the DPD module is implemented in the framework. In the framework, each uncertain parameter is sampled twice: once as if it were an epistemic parameter and once as if it were an aleatory parameter. The sampled value (aleatory or epistemic) used for each parameter in the framework is determined by its type as defined in the input data spreadsheet. If the DPD module is activated, parameters sampled by it are used in lieu of either the epistemic or aleatory sampling.

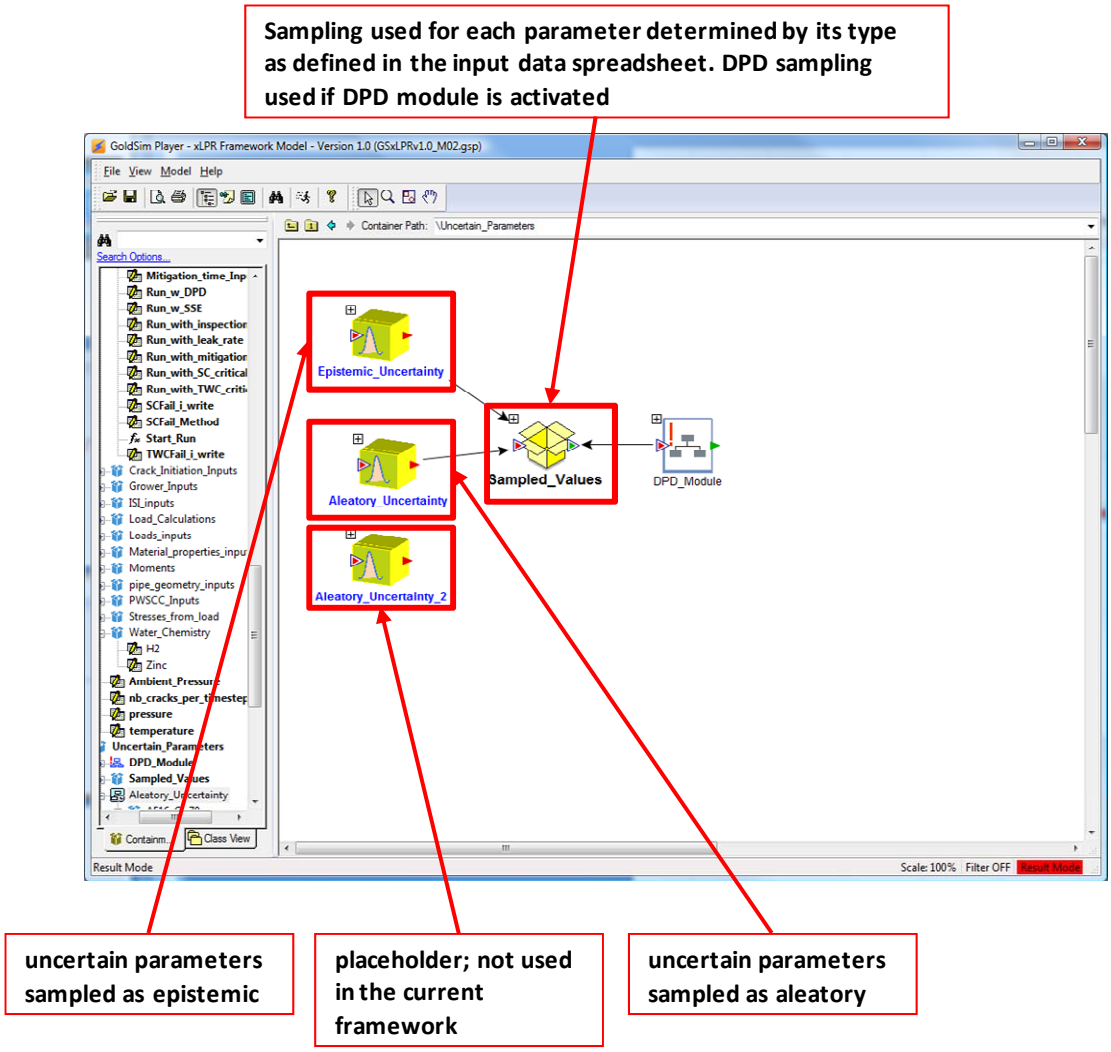


Figure 5. Uncertain_Parameters Container

When the DPD module is not activated, the Uncertainty_Structure container handles the separation of epistemic and aleatory uncertainty within the framework. When the DPD module is activated, all uncertain parameters are treated a epistemic.

In order to incorporate the multiple loops needed to separate the epistemic and aleatory uncertainty -- GoldSim only has a single realization loop -- the following logic is incorporated into the framework (see Figure 6).

$$N_G = N_e \times N_{a1} \times N_{a2}$$

$$X_e = max\left(1, trunc\left(\frac{N_G - 1}{N_{a1} \times N_{a2}}\right) + 1\right)$$

$$X_{a1} = max\left(1, mod\left(trunc\left(\frac{N_G - 1}{N_{a2}}\right), N_{a1}\right) + 1\right)$$

$$X_{a2} = mod(N_G - 1, N_{a2}) + 1$$

where:

N_G	-	total number of GoldSim realizations
N_e	-	total number of epistemic loop realizations
N_{a1}	-	total number of aleatory loop 1 realizations
N_{a2}	-	total number of aleatory loop 2 realizations
X_e	-	the ith epistemic loop realization
X_{a1}	-	the ith aleatory loop 1 realization
X_{a2}	-	the ith aleatory loop 2 realization

This looping structure is defined for one epistemic loop (outer loop) and two aleatory loops (center loop and inner loop). For the xLPR pilot study only the epistemic loop (outer loop) and 1st aleatory loop (inner loop) are used; the 2nd aleatory loop is not used[1]. The default xLPR framework realization settings for the probabilistic case are

$$N_e = 1000$$
$$N_{a1} = 50$$
$$N_{a2} = 1$$
$$N_G = 1000 \times 50 \times 1 = 10{,}000$$

[1] The loop is "deactivated" by setting its total number of realizations equal to 1.

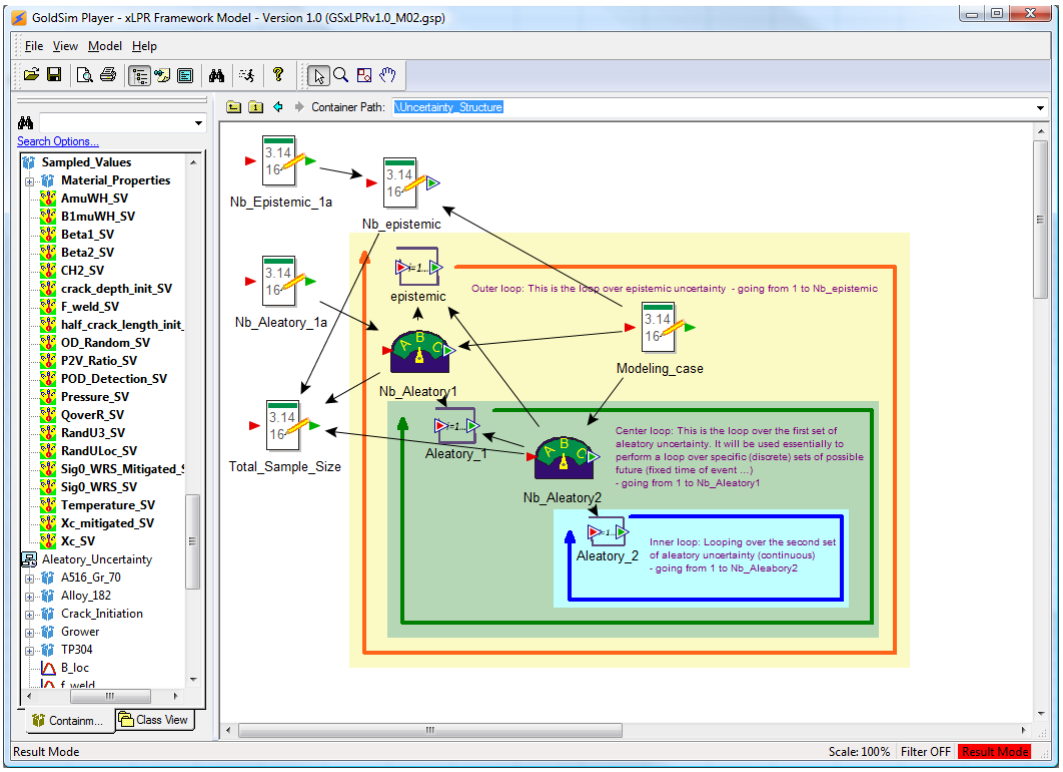


Figure 6. Uncertainty_Structure Container Showing Elements Arranged to Graphically Depict the Framework Looping Structure

The uncertainty sampling when the DPD module is activated is controlled by two parameters. The number of DPD bins determines how many sample sets of the uncertain parameters are sampled by the DPD module. The number of realizations in GoldSim determines how many realizations of the framework are run, with each realization using a randomly-sampled DPD sample set. Note that the number of epistemic and aleatory realizations on the Main Dashboard is not used when the DPD module is activated.

The framework input data is passed to the Time_Zero container (see Figure 7). Also within the Time_Zero container is the Controls container, which contains the elements and logic for the framework's dashboard screens (e.g., Main Dashboard, Results), including the results elements that capture the framework's output. Lastly, the load module which has no temporal variation, is implemented in the Time_Zero container.

The load module calculates the total axial membrane stress, with and without stress due to SSE and with and without weld residual stress. It also calculates the total bending moment with and without SSE. The load module also determines the weld residual stress fitting coefficients for its assumed 3rd order form, boundary conditions, and constraints.

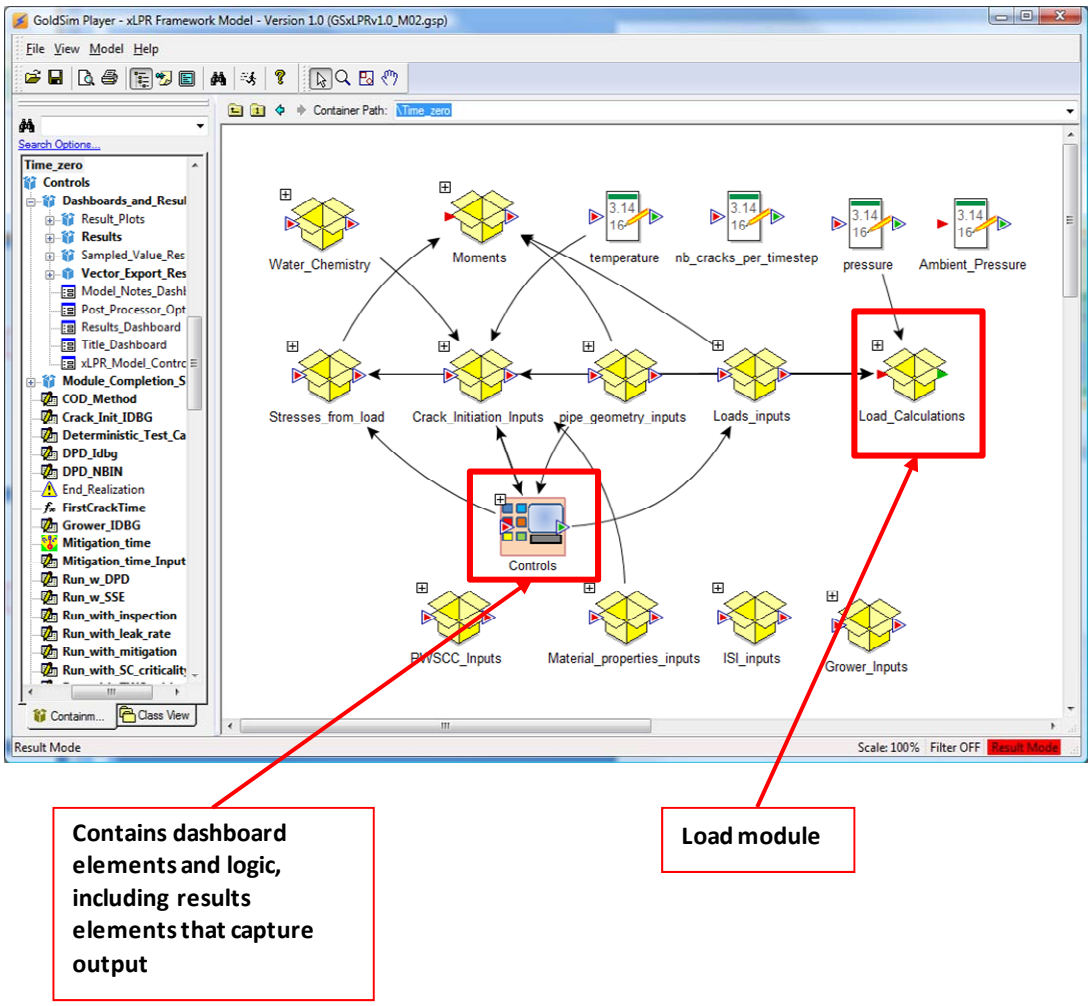


Figure 7. Time_Zero Container

The time loop portion of the framework executes the time-dependent modules in the following order:

- **Crack_Initiation**: Determines if one or more cracks are formed and where those newly-formed cracks are located on the pipe weld.
- **Crack_Growth**: Calculates the growth of cracks over the time step.
- **Coalescence**: Determines if any of the cracks coalesce.
- **Criticality_SC**: Determines if any of the SCs meet the SC criticality criteria. If so, the crack is assumed to now be a TWC.
- **Criticality_TWC**: Determines if any of the TWCs meet the TWC criticality criteria. If so, the pipe is considered ruptured and the realization is terminated.
- **Leakage**: Calculates leakage from the pipe via TWCs
- **Inspection**: Calculates the PND of cracks.

3 FRAMEWORK INSTALLATION, EXECUTION, AND RESULTS

This section explains how to install the software and files needed to run the framework. It then covers starting the software, opening and navigating the framework file, and how the user can set-up and run both deterministic and probabilistic simulations. A discussion of the framework results – monitoring during a run, browsing, exporting, and post processing – is also provided.

3.1 Installing the Framework

The installation instructions in this section will cover installing the framework from the CD provided with the User's Guide or from the xLPR SharePoint site.[2] While both the Framework Model File (run in GoldSim) and the Framework Player File (run in GoldSim Player) are provided to the user, only the GoldSim Player software is provided to the user.

Microsoft Excel is required to be installed on the computer on which the framework will be run. The framework as-installed requires Excel 2007. Excel 2003 can also be used with the framework if a copy of the input data spreadsheet file is created in Excel 2003 and the GoldSim Player framework file is changed to use the Excel 2003 file. Directions for modifying the framework to Excel 2003 versions of Excel are provided in Section 3.2.4 .

3.1.1 Installation from the User's Guide CD

The file listing for the xLPR framework installation CD is given in Table 9.

Table 9. xLPR Framework Installation CD File List

File	Description
GSxLPR_v1.02_M02.gsp	Framework Player File
GSxLPR_v1.02_M02.gsm	Framework Model File
GS Module Set M02.zip	Module Set input data Excel spreadsheet (see Section 2.2.3) Fortran DLLs (see Table 3) post-processing executables and input control files (see Table 5)

[2] A user account is required to access to the xLPR SharePoint site. Contact David Rudland <David.Rudland@nrc.gov> to begin the process for obtaining a user account.

The steps to install the framework are:

(1) Create a directory on the hard drive where the framework is to be installed.
(2) Copy the entire contents of the CD to the newly-created directory.
(3) Go to the GoldSim website <http://goldsim.com/forms/playerdownload.aspx> and download the GoldSim Player installation file to the newly-created directory.
(4) Install the GoldSim Player software by double-clicking on the setup file and following the instructions on the screen (see the GoldSim Player User's Guide [2] for details).

3.1.2 Installation from the xLPR SharePoint Site

The steps for installing the xLPR framework via the xLPR SharePoint site are:

(1) Create a directory on the hard drive where the framework is to be installed.
(2) Go to the GoldSim website <http://goldsim.com/forms/playerdownload.aspx>, click on the "free download" link, fill out the form, and click on the Request GoldSim Player button to download the GoldSim Player installation file to the newly-created directory.
(3) Install the GoldSim Player software by double-clicking on the setup file and following the instructions on the screen (see the GoldSim Player User's Guide [2] for details).
(4) Go to the xLPR SharePoint site (see Figure 8) <https://websps1.battelle.org/nrcnureg/home/xLPR_CM/default.aspx> and log on.

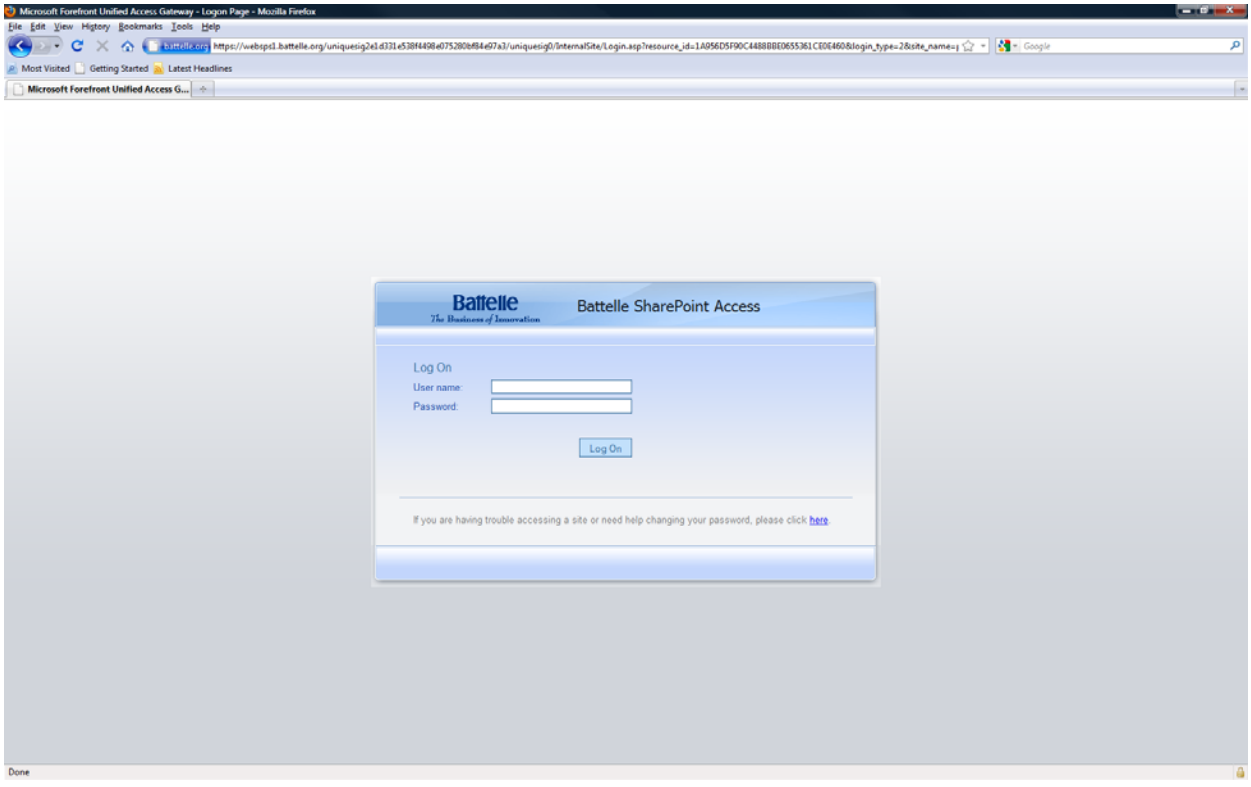


Figure 8. xLPR SharePoint Site Log-In Screen

(5) Under **Links**, click on the **Configuration Management** link (see Figure 9)

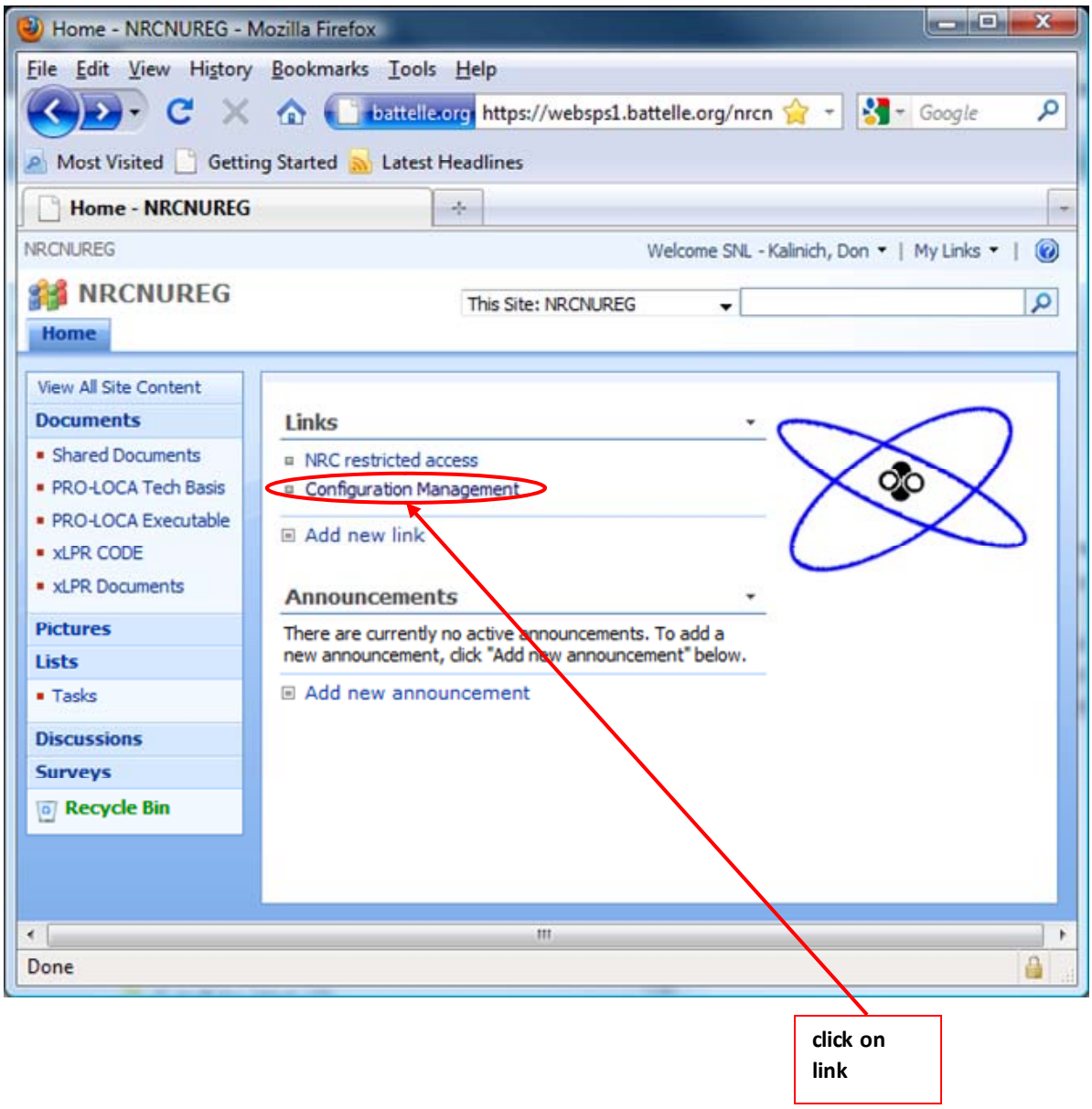


Figure 9. xLPR SharePoint Site Initial Page

(6) Under **Documents**, click on the **Controlled Files** link (see Figure 10)

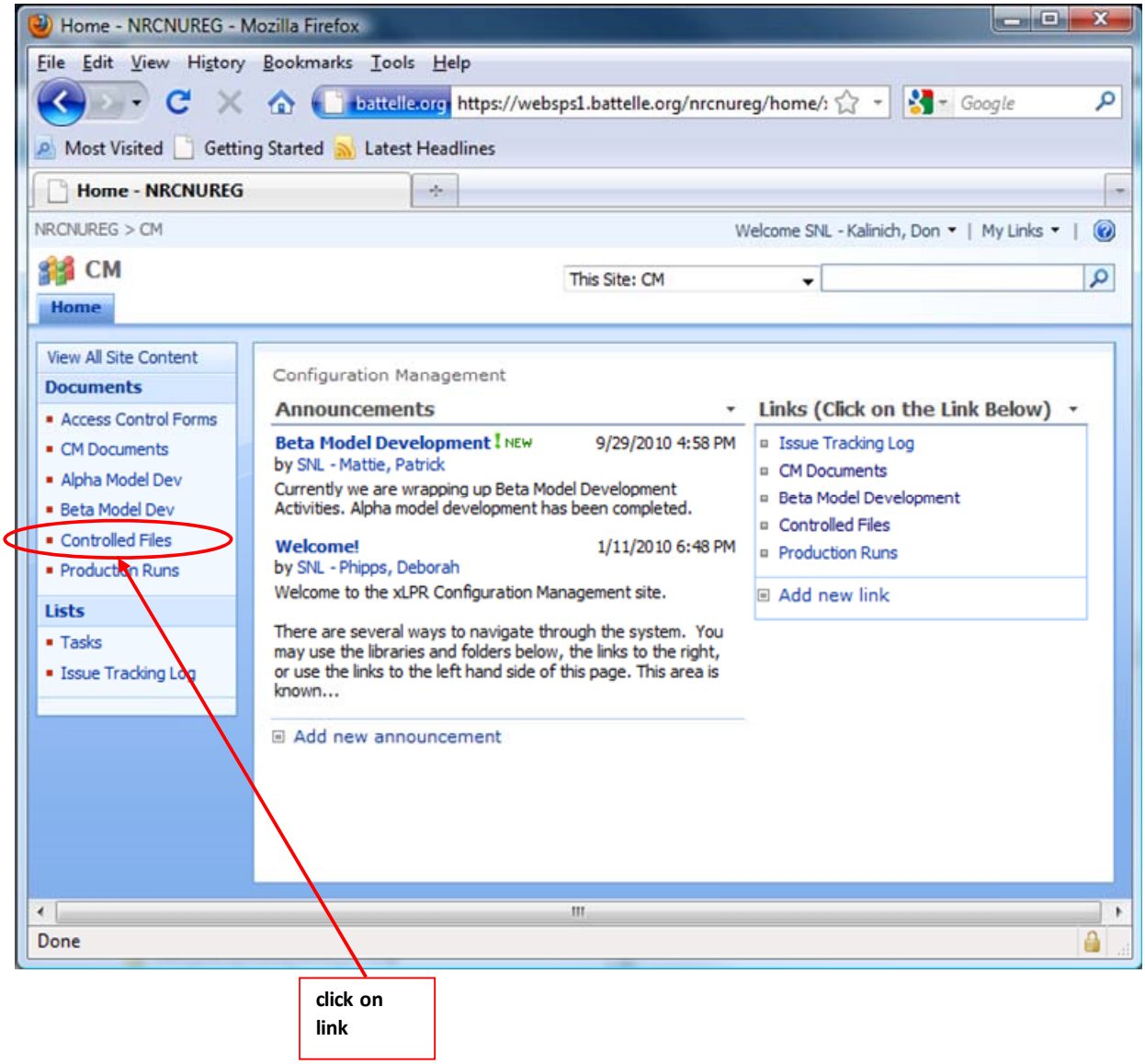


Figure 10. xLPR SharePoint Site CM Page

(7) Click on the **GS_xLPRv1.02** directory icon to access the xLPR framework files. Click on the **Module Set – M02** directory icon to access the module set .zip file (see Figure 11).

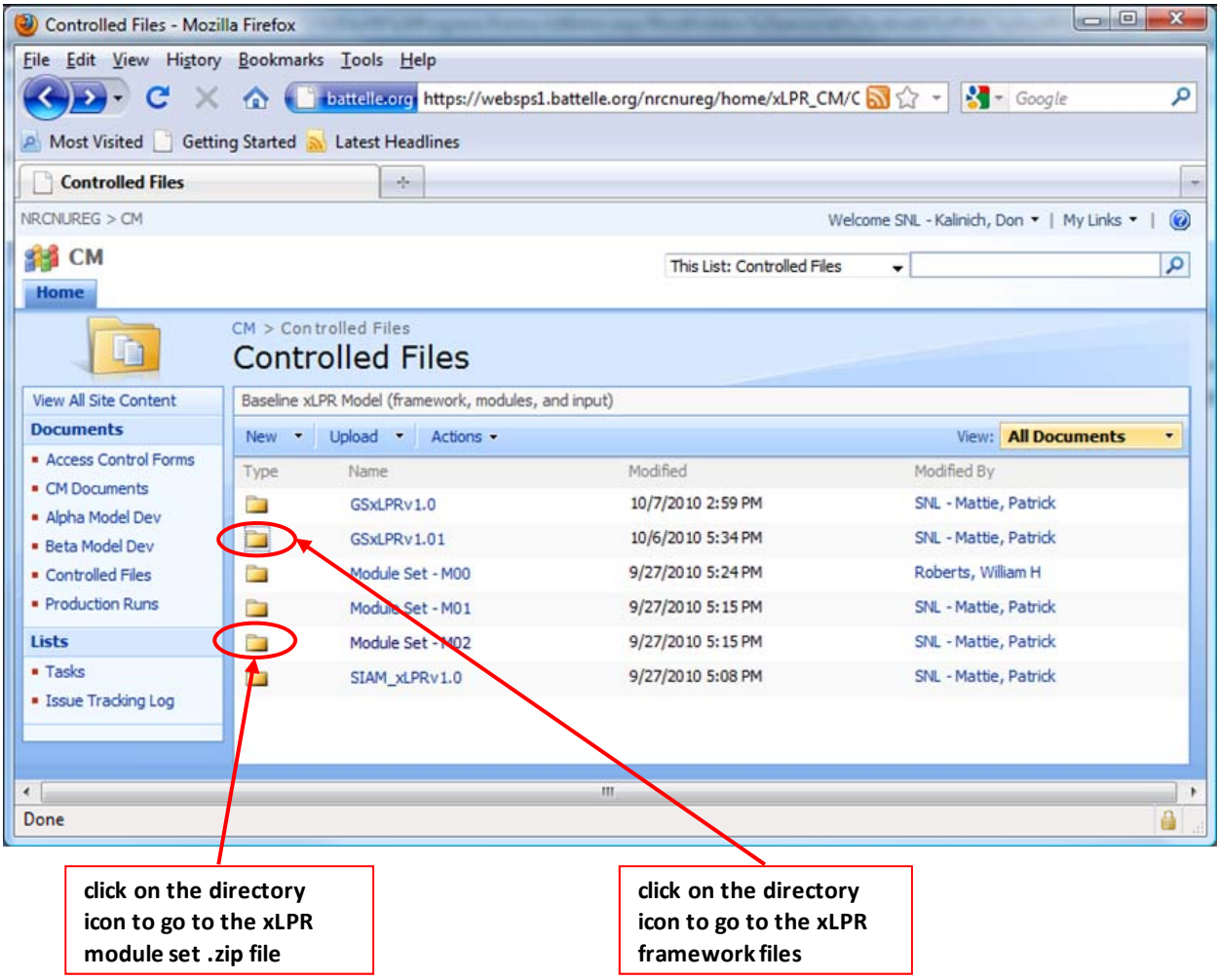


Figure 11. xLPR SharePoint Site Controlled Files Page

(8) Click on any of the file icons; a pop-up dialogue box will appear. Select the "save file" option in the pop-up dialogue box, save the file to the newly-created directory, and unzip it in the newly-created directory (see Figure 12).

Note that there two versions of the framework – a probabilistic version (GSxLPRv1.02_M02) and a deterministic version (GSxLPRv1.02_M02_Deterministic). Each version is available as a GoldSim file (.gsm file extension) or a GoldSim Player file (.gsp file extension). The full file name (including the file extension) will appear in a pop-up box when the cursor is hovered over the file's icon.

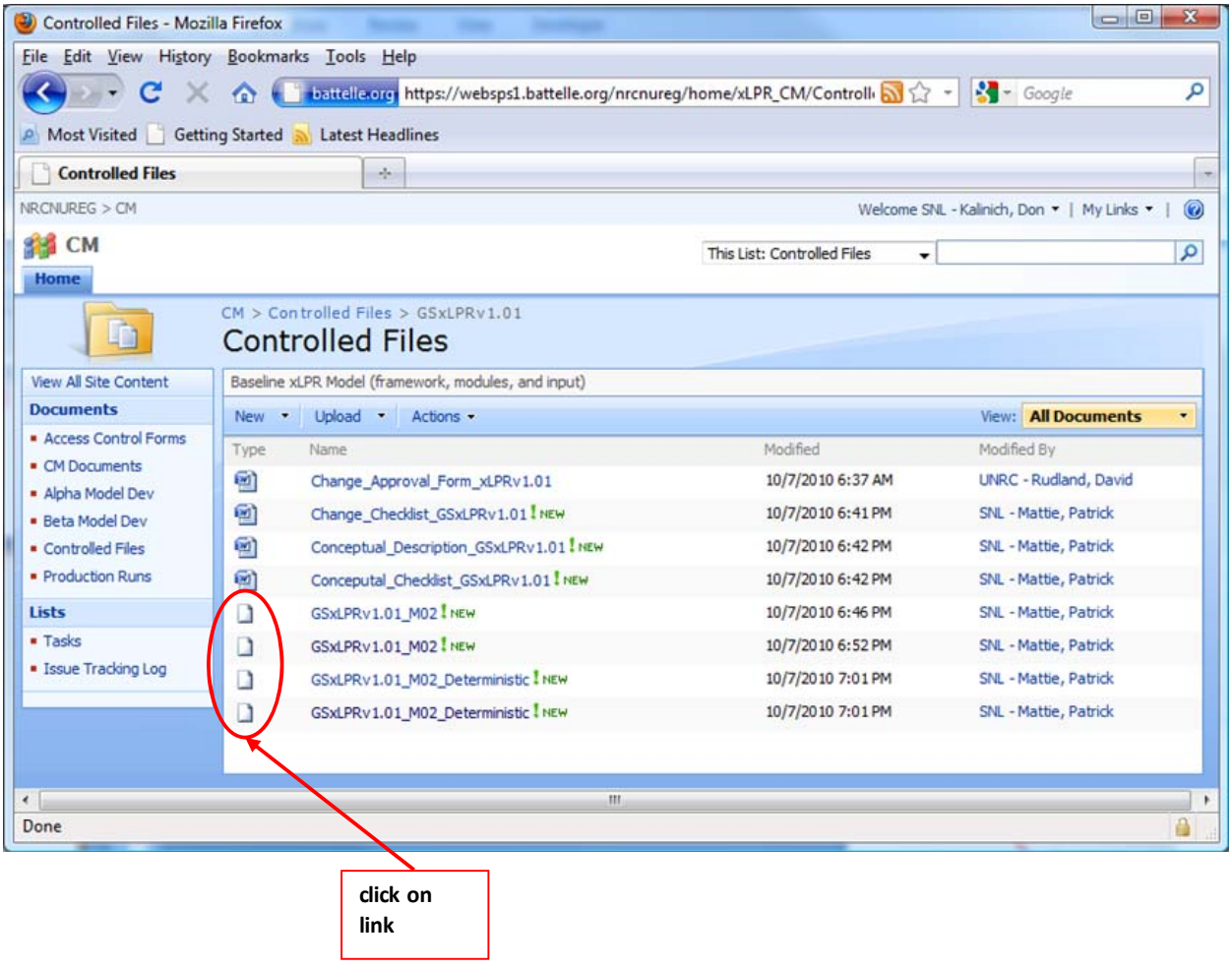


Figure 12. xLPR SharePoint Site Framework Files Directory

(9) Click on the **GS Module Set M02** zipped directory icon; a pop-up dialogue box will appear. Select the "save file" option in the pop-up dialogue box, save the file to the newly-created directory, and unzip it in the newly-created directory (see Figure 13).

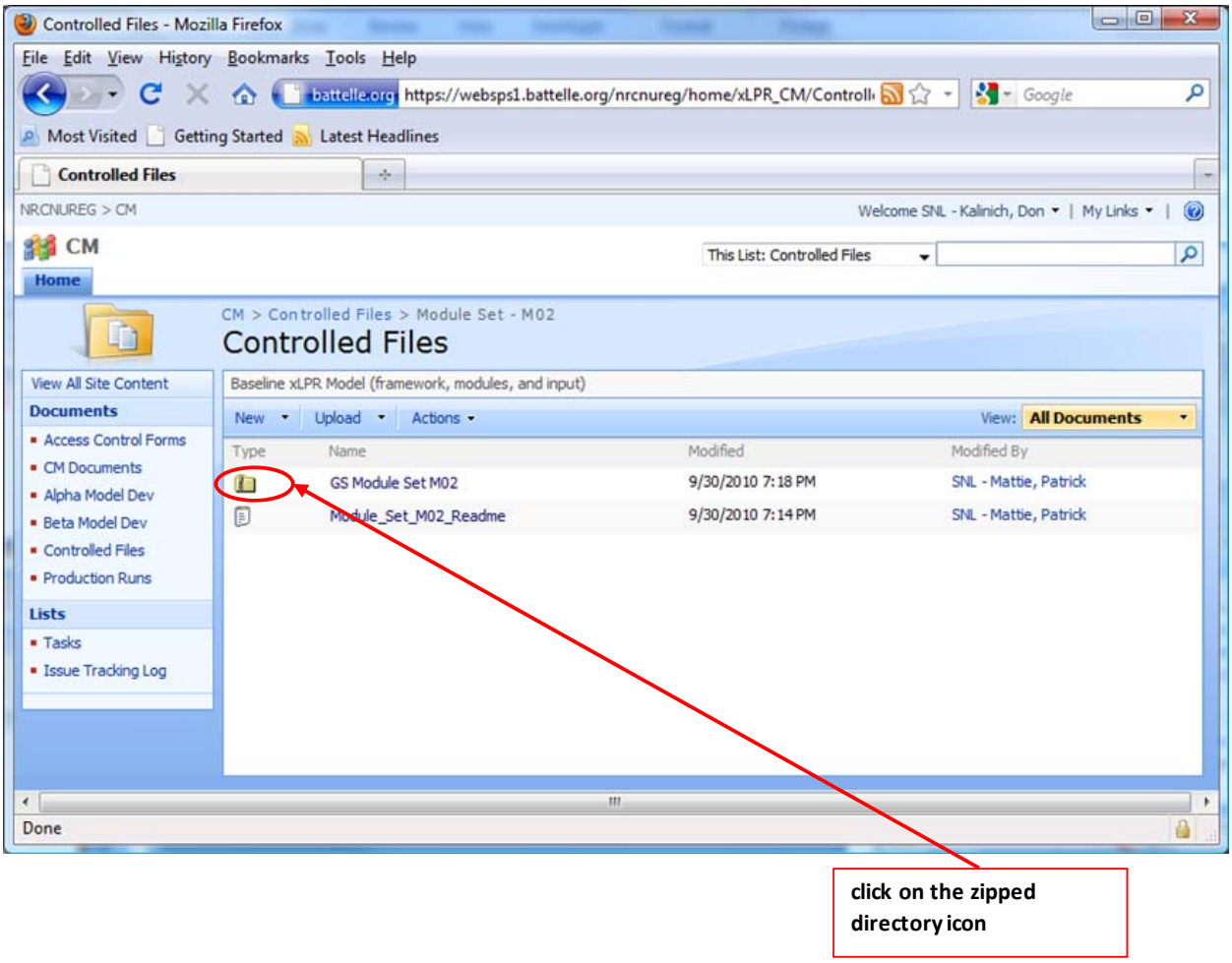


Figure 13. xLPR SharePoint Site Framework Module Set Directory

(10) Extract the files from the GS Module Set M02.zip folder (see Figure 14) in the same directory with the xLPR Framework GoldSim Player file.

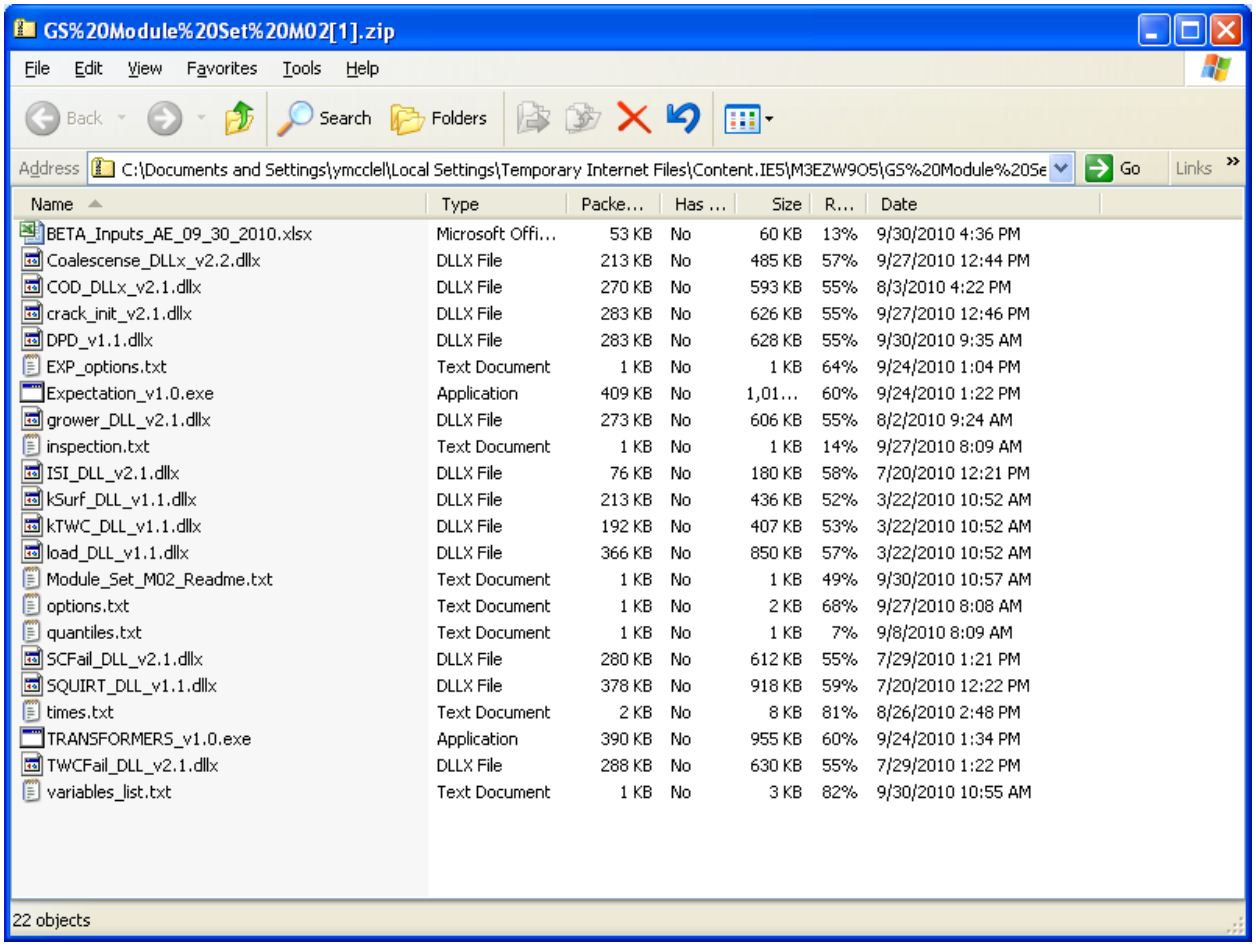


Figure 14. Screen Shot of Unzipped Module Set

3.2 Working with the Framework

3.2.1 Opening the Framework Player File

The framework player file can be opened by either double-clicking on the file (.gsp) or by launching the software and clicking on the **Open Model** button; a browser window will be opened to allow the user to select a Player file (see Figure 15).

Figure 15. GoldSim Player Opening Screen

3.2.2 Understanding and Navigating the Framework File

There are four main "screens" to the framework player file.

- Title Screen
- Main Dashboard
- Browse Model
- Model Notes/Contributors

The title screen appears when the xLPR framework Player file is opened (see Figure 16). There are three buttons on this screen. Clicking on the **Contributors** button takes the user to the Model Notes/Contributors screen. A copy of this User's Guide can be opened by clicking on the **Framework User's Guide** button. Clicking on the **START** button will take the user to the Main Dashboard. The GoldSim Run Controller also appears when the framework player file is opened.

Figure 16. Framework Title Screen

The Main Dashboard (see Figure 17) is where the user can

- change model options and simulation settings (see Section 3.2.5)
- change input data (see Section 3.2.6)
- run the model (see Section 3.2.7)
- view model results (see Section 3.3

A set of buttons is provided on the Main Dashboard which allow the user to

- go to the Title Screen
- go to the Model Notes/Contributors screen
- browse the model
- quit the model (i.e., close the model and shutdown the GoldSim Player software)

A button is also provided to open the Framework User's Guide.

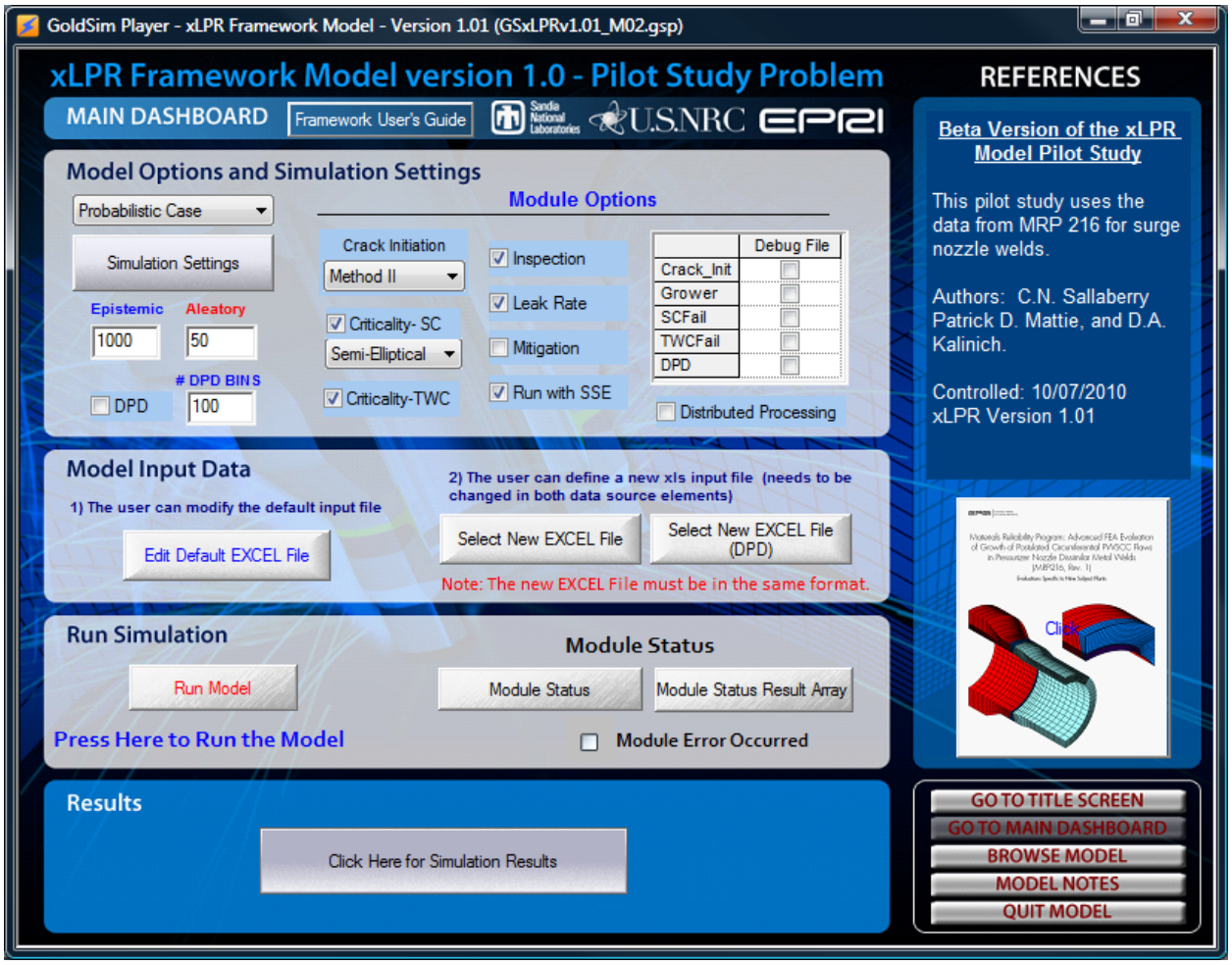


Figure 17. Framework Main Dashboard Screen

The Model Notes/Contributors screen contains (see Figure 18)

- model version information,
- the URL to the GoldSim website from which the GoldSim Player software can be obtained,
- the URL to the location of the xLPR controlled file on the xLPR SharePoint site
- contact information,
- the URL to the location of the Issue Tracking Log for the xLPR framework and its associated modules,
- notes regarding execution of the framework, and
- a listing of the organizations that contributed to the development of the framework.

As with the Main Dashboard there is a set of buttons which allow the user to

- go to the Title Screen,
- go to the Main Dashboard,
- browse the model, and
- quit the model (i.e., close the model and shutdown the GoldSim Player software).

Buttons are also provided that will open the Framework User's Guide and open (using a default web browser) the xLPR CM Share Point site log-in screen.

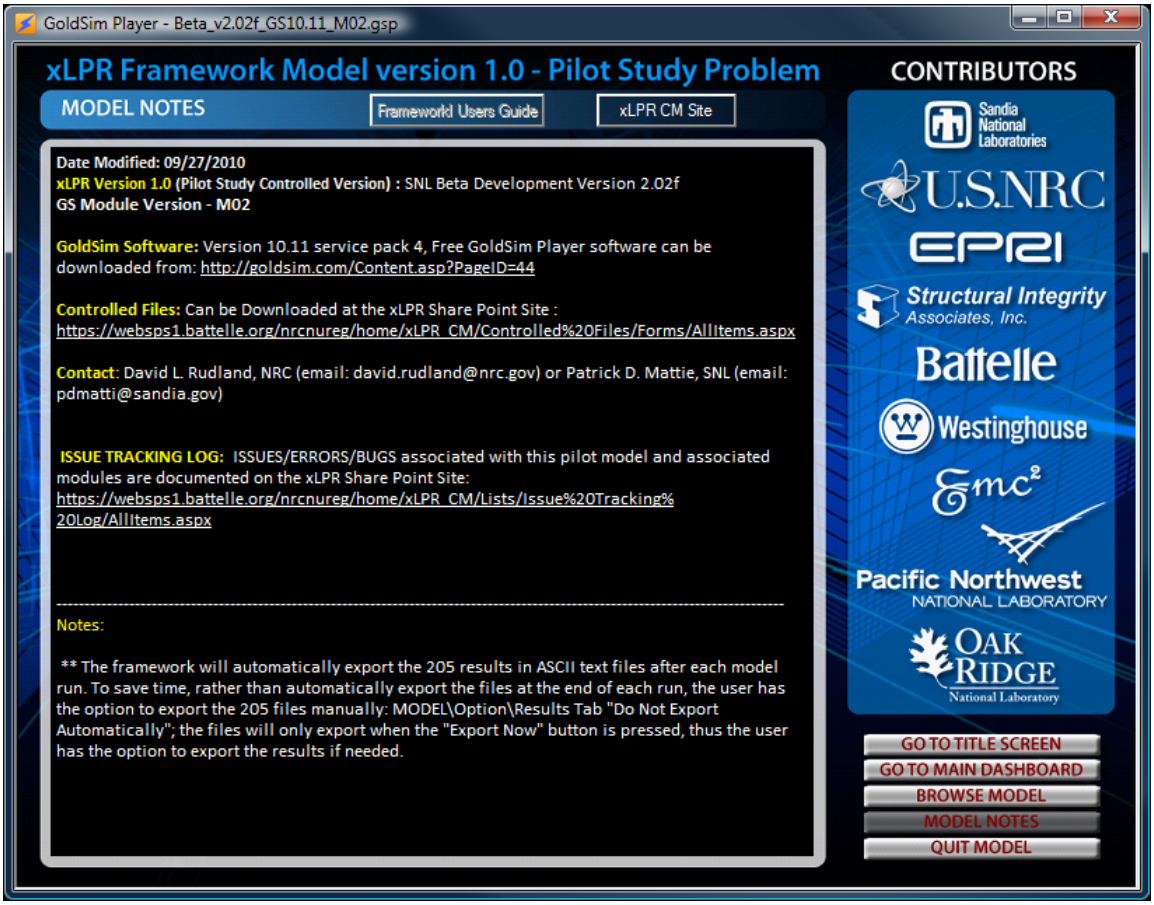


Figure 18. Framework Model Notes/Contributors Screen

Clicking on the **Browse Model** button displays the framework model using the GoldSim Player browser and graphics pane (see Figure 19). The browser displays the model in one of two ways: (1) in a hierarchical manner (referred to as Containment View), similar to the way that files and directories on a computer are organized; or (2) by element type (referred to as Class View). The graphics pane shows the contents of a single container.

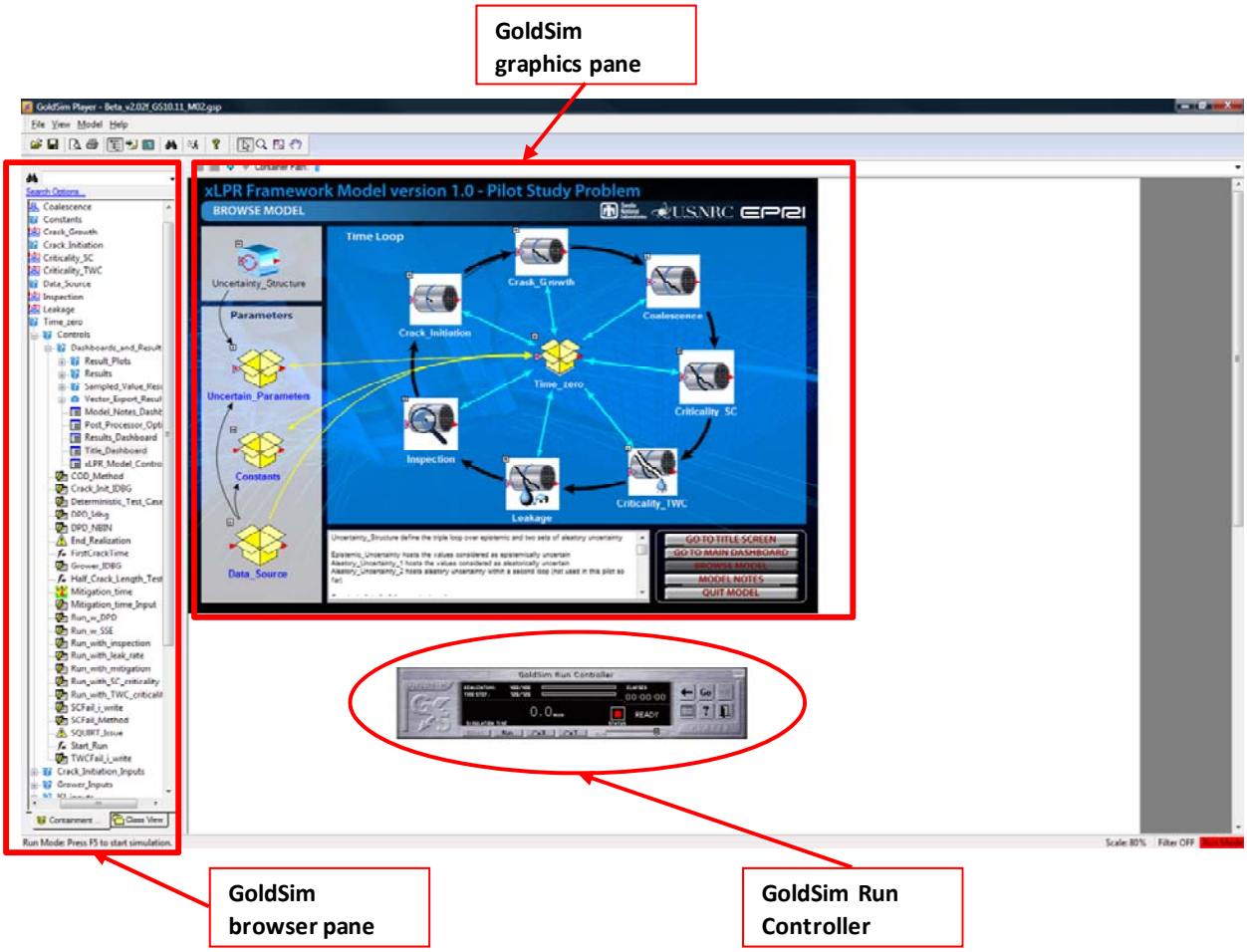


Figure 19. Framework Initial "Browse Model" Screen

3.2.3 Framework Software Settings

As installed, the default setting for exporting results is “Export results after simulation”. For simulations with a large number of realizations the export process can take a significant amount of time. It is recommended that this setting be changed to “Do not export results”. The steps for changing this setting are as follows:

(1) Open the GoldSim Player xLPR file

(2) Press “Alt+m”; a pop-up dialogue box will appear

(3) Select “Options…”

(4) In the “Automatic Export for Results Elements” drop-down box select “Do not export results”

Note that this setting is associated with the GoldSim Player software – not the specific player file. Once it has been made, it will be applied to all player files that are executed on that computer.

Changing this setting will require the user to manually export the results after a simulation has completed. Directions for manually exporting results are provided in Section 3.3.4 .

3.2.4 Input Data Excel Spreadsheet Version

The GoldSim functionality that allows an Excel spreadsheet to be dynamically linked with a GoldSim file or GoldSim Player file requires that the version of Excel in which the spreadsheet was created be installed on the computer on which the GoldSim file or GoldSim Player file is run. The dynamically-linked Excel spreadsheet that is part of the xLPR framework was created in Excel 2007.

To accommodate an older version of Excel (e.g., Excel 97, Excel 2000, Excel 2002, Excel 2003) the user will have to convert the Excel 2007 file to the installed Excel version. The framework file is then changed to use the converted file rather than the Excel 2007 version. The directions for changing the framework to use the converted Excel file are as follows:

(1) Open the GoldSim Player xLPR file

(2) Click the **START** button; the Main Dashboard will appear.

(3) If the model is in “Results” mode (i.e., the player file has been run and contains results) it must first be reset. Click the **Reset** button on the GoldSim controller; click **Yes** to reset the model.

(4) Under "Model Input Data", click on the **Select New EXCEL File** button; a browser window will appear.

(5) Browse to the converted Excel file, select it, and click on the **OK** button.

(6) Under "Model Input Data", click on the **Select New EXCEL File (DPD)** button; a browser window will appear.

(7) Browse to the converted Excel file, select it, and click on the **OK** button.

3.2.5 Framework Options and Simulation Settings

The Main Dashboard screen contains a section (Model Options and Simulation Setting) that allows the user to change certain features of the framework. Note that changes cannot be made to a model with saved results.

Probabilistic/Deterministic Pull-Down: This pull-down menu is only to be used in the deterministic case framework to switch between **Deterministic Case #1** and **Deterministic Case #2**. It is not to be used to change between the deterministic cases and the probabilistic case[3].

Simulation Settings: Clicking on the **Simulation Settings** button opens the **Simulation Settings…** pop-up menu to one its tabs (see Figure 20). For the probabilistic framework the Monte Carlo tab allows the user to:

- change the total number of realizations (this value is the number of DPD realizations, or it must be consistent with the total number of Monte Carlo realizations ($N_e \; x \; N_a$)),
- change the number of histories to save (this value is the number of DPD realizations, or it must be consistent with the total number of Monte Carlo realizations ($N_e \; x \; N_a$)),
- run only a single specific realization,
- change between repeated and non-repeated sampling sequences, and
- change the random seed.

While there is a checkbox for selecting between Latin Hypercube Sampling (checked box) and Monte Carlo sampling (unchecked box) this feature does not work in the framework. Rather, the sampling method is defined within the framework based on the uncertain parameter type (e.g., epistemic or aleatory).

In the framework, uncertain parameters defined as epistemic are sampled using Latin Hypercube Sampling while those defined as aleatory are sampled using Monte Carlo sampling. The type can be changed by modifying the parameter in the input data spreadsheet (see Section 3.2.6).

[3] This functionality may be included in a future version of the framework.

Clicking on the **Time** tab allows the user to access the framework's time settings. Here, the user can change:

- the Time Display Units,
- the simulation duration, and
- the number of time steps and the plot (i.e., output) frequency.

The user can make no changes to the framework in either the **Globals** or **Information** tabs

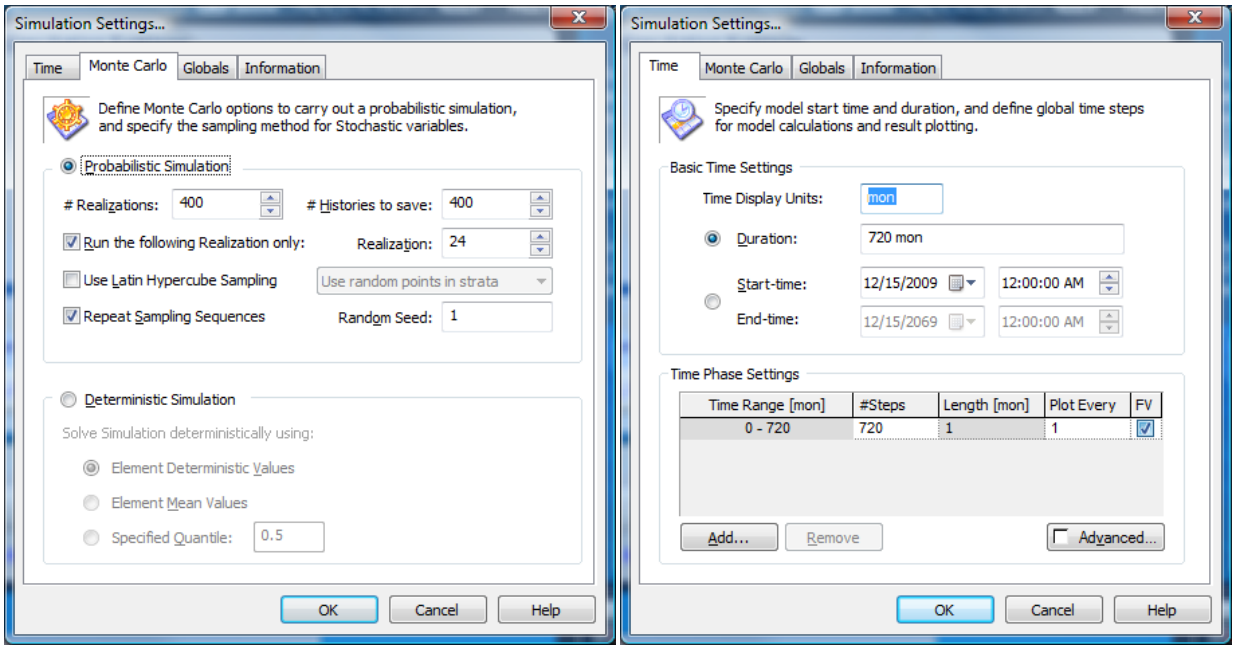


Figure 20. Framework Simulation Settings… Tabs

Epistemic and Aleatory: The two input boxes allow the user to specify the number of epistemic and aleatory realizations. Note that the product of these two values must match the total number of realizations specified in the **Simulation Settings…/Monte Carlo** pop-up dialogue box.

DPD/# DPD Bins: Checking the box causes the framework to use the DPD module to sample the uncertain parameters. The input box allows the user to specify the number of bins used by the DPD module. Note that when the DPD module is used, the number of realizations using the for the framework defined in the **Simulation Settings…** pop-up menu (see Figure 20) is what determines the number of samples that will be randomly pulled from the DPD generated sample matrix. For instance, a DPD sample matrix can be 100 bins, however the framework model can randomly sample the DPD matrix 2,000 times creating 2,000 potentially different realizations since each uncertain parameter value is pulled randomly from the 100 DPD bins. The number of aleatory and epistemic realizations (as specified on the Main Dashboard) is not used.

Crack Initiation: This pull-down menu allows the user to select between the three (Method I, Method II, and Weibull) crack initiation models.

Criticality – SC: Checking the box activates the SC criticality module. The pull-down menu allows the user to select whether SCs are modeled with semi-elliptical or constant depth geometry.

Criticality – TWC: Checking the box activates the TWC criticality module.

Inspection: Checking the box activates the inspection module.

Leak Rate: Checking the box activates the crack leak rate module.

Mitigation: Checking the box activates the crack mitigation module.

Run with SSE: Checking the box causes the framework to run with the effects of the safe shutdown earthquake included in the calculation.

Debug File: A debug file can be written for the following modules by checking its box.

- Crack Initiation
- Grower
- SCFail
- TWCFail
- DPD

3.2.6 Framework Input Data

Framework input is contained in an Excel file. That file has three worksheets

- Uncertain Parameters
- Constants
- DPD Inputs

The user can change any cell in the worksheets that is highlighted in yellow. The user cannot add or subtract input parameters, nor can the user change the distribution type (e.g., normal, uniform, etc.) of an uncertain parameter.

The input to the framework can be modified in two ways. The first is to edit the default Excel input data file. This can be accomplished within the framework by clicking on the **Edit Default EXCEL File** button. The second is to make a copy of the default Excel input data file, modify it, save it with a different name, and link the framework to the new Excel file. This is accomplished by:

(1) clicking the **Select New EXCEL File** button,

(2) using the pop-up browser window to select the new Excel file, and

(3) clicking the **Select New EXCEL File (DPD)** button.

(4) using the pop-up browser window to select the new Excel file

3.2.7 Running the Framework

The framework can be run by:

- clicking on the **Run** button on the GoldSim Run Controller,
- clicking on the **Run Model** button on the Main Dashboard Screen, and
- pressing Alt+m and selecting **Run Model** from the pop-up menu.

If the framework has been previously run, and hence contains results, a pop-up dialogue box will appear to let the user know that the existing results will be destroyed if the framework is rerun and asks if the user wants to run again (see Figure 21). At this point the user should choose accordingly.

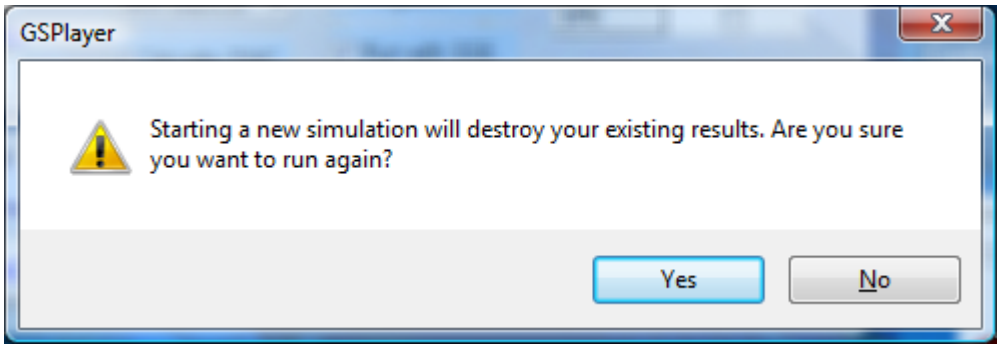


Figure 21. Dialogue Box Shown When Attempting to Run a Framework Player File that Contains Results

A run can be aborted by clicking on the **Abort** button on the GoldSim Run Controller. A pop-up dialogue box will appear to let the user choose whether to keep or discard the available results (see Figure 22). At this point the user should choose accordingly.

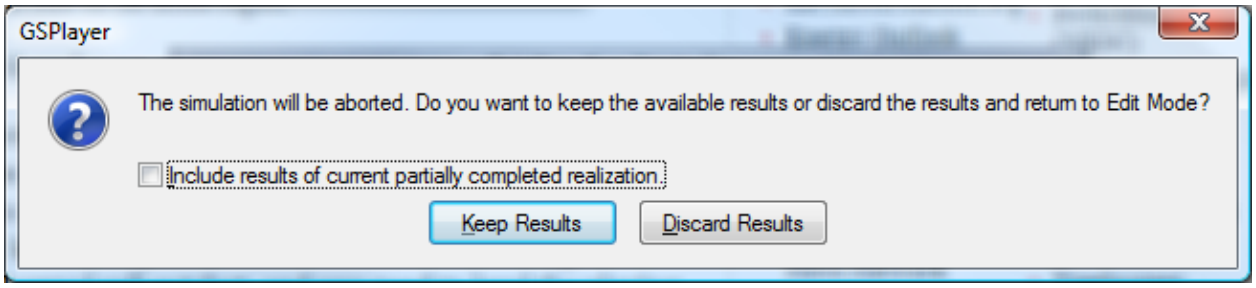


Figure 22. Dialogue Box Shown When Aborting a Running Framework Player File

A pop-up dialogue box appears when the framework has completed its run; the user should click on the **OK** button (see Figure 23).

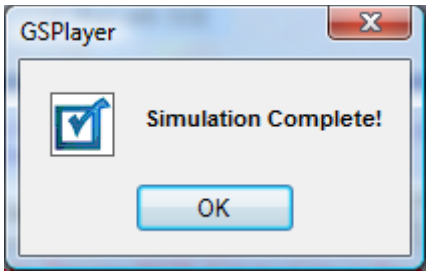


Figure 23. Dialogue Box Shown When a Framework Player File Run is Completed

3.3 Framework Results

Clicking on the **Click Here for Simulation Results** button in the Main Dashboard screen will take the user to the Results screen.

As with the Main Dashboard there is a set of buttons which allow the user to

- go to the Title Screen
- go to the Main Dashboard
- browse the model
- go to the Model Notes
- quit the model (i.e., close the model and shutdown the GoldSim Player software)

Buttons are provided that will open the Framework User's Guide and open (using the computer local computer system's default web browser) the xLPR CM Share Point site log-in screen.

In terms of results, this allows the user to monitor the simulation status as the framework runs, and when the run is completed view plots/tables of results and post-process results.

If the simulation was not started with the Results screen displayed, the user can pause the simulation (click on the "Pause" button on the GoldSim Run Controller), switch to the Results screen, then resume the simulation (click on the "Run" button on the GoldSim Run Controller).

3.3.1 Monitoring Results During a Model Run

Model results can be monitored by the user if the framework is run from the Results screen (e.g., the run is started by the user clicking on the GoldSim Run Controller Run button while the framework is on the Results screen) (see Figure 24). Note that the I/O overhead associated with displaying the Simulation Status results causes the framework to run slower.

Indicators are provided for:

- the number of cracks formed,
- the number of cracks coalesced,
- the number of SCs,
- the number of TWCs,
- the leakage rate ,
- if pipe rupture has occurred and at what time, and
- the maximum number of cracks that could occur.

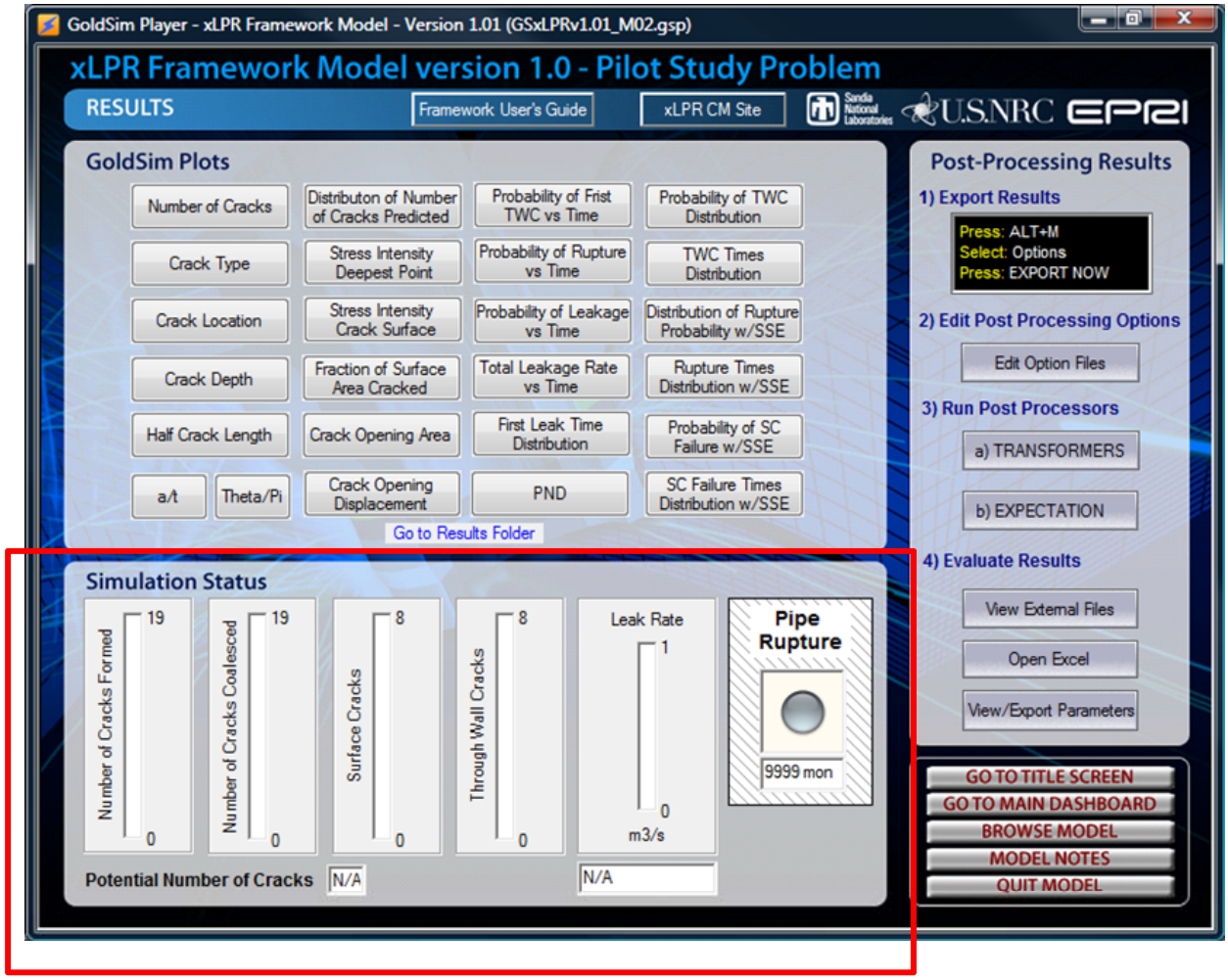


Figure 24. Simulation Status Indicators

3.3.2 Inspecting the Framework for Errors

The GoldSim Player software creates a run-log. The user should inspect the run-log for warning and/or error messages. The framework has been configured to write a message to the run-log

when a module has an error. The run-log can be accessed by pressing "Alt+m" and selecting **View Run Log** from the pop-up menu. The run-log will then be displayed in Notepad and it will be saved as a text file (GoldSim Run Log.txt) in the directory containing the framework GoldSim Player file.

There is an indicator box on the framework's Main Dashboard screen that when checked indicates at least one module had at least one error during the course of the run. Also, there are two buttons (**Module Status** and **Module Status Result Array**) that when clicked provide the user with plot/tabular results of error flag values for the following modules (see Figure 25):

- Crack Initiation
- Coalesce
- Grower
- SCFail
- TWCFail
- COD
- SQUIRT

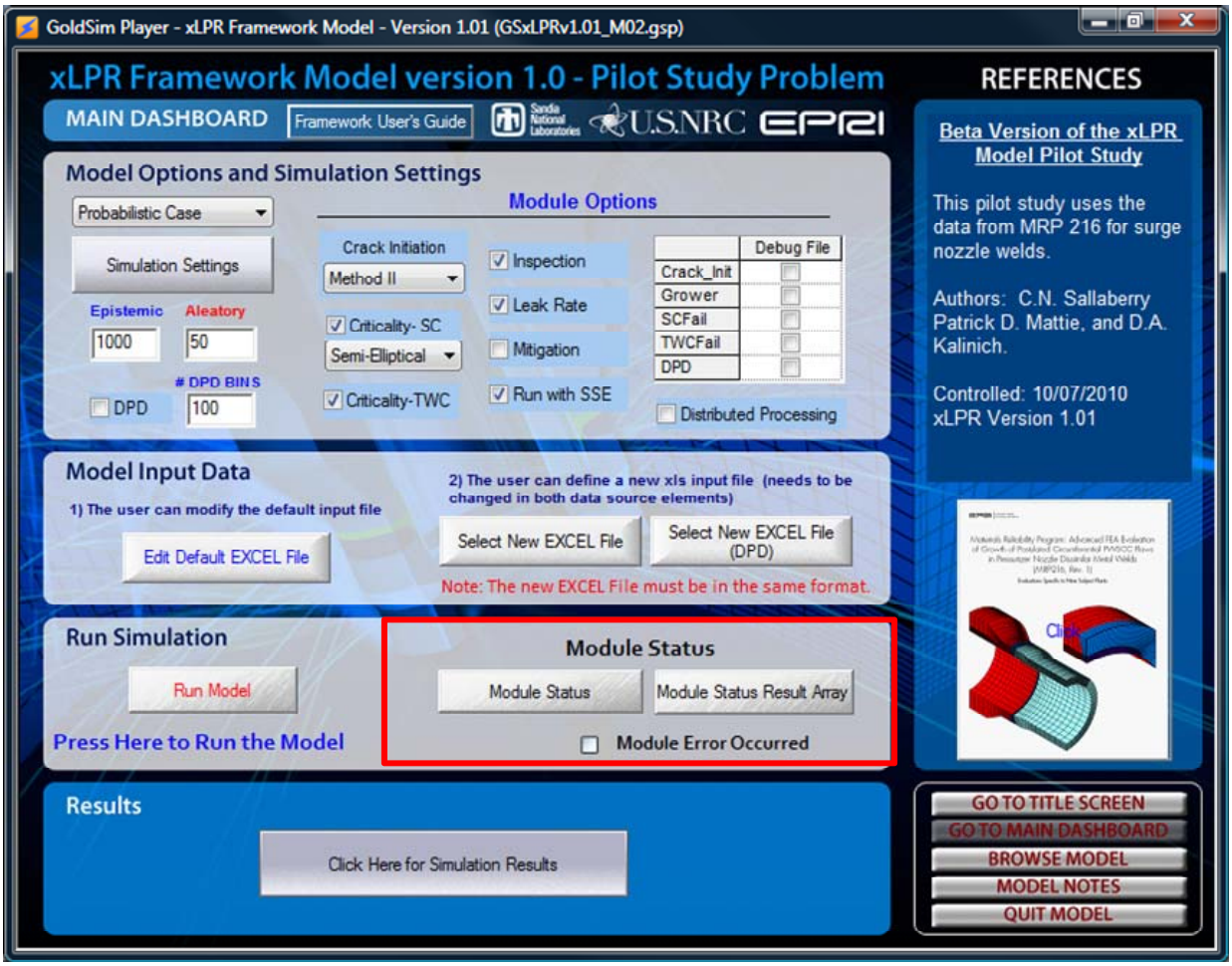


Figure 25. Module Status (Error) Indicators

No error is indicated by an error flag value of 0. Errors are indicated by non-zero error flag values.

Figure 26 shows the plot from the default xLPR probabilistic framework model that appears when the Module Status button is clicked on and realization 365 is selected.

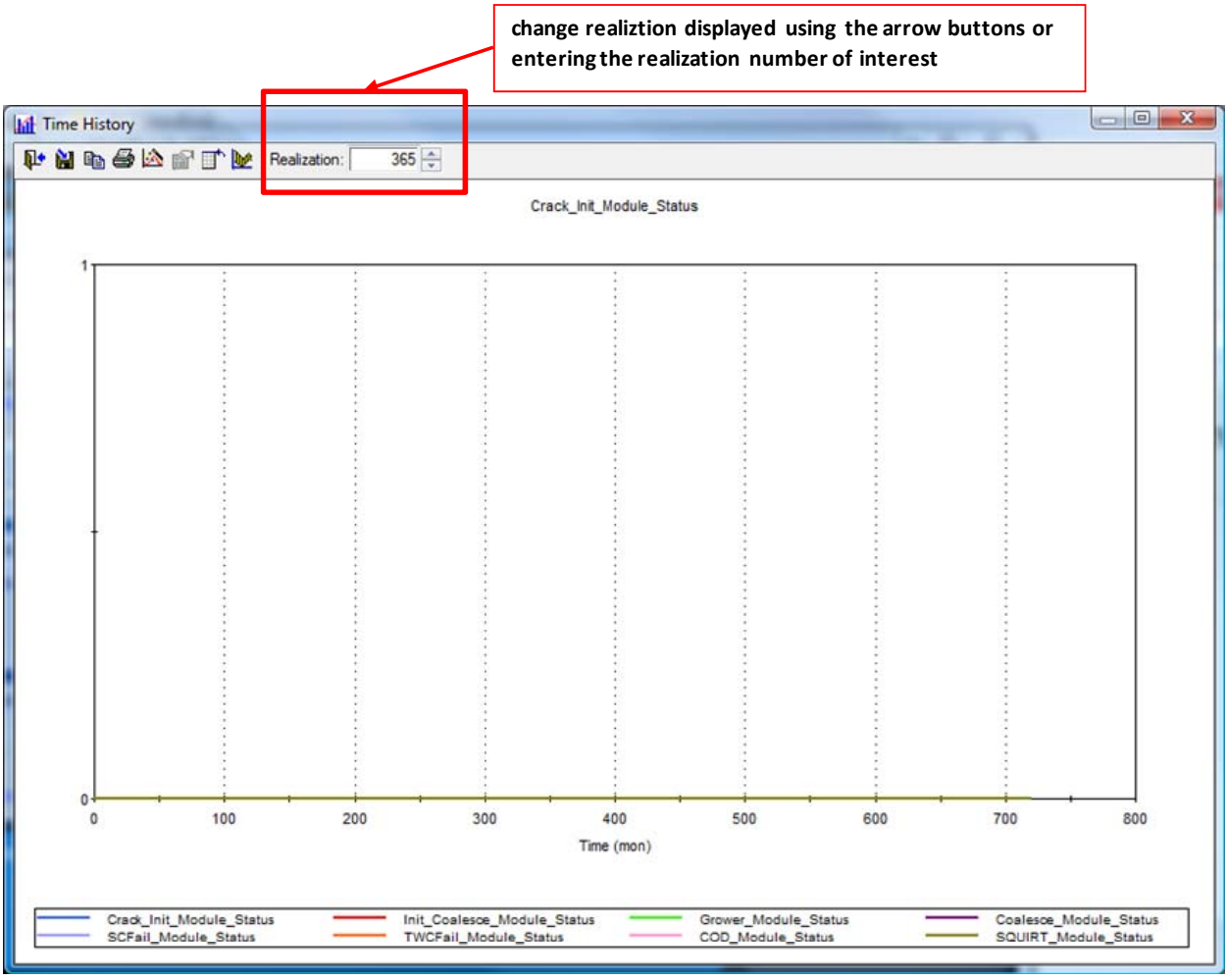


Figure 26. Time History Plot of Module Error Flags (Realization 365)

Figure 27 shows the plot from the default xLPR probabilistic framework model that appears when the Module Status Result Array button is clicked on. The flag shown is the first one in the array (Crack_Init_Module_Flag); the flags for the other modules can be displayed by selecting the module flag of interest from the pull-down menu (see Figure 28). Note that in this instance the PDF plot shows that all flag values are zero. Clicking on the Table View button (see Figure 28) will display the results for all of the module flags for all realizations (see Figure 29).

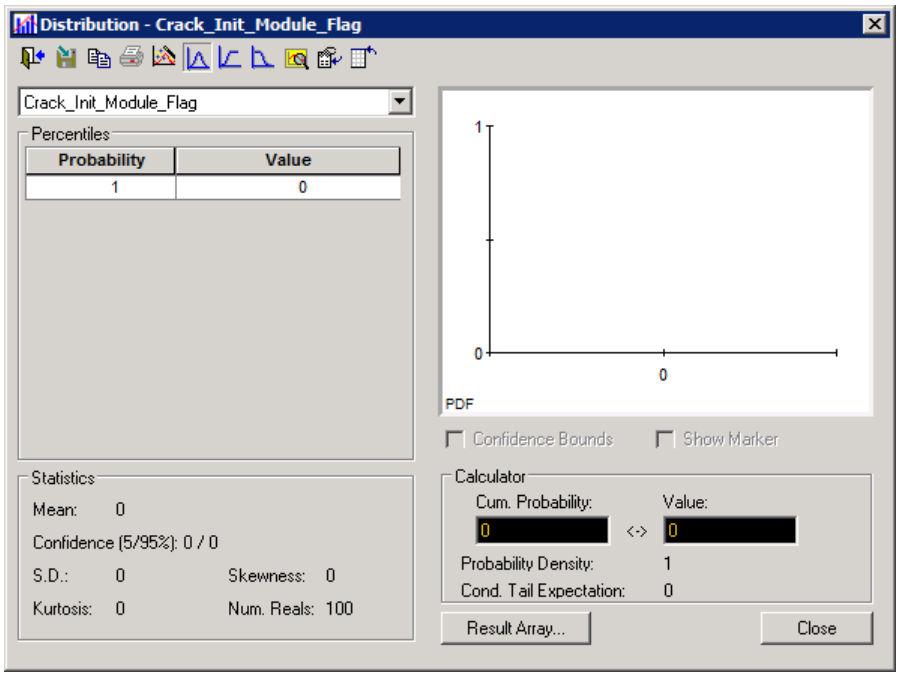


Figure 27. Module Status Results Array Distribution Plot

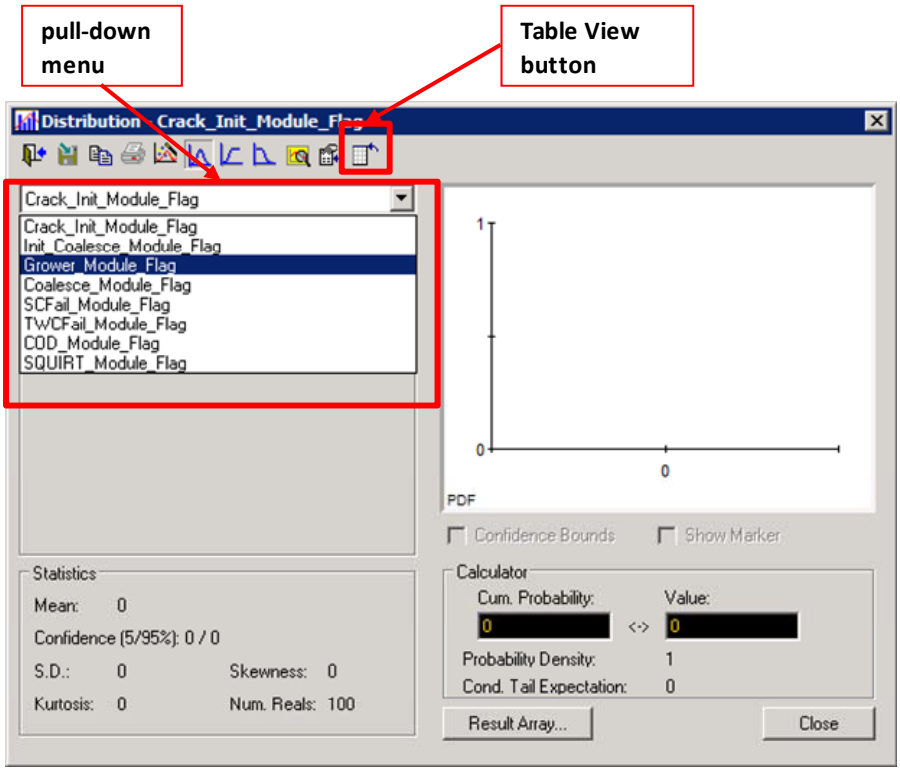
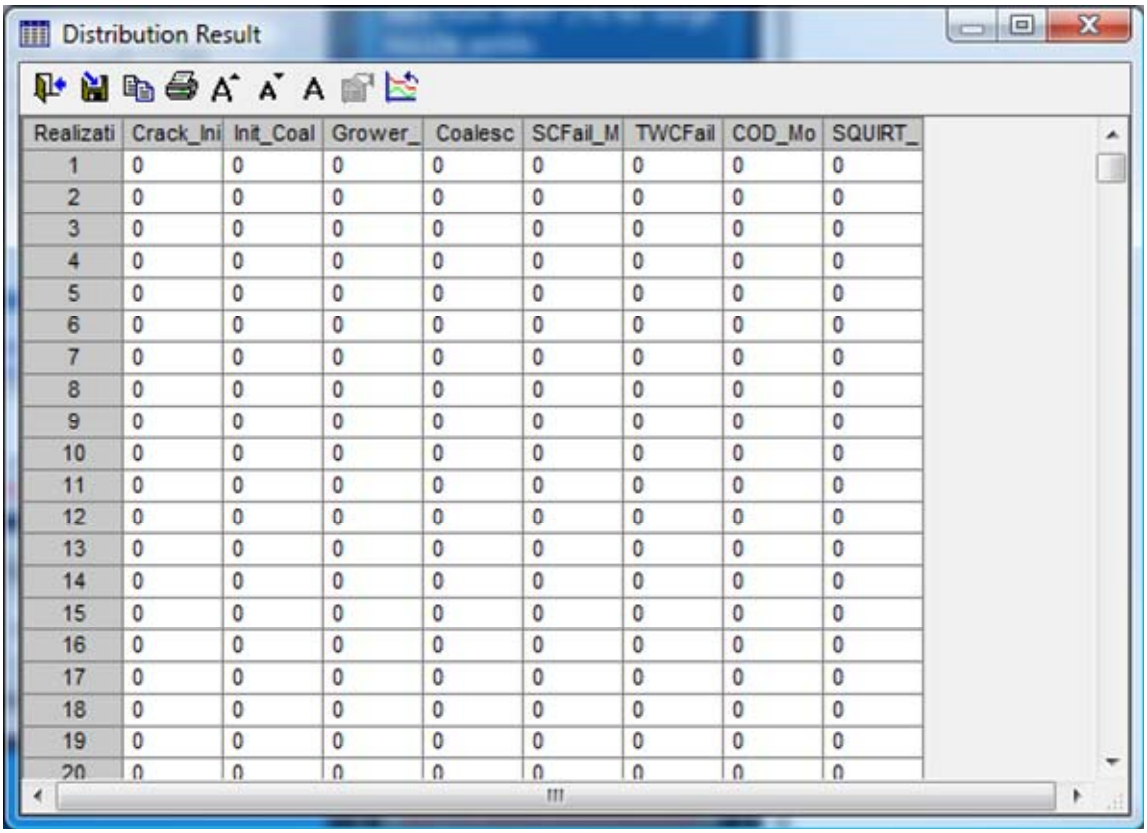


Figure 28. Module Status Results Array – Pull-Down Menu



Realizati	Crack_Ini	Init_Coal	Grower_	Coalesc	SCFail_M	TWCFail	COD_Mo	SQUIRT_
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0

Figure 29. Module Status Results Array – Table View

3.3.3 Browsing the Results

Plots/tables[4] of the results listed in Table 10 can be displayed by clicking on the GoldSim Plots buttons provided on the Results screen (see Figure 30). A link is also provided that will take the user to the results container within the framework model.

[4] See the GoldSim User's Guide, Chapter 8 [1] for details on the viewing features of a GoldSim Results element.

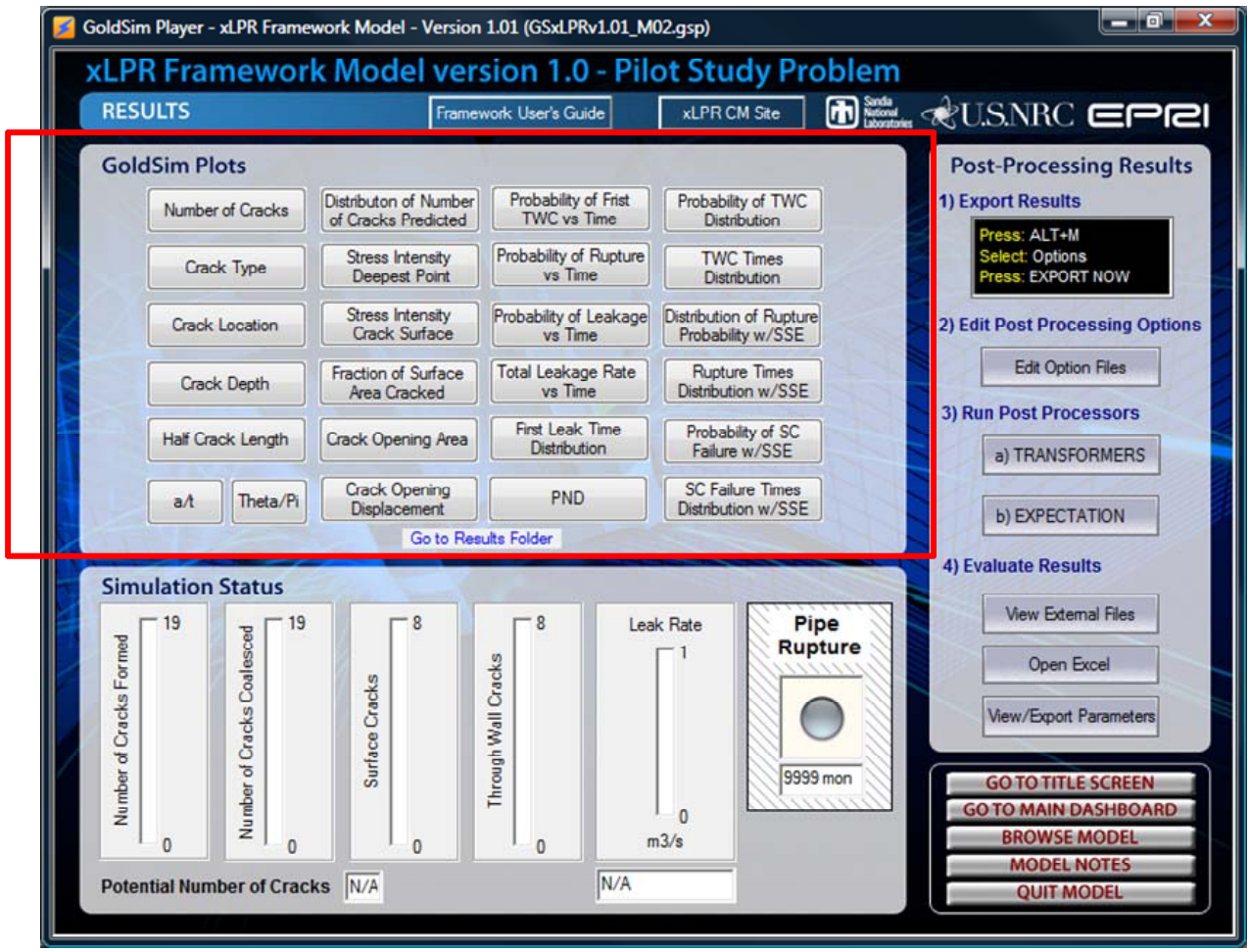


Figure 30. Results Plot Buttons

Table 10. xLPR Framework Results Plots/Tables

Number of Cracks	Probability of First TWC vs. Time
Crack Type	Probability of Rupture vs. Time
Crack Location	Probability of Leakage vs. Time
Crack Depth	Total Leakage Rate vs. Time
Half Crack Length	First Leak Time Distribution
a/t	PND
Theta/Pi	Probability of TWC Distribution
Distribution of Number of Cracks Predicted	TWC Times Distribution
Stress Intensity Deepest Point	Distribution of Rupture Probability w/SSE
Stress Intensity Crack Surface	Rupture Times Distribution w/SSE
Fraction of Surface Area Cracked	Probability of SC Failure w/SSE
Crack Opening Area	SC Failure Times Distribution w/SSE
Crack Opening Displacement	

When a result button (in this example **Number of Cracks**) is clicked, a plot is displayed (see Figure 31). In this example the variables plotted are:

- number of cracks that could be initiated,
- number of cracks initiated,
- number of SCs,
- number of TWCs, and
- number of cracks that coalesced.

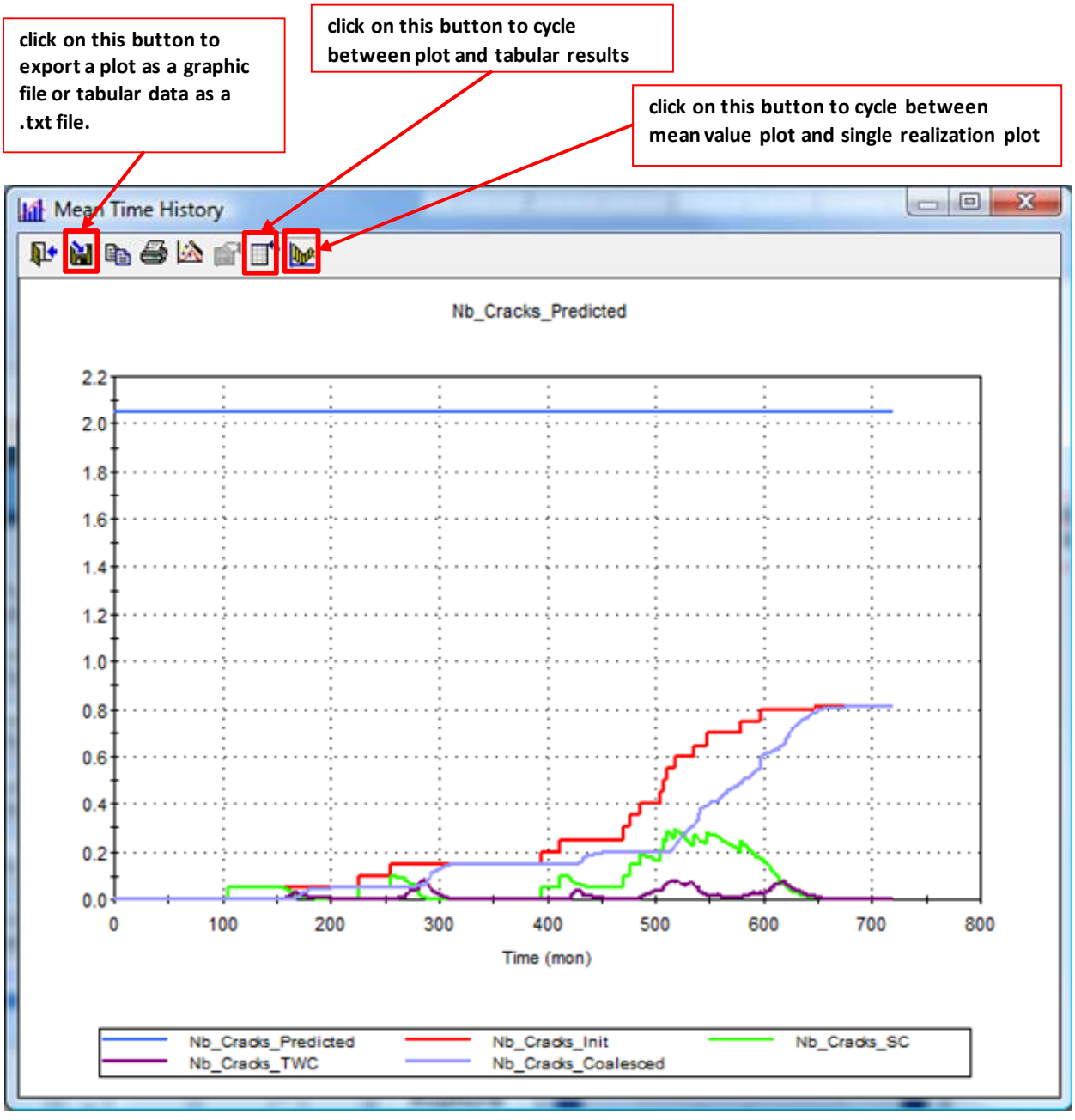


Figure 31. Example Result (Mean Value Plot)

This plot shows the mean time histories results. Clicking "off" the Show Mean Values button will change the plot to show the results for individual realizations (see Figure 32). Clicking on the Table View button will display the results as a table with numeric values (see Figure 33). As with the plot results, the tabular values can be cycled between mean results and single realization results (see Figure 34).

Clicking on the Export button (see Figure 31) allows the user to export the plot as a graphic file; if the results are displayed in the tabular format, clicking on the Export button (see Figure 34) allows the user to export the tabular values to a .txt ASCII file.

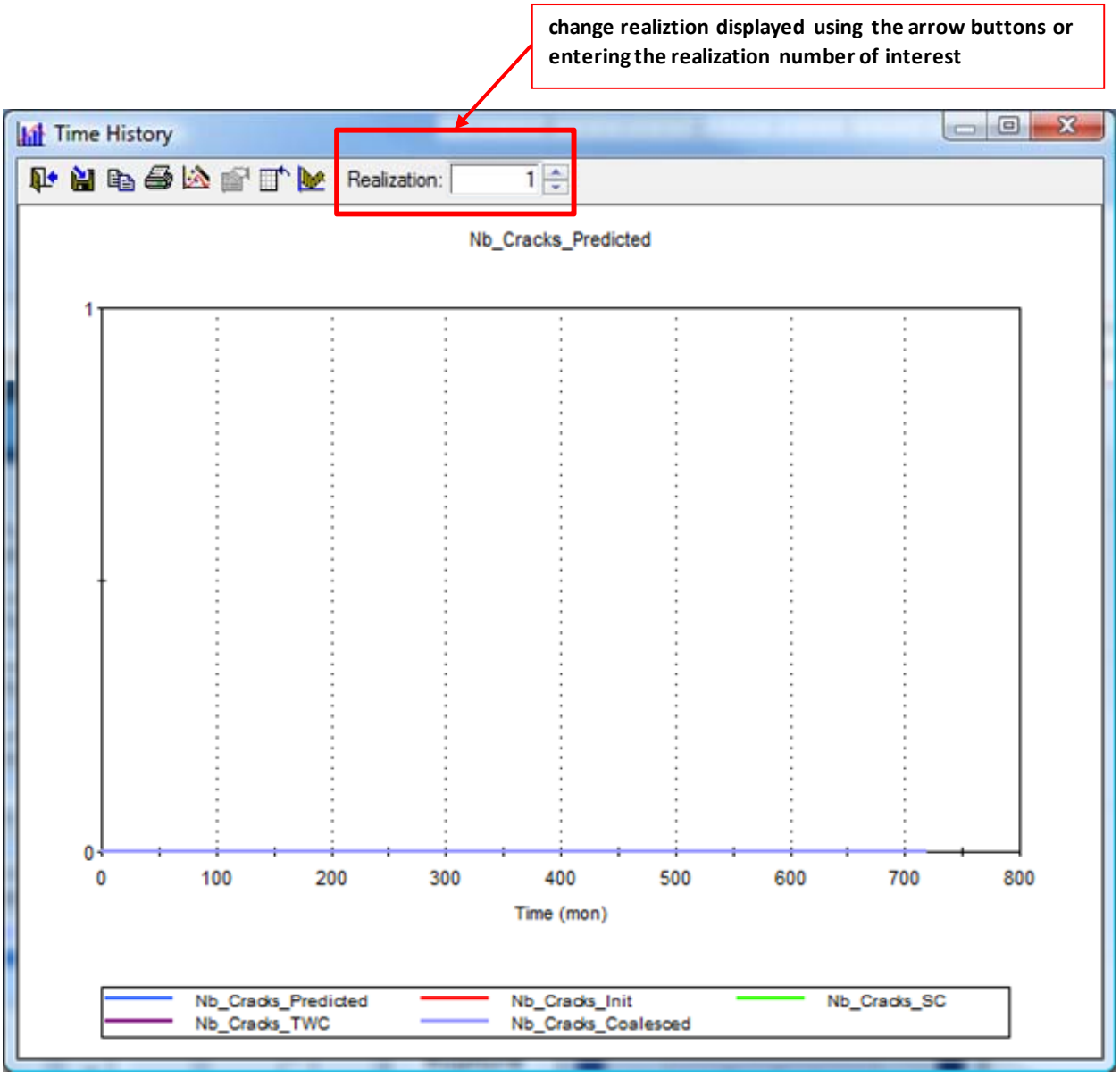
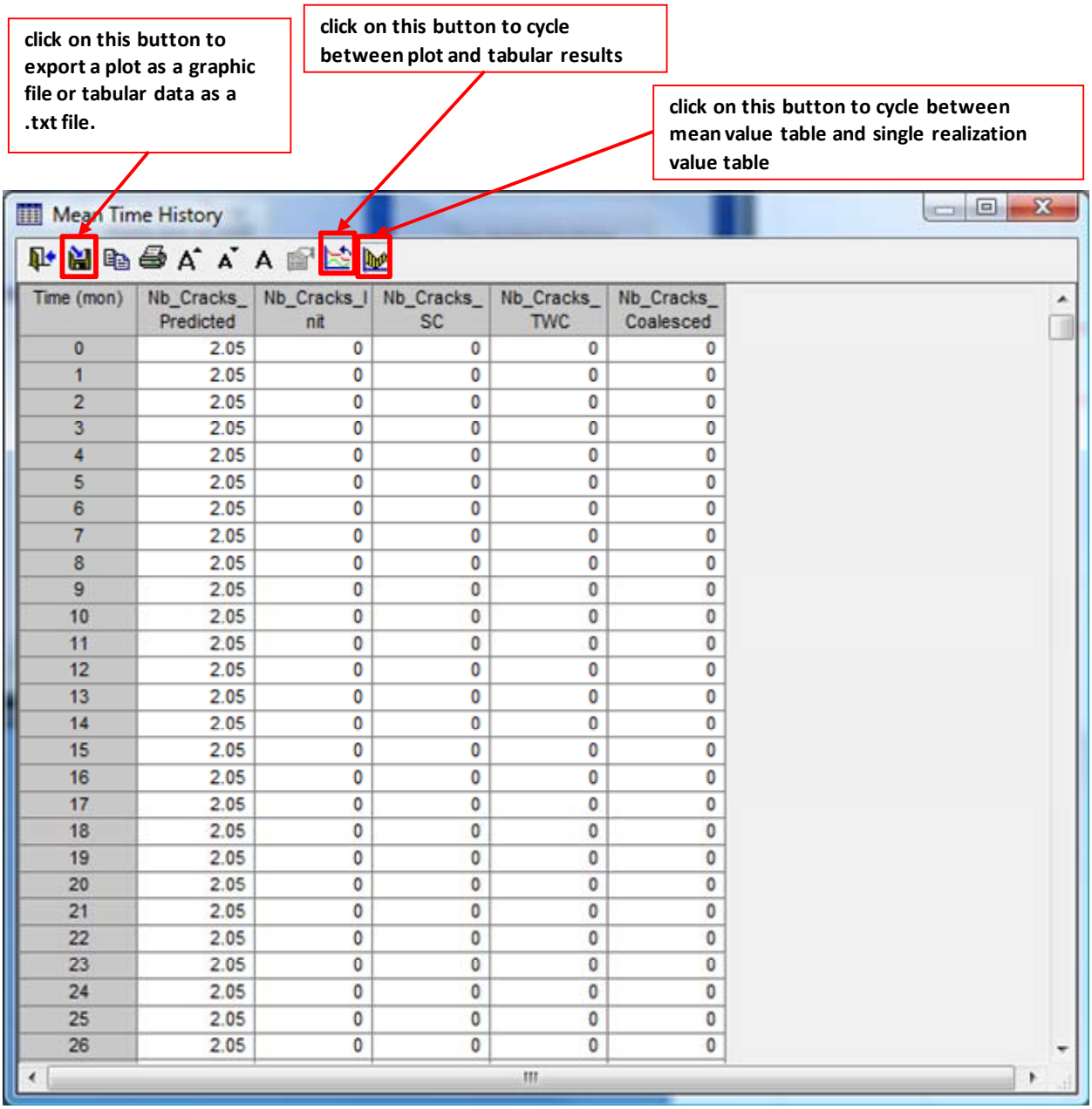
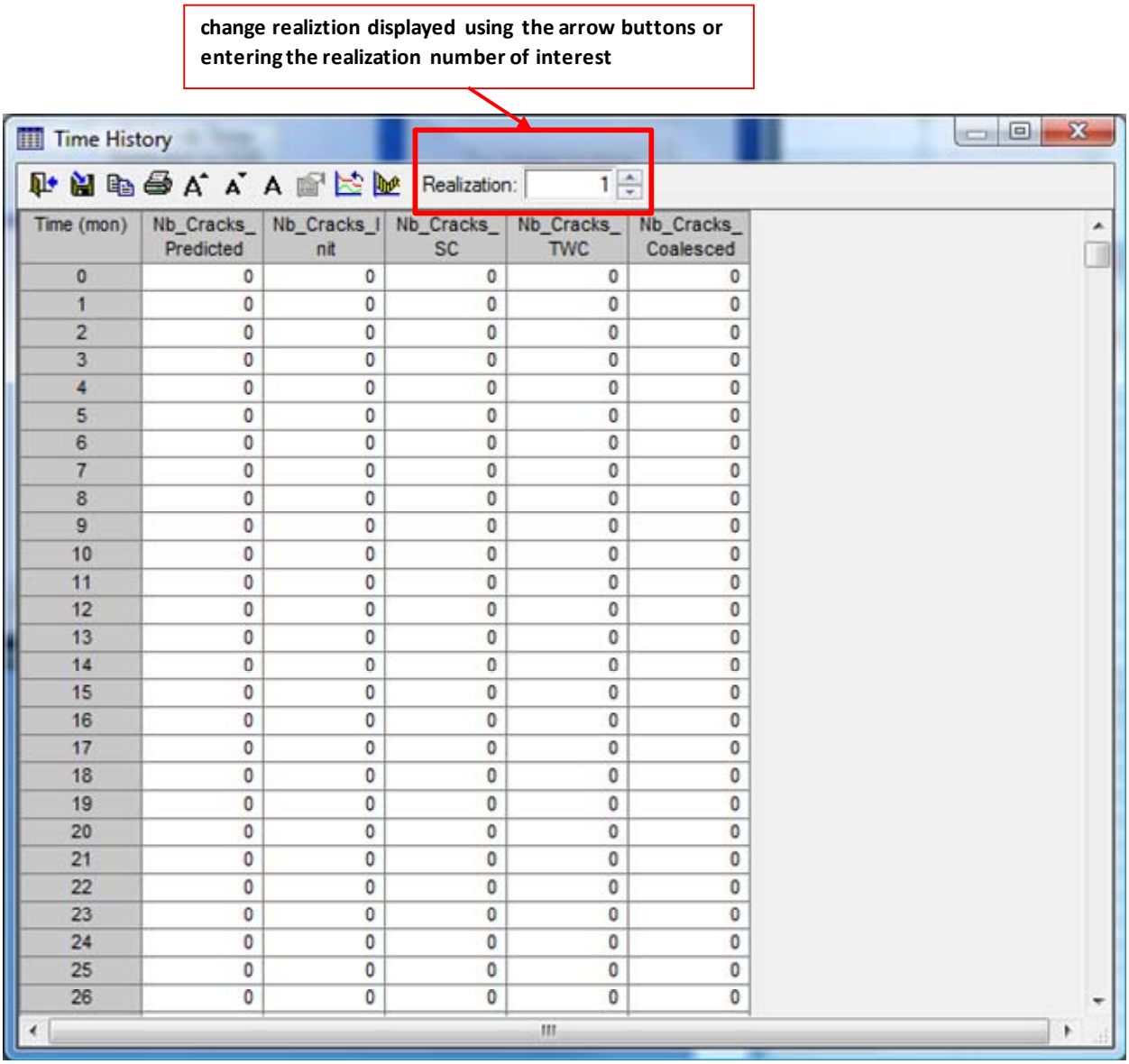


Figure 32. Example Result (Single Realization Plot)



Time (mon)	Nb_Cracks_ Predicted	Nb_Cracks_I nit	Nb_Cracks_ SC	Nb_Cracks_ TWC	Nb_Cracks_ Coalesced
0	2.05	0	0	0	0
1	2.05	0	0	0	0
2	2.05	0	0	0	0
3	2.05	0	0	0	0
4	2.05	0	0	0	0
5	2.05	0	0	0	0
6	2.05	0	0	0	0
7	2.05	0	0	0	0
8	2.05	0	0	0	0
9	2.05	0	0	0	0
10	2.05	0	0	0	0
11	2.05	0	0	0	0
12	2.05	0	0	0	0
13	2.05	0	0	0	0
14	2.05	0	0	0	0
15	2.05	0	0	0	0
16	2.05	0	0	0	0
17	2.05	0	0	0	0
18	2.05	0	0	0	0
19	2.05	0	0	0	0
20	2.05	0	0	0	0
21	2.05	0	0	0	0
22	2.05	0	0	0	0
23	2.05	0	0	0	0
24	2.05	0	0	0	0
25	2.05	0	0	0	0
26	2.05	0	0	0	0

Figure 33. Example Result (Mean Value Table)



Time (mon)	Nb_Cracks_Predicted	Nb_Cracks_Init	Nb_Cracks_SC	Nb_Cracks_TWC	Nb_Cracks_Coalesced
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	0	0

Figure 34. Example Result (Single Realization Value Table)

Note that the saved results of any GoldSim element (i.e., has a green output port (see Figure 35)) can be displayed by left-clicking on the output port. This causes the output interface to pop-up; this interface displays the outputs of the element. Right-clicking on an output causes a pop-up dialogue box to appear; selecting Time Histories… will cause a plot of the output to be displayed.

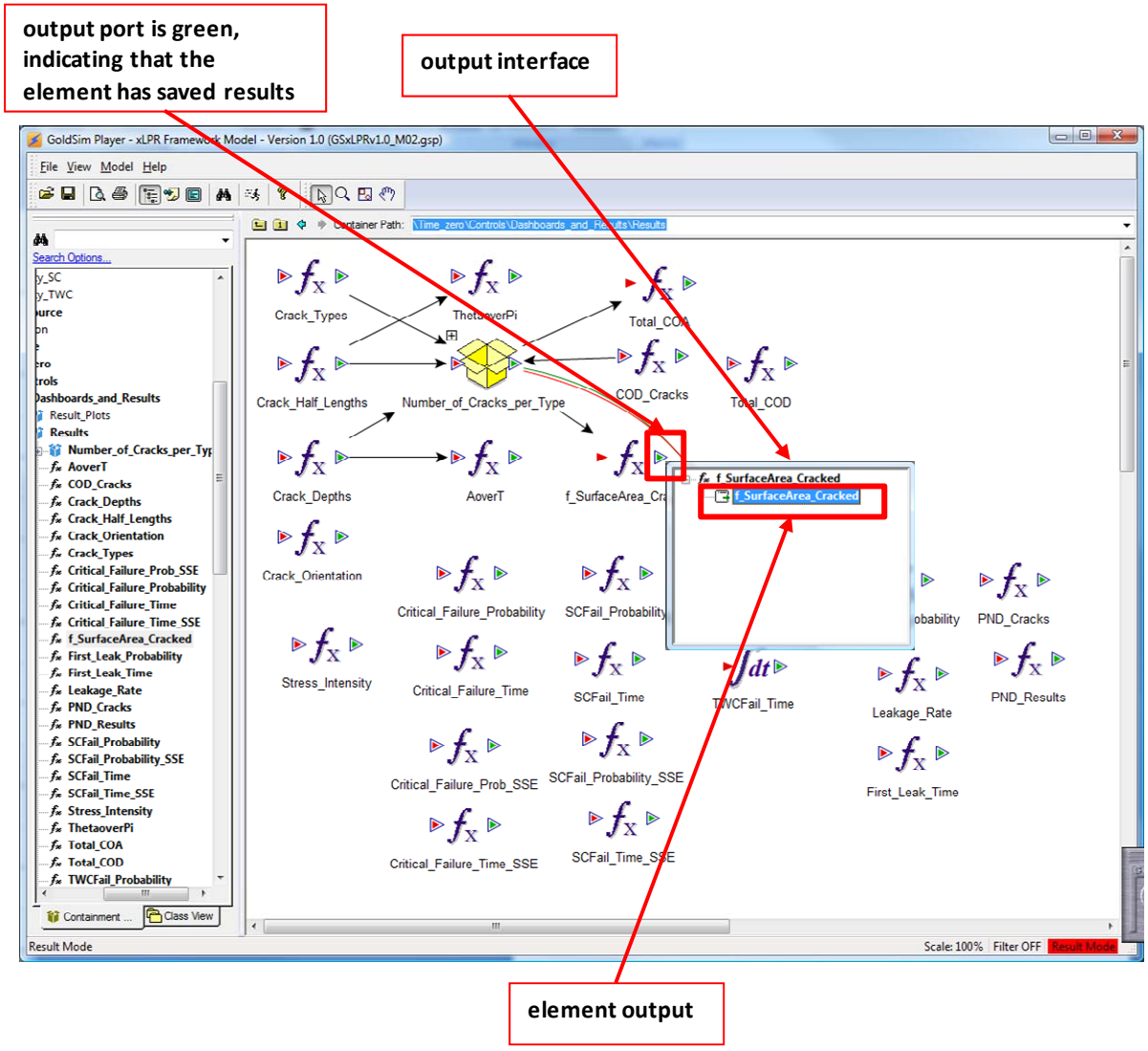


Figure 35. Displaying Element Output Example

The user can also screen which realizations are displayed in a plot or table (see Figure 36).

- press “Alt-m”
- select Screen Realizations… from the pop-up dialogue box
- select/deselect realizations as desired

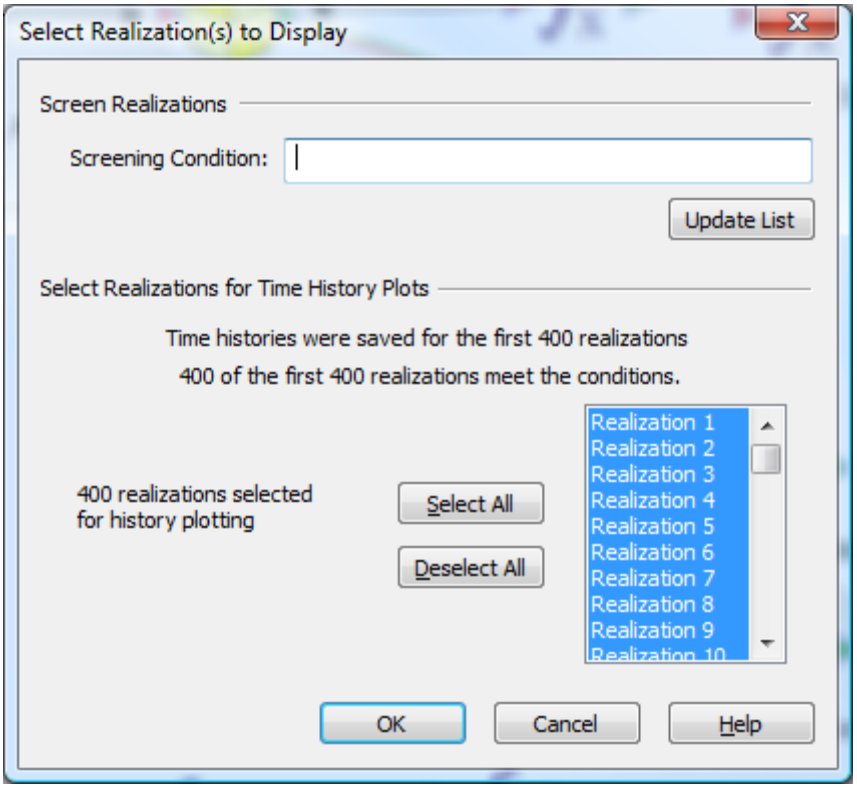


Figure 36. Screen Realizations Pop-Up Dialogue Box

A detailed discussion on displaying results (including details on modifying plots) is provided in both the GoldSim Player User’s Guide [2] and Chapter 8 of the GoldSim User’s Guide [1].

3.3.4 Exporting and Post-Processing Results

After the framework has been run, the post-processing executables TRANSFORMERS and EXPECTATION can be run (see Figure 37). This is accomplished by exporting the results from the framework into a set of ASCII files and running TRANSFORMERS and EXPECTATION (in that order) to produce their respective sets of output files.

Results are exported from the framework into a set of ASCII files by:

(1) pressing “Alt+m”; a pop-up dialogue box will appear,

(2) selecting “Options…”, and

(3) clicking on the **Export Now** button.

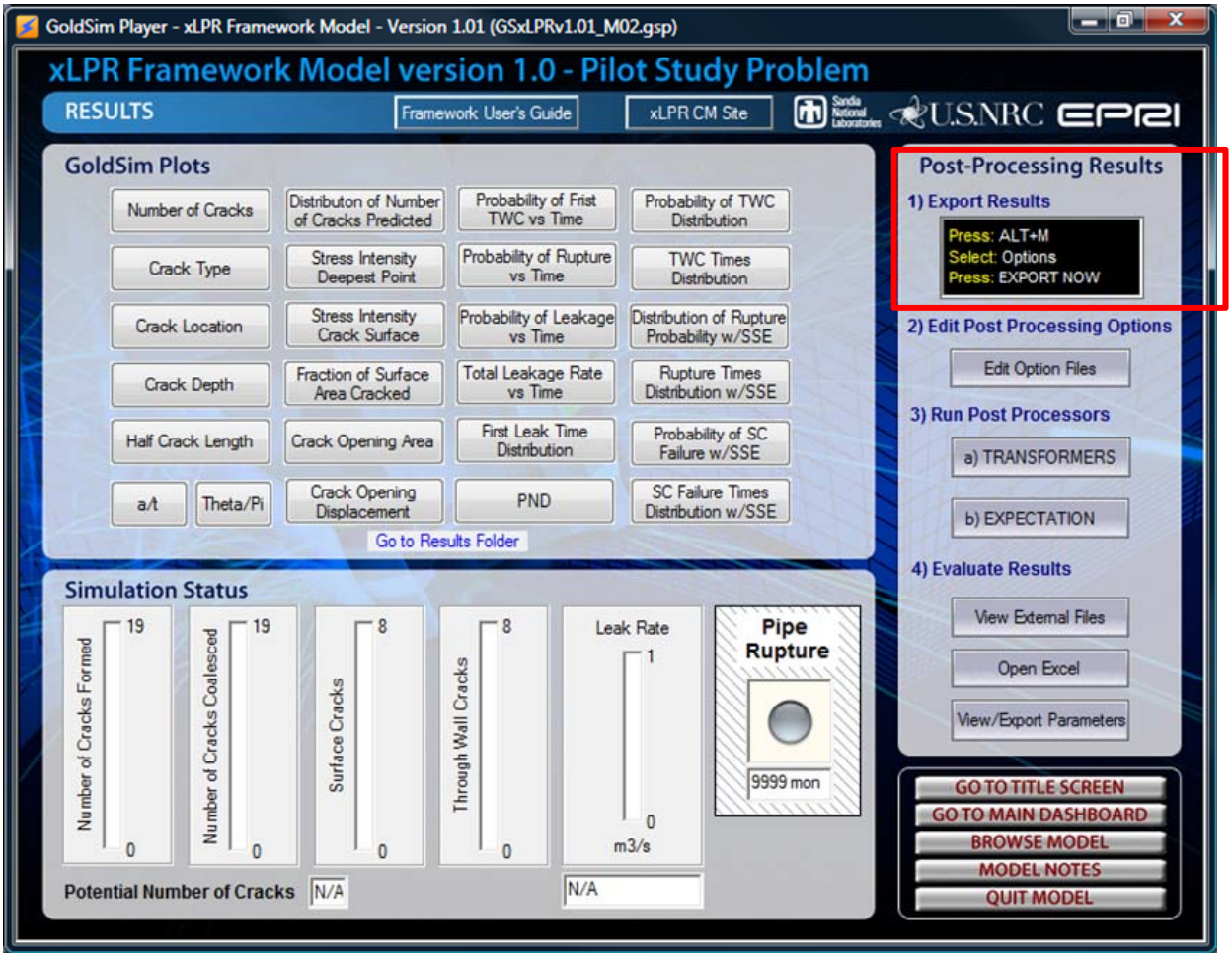


Figure 37. Exporting Results Instructions

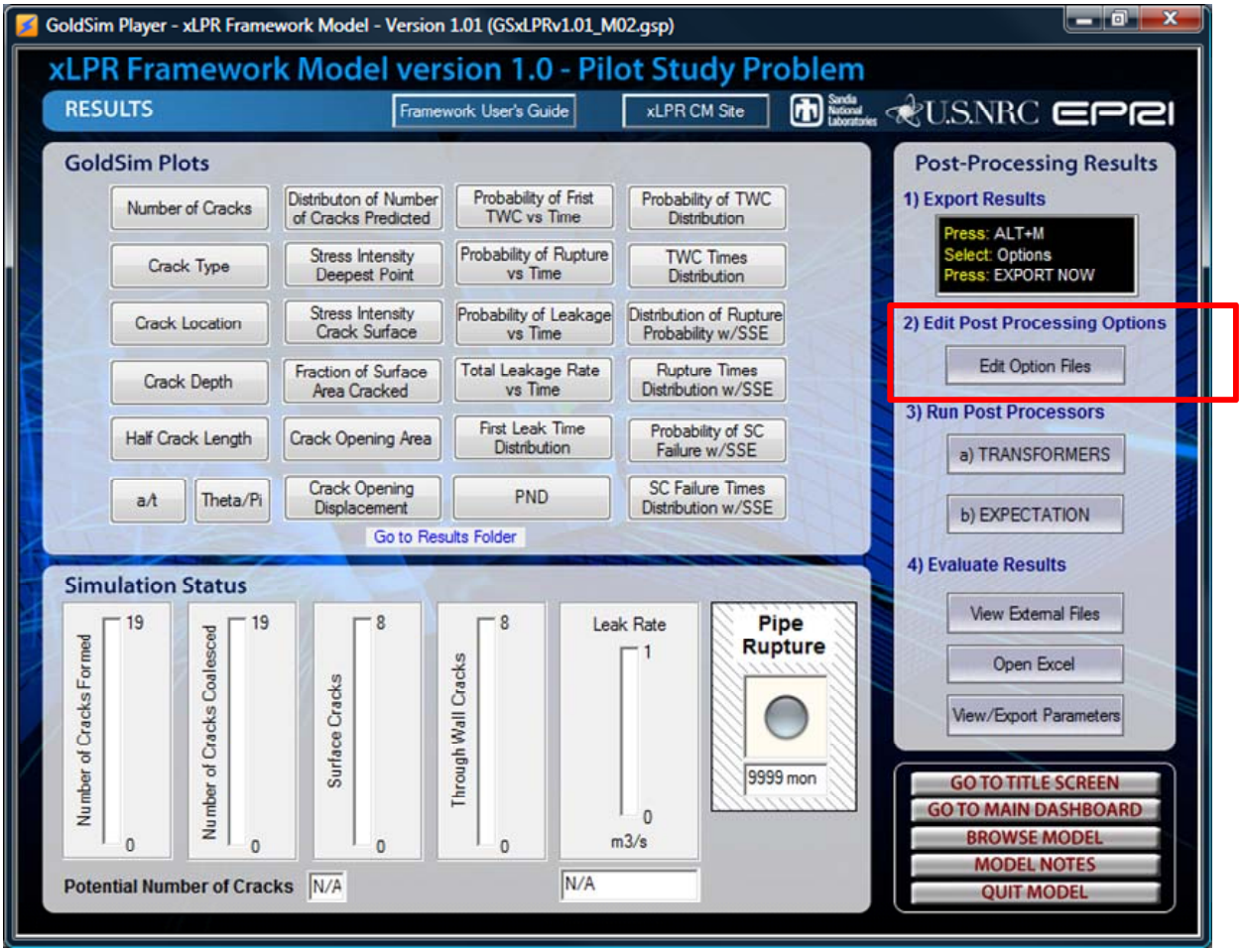


Figure 38. Edit Option Files Button

If the framework has been changed from its default configuration, the user will need to make the appropriate changes to the post-processor input control files (see Section 2.2.6). The user can access these files by clicking on the Edit Option Files (Figure 38), which will take the user to the Post Processing Options screen (see Figure 39). On that screen there is a button for each input control file. Clicking on a file's button will open that file in Notepad, allowing the user to change the file and save it.

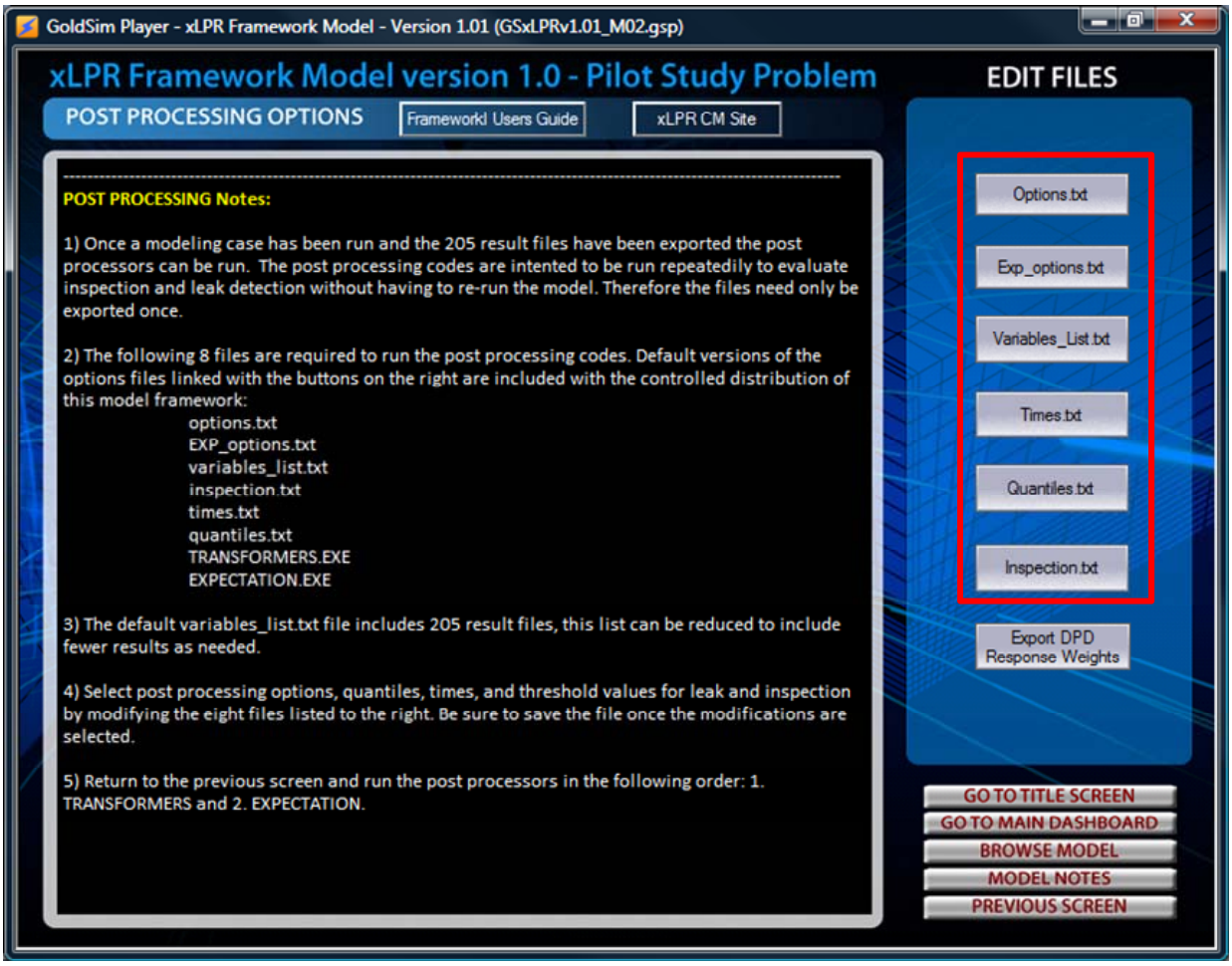


Figure 39. Post-Processing Options Screen -- Edit Files Buttons

In addition, for analysis using the DPD module, the user must export the DPD response weights, by clicking the "Export DPD Response Weights" button on the Post Processing Options screen (see Figure 40) and saving the table as an ASCII text file. This file is required if using the DPD module with importance sampling. The exported file name should be identified in the EXP_Options.txt file as described by input #6 of Table 7 (Section 2.2.6).

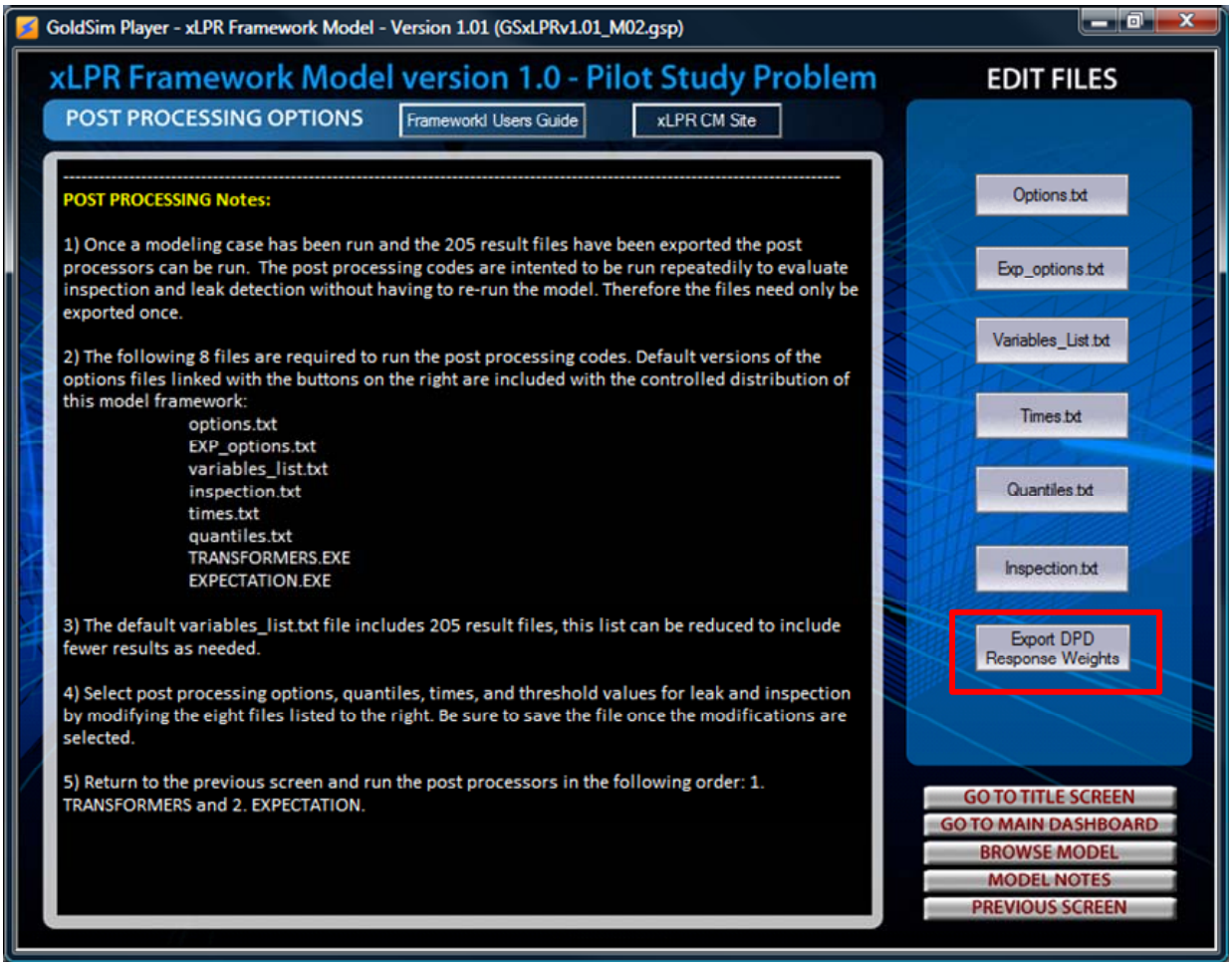


Figure 40. Post-Processing Options Screen -- Export DPD Response Weights Button

Once the results have been exported, the user can run TRANSFORMERS and EXPECTATION (in that order) by clicking on their respective buttons on the Results screen (see Figure 41).

Example input is provided in Appendix E.

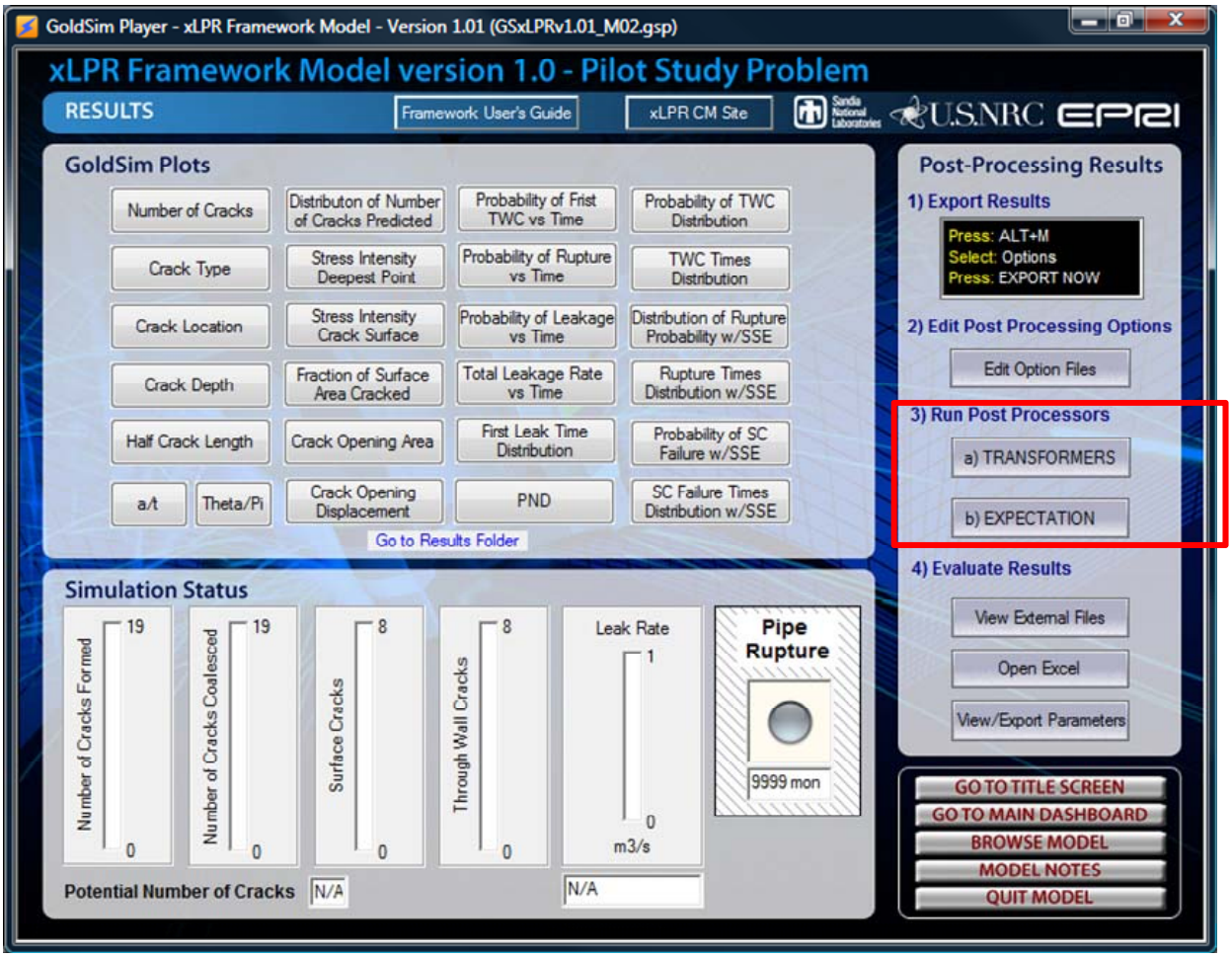


Figure 41. Run Post Processors Buttons

Figure 42 depicts the options available on the results dashboard that can be used to evaluate the results. The View External Files button opens Windows Explorer to the directory containing the framework. The Open Excel button opens Excel. The View/Export Parameters button takes the user to the container in the framework model that contains the GoldSim Results elements corresponding to plots of the uncertain parameters sampled in the associated analysis. These results display the sampled values for all of the model's uncertain parameters and can be exported as ASCII text files for parameter uncertainty studies.

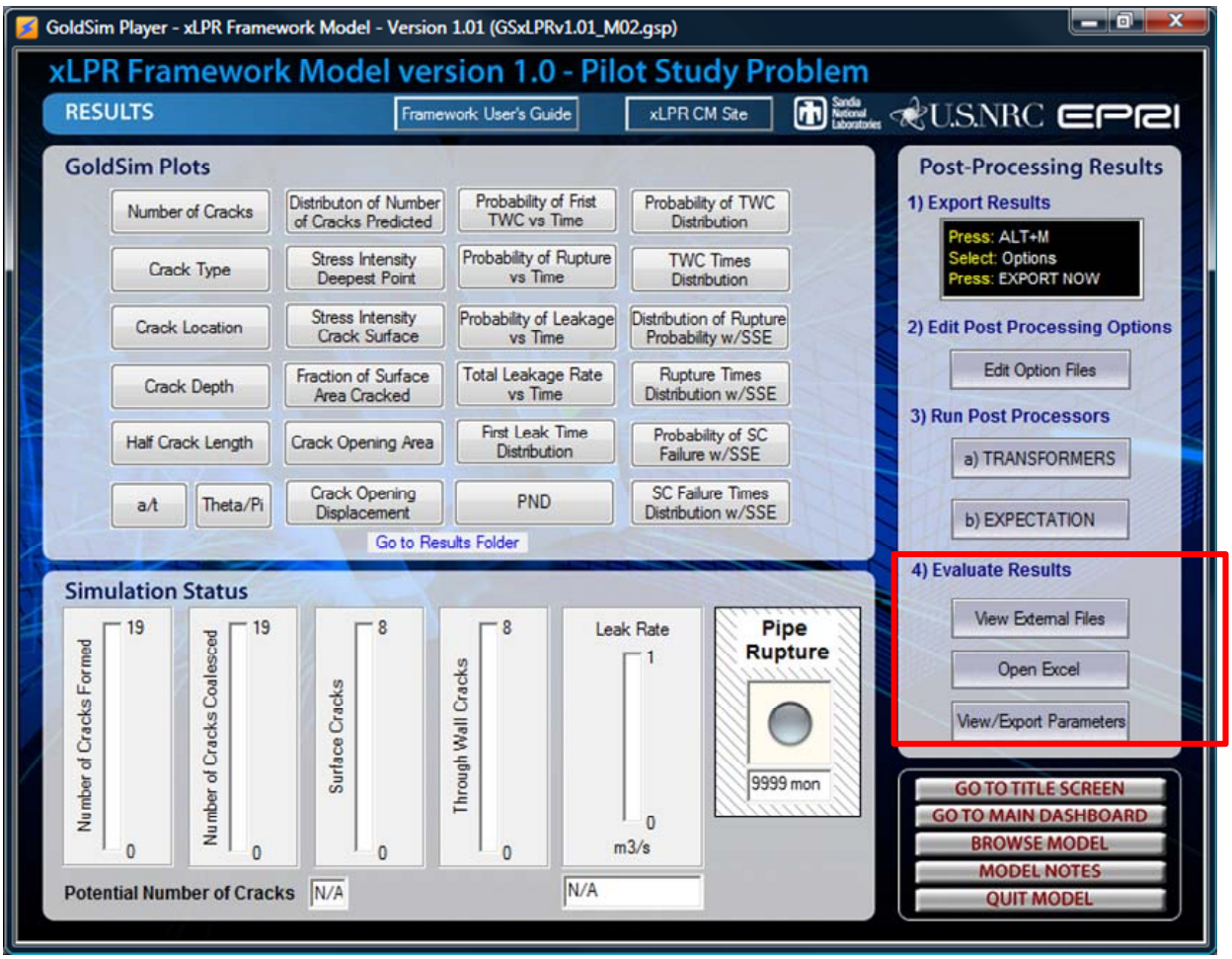


Figure 42. Evaluate Results Buttons

4 CONCLUSIONS

The user is encouraged to download a review a copy of the GoldSim software User's Guide. This intent of this document is to provide guidance on the features of Version 1.0 of the xLPR Framework model. The GoldSim software is a self-documenting code. The user should navigate the model file using the GoldSim software features to trace the model calculations and model logic to thoroughly understand the implementation. Since this is a commercial software package, there are many features that are not described herein that allow the user to screen and plot the results which provide a platform for troubleshooting, debugging, and analysis of model simulations. The .gsm model file is also included so advanced users with a licensed copy of the GoldSim software can modify the elements and capture results that have not be saved as default in the player version.

The framework was developed under the configuration management process developed for the xLPR pilot study. This is a prototype model which meets the requirements of the pilot program and is not expected to have a life cycle of greater than a few years.

5 REFERENCES

1. GoldSim Technology Group LLC, 2009, *GoldSim User's Guide, Volumes 1 & 2, Version 10.0,* GoldSim Technology Group LLC, Issaquah, Washington.

2. GoldSim Technology Group LLC, 2009, *GoldSim Player User's Guide Version 10.0*, GoldSim Technology Group LLC, Issaquah, Washington.

3. Sandia National Laboratories, 2010, *Development, Analysis, and Evaluation of a Commercial Software Framework for the Study of Extremely Low Probability of Rupture (xLPR) Events at Nuclear Power Plants, (draft),* Sandia National Laboratories, Albuquerque, NM. SAND2010-8480.

4. U.S. NRC, 2010, *xLPR v.1.0 Models Report (draft),* U.S. Nuclear Regulatory Commission, Washington, D.C.

APPENDIX A: INSTALLATION TEST CASES

Deterministic test cases using constant input are used to verify the installation which can be also verified against a hand calculation (using EXCEL). The installation test cases verify that the xLPR model framework is operating as expected. Two deterministic analyses are included in the installation package as outlined below.

A.1 Deterministic Test Case #1

Deterministic Analysis #1: Single crack at t = 0 years, with no mitigation. The location of the crack is at the top of the weld (theta = 0 rad). The initial crack depth and half crack length are 0.0015 m and 0.003 m respectively. The input deck for this case is included with the controlled version of the inputs spreadsheet for xLPR.

Screen captures of the following results plots are provided in Figure A-1 through Figure A-7 for comparison to the results produced by the user-installed version of the case.

- Number of Cracks
- Crack Type
- Crack Location
- Crack Depth
- Crack Half Length
- Crack Opening Area
- PND

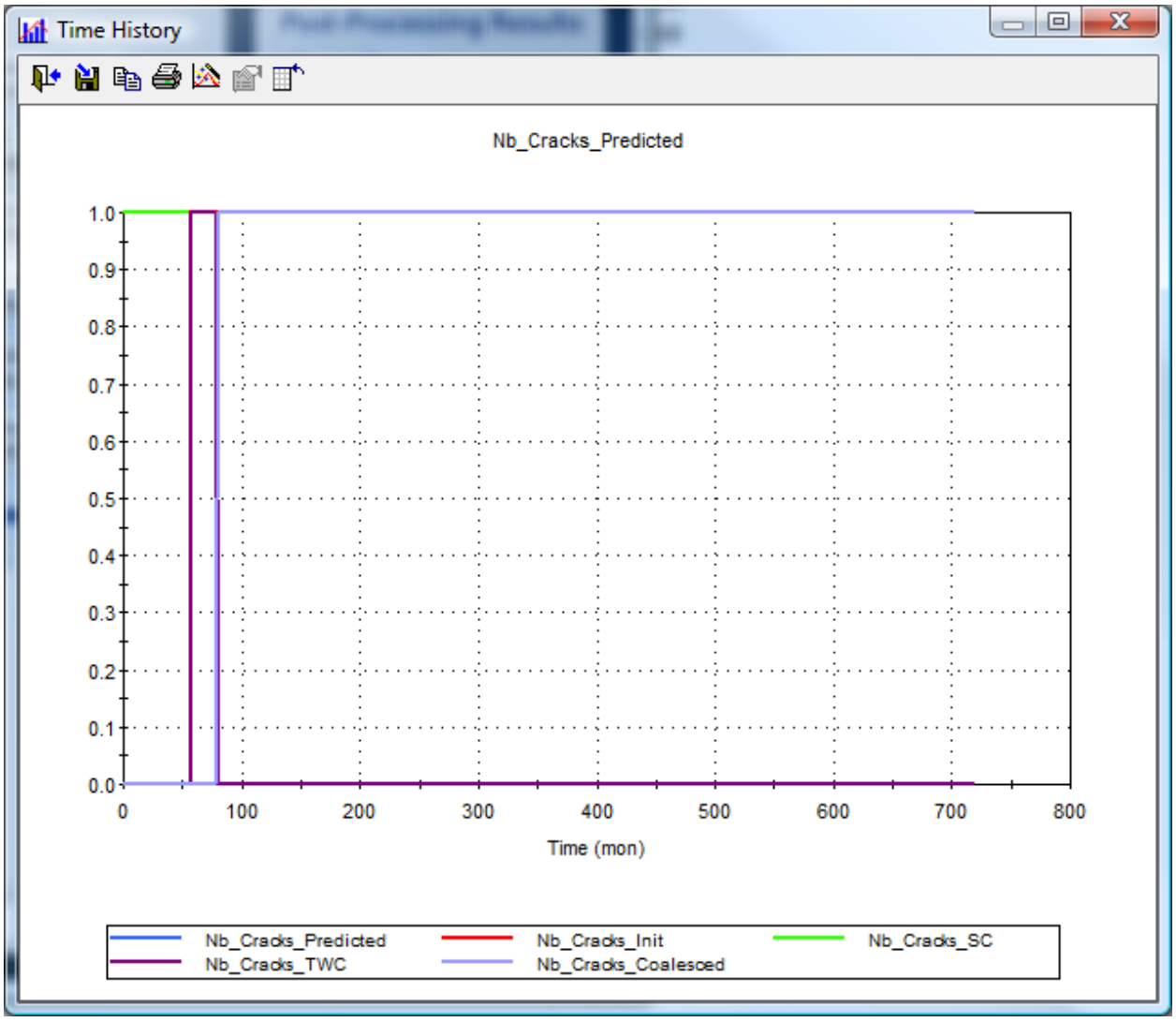


Figure A-1. Deterministic Test Case #1 -- Number of Cracks Results Plot

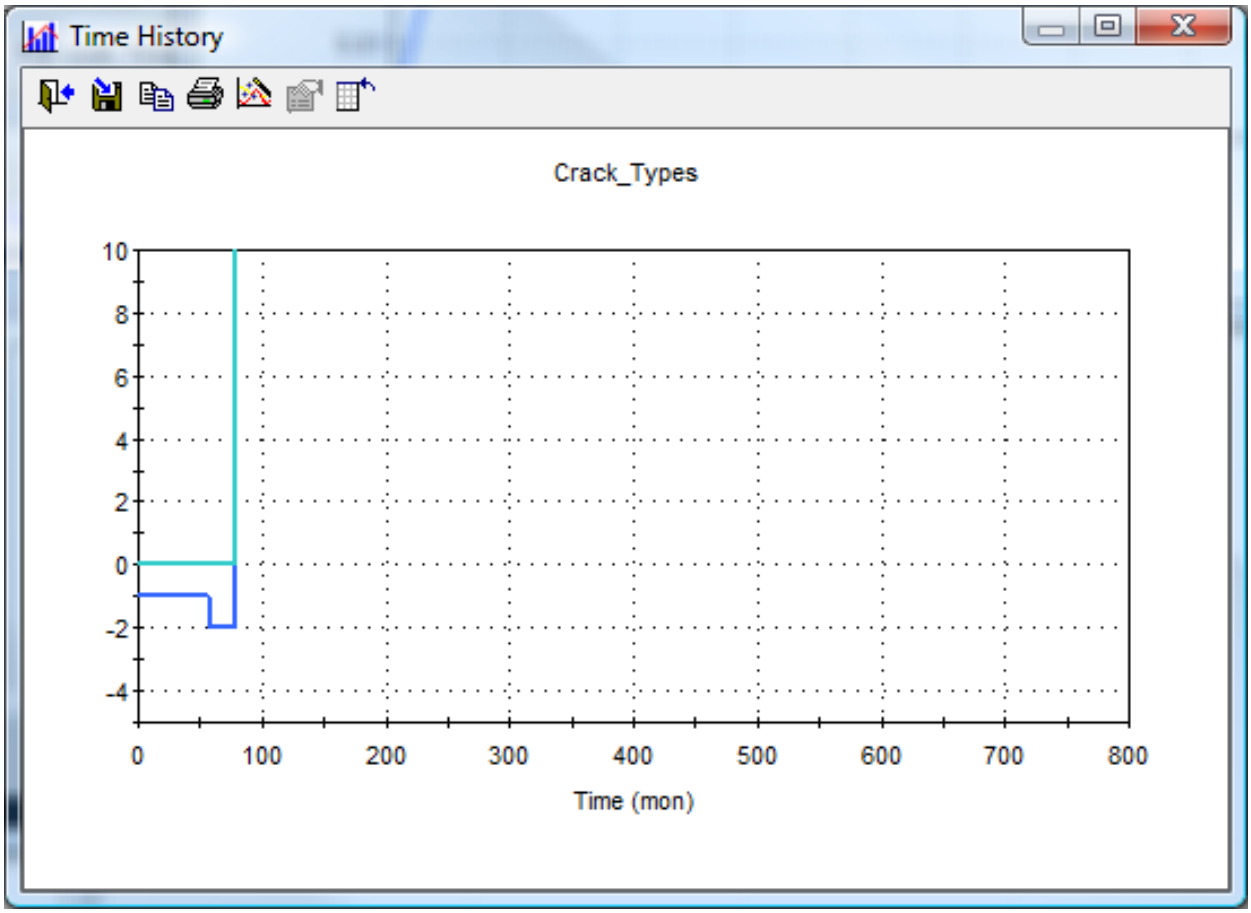


Figure A-2. Deterministic Test Case #1 -- Crack Type Results Plot

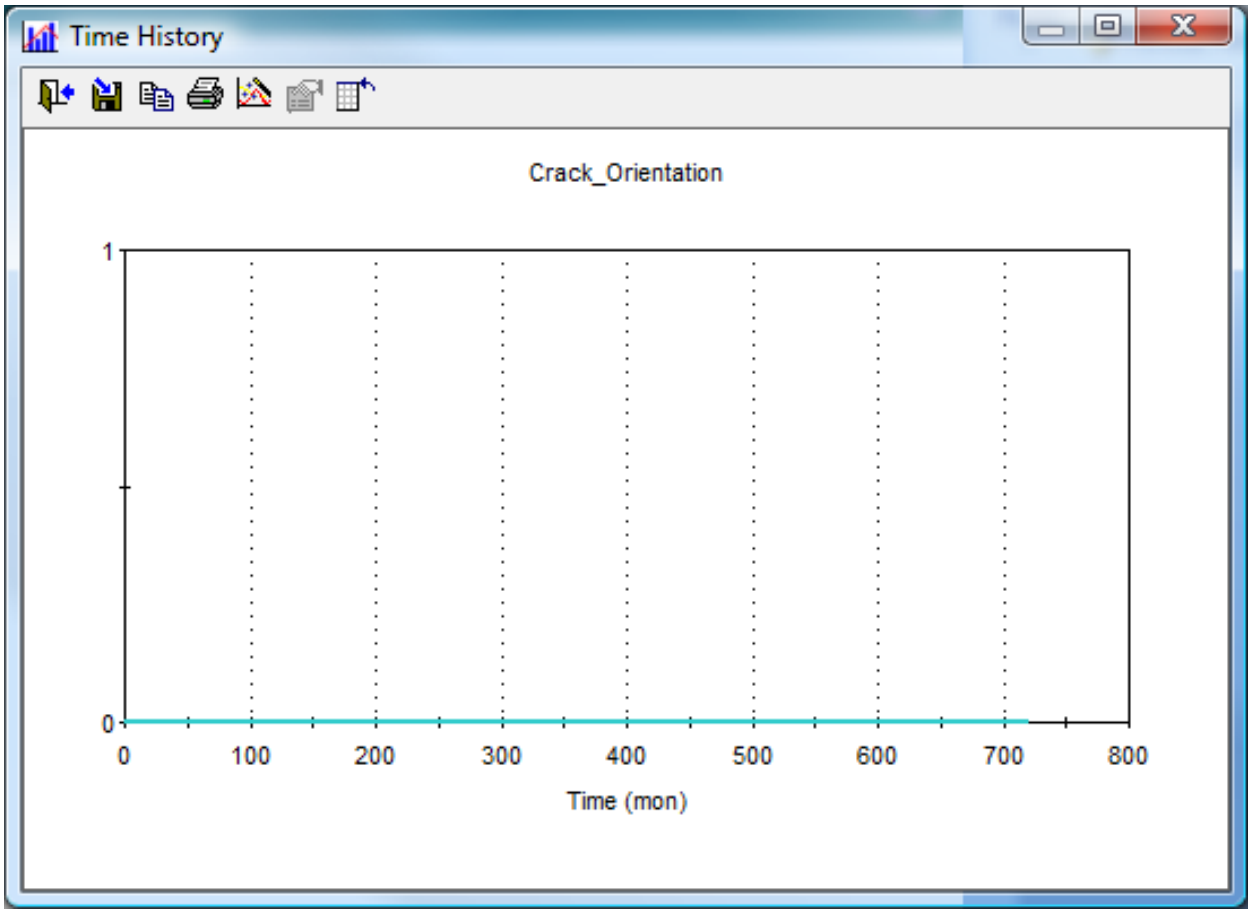


Figure A-3. Deterministic Test Case #1 -- Crack Location Results Plot

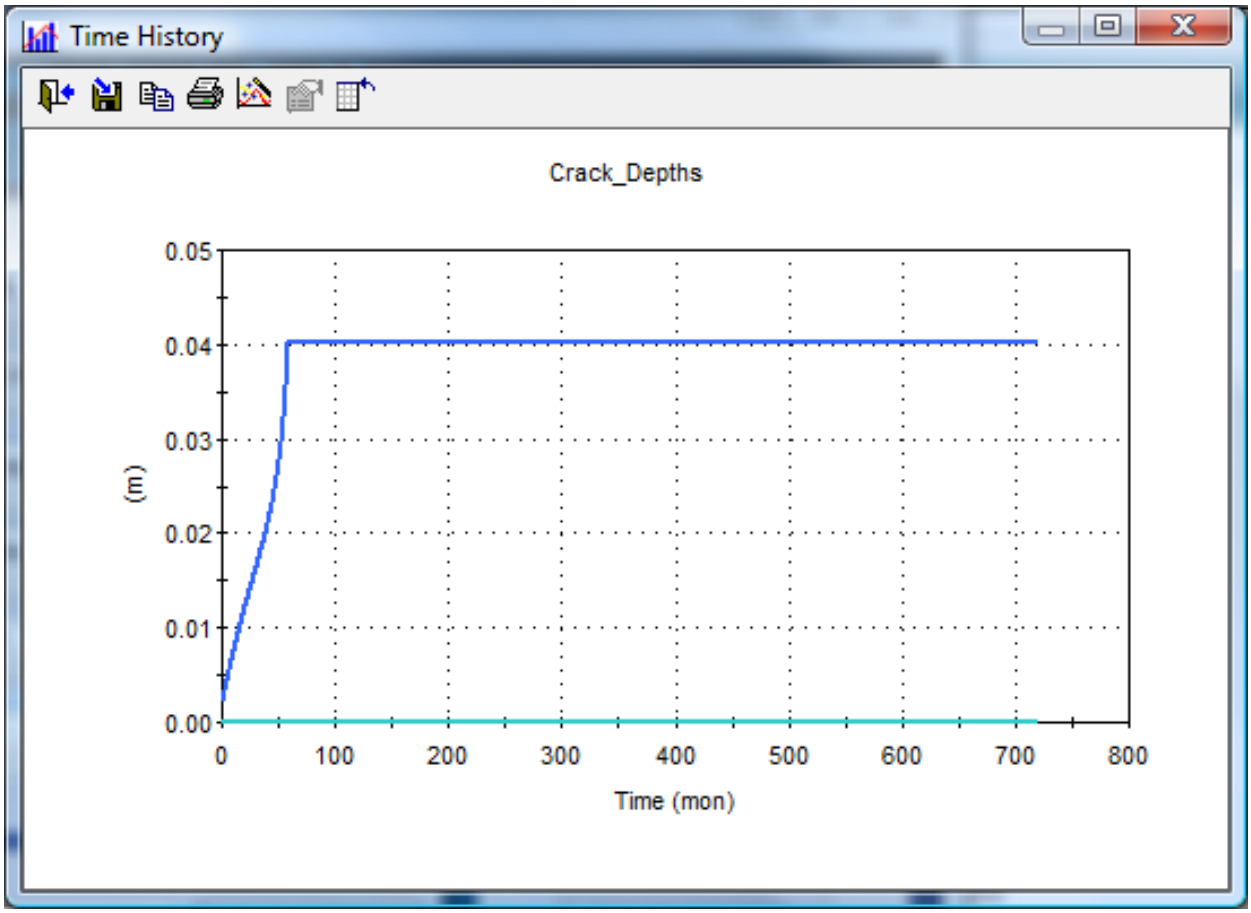


Figure A-4. Deterministic Test Case #1 -- Crack Depth Results Plot

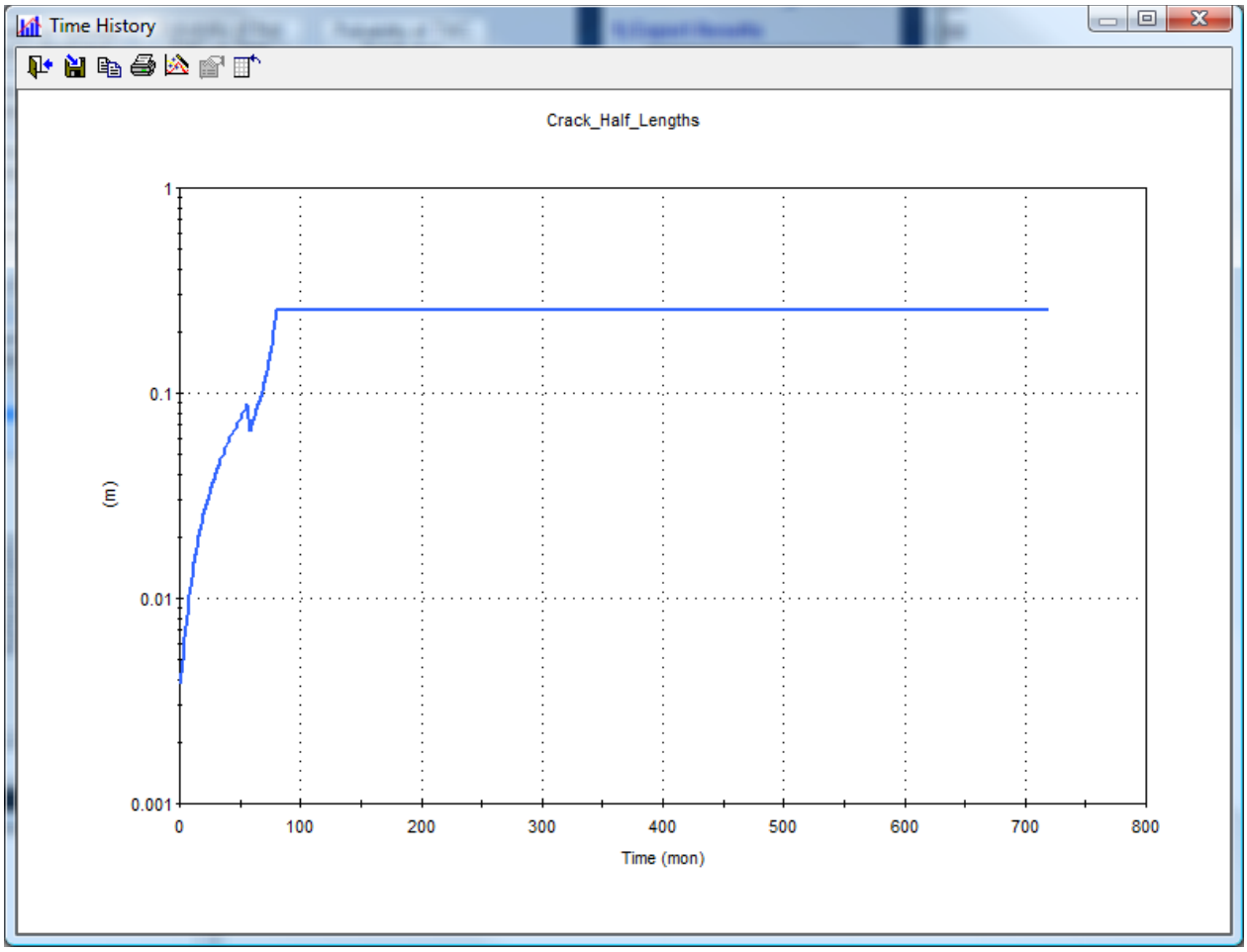


Figure A-5. Deterministic Test Case #1 -- Crack Half Length Results Plot

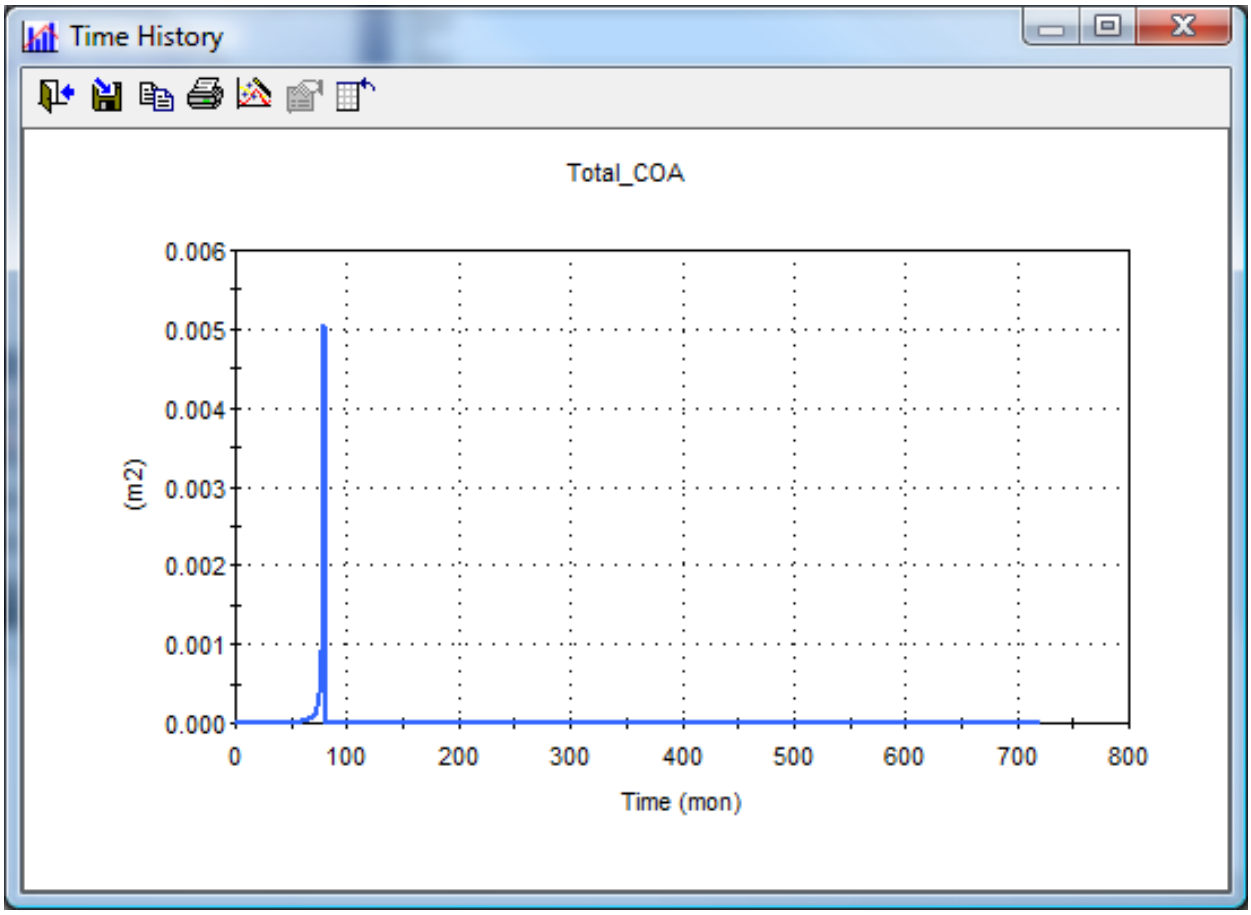


Figure A-6. Deterministic Test Case #1 -- Crack Opening Area Results Plot

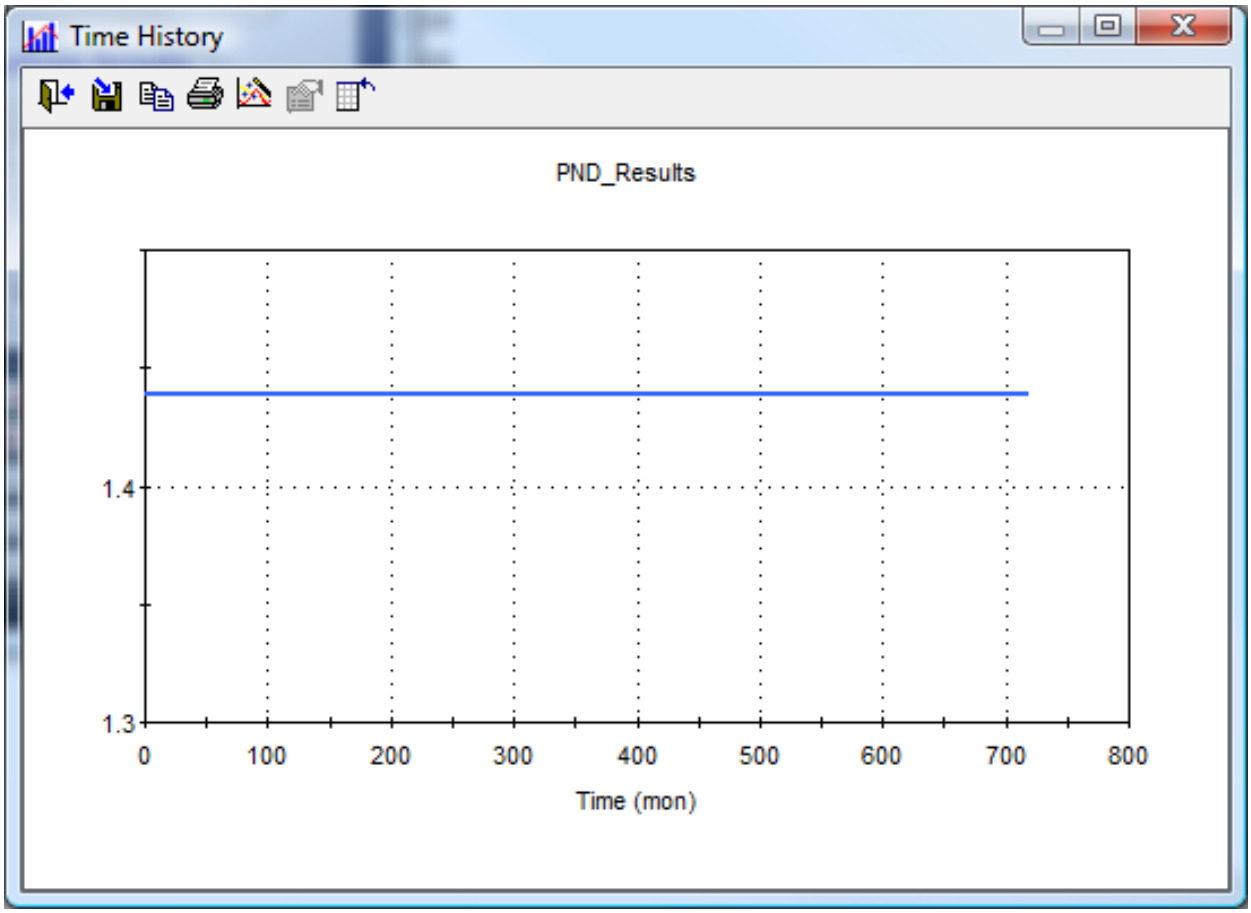


Figure A-7. Deterministic Test Case #1 -- PND Results Plot

A.2 Deterministic Test Case #2

Deterministic Analysis #2: Three cracks at t = 0 years, with no mitigation. This is the same problem as the first deterministic analysis, but with three cracks. The three cracks are the same size as the single crack in the deterministic test case #1, 0.0015 m and 0.003 m for depth and half length respectively. Their respective locations are theta = 0 rad, theta = 0.6 rad and theta = -1 rad. The input deck for this case is included with the controlled version of the inputs spreadsheet for xLPR.

Screen captures of the following results plots are provided in Figure A-8 through Figure A-14 for comparison to the results produced by the user-installed version of the case.

- Number of Cracks
- Crack Type
- Crack Location
- Crack Depth
- Crack Half Length
- Crack Opening Area
- PND

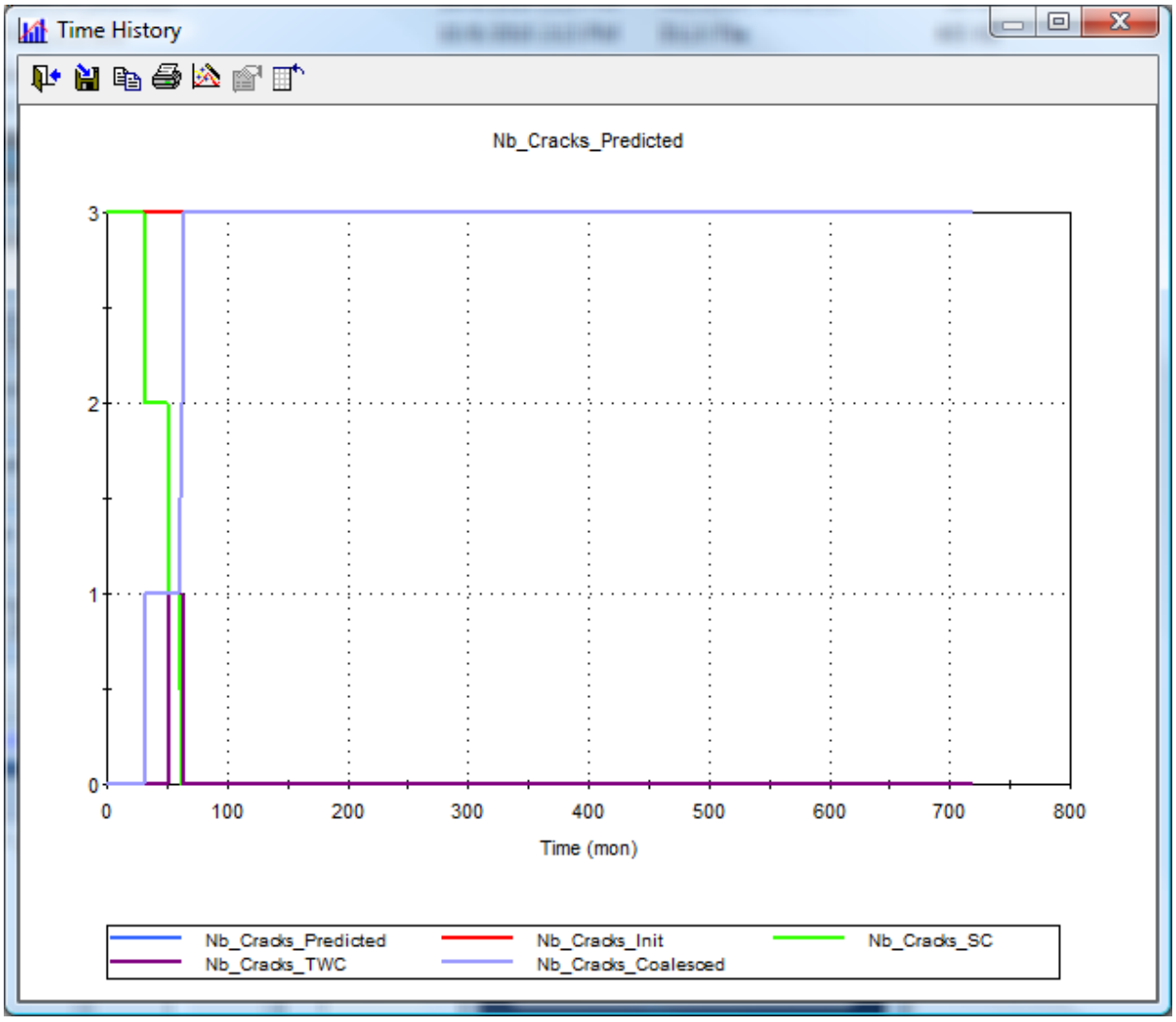


Figure A-8. Deterministic Test Case #2 -- Number of Cracks Results Plot

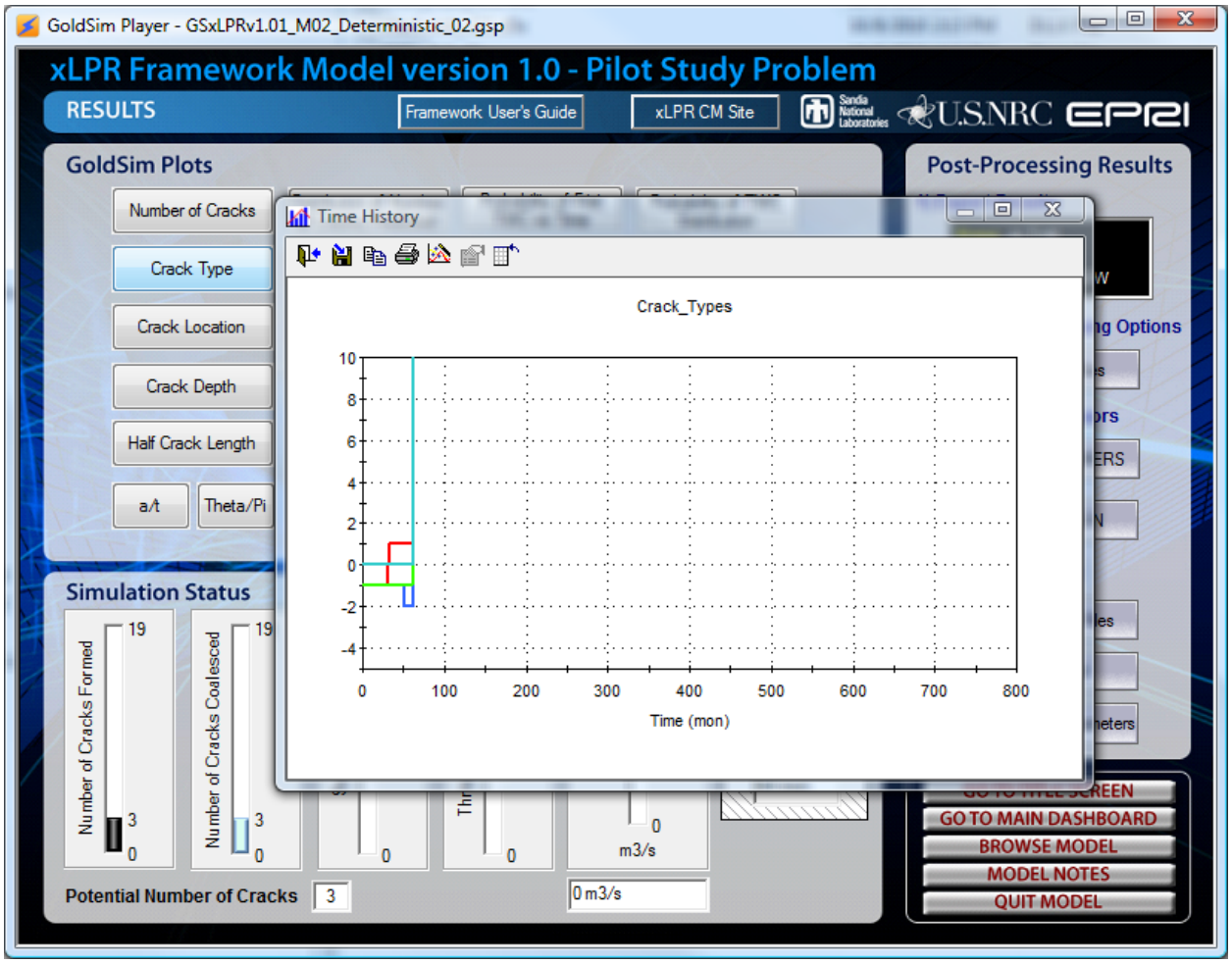


Figure A-9. Deterministic Test Case #2 -- Crack Type Results Plot

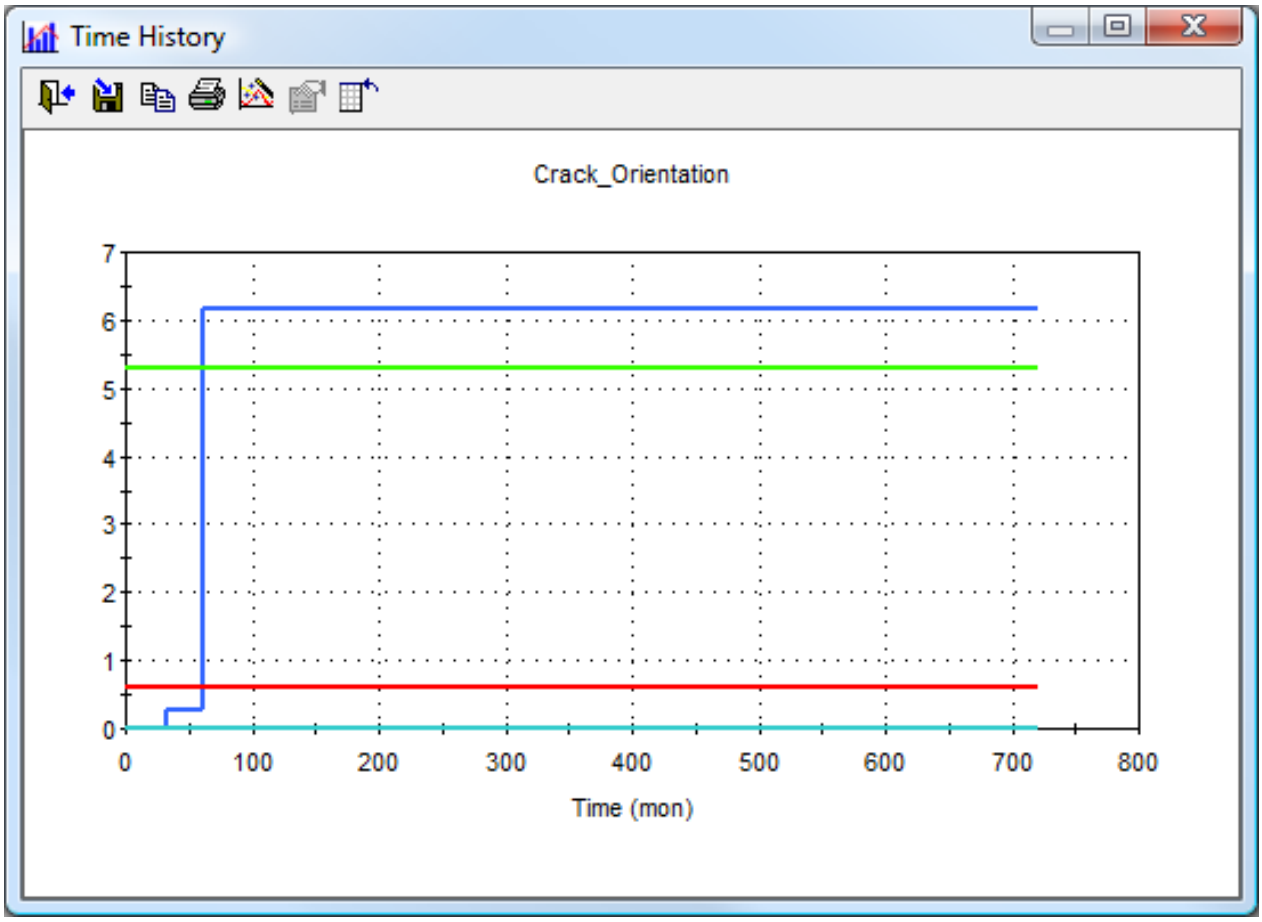


Figure A-10. Deterministic Test Case #2 -- Crack Location Results Plot

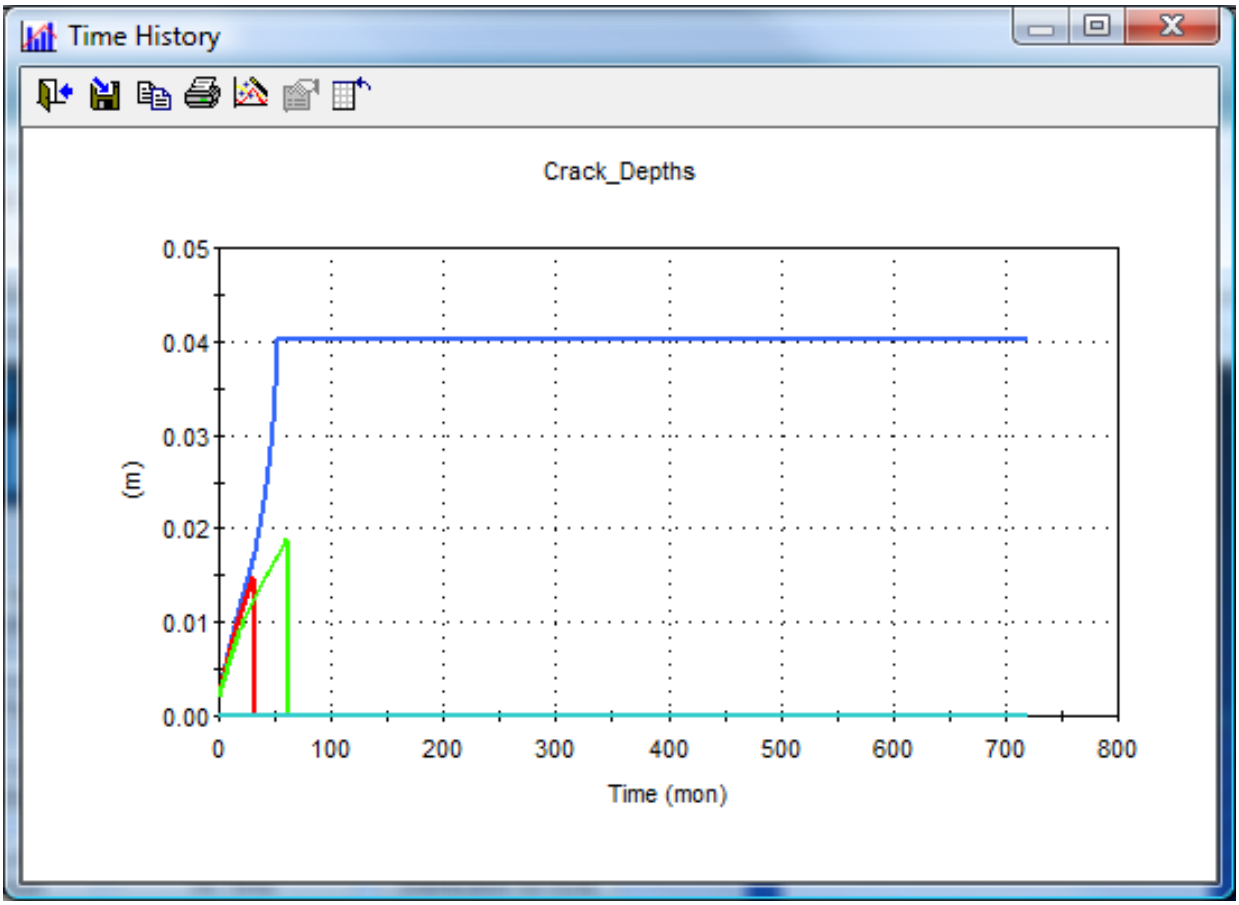


Figure A-11. Deterministic Test Case #2 -- Crack Depth Results Plot

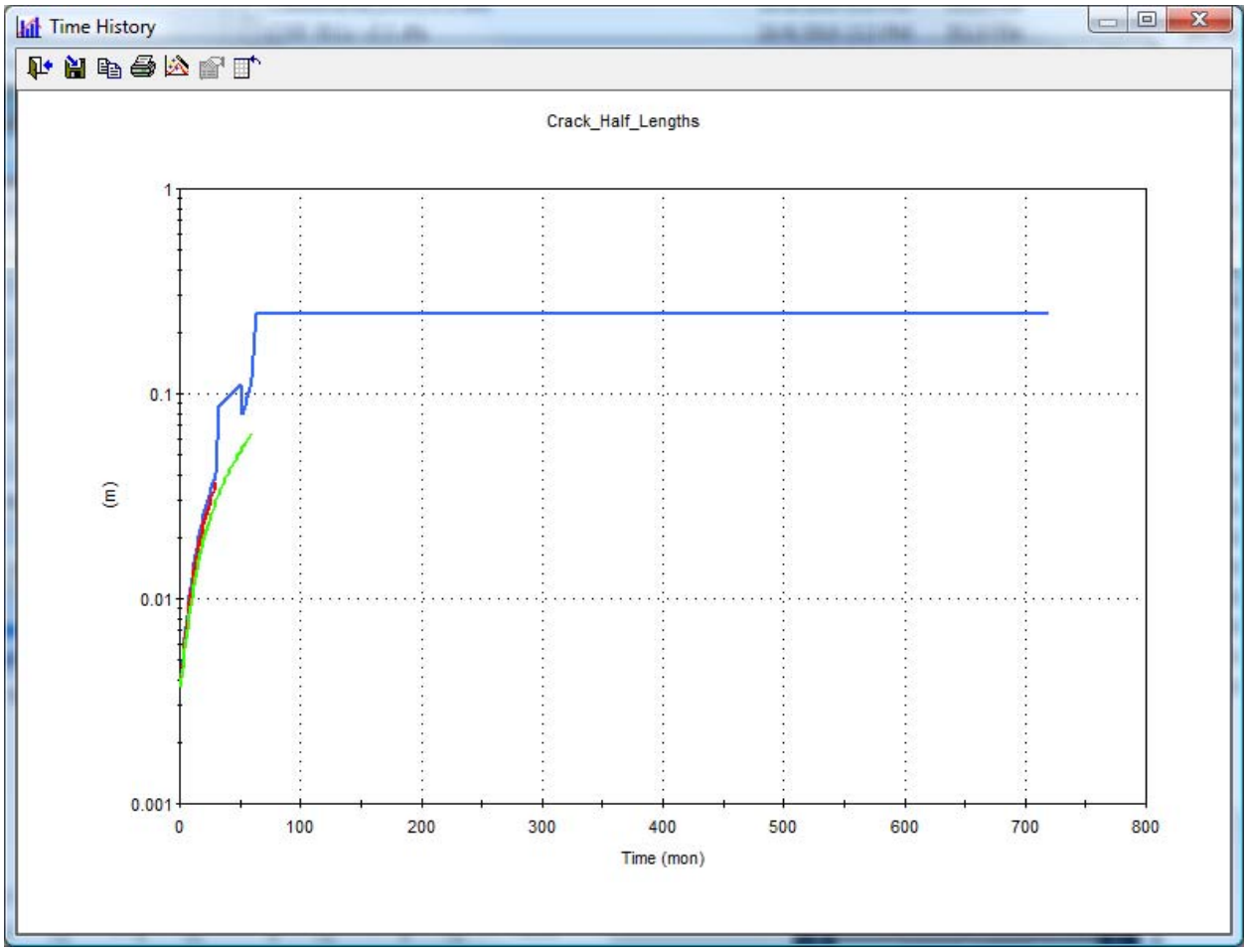


Figure A-12. Deterministic Test Case #2 -- Crack Half Length Results Plot

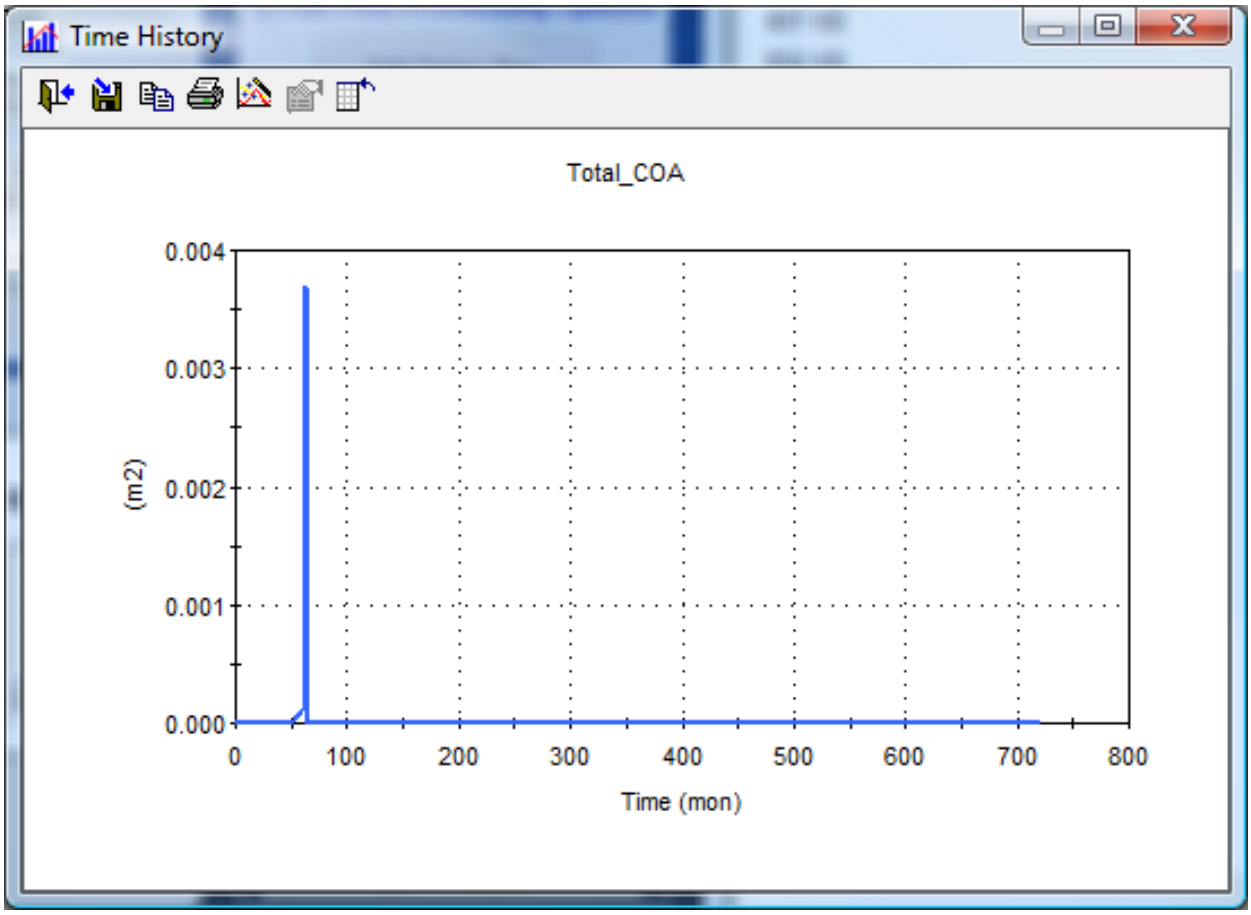


Figure A-13. Deterministic Test Case #2 -- Crack Opening Area Results Plot

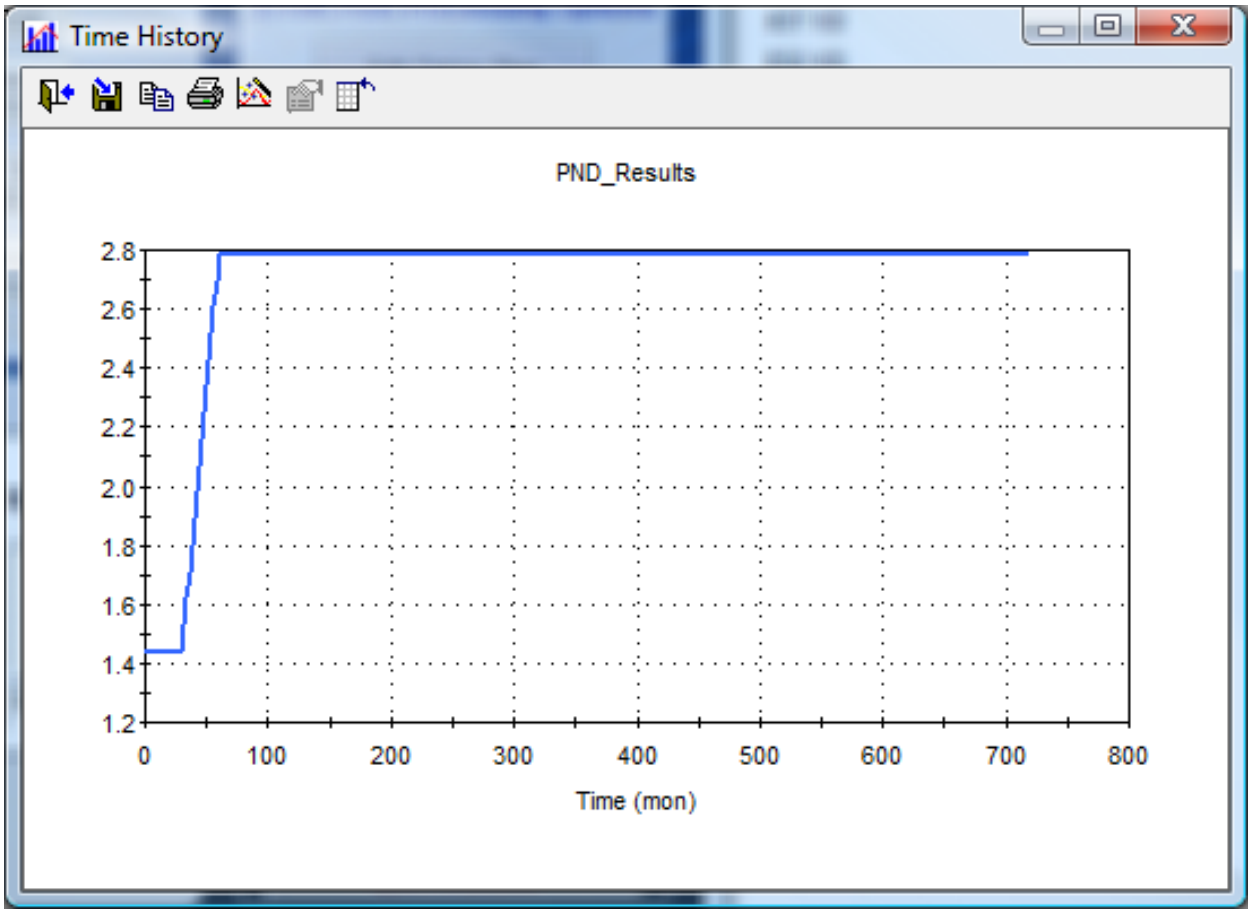


Figure A-14. Deterministic Test Case #2 -- PND Results Plot

APPENDIX B: DVD – VERSION 1.0 MODEL FILES AND TEST CASES

The DVD contains a complete installation package controlled on 10/13/2010. These files are available (at the time of this publication) on an xLPR Program Web accessible file share hosted by Battelle Labs <https://websps1.battelle.org/nrcnureg/home/xLPR_CM/default.aspx>. Listed below are the files contained on the DVD.

Controlled Model\Version 1.02\DVD Files:

BETA_Inputs_AE_09_30_2010.xlsx
Coalescense_DLLx_v2.2.dllx
COD_DLLx_v2.1.dllx
crack_init_v2.1.dllx
DPD_v1.1.dllx
Expectation_v1.0.exe
EXP_options.txt
grower_DLL_v2.1.dllx
GSxLPRv1.0 Users Guide.PDF
GSxLPRv1.02_M02.gsm
GSxLPRv1.02_M02.gsp
GSxLPRv1.02_M02_Deterministic.gsm
GSxLPRv1.02_M02_Deterministic.gsp
inspection.txt
ISI_DLL_v2.1.dllx
kSurf_DLL_v1.1.dllx
kTWC_DLL_v1.1.dllx
load_DLL_v1.1.dllx
options.txt
quantiles.txt
SCFail_DLL_v2.1.dllx
SQUIRT_DLL_v1.1.dllx
times.txt
TRANSFORMERS_v1.0.exe
TWCFail_DLL_v2.1.dllx
variables_list.txt
26 File(s)

APPENDIX C: TRANSFORMERS V1.03 USER'S MANUAL

INTRODUCTION

TRANSFORMERS consists of a set of routines used to post-process one or several variables of interest in order to take into account the effect of routine inspection and leak detection. It is developed in Fortran 90 and run as a desktop executable that can easily be changed to a dynamically linked library (DLL) to be called from another code if desired.

MATHEMATICAL DESCRIPTION

TRANSFORM option

TRANSFORMERS allows the user to transform the data for any variable into an indicator function (a set of 0 and 1) in case the user would be interested in threshold values.

For instance one user could be interested with the time when a crack occurs using crack type and set the value to 1 once a crack type is equal to (-1). Or with the time a rupture occurs using the same file and set the value to 1 once a crack type is equal to 200 (representing a pipe rupture). The user could also want to study the value once reaching a certain threshold (the indicator function is set to 1 if the value is allow or below a certain threshold selected by the user). The transform command can then be used before applying other type of correction to create such indicator function.

The transform options are:

- Transform option (integer = 0, 1 or 2). This option is used to select which kind of limit interest the user.
  - If this option is set to 0, the indicator function will be equal to 1 when the data is below the threshold and 0 otherwise.
  - If this option is set to 1, the indicator function will be equal to 1 when the data is "equal" to the threshold and 0 otherwise. Note that since we work with real numbers, the equality is considered in the following way: $\left|\frac{data-threshold}{threshold}\right| < 10^{-6}$.
  - If this option is set to 2, the indicator function will be equal to 1 when the data is greater than the threshold and 0 otherwise.
- Transform time option (integer = 0 or 1)
  - If the time option is set to 0, then the indicator function is set to 1 only when the data meet the condition.
  - If the time option is set to 1, then the indicator function is set to 1 when the data first meet the condition and at all following time steps.

- Transform threshold (real) – real number used as threshold value. It is valid in Fortran to write an integer (that will be read as a real).

Leak detection correction

The leak detection correction implementation supposes that once a certain leak rate is reached, it will be detected. The weld will then be replaced such that it will never fail.

The total leak rate file is read for each realization (epistemic and aleatory). An indicator function is created such that it is equal to 1 until the threshold value for leak rate detection is reached. Then the indicator function is taken equal to 0.

The correction for leak detection on the studied variable is obtained by multiplying this indicator function to the variable at each time step and for each realization (epistemic and aleatory).

Inspection correction

Inspection detection uses the PND in order to find whether a SC is found or not when inspection occurs. We suppose once again that if a crack is detected, the weld will be replaced in a way that it will never fail again for the remaining duration of the plant life.

The first step is to transform all PND values (for all cracks) such that:

- If the crack is not initiated yet or has coalesced the PND is 1.
- If the crack is a TWC, it will always be detected by inspection and therefore the PND is 0.

For each crack, its type must be known at each time step in order to perform this update.

Several options are discussed in order to take into account inspection correction.

The first option was on the dependency of detection for multiple cracks in the weld. Two methods have been implemented. The first method supposes that each crack has a probability of being detected or not, independent of the presence of other cracks. In this case, the total PND is equal to the multiplication of all PND. Another method proposed was to consider only the lowest value of PND, regardless of the number of cracks. The minimum over all PND is then considered at each time-step. The convolution is controlled with an integer (=1 for the first method and 2 for the second).

Another set of options was considered with respect to the successive inspection. The first method was considering that each inspection was independent of the preceding ones. Then the probability of still having not detected a crack at time T_i (time of the i^{th} inspection) is the multiplication of PND of cracks at time T_j (j=1,…,i). The second method considered that if a crack has not been detected previously, it will not be detected unless it becomes a TWC. The implementation is a little more complex, since it requires checking the crack type at each inspection time and fixing the value of PND to 1 for each crack and for each time step after an

inspection happens and the crack was already present until it becomes TWC or fails. The time dependent option is controlled with an integer (= 0 for the first method and 1 for the second).

INPUTS TO THE CODE

In order to run, the code needs several inputs files listed as follows:

- The file "options.txt" that lists all the options selected as well as the name of the other input files (see description below)
- The total leak rate file (here we consider the sum of leak rate amongst all cracks)
- The crack type over time for each considered crack (the name should be a prefix followed by "_xxx.txt" where xxx refers to the crack number)
- The PND over time for each considered crack (the name should be a prefix followed by "_xxx.txt" where xxx refers to the crack number)
- The inspection times file if the inspection option is selected
- A time file listing the time steps
- One or several files listing the value for the variables of interest
- One file listing all variables filenames if the code loops over several variables (number of variables > 1)

THE "OPTIONS.TXT" FILE

The options.txt file is used to determine which type of analysis is performed. The data that needs to be filled is, in order:

- Prefix that will be common to all filenames (e.g., "xLPR_beta_") – this prefix will be used to distinguish one run from another (a string of maximum length 63)
- Filename listing all variables filenames (a string of maximum length 63)
- Prefix for the PND files (a string of maximum length 63)
- Leak rate filename (a string of maximum length 75)
- Prefix for crack type (ctype) files (a string of maximum length 63)
- Times filename (a string of maximum length 75)
- The number of time steps (integer)
- The sample size (integer) – this corresponds to the total sample size (aleatory * epistemic)
- The number of cracks (integer) that should be consistent with the number of PND and crack type files
- Number of comments in the matrix files (see description in next section)
- Number of columns ignored in matrix files (see description in next section)
- Option if correction with respect to inspection is applied (integer = 0 for no and 1 for yes). The following four options are added only if this option is 1:
    - Name of inspection time file (string of maximum length 22)
    - Number of inspections considered (integer)

    - Dependency type on inspection over time (integer = 0: inspections are independent and their effects are multiplied to each other; 1: inspections are conditional on the first one occurring. If a SC exists and is not found, it will not be detected unless it becomes a TWC)
    - Convolution method on inspection for multiple cracks (integer = 0: take the minimum PND from multiple cracks; 1: multiply the PND)
- Option if correction with respect to leak rate threshold is applied (integer = 0 for no and 1 for yes). The following option is added only if this option is 1:
    - Leak rate threshold (real)
- Option if the output of interest needs to be transformed upfront (integer = 0 for no and 1 for yes). **See section relative to the transform option.** The following three options are added only if this option is 1:
    - Transform option (integer = 0 for "less than", 1 for "=" and 2 for "greater than")
    - Transform time option (integer = 0 means change only the value corresponding to the option; 1 means change everything after this option has been met)
    - Transform threshold (real)
- Option if the output of interest needs to be taken as it is calculated by the framework or if it needs to be transformed such that the maximum over time is taken at each time step (integer = 0 for no change and 1 if max over time step is taken).
- Suffix that will be added to the basename to save output results (character of max size 6). If the keyword "NO" is used, then the same name is used and the variable file will be overwritten.

An example of valid "options.txt" file follows

```
poipoi                  ! prefix for all files
variables_list.txt      ! filename listing all var. filenames
pnd                     ! pnd prefix
lrate_20E_20A.txt       ! leak rate file
ctype                   ! prefix for type file
times.txt               ! times file
721                     ! number of time steps
400                     ! total sample size
3                       ! number of cracks considered
2                       ! number of comments rows
1                       ! number of columns to be ignored
1                       ! inspection tag: 1= inspection, 0 = no inspection
inspection.txt          ! inspection file
3                       ! nb of inspection
0                       ! dependency on inspection (0 = independent; 1 = dependent)
1                       ! convolution on inspection (0 = minimum; 1 = product)
0                       ! leak rate threshold correction: 1 = threshold, 0 = no threshold
1                       ! transform tag: 1 = output data transformed, 0 = no transform
2                       ! transform option 1: 0: <, 1: =, 2: >
1                       ! transform time option
0.0                     ! threshold value for the transformed
1                       ! take max over time or not (1 = max; 0= no change)
_COR                    ! suffix for output file
```

DESCRIPTION OF THE OTHER INPUT FILES

The variable of interest, leak rate over time, PND for each crack, and cracks type files are created and output automatically by the framework. As it is possible that these files may change in the future, the code allows flexibility on the way the data are saved in these files but supposes that all files are consistent.

- All these files have to contain a matrix.
- Values are supposed to be real, but integers are read and transformed as real
- One can have as many rows of comments as wanted. The number of comments row should be assigned appropriately to ignore these comments
- One can then ignore the first *c* columns (for instance if one wants to save the time steps or something else). The number of columns to be ignored should be assigned appropriately to ignore them
- For PND and ctype, the name of the files should be "prefix_xxx.txt". For instance if the prefix specified for PND files in options.txt is "pnd" all the files considered should be named pnd_001.txt, pnd_002.txt … and so one.
- The number of cracks considered has to be consistent with the number of PND and ctype files.

A few other specific files need to be created in order to run the code. The times file consists of one column listing all the time steps. This number can have any unit (years or months for instance) and the code will work as long as it is consistent with the unit used in the inspection file. The inspection file consists of one column listing the inspection times. The unit has to be consistent with the time file described above. Moreover, the number of inspections in options.txt has to be consistent with the number of inspection times listed here. The variables file lists the variables filenames in one column if multiple variables are considered.

LIST OF SUBROUTINES

The main code (**TRANSFORMERS**) will read all options in the "options.txt" file and the other input files. When it is run it will first apply transform on the output of interest.

The transform option can be used to create a set of variables of interest. For example, one can use the crack type file to study only SCs (option '=' for transform option , '0' for time option, and '-1' for threshold) or only when there is at least one crack (option '<' for transform option,'1' for time option, and '0.0' for threshold) – see description below.

Once the transform is applied, the code will estimate and apply the correction due to leak rate detection if required. It will then do the same thing with inspection if required. The desired result will then be output to a file.

The following subroutines have been developed in order to perform such calculations.

Read matrix is a simple routine reading the matrix file as described in the previous section.

Multiply_correction simply multiplies each element of two matrices (in real) and sends back the resulting matrix. It is used to apply leak rate and inspection corrections.

Indicator_matrix creates a matrix filled with 0 and 1 depending if the threshold is met (1) or not (0). It takes in input as a matrix and its dimensions, then the threshold value, the limit type (0: < , 1: =, 2: >) and a time option. The time option is used to determine if only the cells meeting the condition are changed or if all are changed for all subsequent time steps once the threshold is met. Indicator matrix calls the routine **indicator_vector** that works at a vector level for one realization. This routine is used for transforming the data of the variable of interest and for determining the correction with respect to the threshold value on leak rate.

Update_PND reads the PND and crack type files and insures that: (1) before a crack is created or after it has coalesced, its PND is 1.0 and (2) after a crack becomes a TWC, its PND is 0.0.

PND_Multiple_cracks deals with the consideration of multiple cracks. It reads all PND files and either multiplies or takes the minimum value amongst the considered cracks at a given time step for a given realization, depending on the option on convolution.
PND_Unique_inspection estimates the effect of multiple inspections, considering the crack types and PND values. Depending on the dependency option, the subroutine will either consider the multiple inspections as independent (so each is multiplied) or not. In case of dependency (option = 1) any subsequent inspection of a SC will not lead to its detection, unless it evolves into a TWC.

PND_corrector estimates the correction due to inspection based on the PND files, the times file and the inspection file. The PND files are changed upfront and this subroutine calculates only the result of multiplying correctly all corrections. It calls the subroutine **PND_corrector_vector** that works at a vector level for one realization.

APPENDIX D: EXPECTATION V1.06 USER'S MANUAL

INTRODUCTION

EXPECTATION is a code used to post-process one or several uncertain variables estimated with separation of aleatory and epistemic uncertainty. The aleatory uncertainty treatment is done using a Simple Monte Carlo sampling (SMS) procedure. The code averages over aleatory uncertainty, then calculates mean and quantiles over the expected values. As an option, CCDFs and statistics (i.e. mean and quantiles) can be estimated. Since Version 1.03, the code allows the use of importance sampling for aleatory uncertainty and/or epistemic uncertainty. Version 1.04 adds a possible normalization of the weights if necessary.

Maximum epistemic sample size has been extended in Version 1.06. It is now set to 1 million for expected value and 500,000 for CCDF (due to the fact that 2 columns are saved per CCDF). It is developed in Fortran 90 and run as a desktop executable that can easily be changed as a library.

MATHEMATICAL DESCRIPTION

The mathematical model for EXPECTATION is simple. The aleatory uncertainty is sampled randomly using SMS. The average over aleatory uncertainty is obtained by summing all values in an epistemic set and dividing by the aleatory sample size. In case of importance sampling, a weighted sum is used. This operation transforms the data (n_aleatory x n_epistemic vs. time) into expected data (n_epistemic vs.time). For statistics, this expected data is then sorted over epistemic runs to estimate quantiles.

The code also generates CCDFs at a selected time step. The CCDFs are created as follows. For each epistemic set, a CCDF is created. It is done by sorting the data from smallest to largest and associating with them a probability from 1 to 0 decrementing by 1/n_aleatory (or by the associated weight in case of importance sampling). Statistics over CCDFs are calculated and saved in a separate file.

INPUTS TO THE CODE

In order to run, the code needs several input files listed as follows:

- The file "EXP_options.txt" that lists all the options selected as well as the name of the other input files (see description below)
- A quantile file (list of quantiles that will be estimated)
- A time file listing the time steps
- Files listing the value for the variables of interest
- File listing the prefix names for the variables of interest

- A weight file for aleatory uncertainty if importance sampling is used (optional)
- A weight file for epistemic uncertainty if importance sampling is used (optional)

THE "EXP_OPTIONS.TXT" FILE

The EXP_options.txt file is used to determine which type of analysis is performed. The data that needs to be filled is, in order:

- Prefix that will be common to all filenames (e.g., "xLPR_beta_") – this prefix will be used to distinguish one run from another (a string of maximum length 63)
- Filename listing all variables filenames (a string of maximum length 63)
- Filename for the quantiles desired (a string of maximum length 75)
- Times filename (a string of maximum length 75)
- Epistemic sample size (integer of maximum 7 digits) – the aleatory sample size is not used here
- Filename for the epistemic weights in case of importance sampling (a string of max length 63) – the prefix will be added to this filename. The keyword "NO" means that there is no importance sampling used.
- Number of time steps (integer)
- Option to normalize the epistemic weights if needed (character "Y" or "N")
- Number of comments in the matrix files (integer: see description in next section)
- Number of columns ignored in the matrix files (integer: see description in next section)
- Aleatory sample size (integer)
- Filename for the aleatory weights in case of importance sampling (a string of max length 63) – the prefix will be added to this filename. The keyword "NO" means that there is no importance sampling over aleatory uncertainty
- An option to normalize the aleatory weights needed (character "Y" or "N")
- Number of quantiles considered (integer: has to be consistent with the number reported in the quantile file)
- Log info number (integer used for debugging – save different information in log file – can be set to 0)
- Suffix name (string of size 6) expected at the end of the variable file. The keyword "NO" means there will be no suffix.
- Option if CCDFs are saved in a file (integer = 0 for no and 1 for yes). The following three options are added only if this option is 1:
    - Time step considered (real – has to be consistent with values used in the time file)
    - Number of discretization values for the statistics (makes the statistics more or less smooth)
    - Discretization type used for the statistics (1: linear; 2: logarithmic)

An example of valid “EXP_options.txt” file follows.

```
poipoi_                 ! prefix for all files
variables_list.txt      ! filename listing all var. filenames
quantiles.txt           ! file listing the quantiles of interest
times.txt               ! times file
20                      ! epistemic sample size
NO                      ! epistemic uncertainty weight file (NO = no weight)
N                       ! weight normalization for epistemic (Y or  N)
721                     ! number of time steps
2                       ! number of comments rows
1                       ! number of columns to be ignored
20                      ! aleatory sample size
NO                      ! aleatory uncertainty weight file (NO = no weight)
N                       ! weight normalization for aleatory (Y or  N)
3                       ! number of quantiles considered
6                       ! log info number (for debugging)
NO                      ! suffix (NO = no suffix)
1                       ! CCDF option (0 = no, 1= yes)
50.00                   ! time to use for CCDF (optional)
30                      ! number of discretization points
1                       ! discretization type for the stats (1: linear; 2: log)
```

DESCRIPTION OF THE OTHER INPUT FILES

The variables of interest files are either created by the framework or come from other post-processing tools. As it is possible that these files may change in the future, the code allows some flexibility on the way the data are saved in these files but supposes that all files are consistent.

- If the files went through TRANSFORMERS first and have a suffixed name, then the same suffix has to be used here.
- All the files have to contain a matrix. The two options referring to “matrix files” are referring these files.
- Values are supposed to be real, but integers are read and transformed as real.
- One can have as many rows of comments as wanted. The number of comments row should be assigned appropriately to ignore these comments.
- One can then ignore the first *c* columns (for instance if one wants to save the time steps or something else). The number of columns to be ignored should be assigned appropriately to ignore them.

Some specific files need to be created in order to run the code. The times file consists of one column listing all the time steps. This number can have any unit (years or months for instance) and the code will work as long as it is consistent with the unit used in the inspection file. The quantile file consists of one column listing the quantile that needs to be estimated. The number of quantiles reported in EXP_options.txt has to be consistent with the number of inspection times listed here. The variable file (listing the file name if the code will be used for several variables) consists of one column listing of **variables** names.

Response weight files are required in case of importance sampling for epistemic or aleatory uncertainty. They are considered to have the same structure as the variables files, i.e. the same number of comments rows and column. The information used is displayed as a column of *n* weights, *n* corresponding respectively to the epistemic and aleatory sample size.

LIST OF RESULT FILES

The result filename will be based on the information indicated in "EXP_options.txt". The files are described below, where *Basename* stands for the string of characters selected by the user.

Basename = 'Prefix'+'variable name'+'suffix'
*Basename*_exp.txt will store the result of the expected value over aleatory uncertainty
*Basename*_stat.txt will store the statistics (mean and quantiles) on the expected values
Basename _log.txt is used for debugging
*Basename*_CCDF_xxxxxxx.txt will store the CCDFs if the option CCDF is selected, xxxxxxx will store the time steps selected as a 7 digit integer
*Basename*_stat_xxxxxxx.txt will store the statistics (mean and quantiles) on the CCDFs

LIST OF SUBROUTINES

The main code (**EXPECTATION**) will read all options in the "EXP_options.txt" file and the other input files. Once done, it will estimate the expected value over aleatory uncertainty and related statistics. If the option is selected, it will also create a CCDF file as well as estimate the statistics, stored in a separate file.

Shell_sort sorts an array by using a shell sort technique. The algorithm comes from Numerical Recipes.

Shell_sort_2 sorts two arrays based on the value of one using shell sort technique. The algorithm extended from the preceding.

Linear_interpolation estimates a linear interpolation.

APPENDIX E: POST-PROCESSING FORM TEMPLATES AND DESCRIPTION

Example Input Option Files for TRANSFROMERS and EXPECTATION Post Processing Codes:

Post Processing Cases		Description
TRANSFORMER Options [Options.txt]		
Option 0	No Inspection or Leakage	Transform does not have to be run.
Option 1	With Inspection	Transformer options to include inspection times and number which will reduce the probability of TWC and rupture.
Option 2	With Leak Detection	Transformer options to include inspection times and number which will reduce the probability of rupture.
Option 3	With Inspection & Leak Detection	Combination of Options 1 & 2.
Option 4	Indicator Function	Optional advanced feature that allows the user to transform data in the Variable_list.txt file based on a transform value. Used on single files or a file set with same units. See Post Processor Documentation.
EXPECTATION Options [Exp_Options.txt]		
Average over Aleatory		Inner/Outer Looping Case. The post processor calculates one result for each epistemic sample that is an average over the inner aleatory samples. [Expectation options]
DPD with Importance Sampling		DPD does not separate the aleatory and epistemic uncertainties; therefore the total number of realizations is the epistemic number. In addition the DPD weighting must be applied to the responses to account for the importance sampling. [Expectation options]

Post Processor Output:

TRANSFORMER Output	
Optional suffix added to file names in the Variable_List.txt file (e.g., _COR)	Transformed data based upon inspection, leak detection or transform function
EXPECTATIONS Output	
*Basename*_exp.txt	Result of the expected value over aleatory uncertainty (column = number of epistemic samples + 1 for time steps)
*Basename*_stat.txt	The statistics (mean and quantiles) on the expected values listed in the quantile.txt file
*Basename*_log.txt	Debug file out
*Basename*_CCDF_xxxxxxx.txt	Store the CCDFs if the option CCDF is selected, xxxxxxx will store the time step selected as a 7 digit integer

Example 1*: Options.txt file. 400 realizations (20 epistemic 20 aleatory case, 400 DPD Samples, etc).

```
GSxLPRv1.02_              ! prefix (model name) for output files
variables_list.txt        ! variable filename or filename listing all var. filenames
pnd                       ! pnd prefix
TLR_001.txt               ! leak rate file
CT                        ! prefix for crack type file
times.txt                 ! times file (must be same number of times used in the simulation)
361                       ! number of time steps used in the simulation
400                       ! total sample size (epistemic * aleatory)
19                        ! number of cracks considered
3                         ! number of comments rows
1                         ! number of columns to be ignored
0                         ! inspection tag: 1 = inspection, 0 = no inspection
0                         ! leak rate threshold correction: 1 = threshold, 0=no threshold
0                         ! transform tag: 1 = output data transformed, 0 = no transform
0                         ! take max over time or not (1 = max; 0= no change)
_COR                      ! suffix for output file
```

* it is not necessary to run transformer in this case, but EXP_options.txt should be modified to increase the number of comment rows = 3 and eliminate the suffix for the file names.

Example Option 1 with Inspection*: Options.txt file

```
GSxLPRv1.02_              ! prefix (model name) for output files
variables_list.txt        ! variable filename or filename listing all var. filenames
pnd                       ! pnd prefix
TLR_001.txt               ! leak rate file
CT                        ! prefix for type file
times.txt                 ! times file
361                       ! number of time steps
400                       ! total sample size
19                        ! number of cracks considered
3                         ! number of comments rows
1                         ! number of columns to be ignored
1                         ! inspection tag: 1 = inspection, 0 = no inspection
inspection.txt            ! inspection file
3                         ! nb of inspection
0                         ! dependency on inspection (0 = independent; 1 = dependent)
1                         ! convolution on inspection (0 = minimum; 1 = product)
0                         ! leak rate threshold correction: 1 = threshold, 0=no threshold
0                         ! transform tag: 1 = output data transformed, 0 = no transform
0                         ! take max over time or not (1 = max; 0= no change)
_INP                      ! suffix for output file
```

* must have an inspection file. The inspection file consists on one column listing the inspection time. The unit has to be consistent with the times.txt file. Moreover, the number of inspections in options.txt has to be consistent with the number of inspection times listed in the inspection.txt file.

Example Option 2 with Leak Detection*: Options.txt file

```
GSxLPRv1.02_            ! prefix (model name) for output files
variables_list.txt      ! variable filename or filename listing all var. filenames
pnd                     ! pnd prefix
TLR_001.txt             ! leak rate file
CT                      ! prefix for type file
times.txt               ! times file
361                     ! number of time steps
400                     ! total sample size
19                      ! number of cracks considered
3                       ! number of comments rows
1                       ! number of columns to be ignored
0                       ! inspection tag: 1 = inspection, 0 = no inspection
1                       ! leak rate threshold correction: 1 = threshold, 0 = no threshold
 6.3e-04                ! leak rate threshold value (real)
0                       ! transform tag: 1 = output data transformed, 0 = no transform
0                       ! take max over time or not (1 = max; 0= no change)
_LRT                    ! suffix for output file
```

* must have an inspection file. The inspection file consists of one column listing the inspection time. The unit has to be consistent with the times.txt file. Moreover, the number of inspections in options.txt has to be consistent with the number of inspection times listed in the inspection.txt file.

Example Option 3 with Inspection* & Leak Detection: Options.txt file

```
GSxLPRv1.02_            ! prefix (model name) for output files
variables_list.txt      ! variable filename or filename listing all var. filenames
pnd                     ! pnd prefix
TLR_001.txt             ! leak rate file
CT                      ! prefix for type file
times.txt               ! times file
361                     ! number of time steps
400                     ! total sample size
19                      ! number of cracks considered
3                       ! number of comments rows
1                       ! number of columns to be ignored
1                       ! inspection tag: 1 = inspection, 0 = no inspection
inspection.txt          ! inspection file
3                       ! nb of inspection
0                       ! dependency on inspection (0 =  independent; 1 = dependent)
1                       ! convolution on inspection (0 = minimum ; 1 = product)
1                       ! leak rate threshold correction: 1 = threshold, 0 = no threshold
 6.3e-04                ! leak rate threshold value (real)
0                       ! transform tag: 1 = output data transformed, 0 = no transform
0                       ! take max over time or not (1 = max; 0 = no change)
_COR                    ! suffix for output file
```

* must have an inspection file. The inspection file consists of one column listing the inspection time. The unit has to be consistent with the times.txt file. Moreover, the number of inspections in options.txt has to be consistent with the number of inspection times listed in the inspection.txt file.

Example Average over Aleatory: Exp_Options.txt file

```
GSxLPRv1.02_            ! prefix (model name) for output files
variables_list.txt      ! variable filename or filename listing all var. filenames
quantiles.txt           ! file listing the quantiles of interest
times.txt               ! times file
20                      ! epistemic sample size
NO                      ! epistemic uncertainty weight file (NO = no weight)
N                       ! epistemic weight normalization (Y or N)
361                     ! number of time steps
2                       ! number of comments rows (change to 3 if transformers is not run)
1                       ! number of columns to be ignored
20                      ! aleatory sample size
NO                      ! aleatory uncertainty weight file (NO = no weight)
N                       ! aleatory weight normalization (Y or N)
3                       ! number of quantiles considered
6                       ! log info number (for debugging)
_COR                    ! suffix (NO = no suffix)
0                       ! CCDF option (0 = no, 1 = yes)
600.00                  ! time to use for CCDF (optional)
100                     ! number of discretization points
1                       ! discretization type for the stats (1: linear; 2: log)
```

Example DPD with Importance Sampling: Exp_Options.txt file

```
GSxLPRv1.02_            ! prefix (model name) for output files
variables_list.txt      ! variable filename or filename listing all var. filenames
quantiles.txt           ! file listing the quantiles of interest
times.txt               ! times file
400                     ! epistemic sample size
DPD_weight.txt          ! epistemic uncertainty weight file* (NO = no weight)
Y                       ! epistemic weight normalization (Y or N)
361                     ! number of time steps
2                       ! number of comments rows (change to 3 if transformers is not run)
1                       ! number of columns to be ignored
1                       ! aleatory sample size
NO                      ! aleatory uncertainty weight file (NO = no weight)
N                       ! aleatory weight normalization (Y or N)
3                       ! number of quantiles considered
6                       ! log info number (for debugging)
_COR                    ! suffix (NO = no suffix)
0                       ! CCDF option (0 = no, 1 = yes)
600.00                  ! time to use for CCDF (optional)
100                     ! number of discretization points
1                       ! discretization type for the stats (1: linear; 2: log)
```

*need to ensure the weight file has the prefix GSxLRv1.02_ .

52 Output files set for Automatic Export in the GSxLPRv1.02 Framework

GSxLPRv1.02_CT _001.txt to _019.txt [19 files]	Crack Type vs. Time (0 no crack, -1 SC, -2 TWC, 1-19 crack number of crack that it coalesced with, 200 rupture)
GSxLPRv1.02_PND_001.txt to _019.txt [19 files]	PND of a SC vs. Time (1 when no crack exists, PND for SC, 0 for TWC, 1 for rupture)
GSxLPRv1.02_FSA_001.txt	Fraction of surface area cracked vs. Time (all cracks)
GSxLPRv1.02_TLR_001.txt	Total leak rate (m^3/s) vs. Time (all cracks)
GSxLPRv1.02_CFO_001.txt	Probability of Critical Failure (rupture) vs. Time
GSxLPRv1.02_CFT_001.txt	Critical Failure Times (rupture) vs. Time
GSxLPRv1.02_FLO_001.txt	First Leak Probability vs. Time
GSxLPRv1.02_FLT_001.txt	First Leak Times vs. Time
GSxLPRv1.02_SFO_001.txt	First SC Probability vs. Time
GSxLPRv1.02_SFT_001.txt	First SC Times vs. Time
GSxLPRv1.02_CFOS_001.txt	Probability of Critical Failure (rupture) vs. Time - w/SSE
GSxLPRv1.02_CFTS_001.txt	Critical Failure Times (rupture) vs. Time - w/SSE
GSxLPRv1.02_SFTS_001.txt	First SC Probability vs. Time - w/SSE
GSxLPRv1.02_SFOS_001.txt	First SC Times vs. Time - w/SSE
GSxLPRv1.02_TFO_001.txt	First TWC Probability vs. Time
GSxLPRv1.02_TFT_001.txt	First TWC Times vs. Time
Total Number of files = 52	

DISTRIBUTION

External Distribution

1 Aladar Csontos
Division of Engineering
Office of Nuclear Regulatory Research U.S. Nuclear Regulatory Commission
Mailstop: C-5A24M
Washington, DC 20555-0001
United States of America

2 David Rudland
Office of Nuclear Regulatory Research
Division of Engineering 21 Church Street, Room 5C04
Mail Stop: CSB-5CA24
Rockville, MD 20852
United States of America

2 Craig Harrington
Electric Power Research Institute (EPRI)
3010 LBJ Freeway, Suite 300
Dallas, TX 75234
Send to: 1110 Cherrywood Drive, Cleburne, TX 76033

Sandia National Laboratories Internal Distribution

1	MS0748	Randall O. Gauntt. Mgr.	6232
1	MS0748	Donald A. Kalinich	6232
11	MS0749	Patrick Mattie	6232
1	MS1369	Cedric Sallaberry	6224
1	MS0899	Technical Library (electronic only)	9536