PHD

Hydraulic system analysis by the method of characteristics.

Skarbek-Wazynski, C. M.

Award date:
1981

Awarding institution:
University of Bath

HYDRAULIC SYSTEM ANALYSIS BY THE METHOD OF CHARACTERISTICS

Submitted by

C.M. SKARBEK-WAZYNSKI

For the degree of PhD
of the University of Bath
1981

P A R T   I

SUMMARY

001. This thesis describes the design, development and testing of a distributed parameter hydraulic system simulation program based on the method of characteristics, and is intended to extend and complement the work being carried out at Bath University on the computer aided design of fluid power systems on small computers.

002. The first part of the thesis is an extensive literature review of distributed parameter techniques and related topics, and represents a stock-taking of current simulation methods and their applicability to fluid power system modelling.

003. The method of characteristics is a numerical technique for analysing wave propagation effects in the time domain. A general program structure was designed whereby various systems could be analysed by subroutines modelling the behaviour of individual hydraulic components linked together by pipe models based on the method of characteristics.

004. General aspects of the program operation were tested by simulating a hydrostatic transmission, good correlation was obtained with analytical results, and with a lumped parameter simulation. More specific problems of component modelling were investigated by simulating a Barmag type, 3 port pressure compensated flow control valve.

005. The program was applied to the analysis of pump generated pressure ripple. Good agreement was obtained with experimental results demonstrating the ability of the method of characteristics, as programmed, to accurately predict high frequency effects in hydraulic systems. The program providing an alternative tool for analysing fluid borne noise, which is especially suitable for situations where transient effects are important.

006. The method of characteristics is not ideal for general hydraulic system simulation and the recommendations for future work

include a scheme for incorporating it into the existing lumped parameter simulation language (HGSP) developed at Bath University.

ACKNOWLEDGEMENTS

I would like to express my thanks to my supervisor Professor D.E. Bowns for his help guidance and encouragement, and to Mr. F.W. Baggett and Mr. P.J. Cleife of Commercial Hydraulics (Gloucester) Limited who sponsored this project.

Many thanks to: Dr. W. Richards, Dr. L. Bonson, Dr. A. Spence who gave valuable advice on the numerical aspects of this work. Dr. K. Edge, Dr. D.G. Tilley, Mr. S. Tomlinson and Dr. A.C. Rolfe who gave advice and assistance on the hydraulic, control and computing aspects. And to Mr. M.D. Butler and Mr. O. Baker who provided experimental data for validating some of the computer models.

Thanks also to Miss Gail Fivash who typed the manuscript.

CONTENTS

NOMENCLATURE

a	tube radius	M
A	pipe area	M^2
A_p	pipe area	M^2
A_o	orifice area	M^2
c	wavespeed	M/S
C	coefficient of capacitance	M^4/N
C_1	constraint factor	-
C_D	discharge coefficient	-
C_o	adiabatic capacitance per unit length	M^3/N
C_p	heat capacity, constant pressure	-
C_v	heat capacity, constant volume	-
$C(\Delta t)$	numerically computed value	-
d	pipe diameter	M
E	Young's modulus	N
f	friction factor	-
f	viscous friction coefficient	Ns/M
F	equation constant	-
Fc	spring preload	N
g	acceleration due to gravity	M/S^2
$G(s)$	arbitrary transfer function	-
h	pressure head	M
ha	function defined as $ha = a(\omega/\nu)^{\frac{1}{2}}$	-
H	equations constant	-
j	imaginary operator $j = \sqrt{-1}$	-
J_o	Bessel function of the first kind	-
J_1	Bessel function of the first kind	-
K_{LIN}	linearised flow coefficient	Ns/M^5
K_{NL}	flow coefficient	Ns/M^5
K_R	linearised flow coefficient	Ns/M^5
ℓ	length	M
L	coefficient of inertance	Ns^2/M^6
L_o	nominal series impedance per unit length	Ns/M^6
M	spool mass	Kg
P	pressure	N/M^2
P	order of integration method	-
P_c	spring chamber pressure	N/M^2

P_D	downstream pressure	N/M^2
P_f	frictional pressure drop	N/M^2
P_u	upstream pressure	N/M^2
Q	volumetric flow	M^3/S
Q_{ID}	ideal pump flow	M^3/S
Q_T	bypass flow	M^3/S
r	pipe radius	M
R	frictional pressure drop/unit length/ unit flow	Ns/M^6
S	laplace operator	s^{-1}
t	time	S
T	pipewall thickness	M
T	time constant	S
T_p	half pipe period	S
v	flow velocity	M/S
$\bar{v}$	averaged flow velocity at a crossection	M/S
v_x	flow velocity in x direction	M/S
$V(t)$	flow velocity as a function of time	M/S
V_{CH}	spring chamber volume	M^3
$W(t-u)$	weighting function in convolution integral	-
x	axial distance	M
y	preset orifice opening	M
$Y(s)$	shunt admittance	M^4/Ns
Z_o	characteristic impedance	Ns/M^5
Z_s	source impedance	Ns/M^5
Z_T	termination impedance	Ns/M^5
Z_T	instantaneous valve impedance	Ns/M^5
Z_{TSS}	steady state valve impedance	Ns/M^5
$Z(s)$	series impedance	Ns/M^6
α	attenuation coefficient	-
β_E	effective bulk modulus	N/M^2
β_F	fluid bulk modulus	N/M^2
γ	propagation constant	M^{-1}
$\Gamma(s)$	propagation operator	-
Δt	timestep	S
Δx	distance between calculation points	M

ζ	damping ratio	-
ρ	fluid density	Kg/M^3
ρ_T	source reflection coefficient	-
ρ_s	termination reflection coefficient	-
ν	fluid kinematic viscosity	M^2/S
To	frictional shear stress	N/M^2
τ	dimensionless time $\tau = (\nu/a^2)t$	-
σ	cavitation number	-
σo	time averaged Prandtl number	-
ω	angular frequency	rad/s
ω_n	natural frequency	rad/s

Suffices

O	initial or original value or time averaged value
O	conditions at outlet
x,t	parameter evaluated at location x, time t
x_o, t_o	parameter evaluated at some initial value of x and t
I	conditions at inlet
R	forward characteristic
S	backward characteristic
E	equivalent volume pipeline

1. INTRODUCTION

101. As increasing demands are made of hydraulic systems in terms of efficiency and controllability the hydraulics engineer can no longer rely on steady state design and past experience. To remain competitive and to reduce costly hardware development time the designer must venture into the field of dynamic analysis and ultimately computer simulation. Any dynamic system consists of elements which obey certain physical laws such as continuity, Newton's second law, compressibility, etc. A system is represented as a number of analytical expressions based on these laws and digital simulation involves solving this set of expressions numerically.

102. The classical computer approach is to rearrange the analytical expressions into a set of first order differential equations which are then solved using a numerical integration technique. The classical approach lacks versatility and is generally only useful for small system simulations. A major disadvantage is that the addition of any components to the system involves reformulating all the equations and re-programming.

103. For an engineer modular programming is the most suitable approach. A system is presented to the computer as a number of modules, each module containing a typical function such as; a first order system, a limit, a hysteresis function etc. The user no longer has to synthesise and manipulate a complete set of equations. Each module contains functions describing the behaviour of elements within the system and the computer software links up the modules to obtain the overall system performance.

104. Development of general purpose dynamic system simulation programs began in the early 1950's and over the years a number of languages has been issued for use under various operating systems, each offering certain specific facilities [1.1 , 1.2]. Early programs were attempts at emulating the solution procedure of an analogue computer but on a digital machine, with all the advantages of increased capacity and versatility that digital computing has to offer. Subsequent conceptual innovations led to visualising dynamic systems as discrete change models or as continuous change models, with separate groups of languages based

on those concepts. A continuous change model is one where the system is considered to be a continuous flow of information handled in total, whereas a discrete change model handles information in discrete packages determined by components or elements or sub-systems. The type of language adopted would depend on the problem being analysed. Today probably the most commonly known simulation language is CSMP (Continuous Systems Modelling Program) which is available at most large computer facilities. CSMP is compatible with standard Fortran functions giving a very powerful problem orientated language [1.3]. However, general languages are not ideal for the rather more specific task of simulating fluid power systems. Many of the modules provided by the language are not required, whereas modules representing certain common hydraulic components are not available [1.4]. Furthermore the increasing number of powerful mini-computers in industry has fostered the desire to perform serious simulation work on these machines. The large storage requirements of general purpose languages, portions of which are redundant, discourages their use on small computers.

105. A logical progression is the development of programs in which the modules represent the characteristics of fluid power components such as pumps valves and motors rather than abstract mathematical functions. With regard to hydraulic system simulation the method used to link component models is of fundamental importance in the construction of a general program. Linking is essentially the transfer of information within the program such that the contribution of each module to the overall performance is compatible with the contribution of every other module. The physical links between components in hydraulic systems are pipelines, and in this respect the programming method reflects the physical structure of the system, because the linking procedure depends on the method chosen to model the pipelines.

106. When a pipe is treated as a simple reservoir of compressible fluid, neglecting wave effects, the pressures and flows are then related by a simple first order differential equation which may be solved simultaneously with all the other equations describing the system. The properties of the system are considered to be lumped together hence the name lumped parameter modelling or linking. Since information is handled en masse lumped parameter programs come into the category of continuous change systems.

107. A more sophisticated pipe model is one which takes into account wave propagation and various dissipative mechanisms. This involves solving the waterhammer equations, the pressures and flows are no longer simply a function of time but also depend on the location within the pipeline. The major methods of solution are reviewed in Chapter 3 and are given the general name of distributed parameter models since the properties of a pipeline are distributed along its length. Likewise the properties of system components are distributed about a network of pipes. A simultaneous solution is complicated therefore numerical methods tend to deal with components sequentially and so may be categorised as discrete change models.

108. The purpose of this project was to investigate the potential use of distributed parameter linking within a general purpose hydraulic system simulation program specifically designed for use on small computers. In the early 1970's the MacDonnel Douglas aircraft corporation issued a set of modular programs to analyse all aspects of aircraft hydraulic system performance [1.5]. One of these programs, HYTRAN (hydraulic transient analysis), is a distributed parameter model which is based on the method of characteristics, a technique for solving the waterhammer equations. HYTRAN is an extremely large and sophisticated package providing facilities rarely required outside the aircraft industry. However this prompted interest in using the method of characteristics on mini-computers, with the intention of contributing to the work being done at Bath University in the field of hydraulic system simulation. Currently a fully interactive simulation package (HGSP) is being developed at the fluid power centre, Bath University [1.6]. The program uses lumped parameter linking and has been designed for use on the DEC PDP11/34 mini-computer.

109. The major part of this thesis is devoted to the development of a method of characteristics based program structure and to specific tests and simulations performed using the program. Initially the program was tested by simulating a hydrostatic boat transmission for which lumped parameter simulation results were available. Further tests were carried out by simulating a three port pressure compensated flow control valve (Barmag type) with particular emphasis on the numerical aspects of solving the component equations. Also a model was developed to simulate the effects of flow ripple produced by hydrostatic pumps. The purpose was to create a time domain model of pump generated pressure ripple in pipe systems. Cenerally frequency domain models are used

but these have certain disadvantages when transient conditions arise. The final stage of the project consists of proposed schemes for including pipe models based on the method of characteristics in the HGSP package to increase its capacity and versatility.

The layout of the thesis is as follows:-

110. Chapter two is an extensive review of the major methods of modelling pipelines with distributed parameters, outlining the obsolete arithmetical and graphical procedures, with a more detailed account of modern frequency domain methods and the method of characteristics. Finite element techniques are mentioned although these are normally used for more complex problems in fluid dynamics.

111. Chapter three deals with certain problem areas encountered in fluid power systems, namely fluid friction, cavitation, minor losses etc., and how the distributed parameter methods are modified to cope with these difficulties.

112. The main features of the Bath University simulation package HGSP are described in chapter four. A brief discussion of the MacDonnel Douglas programs is included.

113. Chapters five, six and seven describe the work on the transmission model, the valve model and the pump ripple model respectively.

114. Chapters eight and nine contain a general discussion, conclusions and recommendations for the future. More specific conclusions relating to the various simulations are found in the chapters describing those simulations.

115. The development of programs has been an essential part of the work described. However in order for the programs to be of maximum use to other researchers and engineers they have presented as a separate section (part II of this thesis) and are written up in the form of a users manual.

2. MODELLING HYDRAULIC TRANSMISSION LINES WITH DISTRIBUTED PARAMETERS

201. The flow of a compressible fluid in a distensible pipeline is described by considering continuity and momentum. The resulting pair of equations are known as the water hammer equations.

2.1 Water hammer equations

dynamic (momentum) equation

$$\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{\partial v}{\partial t} + \frac{v\partial v}{\partial x} + \frac{2fv|v|}{d} = 0 \qquad (2.1)$$

continuity equation

$$\frac{v}{\rho}\frac{\partial p}{\partial x} + c^2\frac{\partial v}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial t} = 0 \qquad (2.2)$$

202. The water hammer equations can be found in a number of text books; {B1 , B2}, and are a special case of the Navier Stokes equations. Their derivation is based on a number of assumptions.

1 The flow is isothermal or has relatively small temperature gradients, a valid assumption in the case of liquid flows at relatively low velocities.

2 The flow is one dimensional. Valid for liquids flowing in relatively rigid tubes.

3 The static pressures does not fall below the liquid vapour pressure, in other words cavitation is not permitted anywhere within the pipeline, furthermore the pipe must run full at all times.

This assumption is frequently violated in practical hydraulic systems. Cavitation and aeration are common problems. When aeration is slight the water hammer equations may still be used provided the wavespeed c is altered to take account of the presence of air bubbles. Vapourous cavitation may manifest itself as a region of two phase flow, a vapour bubble or column separation. Several authors have developed techniques to take cavitation into account and this work will be discussed in greater detail below.

4 Pipes are of circular crossection. The bulk modulus of the fluid and the Young's modulus of the pipe material are assumed constant.

The derivation of the water hammer equations includes the compressibility of the fluid and pipe wall distension, but the effect should be limited such that the liquid density and the pipe diameter do not change by more than approximately 1% an assumption which is valid for liquids flowing in relatively rigid pipes.

5 The stresses in the pipes are below the elastic limit of the pipe wall material, so that no plastic deformation occurs.

In fact recent work by Youngdahl et al. [2.1] has shown how the effect of plastic deformation on fluid transients may be taken into account. This effect is important in analysing the progress of a pressure transient through nuclear power plant, but in fluid power applications it is irrelevant since a pipeline which has been plastically deformed is considered to have failed.

6 The pipe and the liquid are perfectly elastic. The implication of this assumption is that no energy is lost as a result of repeatedly straining the fluid and pipe. The only energy dissipation mechanism taken into account by the water hammer equations is the viscous shearing at the pipe walls due to fluid friction. This assumption is valid when dealing with metal pipes but raises doubt for flow through hydraulic hose. A hose is a complex structure consisting of a polymer base reinforced by a textile or wire braid. Under dynamic conditions hoses display certain visco-elastic properties which assist in the dissipation of energy. A theoretical model for wave propagation through fluid filled hydraulic hose has been developed by Longmore

[2.2] and will be discussed further in Chapter three.

203. Before describing the various techniques used to solve the water hammer equations it is interesting to note the physical significance of the various terms in both equations.

Dynamic equation

$$\underbrace{\frac{1}{\rho}\frac{\partial p}{\partial x}}_{\text{Potential gradient over element}} + \underbrace{\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x}}_{\text{Acceleration over element}} + \underbrace{\frac{2fv|v|}{d}}_{\text{Friction}} = 0$$

Continuity equation

$$\underbrace{\frac{v}{\rho}\frac{\partial p}{\partial x} + c^2\frac{\partial v}{\partial x}}_{\text{Net inflow into element}} + \underbrace{\frac{1}{\rho}\frac{\partial p}{\partial t}}_{\text{Accumulation of flow within element (compressibility)}} = 0$$

The friction factor f used in the momentum equation is defined by the Darcy formula for pressure loss due to fluid friction.

Darcy equation $$\Delta p_f = \frac{4f\ell}{d}.\frac{1}{2}\rho v^2 \qquad (2.3)$$

204. Early work on unsteady fluid flow assumed that the friction factor for unsteady flow was the same as that for steady flow. A sufficiently accurate assumption for large diameter pipes and low frequency transients. Under unsteady conditions pipe wall friction does depend on the frequency of the disturbance. At high frequencies in small diameter pipes the friction factor is considerably higher than that predicted by steady state formulae, this results in the rapid attenuation of high frequency transients and models using steady state friction factors have under certain circumstances been found to be severely under damped. Techniques have been developed to model frequency dependent friction, these depend to a certain extent on the methods used to solve the water hammer equations and are discussed in Chapter three.

205. A disturbance propagates down a pipeline at the wavespeed c which is the acoustic velocity for the fluid. Wavespeed depends on the bulk modulus and density of the liquid and is therefore affected by pressure, temperature, the gas content of the liquid and the

elasticity of the pipe walls. The variation of bulk modulus and density with temperature and pressure is a property of the fluid and data is usually available [2.3] {B3}. Empirical formulae have been published which allow values of bulk modulus and density to be calculated for mineral hydraulic oils at any operating temperature and pressure [2.4 2.5].

206. For a liquid flowing in an **infinitely** rigid pipe the wavespeed is determined by the **following equation.**

$$c = (\beta_f/\rho)^{\frac{1}{2}} \qquad (2.4)$$

Pipe wall elasticity reduces the wavespeed and may sometimes be accounted for by using a modified value of bulk modulus in equation (2.4).

$$c = (\beta_E/\rho)^{\frac{1}{2}} \qquad (2.5)$$

β_E is the effective bulk modulus and depends on the fluid bulk modulus β_f and the pipe wall elasticity. The distension of the pipe is a function of the Young's modulus of the pipe material and the stresses generated in the pipe wall during the passage of a transient. The stresses in turn depend on the pipe geometry (thick wall, thin wall, composite) and the degree of constraint applied to the pipe. Various cases are well documented in standard texts {B1 B2}.

207. A general expression for the effective bulk modulus of a thin walled metal pipe is

$$\beta_E = \left(\frac{1}{\beta_F} + \frac{C_1 d}{TE} \right)^{-1} \qquad (2.6)$$

C_1 is a factor which takes into account the effect of the constraint applied to the pipeline. It is a function of Poisson's ratio only. In general Poisson's ratio effects are small compared to other factors affecting wavespeed and C_1 is usually taken as unity except in cases where extreme accuracy is required.

208. The presence of air bubbles in the fluid greatly decreases the effective bulk modulus and hence the wavespeed. A further effect is that the rapid compression and decompression experienced by the bubbles under unsteady conditions results in energy dissipation and a consequent attenuation of the transient. The effects of aeration have been studied by a number of authors {B1 , B2}, [2.6 , 2.7].

209. The simultaneous solution of the two water hammer equations with suitable boundary conditions would allow the calculation of pressure and flow at any instant in time at any point along the pipeline. Unfortunately the water hammer equations are classed as quasi-linear hyperbolic partial differential equations and as such cannot be solved analytically. Over the years a number of techniques have been developed to get around this problem, these include graphical, numerical and simplified analytical methods.

2.2 Arithmetic methods

210. Early work in analysing unsteady flow was directed at solving surge problems in civil engineering systems. The operation of valves and pumps could cause large pressure transients and consequent rupturing of pipelines. In 1893 Joukowsky [2.8] published his analytical and experimental work on the effects of water hammer in the St. Petersbourg water distribution system. The arithmetic technique known as the Allievi interlocking equations was developed by Lorenzo Allievi in 1903 [2.9]. An analytical solution was obtained for the water hammer equations by ignoring friction and all non linear terms.

211. The convective terms, $v\partial v/\partial x$ in the dynamic equation and $(v/\rho)(\partial p/\partial x)$ in the continuity equation, are generally small provided that the flow velocity v is small compared to the wavespeed c. Neglecting friction can only be justified in cases where the frictional pressure drop is a small fraction of the mean static pressure.

The resulting equations are:-

dynamic equation $$\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{\partial v}{\partial t} = 0 \qquad (2.7)$$

continuity equation $$c^2\frac{\partial v}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial t} = 0 \qquad (2.8)$$

Civil engineers usually express pressures as heads ($P = \rho gh$) and the original equations were presented in this form;

dynamic equation $$\frac{\partial v}{\partial t} = -g\frac{\partial h}{\partial x} \qquad (2.9)$$

continuity equation $$\frac{\partial v}{\partial x} = -\frac{g}{c^2}\frac{\partial h}{\partial t} \qquad (2.10)$$

Differentiating 2.9 with respect to x and 2.10 with respect to t and combining gives the classical form of the wave equation

wave equation $$\frac{\partial^2 h}{\partial t^2} = c^2\frac{\partial^2 h}{\partial x^2} \qquad (2.11)$$

The wave equation has an analytical solution defined by the Reimann equations

Reimann head equation $$h = h_o + F\left(t + \frac{x}{c}\right) + f\left(t - \frac{x}{c}\right) \qquad (2.12)$$

Reimann velocity equation $$v = v_o - \frac{g}{c}\left[F\left(t + \frac{x}{c}\right) - f\left(t - \frac{x}{c}\right)\right] \qquad (2.13)$$

212. The functions $F\left(t + \frac{x}{c}\right)$ and $f\left(t - \frac{x}{c}\right)$ have the dimensions of head and represent pressure waves travelling at wavespeed c, upstream and downstream respectively. The Allievi interlocking equations are obtained by eliminating the functions F and f from the Reimann equations and producing a family of equations which can be solved, with the boundary conditions, in a series of steps, the solution from one step being used in the next step and so on. The derivation of the interlocking equations is clearly presented by Fox {B1} pages 30-36. This technique was extensively used until the 1930's, however it was cumbersome when dealing with networks or complicated hydraulic controls and was replaced by graphical methods.

2.3 Graphical methods

213. The Reimann equations involve four variables; pressure or head, flow velocity, distance along the pipe and time. Changes in pressure and velocity are caused by the passage of a wave travelling along the pipe at the wavespeed c. The location of a wave can simply be calculated by:-

$$x = x_o + c(t - t_o) \tag{2.14}$$

where x is the location of the wave at time t and x_o is the known position of the wave at a previous time t_0. By using this simple relationship between time and distance it is possible to manipulate the Reimann equations into a form such that the pressure or head is expressed as a pair of straight line functions of flow velocity.

$$h_{x,t} - h_{x_o, t_o} = \pm \frac{c}{g} (v_{x,t} - v_{x_o t_o}) \tag{2.15}$$

214. Equation (2.15) represents two straight lines, in the h~v plane, of equal but opposite slope. The line of positive slope represents a wave travelling upsteam, the line of negative slope represents a wave travelling downstream. Movement along one of these lines implies not only movement in the h~v plane but also movement in the x~t plane as the wave progresses down the pipe. If the pressure and flow characteristics of the hydraulic components, which form the boundary conditions, are plotted on the h~v plane it is possible to follow the progress of a transient through a system and obtain a plot of the pressure and flow with respect to time at various locations in the system, in other words, to obtain the transient response of the system. {B1 Fox page 39-54}.

215. Graphical techniques were developed by Angus [2.10], Schnyder [2.12] and Begeron [2.11] in the 1930's. Further details of these methods may be found in a number of texts; {B1 , B2 , B4}.

216. Graphical methods were a considerable advance in the analysis of surge. Although they were based on the frictionless Reimann equations it was possible to approximate frictional effects by lumping the frictional pressure drops at one end of a pipeline and treating it as a throttle. Greater accuracy was obtained by having several throttles

along the length of the pipe, however this made graphical analysis extremely complicated and time consuming. Graphical methods were capable of dealing with difficult boundary conditions and remained in general use up to the early 1960's when computer based techniques took over. A number of papers dealing with graphical analysis were presented at the 1965 I.Mech.E. symposium on surges in pipelines, [2.13 , 2.14]. Graphical analysis can be, and has been computerised, [2.15], however other techniques such as the method of characteristics, are more powerful.

217. It is worth repeating that both the arithmetic methods and the graphical methods were developed for large scale civil engineering applications and they were manual techniques requiring a great deal of skill and labour to produce the quality of results needed for the design of large piping systems. A simultaneous study was being carried out by electrical engineers for the solution of electrical transmission line problems. The basic equations for wave effects in fluid pipelines are very similar to those used for electrical transmission lines, and the techniques developed by electrical engineers were adopted in the study of fluid lines primarily in control engineering applications.

2.4 Frequency domain methods

218. The basic theory of wave effects in electrical transmission lines was advanced by Lord Kelvin in 1885 and was developed further by Heaviside in the late 1880's. Constantinesco first applied this theory to liquid pipelines in 1922 {B5}. Early work by Wood 1973 and Rich 1945 used Heaviside's operational theory to develop pipe models with linearised friction, Iberall in 1950 included viscosity and heat transfer effects [2.16]. In adapting the electrical theory to the analysis of fluid transmission lines two different methodologies were developed, however both were based on the same principals. Namely, various assumptions were made, depending on the complexity of the model, enabling an analytical solution to be found using Laplace transformation, in other words the solution was found in the frequency domain.

219. The distinction between the two methodologies during their course of development is not clear cut. However with particular reference to fluid power systems, one method is directed at the analysis of fluid borne noise and the other method led to the block diagram representation of a fluid pipeline for control engineering purposes.

220. Noise analysis is the study of a steady oscillation caused by a forcing function such as the flow ripple generated by a positive displacement pump. The method does not provide any information about the initial transient conditions which exist before the steady oscillation is established, this technique is called the Impedance method. The block diagram approach proposed by Ezekiel and Paynter in 1956 [2.17] is concerned primarily with treating a pipeline as a component which may be included in a block diagram of a dynamic system. Considerable work has been done on finding the inverse Laplace transforms for various block diagram pipe models with the aim of predicting transient responses.

2.4.1 The block diagram approach

221. A pipeline affects the system to which it is connected only in so far as the pressures and flows vary at the ends of the pipe. The internal behaviour determines end conditions but is of no interest to the analyst, consequently the pipe may be represented as a black box with pressures and flows at its inlet and outlet, (Figure 2.1). This is commonly called the two port four terminal representation. A mathematical relationship is required between the four variables so that the component block can be expressed as a transfer function or a combination of transfer functions. Models have been developed which take account of linearised friction, there are also those with viscosity and heat transfer effects [2.16]. These more sophisticated models follow the same format as the lossless case. However the mathematics of the transfer functions is much more complicated.

222. The solution of the lossless linearised water hammer equations (2.7) and (2.8) are expressed as a set of time difference formulae {B6 page 86} which may be manipulated into the form.

$$P_O(t) + Z_O Q_O(t) = P_I(t - T_p) + Z_O Q_I(t - T_p) \qquad (2.16)$$

$$P_I(t) - Z_O Q_I(t) = P_O(t - T_p) - Z_O Q_O(t - T_p) \qquad (2.17)$$

Where Z_O is called the characteristic impedance of the pipe ($Z_O = \rho c/A$) and T_p is the time taken for a wave to travel the length of the pipeline

223. Equations (2.16) and (2.17) describe the progress of waves in a pipeline and the physical significance of the terms may be demonstrated by noting that $P_I(t - T_p)$ is the function $P_I(t)$ delayed by T_p the time required for a wave to travel the length of the pipeline. In Laplace notation a time delay is represented as e^{-Ts}, where T is the delay time and s is the Laplace operator. Hence the function $P_I(t - T_p)$ may be written as $P_I(t)e^{-T_p s}$ and similarly for all other functions in $(t - T_p)$.

224. In more general terms the delay term is written as $e^{-\Gamma(s)}$, where $\Gamma(s)$ is called the propagation operator. For a lossless line $\Gamma(s) = T_p s = ls/c$. Using the new notation equations (2.16) and (2.17) may be rewritten. Omitting the (t) since all the variables in P_I, P_o, Q_I and Q_o are evaluated at time t.

$$P_o + Z_oQ_o = P_Ie^{-\Gamma(s)} + Z_oQ_Ie^{-\Gamma(s)} \tag{2.18}$$

$$P_I - Z_oQ_I = P_oe^{-\Gamma(s)} - Z_oQ_oe^{-\Gamma(s)} \tag{2.19}$$

Equations (2.18) and (2.19) may be represented as a block diagram provided that any two variables are known so long as they do not occur at the same end of the system, this gives rise to four possible configurations (Figure 2.1(b)), where the known variables are treated as inputs to the component block and the unknown variables as outputs.

225. Considering configuration III (Figure 1.1(b)), Q_I and P_o are known, therefore equations (2.18) and (2.19) may be rearranged as follows:-

26 becomes $$Z_oQ_o = (P_I + Z_oQ_I)e^{-\Gamma(s)} - P_o \tag{2.20}$$

27 becomes $$P_I = (P_o - Z_oQ_o)e^{-\Gamma(s)} + Z_oQ_I \tag{2.21}$$

The block diagram from configuration 3 may be drawn directly from equations (2.20) and (2.21) (Figure 2.1(c)). Similar block diagrams may be drawn for the other three configurations [2.16].

226. In general all the properties of the pipeline are described by the characteristic impedance and the propagation operator. For a lossless line these terms are a real function and a pure time delay respectively. For more sophisticated models with friction, viscosity and heat transfer, the overall form of the block diagram remains the same however the characteristic impedance becomes a complex function and the propagation operator models attenuation of the wave as well as time delay.

227. Computer models based on the block diagram representation may be included in high level simulation languages such as CSMP (continuous system modelling program) [2.18].

2.4.2 The impedance method

228. The fundamental cause of fluid borne noise in hydraulic systems is the pulsating flow generated by positive displacement pumps or motors. Pumps produce a mean flow level with a superimposed periodic fluctuation. Fluid borne noise is defined as the magnitude of the pressure fluctuation produced as a result of the flow ripple interacting with the hydraulic system. The flow fluctuation is therefore the forcing function to which the system reacts, the ripple waveform is usually quite complicated however it can be split up by Fourier analysis into a series of sine waves, the fundamental being at the pumping frequency. The pressure ripple produced can be calculated by adding the system response due to each harmonic.

229. The theory to be presented is based on the simplified water hammer equations with linearised pipe friction [2.19] .

dynamic equation $$-\frac{\partial p}{\partial x} = \frac{\rho}{A}\frac{\partial Q}{\partial t} + Q.R. \qquad (2.22)$$

continuity equation $$-\frac{\partial Q}{\partial x} = \frac{A}{\beta}\frac{\partial p}{\partial t} \qquad (2.23)$$

By analogy with electrical engineering the following terms are defined:-

$$\frac{\rho}{A} = L \quad \text{and} \quad \frac{A}{\beta} = C$$

L is called the coefficient of inertance; equivalent to electrical inductance, C is called the coefficient of capacitance; equivalent to electrical capacitance, R is defined as the pressure drop/ unit length/unit flow under laminar flow conditions and is equivalent to electrical resistance. The assumption of linear friction is justified by noting that the flow ripple is a small perturbation about a mean level.

230. Equations (2.22) and (2.23) can be combined to give a pair of wave equations:

$$\frac{\partial^2 Q}{\partial x^2} = \gamma^2 Q \tag{2.24}$$

$$\frac{\partial^2 P}{\partial x^2} = \gamma^2 P \tag{2.25}$$

which have a general solution

$$P = Fe^{-\gamma x} + He^{\gamma x} \tag{2.26}$$

$$Q = \frac{1}{Z_o}(Fe^{-\gamma x} - He^{\gamma x}) \tag{2.27}$$

where $\gamma = (RCs + LCs^2)^{\frac{1}{2}}$ (2.28)

$$Z_o = \left[(R + Ls)/Cs\right]^{\frac{1}{2}} \tag{2.29}$$

γ is called the propagation constant, Z_o is called the characteristic impedance of the pipe. By analogy with electrical engineering, impedance is defined as $Z = P/Q$, where P and Q may be vector quantities.

231. Equations (2.26) and (2.27) are essentially very similar to the Reimann equations. The physical significance is best appreciated by considering the frictionless case. When friction is neglected $\gamma = s/c$ and $Z_o = \rho c/A$. In this case the expressions $e^{-\gamma x}$ and $e^{+\gamma x}$ becomes $e^{\mp \frac{sx}{c}}$, the former represents a delay of $^x/c$, in other words it describes the progress of a wave in the direction of increasing x, the latter represents a delay of $^{-x}/c$. The minus sign may be

visualised as a reversal of direction, the expression $e^{sx/c}$ therefore represents a wave moving in the direction of decreasing x. The constants F and H represent the magnitudes of the waves and can be evaluated for a particular set of boundary conditions.

232. As a matter of interest it can be shown that if the value of x is set to zero and ℓ to signify conditions at pipe inlet and outlet respectively equations (2.26) and (2.27) can be manipulated into exactly the same form as equations (2.18) and (2.19) in the previous section which shows that the mathematical basis of the block diagram approach is identical to that of the impedance method.

233. A pair of equations for pressure and flow at any point in the pipeline can be developed by considering the generalised system shown in Figure (2.2). The hydraulic system at either end of the pipeline is represented in terms of source impedance and termination impedance, the input is the known flow Q_s. By considering the boundary conditions the constants F and H are eliminated from equations (2.26) and (2.27) giving the following expressions.

$$P = \frac{Q_s Z_o}{(1 + Z_o/Z_s)} \left\{ \frac{e^{-\gamma x} + \rho_T e^{-\gamma(2\ell - x)}}{1 - \rho_T \rho_s e^{-2\gamma\ell}} \right\} \qquad (2.30)$$

$$Q = \frac{Q_s}{(1 + Z_o/Z_s)} \left\{ \frac{e^{-\gamma x} - \rho_T e^{-\gamma(2\ell - x)}}{1 - \rho_T \rho_s e^{-2\gamma\ell}} \right\} \qquad (2.31)$$

where $\rho_T = \dfrac{Z_T - Z_o}{Z_T + Z_o}$ and $\rho_s = \dfrac{Z_s - Z_o}{Z_T + Z_o}$

ρ_s and ρ_T are the **reflection coefficients** at the source and at the termination respectively. These coefficients give the relative magnitude of the reflected wave compared to the incident wave. Equations (2.30) and (2.31) are quite general and apply both to harmonic and to transient disturbances.

234. When considering harmonic inputs the propagation constant γ is replaced by the expression $(\alpha + j\omega/c)$, this is equivalent to replacing the Laplace operator s by the term $j\omega$, i.e. performing a Fourier transformation. The term α is the attenuation due to friction and the term $j\omega/c$ models the delay and gives a measure of the phase change as the wave progresses along the pipe, ω is the frequency of the forcing function Q_s.

235. For harmonic disturbance the pressure equation (equ. 2.30) becomes

$$P = \frac{Q_s Z_o}{(1 + Z_o/Z_s)} \left\{ \frac{e^{-\alpha x} e^{-j\omega x/c} + \rho_T e^{-\alpha(2\ell - x)} e^{-j\omega(2\ell - x)/c}}{1 - \rho_T \rho_s e^{-2\alpha\ell} e^{-j2\omega\ell/c}} \right\} \qquad (2.32)$$

Equation (2.32) may be used to calculate the pressure ripple at any point x in the pipe produced by the input flow fluctuation. As mentioned before the flow ripple can be expressed as the sum of a number of harmonics, usually 10 harmonics is sufficient. Each harmonic may be expressed as a vector with amplitude and phase $|Q_R| e^{j\omega_R t}$, Q_R is the magnitude of the R^{th} harmonic and ω_R is its frequency. Substituting $|Q_R| e^{j\omega_R t}$ for Q_s in equation (2.32) the response of the pipe to the R^{th} harmonic is calculated giving the R^{th} harmonic of the pressure ripple. Repeating the calculation for all the flow harmonics gives a series of pressure harmonics which when added vectorially give the resultant pressure ripple waveform. Generally the frequency spectrum of the pressure ripple is more useful in noise work than the actual ripple waveform.

236. Although equation (2.32) uses complex numbers and appears quite complicated it is relatively easy to evaluate by computer. The difficult part lies in finding values for the flow source ripple and the source and termination impedances. The values of Q_s and Z_s vary from pump to pump and are usually determined experimentally [2.20 , 2.21]. The termination impedance is the impedance of the rest of the hydraulic system as seen by the pipe being analysed. In work on noise ratings for pumps the termination is usually a simple orifice, however for more complex systems evaluation of the termination impedance presents considerable problems.

2.5 The method of characteristics

2.5.1 Modelling a pipeline

237. The method of characteristics is a technique where the two partial differential water hammer equations are converted into four total differential equations which are then solved numerically. The technique was devised by B. Reimann in 1860 and was applied to the study of surface flow and unsteady flow of gases by J. Massau in the 1890's. Initially the development of the method of characteristics was closely related to that of the graphical techniques. Essentially the early formulations of the method of characteristics were equivalent to the Schnyder-Bergeron graphical methods. In 1954 Gray [2.22] demonstrated the relationship between characteristic methods and graphical methods and presented a technique for using characteristics to solve surge problems in pipelines, the computation was performed by hand. By the middle of the 1960's computers were applied to the method of characteristics [2.23 , 2.24]. The power of the technique was demonstrated and it began to replace graphical analysis in civil engineering. Over the years various improvements were made to the computational technique to allow convenient and accurate solution of surge problems [2.25 , 2.26 , 2.27 , 2.28]. The general acceptance of the method was shown by the 1972 B.H.R.A. International Conference on pressure surges, where a number of papers presented work based on the method of characteristics [2.29].

238. The ability of the method of characteristics to handle non-linearities, friction, cavitation, etc. has led to its use in many fields. When studying the flow of gases thermodynamic effects may be included by considering the equation of state when deriving the characteristic form. {B1 page 147}. However this is not usually required when dealing with the flow of liquids. A detailed development of the method of characteristics will be given here since the technique forms the basis of the work described in this thesis.

239. The two partial differential equations may be expressed as four ordinary differential equations which define the propagation paths of waves and the variation of pressures and flows along the paths of the waves.

The water hammer equations

dynamic $$\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{\partial v}{\partial t} + \frac{v\partial v}{\partial x} + \frac{2fv|v|}{d} = 0 \qquad (2.1)$$

continuity $$\frac{v}{\rho}\frac{\partial p}{\partial x} + \frac{c^2\partial v}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial t} = 0 \qquad (2.2)$$

Characteristic form of the water hammer equations

forward compatibility equation $$\frac{1}{\rho c}\frac{dp}{dt} + \frac{dv}{dt} + \frac{2fv|v|}{d} = 0 \qquad (2.33)$$

backward compatibility equation $$-\frac{1}{\rho c}\frac{dp}{dt} + \frac{dv}{dt} + \frac{2fv|v|}{d} = 0 \qquad (2.34)$$

forward characteristic $$\frac{dx}{dt} = v + c \qquad (2.35)$$

backward characteristic $$\frac{dx}{dt} = v - c \qquad (2.36)$$

The derivation of the characteristics form of the water hammer equations is given in a number of texts {B1 , B2 , B7}.

240. Equation (2.35) and (2.36) define what are called the characteristic lines in the time distance plane (Figure 2.3). Equation (2.35) describes the progress of a wave in the direction of increasing x, it defines the curve AB in Figure (2.3) which is called the forward characteristic from A. At any point along this line pressure and flow conditions are related by equation (2.33). Similarly equation (2.36) describes the progress of a wave in the direction of decreasing x, it defines the line AC in Figure (2.3) which is called the backward characteristic from A. At any point along AC pressures and flows are related by equation (2.34). Equations (2.33) and (2.34) are called the compatibility equations, or the characteristic equations.

241. In effect this means that equations (2.35) and 2.36) define the location in time and space of waves propagating from a given point and equations (2.33) and 2.34) define the way in which the pressures and flows vary with time along the path of these waves.

242. The characteristic lines are curves because in real systems the flow velocity v and the wavespeed c are not constants. However for a great many hydraulic applications flow velocities are small compared to wavespeed and furthermore the variation in wavespeed is small, in which case the characteristics can be defined as follows.

$$\text{forward characteristic} \quad \frac{dx}{dt} = +c \tag{2.37}$$

$$\text{backward characteristic} \quad \frac{dx}{dt} = -c \tag{2.38}$$

Equations (2.37) and (2.38) are straight lines in the time-distance plane. This simplification is equivalent to neglecting the non-linear $v\partial v/\partial x$ term in the dynamic equation of water hammer and the $(v/\rho)(\partial p/\partial x)$ term in the continuity equation.

243. To solve the compatibility equations consider the effects of waves propagating from two separate points R and s in a pipeline (Figure 2.4). The forward characteristic from R intersects the backward characteristic from S at point P, therefore at point P the two compatibility equations apply simultaneously and may be solved by writing them in finite difference form.

(2.33) becomes:-

$$\frac{1}{\rho c}(P_P - P_R) + (v_P - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d} = 0 \tag{2.39}$$

(2.34) becomes:-

$$-\frac{1}{\rho c}(P_P - P_S) + (v_P - v_S) \quad \frac{2f_S v_S |v_S| \Delta t}{d} = 0 \tag{2.40}$$

$$\text{Forward characteristic} \quad \frac{\Delta x}{\Delta t} = +c \tag{2.41}$$

$$\text{Backward characteristic} \quad \frac{\Delta x}{\Delta t} = -c \tag{2.42}$$

It is assumed that the values of pressure, flow velocity and friction factor are known at points R and S , (P_R, v_R, f_R, P_S, v_S and f_S are known) these are the initial conditions or the results of previous calculation. Therefore equations (2.39) and (2.40) can be solved for P_P and v_P, the pressure and flow velocity at point P. Hence a knowledge of flow conditions at two points in a pipeline make it possible to calculate the pressures and flows at an intermediate point a time interval Δt later, where Δt is determined by the wave-speed in the pipe and distance apart of the two points at which conditions are known. This provides a basis for modelling pipelines and linking component models in a system simulation.

244. Generally the compatibility equations are solved by using a first order finite difference approximation, it is possible to use a second order however this involves iteration. A method known as Roberts technique [2.24 page 45] can be used to give second order accuracy using first order formulae and an extrapolation procedure. This is more efficient than using a second order finite difference formula directly. However for most problems a first order method is sufficiently accurate.

245. Expressing the characteristic equations in finite difference form implies that the solution of a system will be carried out at discrete points in the time distance plane. To model a pipeline it is necessary to define a number of equispaced calculation points along the length of the pipe at which pressures and flows will be evaluated.

246. Consider a pipe connecting two components which is divided into a number of Δx intervals (Figure 2.5). If at time t pressure and flows are known at points A, B, C, D, and E, characteristics with gradients $\Delta t/\Delta x = \pm 1/c$ may be drawn from these points. Where these lines cross at B′ , C′ , and D′ at time t + Δt, two compatibility equations apply so pressures and flows may be calculated. At the ends of the pipeline at points A′ and E′, only one characteristic line passes through each point and so only one compatibility equation applies. Therefore to obtain a solution at A′ and E′ the boundary conditions of the components must be considered. At each boundary there are two unknowns, pressure and flow velocity, but only one compatibility equation, the component model therefore must either supply the value of one of the unknowns, or supply another equation which may be solved

simultaneously with the compatibility equation for both the unknowns. Once conditions at A', B', C', D' and E' have been established the procedure described above may be repeated to calculate conditions a further timestep Δt later, and so on for as many timesteps as required.

2.5.2 Modelling a system

247. The steps taken to calculate the pressures and flows in a pipeline with components forming boundary conditions can be extended directly to model a complete system.

248. Consider a system of three components linked by pipes Figure 2.6a. Each pipe is divided into a number of Δx intervals. The system is represented on the time distance plane in Figure 2.6b. A grid is drawn on the t-x plane so that the intersection points indicate where the characteristic lines cross and therefore where direct calculation of pressure and flow is possible. (For clarity only a few of the characteristic lines have been included in Figure 2.6b.) To obtain a complete solution for the system at a given time level, the pressures and flows must be known at each calculation point of the previous time level. Conditions at interval points in each pipe are calculated by solving the two intersecting compatibility equations at each point. Conditions at boundary points are obtained from the component models. The ways in which components are modelled depend largely on the complexity of the mathematics required to describe the performance of the component and the complexity of the mathematics in turn depends on the application and the interests of the analyst. A more detailed discussion on the treatment of components will be given in the next section.

249. The simulation process is performed on a computer by repeatedly calling subroutines which model the components and pipes. A call to the subroutine for component 1 evaluates pressures and flows at calculation points A and M. Similarly the subroutine for component 2 solves for conditions at D and E and the subroutine for component 3 solves for conditions at H and I. A single pipe modelling subroutine is used to calculate conditions at all internal points in the pipelines. A separate call to the subroutine is made for each pipe. A sequence of call statements to component subroutines and the pipe subroutine calculates conditions throughout the system at a given time level. The solution then progresses one time step, the sequence of calls is repeated, using values from the previous time level to set up the

compatibility equations and the component equations. The order in which the subroutines are called is not important since the individual pipe or component subroutines do not require any information being calculated by other subroutines at a given time level.

250. The simulation starts with the system in a steady state at time level 1, where the pressures and flows at all calculation points are known. A disturbance is now initiated, at say component 1, which propagates from component 1 at the wavespeeds in pipes 1 and 3. The heavy dotted line plots the progress of the transient through the system. The points marked with a cross are locations in the t-x plane where pressures and flows are first altered by the disturbance, at points below the dotted line no effect of the disturbance is felt. Components 2 and 3 are unaware of events happening at component 1 until the wave reaches them. The method of characteristics models the time delay in the effect of a disturbance in the time domain, as do the now obsolete arithmetic and graphical methods, as opposed to the frequency domain methods where delay is implied as a phase shift at a given frequency.

251. The scheme described above for modelling systems shows the essential feature of a method of characteristics simulation. Certain features of real systems require a more complicated treatment and some of the different ways of applying the method of characteristics are outlined in the next section.

2.5.3 Different formulations of the method of characteristics

252. The condition that each pipeline in the system be divided into an integer number of Δx intervals does present certain problems. The more Δx intervals there are in a pipe the more accurate the pipe model. However the choice of a Δx implies a choice of Δt since $\Delta x/\Delta t = \pm c$. A small Δx implies a small Δt and therefore a large number of steps required to execute a simulation of a given duration, so as usual increased accuracy is paid for by increased execution time. Generally the choice of timestep is made by selecting the shortest pipe in the system which is to be modelled with distributed parameters and dividing

this pipe into a minimum of two Δx intervals. Then, on the basis of the value of wavespeed in that pipe the timestep Δt is calculated. Usually all pipe models operate to the same timestep so that when the solution of each pipe and component is advanced one step the results are synchronised. Using the value of Δt the corresponding value of Δx in each pipe can be calculated, it is very unlikely that the length of each pipe will be exactly divisible by the value of Δx to give an integer number of intervals.

253. This problem may be resolved in several ways. The pipe length may be altered to allow an integer number of intervals, alternatively the wavespeed may be slightly altered to give a value of Δx which is an exact integer divisor of the pipe length. Both these simple methods introduce inaccuracies into the simulation by altering the pipe period. The error is relatively easy to calculate beforehand and the user can select a value of Δt which gives acceptable errors.

Trikha [2.27] proposes a method where different timesteps are used in each pipe thereby completely circumventing the problem of awkward pipe lengths. The discrepancy between timesteps being compensated for in the boundary component models. Another method for accommodating different pipe lengths is a more sophisticated way of solving the integrated compatibility equations, and is called the method of specified time intervals. This method is also capable of dealing with a slight variation in wavespeed and can use the more accurate characteristic equations $dx/dt = v \pm c$ (equations 2.35 and 2.36) {B1 , B2}.

254. The time distance plane is divided into a regular rectangular grid (Figure 2.7). The purpose of the method is to find the values of the pressures and flows at the nodes of the grid. With reference to Figure (2.7), assuming all conditions are known at the nodes of the grid at time level 1 , i.e. at points A, B, and C. The two characteristic lines passing through P intersect the grid between A-B at R and B-C at S, this is because the grid is deliberatly drawn with Δt less than $\Delta x/(v + c)$. The values of pressure and flow at R and S are obtained by linear interpolation. By solving the compatibility equations, based on the values at R and S, the conditions at P are calculated.

255. The method of specified intervals may be applied in several ways depending on the required accuracy of the solution. Initially the gradient of the characteristics is calculated on the known value of v and c at node B, using this value of gradient the intersection points R and S are found, values of pressure and flow are interpolated and the conditions at P are found. For increased accuracy an iterative process can be devised where the characteristic gradient is updated by values of v and c at point R for the forward characteristic and values of v and c at point S for the backward characteristic. The new gradients define new intersection points and repeated interpolation and calculation of conditions at P is required. Alternative schemes use averaged values of v and c at points P and R for the forward characteristic gradient and similarly averaged values at points P and S for the backward characteristic gradient.

256. Specified intervals is a very powerful method however it must be applied with great care. The strength of the technique is in its ability to handle different characteristic gradients at all nodes in the t-x plane and thereby take account of different pipe lengths and varying flow velocities and wavespeeds. The disadvantages are the obvious complications in computing and the possibility of instability. The condition that $\Delta t < \Delta x/(v + c)$ is known as the COURANT and LEWY stability criterion. Referring to Figure 2.7, if the stability criterion is not satisfied the characteristic gradient will intersect outside A-B, making interpolation impossible. Interpolation and iteration produce numerical errors which introduce an artificial damping to the solution. Higher order interpolations are possible, these improve accuracy at the expense of added complication.

257. Several other formulations of the method of characteristics have been developed, each has certain advantages under specific circumstances. One major technique is known as the characteristic grid which avoids interpolation by calculating the solution at irregular positions in the time distance plane. However, results presented in this way are awkward to use and some interpolation is necessary. STREETER and WYLIE B2 describe the method and discuss its advantages and disadvantages with respect to a regular rectangular grid method.

258. Irrespective of which formulation of the method of characteristics is used it is necessary to model components which form the boundary conditions.

2.5.4 Component modelling within a method of characteristics system simulation

259. As mentioned before only one integrated compatibility equation applies at a pipe boundary and this is an equation in two unknowns, pressure and flow. To make a solution possible either one of the two unknowns must be specified, or another equation must be supplied which relates pressure and flow. The approach adopted depends largely on the complexity of the component and the specific interest of the analyst. Component models can be divided into three broad categories; steady state models where the component behaviour is described by algebraic equations, simple dynamic models where differential equations can be expressed as algebraic finite difference formulae, and more complicated dynamic models where finite difference integration is inadequate and a more sophisticated numerical integration technique must be used.

(a) Steady state models

260. Steady state models are used in circumstances where the component response is very rapid compared to the system response, its dynamic effects are negligible and it may be considered to act instantaneously. In general the steady state performance of any hydraulic component can be expressed as an algebraic relationship between pressures and flows at the ports of the component and various internal parameters decribing the physical structure of the component. Each pipe connection supplies the model with a compatibility equation which in its finite difference form is an algebraic equation. The simultaneous solution of the algebraic component equations with the relevant compatibility equations defines the conditions of pressure and flow at the component as determined by the interaction of the component and pipeline.

261. The method used to solve the simultaneous equations depends on the complexity of the situation. For very simple systems of three or four linear equations it is possible to solve the equations by hand and obtain expressions for each of the unknowns. These expressions may be programmed directly into a component model subroutine.

For linear sets of equations with a large number of unknowns, the equations may be expressed as a matrix and solved using a standard matrix solving technique such as Gaussian Elimination. Where the equations are non linear a numerical non linear equation solver (Newton Raphson) may be used, or the equations may be linearised using small perturbations. Each component has to be considered on its own merits and the solving technique chosen on the basis of accuracy, ease of programming and computer time required to obtain a solution.

(b) Simple dynamic components

262. The basic integration step length for a method of characteristics system simulation is the value of Δt selected by considering the shortest pipe in the system. For some dynamic components this value of Δt may be sufficiently small to give reasonably accurate simulation using a Simple Euler, or a Backward Euler integration. Under these circumstances the differential equations describing the dynamics of the component may be expressed as algebraic finite difference equations, and the solution performed as described above.

(c) Complicated dynamic components

263. When component dynamics are described by a stiff system of differential equations simple explicit integration techniques are inadequate. More sophisticated methods are required and expressing the integrated equations in algebraic form is not feasible. In this case the component model subroutine simply supplies a value of one of the required unknowns which enables the other unknown to be calculated directly from the integrated compatibility equation. The following simple scheme describes one way of interfacing a dynamic component model with a method of characteristics pipeline model.

264. Consider a case where a component with complex dynamics is connected to one pipeline. Figure 2.8a. The interaction between the pipe and component may be represented as a transfer of information. In this example solving the component equations yields a value of pressure and solving the pipe compatibility equation gives a value of flow. Figure 2.8b shows how the integration schemes for pipe and component may be interfaced. The end of the pipe is shown on the t~x plane, conditions at O and S are known and the pipe integration step length is Δt. The simplest approach to adopt would be to consider the flow

Q_t as a constant over one Δt interval as far as the component model is concerned. Using the known values of P_t and Q_t the component equations could be integrated in a number of steps resulting eventually in a value of pressure at time $t + \Delta t$, ($P_{t + \Delta t}$). At time $t + \Delta t$ the compatibility equation associated with the backward characteristic from S applies at the boundary. Using the now known value of $P_{t + \Delta t}$ in the compatibility equation allows $Q_{t + \Delta t}$ to be calculated and so the boundary condition is fully defined. Accuracy may be improved by updating the value of Q at every integration step. In figure 2.8b, the end of the first integration step δt, coincides with a backward characteristic from point A. Conditions at A may be interpolated from known values at points O and S. Solving the compatibility equation associated with the characteristic from A gives a new value of Q which may be used in the next component integration step. This process may be repeated for as many steps as are required to integrate over the Δt interval.

265. In this scheme the integration technique used is immaterial, a great many numerical integration methods have been developed and the choice depends on factors such as; accuracy required, computing time, complexity of the algorithm and so on.

2.6 Other numerical methods

266. There are a wide variety of finite difference methods which may be used to solve the **water hammer equations directly, without conversion into** the characteristic form. STREETER [2.25] describes a typical method called the centred implicit method and compares it to the method of characteristics. The method involves dividing a pipeline into a number of cells or elements, where each cell is described by two finite difference equations. To obtain a solution for the pipeline over one timestep all equations describing the cells plus two boundary conditions have to be solved simultaneously. The equations are non linear and are usually solved using a Newton-Raphson technique. The implicit method allows various values of Δt and Δx to be used throughout the simulation and it does not suffer the restriction that $\Delta t \leqslant \Delta x(v + c)$ as does the method of characteristics, also friction is modelled extremely accurately. However the implicit method does not simulate sharp transientsaccurately and it can suffer from instability. From a computing point of view the method of characteristics is simpler to program and more economical for the same values of Δx and Δt.

267. There are many similar methods such as the Lax-Wendroff method, the leap frog method etc. etc. However Fox B1 expresses the opinion that for water hammer problems the method of characteristics using specified intervals is superior.

268. Finite element analysis is another method which is finding wide acceptance in fluid mechanics. Originally developed for structural analysis it was recognised that the method is applicable to a wide range of problems in continuum mechanics. Equations in fluid dynamics such as the fundamental Navier Stokes equations or the more specific water hammer equations describe the continuous variation between parameters such as pressure, flow, etc. The analytic solution would involve finding an expression which allows, for a given set of boundary conditions, the values of pressure, flow, etc. to be calculated at any point in the pressure~flow~time~distance plane (called the solution field). Such an expression would be a continuous solution. Unfortunately no such analytical expression is available. In the finite element method the solution is carried out by dividing the solution field into a number of elements where the mathematical description of each element is considerably simpler than the mathematical description of the whole field. The solution is no longer continuous but is limited to nodes located on the element boundaries where the values of independent variables are evaluated. The properties of each element must be known so that it is possible to express the variation of an independent variable in terms of the properties of the element and the values of the variables at the nodes. One strength of the finite element method is that it can handle a large number of independent variables simultaneously by expressing them in matrix form and manipulating them using matrix algebra.

269. Once the behaviour of each element is known, the overall solution is obtained by noting that a node is generally common to a number of elements, therefore the solution of these elements must give the same values at the common node. Also certain nodes are located at the boundaries of the system where boundary conditions of pressure and flow, etc. will be imposed. The continuity condition at common nodes and the boundary conditions are expressed mathematically by assembling the individual element matrix equations into an overall system matrix equation, and the solution of this equation gives values of the independent variables at each node. A good introduction to finite

element analysis is given in several text books {B8 - B11}.

270. Finite elements is a very powerful computer method which is best suited to very complex multidimensional problems. The solution of the one dimensional form of the water hammer equations is too simple a problem to warrant using such an advanced technique.

3. DIFFICULTIES ENCOUNTERED WHEN MODELLING PIPELINES

3.1 Frequency dependent friction

301. Frictional pressure drop is due to energy dissipation caused by viscous shearing of the fluid. In both laminar and turbulent flow the shearing occurs in the boundary layer very close to the pipe walls. Zielke [3.1] explained the unsteady frictional effect as follows: The boundary layer and the fluid at the centre of the pipe are affected differently by a time varying pressure gradient. Frictional forces are dominant in the boundary layer, inertia forces are small, therefore the velocity of the fluid near the pipe wall is in phase with the pressure gradient fluctuation. At the centre of the pipe inertia forces dominate and so it is the acceleration of the fluid rather than the velocity which is in phase with the pressure gradient fluctuation.

302. A sudden change of pressure gradient (high frequency effect) will first alter the fluid velocity in the boundary layer increasing the shear stress before there is a significant change of fluid velocity at the pipe centre. Rapid changes of flow velocity in the boundary layer increase energy dissipation, but because the layer is very thin, the mean flow velocity is unaffected. This is why high frequency components of a disturbance are attenuated more rapidly than low frequency components. Steady state friction equations neglect the velocity fluctuations at the pipe wall basing their calculations on the mean flow as a result frictional energy dissipation under unsteady conditions is underestimated.

303. Early work on unsteady flow was concerned with water hammer analysis in large bore pipelines usually carrying water, frictional pressure drops were small and the steady state friction equations gave adequate results. The development of fluid control systems with small diameter pipes carrying viscous fluids (oils) required accurate modelling of friction. An associated problem was frequency dependent compliance due to heat transfer in pneumatic lines. Iberall (1950) and Nichols (1962) derived frequency response characteristics of fluid lines with frequency dependent friction and heat transfer. Brown 1962 [3.2] extended the work to obtain the

transient response.

304. Their models were based on a one dimensional form of the Navier Stokes equation

$$\nu_o \frac{1}{r} \frac{\partial}{\partial r} \left[\frac{r \, \partial v_x}{\partial r} \right] - \frac{\partial v_x}{\partial t} = \frac{1}{\rho_o} \frac{\partial p}{\partial x} \qquad (3.1)$$

(suffix O indicates time averaged value)

v_x is the axial velocity of the fluid. Radial velocity components are negligible. This equation is one dimensional only in pressure, i.e. it is assumed that the pressure is constant at a crossection. Changes in velocity profile are described by the $\partial v_x/\partial r$ term. This equation is only valid for small amplitude laminar disturbances.

305. The wave equations given in the previous section are:-

$$\frac{\partial^2 Q}{\partial x^2} = \gamma^2 Q \qquad (2.24)$$

$$\frac{\partial^2 P}{\partial x^2} = \gamma^2 P \qquad (2.25)$$

and may be written in the form:-

$$Q = \frac{1}{Z(s)\ Y(s)} \frac{\partial^2 Q}{\partial x^2} \qquad (3.2)$$

$$P = \frac{1}{Z(s)\ Y(s)} \frac{\partial^2 P}{\partial x^2} \qquad (3.3)$$

since $\gamma = \sqrt{Z(s)\ Y(s)}$

Z(s) is the series impedance of a unit length of line and represents momentum effects. Y(s) is the shunt admittance of a unit length of line and represents compressibility effects. The characteristic impedance used in the previous section is given by $[Z(s)/Y(s)]^{\frac{1}{2}}$ and the propagation operator $\Gamma(s)$ is given by $\Gamma(s) = \ell\sqrt{Z(s)\ Y(s)} = \ell\gamma$.

306. Brown took a fluid pipeline as a series of tiny elements of length dx inside which the fluid is considered incompressible (Figure 3.1) with shunt admittances in parallel to take account of compressibility. By taking Laplace transforms of equation (3.1) and finding a general solution in the s domain then combining with equation (3.2) and (3.3) Brown obtained expressions for Z(s) and Y(s)

$$Z(s) = L_o s \Bigg/ \left[1 - \frac{2J_1\left(ja\sqrt{s/\nu_o}\right)}{ja\sqrt{s/\nu_o}\ J_o\left(ja\sqrt{s/\nu_o}\right)}\right] \qquad (3.4)$$

$$Y(s) = C_o s \left[1 + 2\left(\frac{Cp}{Cv} - 1\right) \frac{J_1\left(ja\sqrt{\frac{\sigma o s}{\nu_o}}\right)}{ja\sqrt{\frac{\sigma o s}{\nu_o}}\ J_o\left(ja\sqrt{\frac{\sigma o s}{\nu_o}}\right)}\right] \qquad (3.5)$$

Equations (3.4) and (3.5) are exactly the same as those derived by Iberall and Nichols except theirs have $j\omega$ substituted for s. In this work we are concerned primarily with pipes carrying liquids in which case $Cp/Cv \simeq 1$ and the expression for Y(s) is greatly simplified to:-

$$Y(s) = C_o s \qquad (3.6)$$

where

$$C_o = \frac{\pi a^2 \rho_o}{\beta}$$

307. The characteristic impedance and the propagation operator can be evaluated using the expression for Z(s) and Y(s) (equations (3.4) and (3.6)). And then the pressures and flows in the pipe can be evaluated using equations (2.18) and (2.19) of the previous section

$$P_o + Z_o\, Q_o = P_I e^{-\Gamma(s)} + Z_o\, Q_I e^{-\Gamma(s)} \qquad (2.18)$$

$$P_I - Z_o Q_I = P_o e^{-\Gamma(s)} - Z_o Q_o e^{-\Gamma(s)} \qquad (2.19)$$

The procedure for obtaining a solution in the frequency domain involves substitution of $j\omega$ for s and representing equations (2.18) and (2.19) as a 4 terminal 2 port network. The transfer functions for such an approach are given by Goodson and Leonard [2.16].

308. Calculating the transient response is considerably more difficult. The characteristic impedance is given by a complicated expression which makes analytical inverse Laplace transformation impossible. Therefore the solution given by equations (2.18) and (2.19) cannot be transformed to the time domain directly.

309. To make an analytical time domain solution possible Brown used a simplifying equation for the Bessel functions in expressions (3.4) and (3.5). He derived approximate expressions for the characteristic impedance Z_o and the propagation operator $\Gamma(s)$ which are only valid for high frequencies or short transient times. Brown only considered the response of a semi-infinite line, in other words, only the behaviour of waves travelling in the downstream direction was taken into account. This approach is valid either for very long lines or for short transients where the disturbance at a particular point has died down before any reflections have had time to return. The high frequency approximation made inverse Laplace transformation possible and expressions were derived for pressure transients due to an impulse and step input.

For a semi-infinite line equation (2.18) and (2.19) become

$$\frac{P_{downstream}}{P_{upstream}} = e^{-\Gamma(s)} \qquad (3.7)$$

The approximation gives good results for non dimensional frequencies $a^2\omega/\nu \geqslant 10$. Which implies that the responses are only accurate for the high frequency components of the input function.

Holmboe and Rouleau (1967) [3.3] presented experimental results independantly verifying Brown's approach.

310. In 1965 Brown and Nelson extended this work to cover the entire frequency range [3.4] Simplifying expressions were used for the Bessel functions in the high frequency and low frequency ranges making analytical solutions possible. The mid frequency range could not be simplified and a numerical procedure was used to convert the frequency response results to step response in this range. The authors stated that this "proved to be a very complex and tricky business". Again the results only applied to semi-infinite lines, however examples were given of how reflections can be taken into account by superposition.

311. A different approach to transients was tried by Oldenburger and Goodson (1964) [3.5]. The work in this paper deals only with pipe models incorporating linear friction although in the discussion the authors show how their technique can be applied to pipes with frequency dependent friction. The solution of the water hammer equations is presented as a transfer matrix with hyperbolic functions, this is another way of representing a 4 terminal two port network [2.16]. The transfer matrix can be combined with suitable boundary conditions to give an overall transfer function for a simple system consisting of a pipe with some sort of terminations, the technique is described in references [3.5 and 2.16]. The system transfer functions are in terms of hyperbolic functions. Oldenburger and Goodson express the hyperbolic terms as an infinite series of products. Retaining only a few terms of the infinite products gives an approximate transfer function which can be inverse Laplace transformed to obtain the transient response of the system.

312. Unlike Brown's technique this approach takes account of reflected waves automatically, however, as Brown pointed out in the discussion to this paper, the number of terms required in the approximation depends on the band width of the solution required. The larger the band width required the more terms are required in the infinite products which makes inverse Laplace transformation very involved and results in complicated expressions for the transient response, therefore this

method is only practical in cases of narrow band excitation.

313. Later that same year (September 1964) D'Souza and Oldenburger [3.6] presented a set of frequency response expressions derived from the solution of the same one dimensional form of the Navier Stokes equation as used by Brown. It was concluded that the dynamics of a line are characterised by the parameter $a.(\omega/\nu)^{\frac{1}{2}}$, and when this tends to infinity as in the case of high frequency oscillation of a low viscosity liquid in a large diameter pipe, the friction may be neglected in the frequency response analysis. Unfortunately although the paper describes experimental verification of the theoretical expressions no indication is given for a value of $a.(\omega/\nu)^{\frac{1}{2}}$ which is large enough to be considered infinite for practical purposes.

314. Also in 1964 Foster and Parker [3.7] developed their theory for the frequency response of a hydraulic pipeline. The work was based on the one dimensional form of the Navier Stokes equation equation (3.1) , however the equation of continuity and momentum were used in the following form.

Momentum $$-\frac{\partial p}{\partial x} = \rho' \frac{\partial \bar{v}}{\partial t} + k'\bar{v} \qquad (3.8)$$

Continuity $$-\frac{\partial \bar{v}}{\partial x} = \frac{1}{\beta_E} \frac{\partial p}{\partial t} \qquad (3.9)$$

This is the standard form of the water hammer equations with linear pipe friction, equivalent to equations (2.22) and (2.23) in the previous section. The ρ' term is an apparent density and takes account of inertia effects under oscillatory flow and k' in the oscillatory friction factor representing the effects of viscosity. The values of ρ' and k' are determined by the parameter $a(\omega/\nu)^{\frac{1}{2}}$ denoted by ha. Three ranges are considered; low frequency or high viscosity where $ha \leq 1$, high frequency or low viscosity where $ha \geq 10$, and a transitional stage for values of ha between one and ten. The variables ρ' and k' are evaluated using different expressions in each of the three frequency ranges, thus making equation (3.8) equivalent to the one dimensional Navier Stokes equation. This approach to modelling fluid friction is now widely used in noise analysis work using the impedance method.

315. Over the years various approximate analytical techniques were tried for transient simulation. Brown's and Olderburger's methods were considered cumbersome and Karam (1972) [3.8] proposed a simple model for calculating the step response of semi-infinite lines. Three frequency ranges were defined and by inverse transforming the simplified frequency response expressions in the high and low frequency ranges Karam obtained an function in the time domain for the pressure ratio between an upstream and downstream station for a step input. No transform was available for the mid-frequency range and it was assumed that the low and high frequency results would overlap to cover the mid range. Karam's expression was fairly simple and could be used to calculate the response of a line without the aid of a computer. In 1973 Karam and Leonard [3.9] proposed a simple model for a finite line with arbitrary terminations. They considered the three dominant characteristics of a transmission line as: the delay of a signal, its attenuation, its dispesion of high frequencies. The model was proposed in the form of a 4 terminal 2 port network figure (3.2). Lumped resistance is used at the upstream port and the characteristic impedance is treated as a constant although strictly it is a function of time. This assumption is justified under some conditions and experimental verification was presented, however the generality of the model is weakened and results could be no more accurate than those calculated from a frictionless model (Trikha's comment).

316. Further refinements were published by Karam and Tindall (1975) [3.10]. The line resistance and the time constant of the 1st order lag in figure (3.2) were non-dimensionalised and their values adjusted empirically until a best fit was obtained between the results of the approximate model and those of a more complete model. The approximate solution covered slow transients, where friction was determined by a non dimensional form of the steady state equations for laminar and turbulent flow, and fast transients, where resistance was approximated by an empirical equation. The authors conceded that no criterion was available to judge what was a fast transient and what was slow, and the only model to their knowledge which automatically handled fast and slow transients was that proposed by Trikha (this model is discussed further below). However the claim was made that given suitable

transitional criteria their model would be more economical in computation.

317. The difficulty of calculating transient response by analytical methods prompted the development of numerical techniques for use with the method of characteristics. Zielke (1968) [3.1] used from boundary layer theory, the equation of motion for parallel axisymmetric flow of an incompressible fluid to derive a friction term for unsteady laminar flow.

$$\frac{\partial^2 v}{\partial r} + \frac{1}{r}\frac{\partial v}{\partial r} - \frac{1}{\nu}\frac{\partial v}{\partial t} = \frac{1}{\nu\rho}\frac{\partial p}{\partial x} \qquad (3.10)$$

$\partial p/\partial x$ is the pressure gradient producing the unsteady flow fluctuations, therefore $\partial p/\partial x$ is a function of time.

318. The partial differential equation (3.10) is Laplace transformed to yield an ordinary differential equation in the s domain. This has an analytical solution which expresses the transform of flow velocity as a function of the transformed pressure gradient and some Bessel functions. By manipulating the solution in the s domain Zielke obtained a transform which relates the wall shear stress to the acceleration of the fluid. The acceleration is taken as the rate of change of mean velocity at a crossection.

Inverse Laplace transformation yields the following expression:-

$$To(t) = \frac{4\rho\nu V(t)}{a} + \frac{2\rho\nu}{a}\int_o^t \frac{\partial V(u)}{\partial t}.W(t-u)du \qquad (3.11)$$

To is the frictional shear stress at the pipe wall. V(t) is the instantaneous mean velocity at a crossection. The first term on the right hand side of equation (3.11) represents the steady shear stress due to the mean velocity, the second term is the unsteady shear due to fluctuations in the mean flow. W is a known function of dimensionless time τ ($\tau = (\nu/a^2)t$) and can be calculated from a series for a given value of τ. The second term is in the form of a convolution integral which can be visualised as certain weights being given to past velocity changes to calculate their contribution to the total shear stress. Once the frictional shear stress is known the corresponding frictional pressure drop/unit length can easily be calculated by $P_f = 2\tau_o/a$.

319. If the method of characteristics is used to calculate the transient response, then the flow history at a number of calculation points is known and equation (3.11) can be expressed in finite difference form where the convolution integral is expressed as the summation of a series of terms containing previously calculated flow values. Zielke gives a criterion to help judge how many terms are required in this series, or in other words, how many values of the previous flow velocities have to be stored at each calculation point, this number can be quite high. For an average viscosity hydraulic oil (30cS) flowing in a 12 mm pipe with an integration step length Δt of 0.001 sec, about 100 values of flow have to be stored for each calculation point. The smaller Δt gets the greater the number of flow values required. Large systems or fast acting systems where Δt has to be small can require huge amounts of computer storage when modelling unsteady friction in this way.

320. Brown (1969) [3.11] reformulated Zielke's method in a more general form and extended its application in conjunction with the method of characteristics to cover a class of problems represented by semi hyperbolic partial differential equations. He applied the method to frequency dependent heat transfer as well as friction, however the method still used weighting functions and needed a considerable amount of computer storage. It was not until 1975 that a practical formulation of Zielke's method was made available.

321. Trikha [3.12] developed an approximate expression for the weighting function $W(\tau)$ which decreased the required computer storage quite dramatically. The weighting function was approximated by the sum of three exponential functions, this reduced the storage requirement to only four values at every calculation point along the length of a pipe. Comparison with frequency domain methods showed that Trikha's approximate method starts to loose accuracy for values of

the dimensionless parameter $a(\omega/\nu)^{\frac{1}{2}}$ greater than 150. For an average viscosity hydraulic oil (30cS) flowing in a 12 mm diameter pipe this corresponds to a maximum frequency of about 3000 Hz. If accuracy is required at higher frequencies more terms can be used to approximate the weighting function. Trikha's work is described in greater detail in the program documentation for the fluid friction subroutine.

322. All the work described so far is based on the direct or indirect solution of the one dimensional form of the Navier Stokes equation, and as such only applied for small amplitude disturbances in a laminar flow.

323. Frequency behaviour with turbulent flow was investigated by Brown, et al. in 1969 [3.13]. The analysis was semi-empirical and quite complicated. Solutions were presented for three frequency ranges employing a two or three region boundary layer model (three region model required at high frequencies). Expressions for various break frequencies were given to indicate which model was most suitable in a given frequency range. The results were presented in the frequency domain.

324. Wood and Funk (1970) [3.14] presented a simpler single boundary layer model for transient analysis of turbulent flow. It was assumed that all viscous effects occurred in a laminar boundary layer at the pipe wall whilst there was slug flow in the centre of the pipe. A knowledge of flow history was required and in this respect the model was similar to Zielke's and Trikha's models for laminar flow. A further paper in 1974 [3.15] applied the single boundary layer model to sinusoidal disturbances.

325. Trikha [3.12] suggested, with reservations, that since frequency dependent losses occur in the boundary layer, equation (3.11) could be used to model frequency effects under turbulent flow provided that the 1st term of the RHS, the steady state laminar shear is replaced by the corresponding equation for the steady state turbulent shear. With the implicit assumption that the contribution of unsteady shear is the same for both laminar and turbulent flow. As a first approximation

this approach seems reasonable allowing turbulent flow some degree of frequency dependance although experimental verification would be required before this method could be used with confidence.

3.2 Vaporous cavitation and gas release

3.2.1 Introduction

326. All liquids contain dissolved gases, the amount held in solution depends on the respective properties of the liquid and gas and on the pressure, called the saturation pressure, at which they were exposed to one another. For most fluid power applications the gas in question is air, predominantly nitrogen, dissolved at atmospheric pressure. If at any point in the system the static pressure drops below the saturation pressure a certain amount of air will come out of solution in the form of bubbles. The process is not instantaneous and the condition of low pressure must be sustained for some time before bubbles appear. Likewise when a gas bubble passes into a region of high pressure re-solution takes a finite time. Generally the rate of gas evolution is faster than the rate of solution and in most of the work discussed below it is assumed that once a gas bubble has formed, it does not re-dissolve.

327. Vaporous cavitation results only when the static pressure in a system drops below the vapour pressure of the liquid. Local boiling occurs and a vapour bubble is evolved very rapidly. The collapse of a bubble when it passes into a region of high pressure is also very rapid giving rise to large local pressures. Should bubble collapse occur near to a metal surface the pressures generated can be sufficiently violent to knock grains of material from the surface producing the easily recognisable cavitation damage often seen on propellers, in turbomachinery and in high pressure valves. The almost instantaneous nature of bubble evolution and collapse distinguish cavitation from aeration. The former is very violent producing high frequency pressure fluctuations which cause significant damage and produce a sharp crackling noise. The fluctuations are localised and rapidly damped out by frequency dependent friction, and

it has been shown, have little effect on the overall transient propagation in the system. Aeration is a more gentle phenomenon characterised by a hissing noise, local effects are not so drastic however the persistence of gas bubbles once evolved affects the wavespeed and causes significant changes in the transient behaviour.

328. The vapour pressure of hydraulic oils is very low thus vaporous cavitation is almost inveriably preceeded by some degree of gas release which has often led to the confusion of the two phonomena. The formation of bubbles be they gas or vapour depends on the presence of microsopic nuclei, either particles of contaminant or tiny gas bubbles in the fluid or attached to the pipe walls. Tests have shown that in the absence of micronuclei liquids can sustain considerable negative pressures or tensions. It is fair to assume that in industrial hydraulic fluids there are always sufficient micronuclei present to ensure that cavitation or air release will always occur if pressures are low enough.

329. This section deals with two areas of system modelling where cavitation and aeration may affect the accuracy of a simulation. Firstly flow in pipelines where the presence of bubbles affects the wavespeed and the dynamic energy dissipation. And secondly in flow through small orifices where the interchange between static head and velocity head as the fluid accelerates may result in very low local pressures and the evolution of vapour and air bubbles despite the fact that the nominal upstream and downstream pressures are well above the saturation pressure.

3.2.2 Cavitation in pipelines

330. In 1967 Baltzer [3.16] proposed a model based on the method of characteristics where a vapour cavity manifested itself as a long bubble which did not entirely fill the bore of the pipe such that there was a volume of liquid below the bubble with a free surface. The solution procedure involved treating the pipe as a system with

regions of full pipe flow and regions of free surface flow. The model was tested against an experimental set up where the flow in a pipeline was disturbed by the rapid closure of a valve, the fluid downstream experienced a rapid decrease in pressure a low pressure wave propagated down the pipeline to a volume tank which reflected a high pressure wave back to the valve. The study predicted the first high pressure peak accurately but the phasing and amplitude of subsequent peaks was in error.

331. Weyler et al [3.17] published an improved model in 1971, taking account of the diffusion of air into the vapour cavity. Energy losses due to repeated expansion and contraction of the bubble were treated as a dissipative mechanism analagous to fluid friction. Employing the method of characteristics the bubble expansion rate and internal pressure were used as velocity and pressure boundary conditions for the regions where the pipe ran full. Baltzer's and Weyler's models were concerned with the flow of water plus dissolved air in long (100 ft. to 300 ft.) pipelines.

332. A study by Swaffield (1972) [3.18] on a pipeline carrying aviation kerosene treated the cavity formed as column separation rather than a region with free surface flow. The method of characteristics was used with the cavity expansion and pressure as boundary conditions. Two models were considered, one where the cavity was pure vapour, the other where the cavity was a mixture of vapour and air. Experiment showed that the vapour plus air model was more accurate. The evolution of gas was governed by Henrie's Law which does not take account of the rate of evolution only the amount of gas released at a given pressure. Driels' 1973 [3.19], improved Swaffield's model by including the rate of gas release using the Schweitzer and Szebehely exponential formula. The amplitude and occurrence of the first pressure peak was predicted accurately however amplitude and phase errors were experienced with subsequent peaks.

333. A more complicated model by Wiggert and Sundquist 1973 [3.20]

improved accuracy for the entire transient solution. Vaporous cavitation was assumed to manifest itself as column separation whereas the gas released was treated as small bubbles uniformly spread through the liquid. A momentum equation was derived for the gas-liquid mixture which was converted into three characteristics in the time distance plane. Two of these characteristics were the normal fluid flow characteristics described in chapter 2, the third characteristic was associated with a compatibility equation relating pressure and void fraction. Void fraction is the ratio of the volume of free gas to the volume of the gas-liquid mixture. This compatibility equation was combined with the ideal gas law and a polytropic gas equation and was integrated along the relevant characteristic. Terms within the momentum equation related the rate of gas release using a diffusive bubble growth model attributed to Bousinesque (1905). Column separation due to vaporous cavitation acted as boundary conditions to sections of pipe carrying two phase flow.

334. An alternative method of solution was adopted by Karam 1974 [3.21] by re-working Driels' model into a 2 port - 4 terminal network and including the effect of frequency dependent friction. However since the system for which Driels' model was developed was very slow, a period of 1/2 second, the inclusion of frequency dependent friction has no effect on the accuracy of solution. However Karam's model is quite general and can be used for modelling higher frequency transients.

335. All the above models dealt with the flow of water or kerosene in long pipelines at relatively high Reynold's numbers. The transients were slow and the air released had significant effects. Work by Edge 1975 [3.22] showed that in well designed fluid power systems any air released is flushed out of the system in the first few minutes of operation and thereafter aeration has little effect on the transient behaviour. Yamaguchi et al., 1977 [3.23], investigated column separation in the flow of hydraulic oil at low Reynold's numbers and high frequencies (approximately 750 Hz). One conclusion from their

study was that if a cavity was small and collapsed instantly on the arrival of a high pressure wave the influence of the cavity on the period of the transient was negligible. This implied that the velocity of the cavity-liquid interface was the same as the fluid velocity in the absence of a cavity and the diffusion of gases into the cavity was negligible. Since the transients are fast and the cavity small, it is reasonable to assume that dissolved air does not have time to leave solution.

336. In the light of these findings and those of Edge a very simple cavitation model was adopted for the programs developed in this thesis. At any point in the system if the calculated pressure is negative vaporous cavitation is assumed to be taking place, the program resets the pressure to zero, however the flow velocity and the wavespeed are not modified. The vapour pressure of most oils is very low and setting the pressure to zero introduces negligible error. Physically the model implies a total absence of aeration and represents a vapour cavity which does not impede the progress of a wave.

3.3.3 Flow characteristics of valves under cavitating conditions

337. Generally flow through orifices is characterised by a square law formula of the form;

$$Q = C_d . A \sqrt{\frac{2}{\rho}(P_u - P_D)} \qquad 3.12$$

Where P_u and P_D are pressures upstream and downstream of the orifice and C_d the discharge coefficient. As a liquid passes through an orifice it forms a vena contracta where flow velocity is considerably higher than in the upstream and downstream regions. Consequently the static pressure at the vena contracta is reduced by the velocity head ($\rho V^2/2$) required by Bernoulli's equation. There may be some pressure recovery such that the pressure at the downstream gauge is higher than at the vena contracta. A condition arises where although the downstream pressure is well above vapour pressure, cavitation is occurring at the vena contracta. Researchers in this field have defined a cavitation number σ;

$$\sigma = \frac{P_u - P_D}{\frac{1}{2}\rho V^2} \qquad 3.13$$

which is the ratio of the static pressure opposing cavitation to the dynamic pressure trying to produce it. McCloy and Martin {B12} introduce the concepts of cavitating flow in orifices.

338. The pressure recovery from the vena contracta depends on the geometry of the flow path and the location of the downstream pressure gauge. A number of researchers have investigated flow through different types of orifice and have established criteria for the inception of cavitation in terms of a critical cavitation number. Lichtarowicz et al. [3.24] considered flow through long tube orifices and established equations for the discharge coefficient in terms of the cavitation number. MacLellan et al. [3.25] investigated flow through an idealised spool valve, and Stone [3.26] looked at flow through poppet type valves. Many other papers were published dealing with various orifice configurations [3.27 , 3.28 , 3.29]. The papers by Stone and MacLellan revealed that the discharge coefficient of the orifices tested was very dependent upon the geometry of the downstream chamber. MacLellan changed the position of a downstream boundary wall and observed the flow pattern. Of particular importance was whether the flow formed a free jet, was attached to one of the boundary walls or formed a laminar separation bubble. Cavitation was initiated in the shear layer that formed the surface of the separation bubble.

339. The purpose of this literature search was to try and establish a pattern whereby the discharge coefficient of an orifice in a simulation program could be predicted for various pressure differences across the orifice for both cavitating and non cavitating flow. Most of the published tests were performed with idealised orifices. In an attempt to find some correlation between published results and the

behaviour of real hydraulic components a short series of tests was carried out on a poppet type restrictor valve and a sharp edged circular orifice. The results of these tests were inconclusive and no correlation could be found with published results. In the absence of an extensive test program to establish trends for various types of fluid power valves, if such trends do exist, there is no reliable way of accurately predicting the discharge coefficient of a particular valve. If it is felt that the flow characteristics of a valve have an important effect on the performance of a system, at present only experimental determination of the discharge coefficient over a wide range of pressures is possible.

3.3 Minor losses, bends and pipe vibration

340. Minor losses are relatively easy to deal with when using the method of characteristics. Contractor [3.30] pointed out that localised losses which occur at changes of crossection or at T junctions, etc. may be treated as component models which introduce discontinuities of pressure between pipelines. Generally only steady state losses are known, the validity of using these losses in high frequency flow is doubtful, however it serves as a reasonable first approximation which ensures that at least the final steady state achieved will be accurate.

341. Bends present a different problem since the loss is not concentrated but is distributed along the length of the bend. Classical wave theory states that if the diameter of a pipe is small compared with the wave length of the transient the wave propagation in a curved pipe is the same as in a straight one. However tests by Swaffield [3.31] have shown significant reflection from bends. Dimensional analysis by Swaffield revealed the reflection and transmission coefficients were independent of all physical parameters except the bend geometry. Reflections of some 6% were obtained from pipes with a bend radius to pipe diameter ratio of three. The paper attracted several contributions to the discussion and mechanisms were proposed to account for the unexpectedly large reflections. The analysis resulted in empirical formulae for determining reflection

coefficients in terms of pipe geometry. However viscosity was not included in the dimensional analysis and therefore it cannot be judged whether these formulae may be used for low Reynold's number flows encountered in fluid power systems.

342. In their frequency response analysis of pipe flow D'Souza and Oldenburger 1964 [3.6] developed a model taking account of the longditudinal vibrations of the pipe wall. Experimental validation showed that pipe wall vibration was significant if the frequency of wall vibration was close to the transient frequency of interest. Thorley 1969 [3.32] found evidence of a precursor wave in the liquid ahead of the main water hammer wave caused by and lagging slightly behind the longditudinal tension wave in the pipe wall. Longmore 1977 [2.2] has produced and verified a frequency response model for the vibration of double steel braid hydraulic hoses, taking account of the different Young's modulii of the reinforcements in the axial and in the radial directions and of the energy dissipation in the hose material. The losses in the hose were found to be more than four times greater than those in steel pipes of similar dimensions. Washio et al. 1979 [3.33], have developed a frequency response model in terms of hyperbolic operators for metal pipes and include pipe wall dissipation.

343. For most applications involving fluid flow in metal pipelines the natural frequency of the pipe wall is considerably higher than the highest frequency transient of interest and therefore the pipe vibration may be ignored. However hydraulic hoses present an important additional dissipative mechanism at audible frequencies and future work may require formulating Longmore's model in the time domain for solution with the method of characteristics. A similar approach to that employed by Zielke and Trikha for dynamic friction may be possible here.

4. HYDRAULIC SYSTEM SIMULATION PROGRAMS

401. Two highly developed hydraulic system simulation programs are of particular interest; HYTRAN (MacDonnel Douglas), based on the method of characteristics and designed for dynamic analysis of aircraft hydraulic systems; and HGSP (Bath University) a more general, user orientated program using lumped parameter linking. The recommendations for future work (Chapter 9) are concerned with the HGSP package, therefore the structure and operating procedures of the program are described here in some detail.

402. Bond graphs and large scale simulation techniques are mentioned in the last section of this chapter. Although these are not strictly fluid power simulation programs they are important methods and should feature in a review of modelling techniques.

4.1 HYTRAN (Hydraulic Transient Analysis)

403. HYTRAN (hydraulic transient analysis) is one of a group of programs issued by the MacDonnel Douglas aircraft corporation. The other programs are SSFAN (steady state flow analysis) and HSFR (hydraulic system frequency response). SSFAN calculates the steady state pressures and flows throughout a system under any loading conditions. An iterative procedure is used whereby flows are varied to obtain a pressure balance throughout the whole system. HSFR calculates the system frequency response up to the 10^{th} harmonic using a form of the impedance method [1.5].

404. HYTRAN itself is a very large program really only suitable for use on mainframe computers. The test program included in the documentation requires 320K bytes of *storage on a CDC machine (equivalent to 80K words at 4, eight bit bytes per word). By comparison, at the time of writing, the Bath University school of engineering PDP11/34 had a total storage of 64K bytes. HYTRAN uses program SSFAN to set up initial conditions therefore the format of data storage has to be compatible with the solution procedures used by both programs. Within HYTRAN the method of specified intervals is used with the method of characteristics and solution requires both iteration and interpolation. The component models are solved with a 4^{th} order Runge Kutta integration routine [4.1 , 4.2]. The program structure is modular

* core storage

and different system models can be contructed from a library of component model subroutines. At present the library contains mainly specific aircraft hydraulic components. The program library and facilities for input/output and program alterations are not as advanced as those offered by the HGSP package.

4.2 HGSP (Hydraulic General Simulation Program)

405. The strategy behind HGSP was to develop a simulation language or package which would allow engineers with no formal training in computer programming or mathematical modelling to simulate a wide variety of hydraulic systems. And to provide facilities whereby, with the minimum of effort, the user could change various parameters, or by adding or subtracting components could alter the system configuration. The resulting package is fully interactive, the user operates from a terminal (visual display unit VDU) and is invited by the computer to enter details of the system to be analsyed. At every stage the opportunity is provided to alter input data either to correct errors or to change the system configuration. Output of results is selected by the user and can be in graphical or tabular form. The entire package can be supported by a mini-computer with 64K bytes of store [1.6]. As far as the user is concerned the preliminary work for simulating a system involves studying the component model documentation, selecting suitable models and correctly specifying the links between the models. The package itself writes the simulation program and supervises all input and output, Fig. 4.1 shows the important features of the package.

406. The main program controls interactive input and output and sets up the storage of data and results. Once all the data describing a given system has been input and no further alterations are required a simple command is invoked to run the program generator. The generator produces all the specific program coding, in particular it writes the system description routine which is a sequence of call statements to component model subroutines. A further command starts the integrator which from then on automatically controls the entire simulation. At each step of the solution the integrator calls the component models in the sequence specified by the system description routine. The outputs from the models are integrated simultaneously and the results are loaded into storage arrays via the output subprogram. Final display of results is specified interactively by the user.

407. The integrator uses a modified form of Gears algorithm for solving mathematically stiff systems, i.e. systems with widely varying time constants. Gears algorithm is a variable timestep, variable order differential equation solver. The order and the timestep employed are automatically varied throughout the solution to minimise computing time yet maintain acceptable accuracy. Generally a differential equation solver handles equations of the form

$$\frac{dy_i}{dt} = f_i(t, y_1, y_2, y_3, \ldots, y_n) \quad i = 1,n \qquad 1$$

where at time t the values of y_i are known, f_i are known functions, thus the state variables dy_i/dt may be evaluated and integrated to give values for y_i at time $t + \Delta t$. In the HGSP package state variables are evaluated by calls to the component model subroutines.

408. A component is generally a multiport element (Figure 4.2), each port is associated with a link to another component model. A link represents two parameters, an effort and a flow, one of which is an input to the model the other an output, the choice is arbitrary. Effort parameters are defined as pressures, forces and voltages; flow parameters are defined as flows, velocities and currents. The product of effort and flow is power, therefore each link represents a power bond between one component and another. Power bonding is a useful concept for establishing cause and effect relationships in a form suitable for computer manipulation and has resulted in a whole field of study called Bond Graphs (see below).

409. Before a component model can be solved all the inputs must be known. Since inputs to one model are the outputs from another it is clear that the order in which the component subroutines are called is critical. Therefore the program generator examines each link and works out the correct order when writing the system discription routine. However when preparing the simulation the user must ensure that the outputs from one model are compatible with the inputs to all adjecent models and vice versa.

4.3 Bond graphs and large scale simulation techniques

410. Bond graphs are a formal mathematical system for defining the

topography of any dynamic system. Originally the concept of power bonds was used to establish a cause and effect relationship. However the development of the bond graph language has reached a state of mathematical abstraction that any system may be modelled provided analogous properties to efforts and flows may be identified.

411. A graph is a collection of elements bonded together. Multi-port elements are nodes on the graph, ports are defined as places where an element can interact with other elements. Bonds are formed when pairs of ports are joined together. Basic elements are seen as; sources of effort, sources of flow, capacitances, inertances etc. etc.

412. Rosenberg and Karnop [4.3] give a good introduction to the ideas behind the bond graph language. The field is quite extensive and continuously expanding as new applications are found. Gebben [4.4] published a bond graph bibliography for the years 1961-1976, a section of which deals with applications in hydraulics.

413. Large scale system simulation techniques are based on state-space theory. A system is represented in matrix form as an aggregate of many sub-systems, and manipulative procedures are available to facilitate altering the interconnection of these sub-systems. The technique has several advantages over transfer functions and bond graphs. Iyengar and Fitch [4.5 , 4.6] discuss the method and its application to fluid power systems.

414. Bond graphs and the large scale simulation techniques are complicated and require considerable mathematical skill and programming knowledge. And therefore in this respect they do not satisfy the criterion for user orientated system simulation programs.

5. HYDROSTATIC TRANSMISSION MODELS

501. The purpose of modelling the hydrostatic transmission systems described in this section was to test the performance of the system simulation program devised, based on the method of characteristics. Initially it was intended only to model a transmission used to drive a boat. However a stability problem was encountered and the process of identifying the instability included modelling a simple open loop transmission. This transmission was analysed using basic control theory and the theoretical solution was compared with the numerical solution produced by the method of characteristics program and good agreement was obtained.

502. The boat transmission system had been analysed numerically by Huckvale [5.1] using lumped parameter theory. Huckvale's work served as a yardstick by which the method of characteristics program could be judged. The response of the system and the pipe lengths were such that wave effects were not expected to affect the performance and close agreement was expected between the two simulation methods.

5.1 Simple open loop transmission

503. The transmission consists of a constant speed prime mover driving a variable displacement pump which supplies flow to a motor connected to an inertia plus viscous friction load. (Figure 5.1) The initial steady state for the system was with the pump turning at prime mover speed but deswashed and generating no flow. The motor and load were at rest, the pipes and tanks were pressurised at 5 bar to minimise cavitation at the pump inlet. The system was disturbed by a ramp increase in swash, from zero to full swash in 0.25 seconds. The consequent transient was computed and the results are shown in Figures (5.3) and (5.4), fluid friction was ignored.

504. The simulation programme included the following component model subroutines Figure (5.2).

PUMP	Method of characteristics model of a hydrostatic pump
PIPE	Pipe model using the method of characteristics
MOTOR	Method of characteristics model of a hydrostatic motor

SOURCE	Method of characteristics model of a constant pressure flow source
SINK	Method of characteristics model of a constant pressure flow sink
CONST	Constant speed prime mover model
LOADCH	Simple inertia load model
ZEROF	Zero friction model

The main program structure and each subroutine is described in detail in the program report, part II. All system data is presented in appendix 5.1.

505. Since a relief valve model was not included the maximum pressures developed are very high, 606 bar on the first overshoot and 193 bar on the second, Figure 5.4. The load speed graph (Figure 5.3) gives a better indication of the system dynamics and is a typical 2^{nd} order response. The logarithmic decrement formula was used to calculate the damping ratio using values of overshoots measured from the graphs. Likewise the damped natural frequency was measured directly from the graphs, the value of 15.1 rad/sec being the average of several periods. The damping ratio could not be calculated exactly because of the limited accuracy of the plots but was found to be in the range 0.176–0.210.

506. An analytical solution was found for the system, the analysis (presented in appendix 5.2) neglects the pump inlet and the motor outlet pipes, these operate at low pressure and have negligible effect on overall system dynamics. They are only included in the computer simulation because the pump and motor component model subroutines require both inlet and outlet pipelines.

507. Comparing the analytical results with the computer simulation:

	Computer Solution	Analytical Result	Percentage Error
damped natural frequency	15.1 rad/sec	15.3 rad/sec	1.3%
damping ratio	0.176–0.210	0.186	-5.3%/+12.9%

These results indicate that for this system the computer solution is stable and accurate, the errors quoted above can be attributed mainly to plotting inaccuracies.

5.2 Closed loop hydrostatic boat transmission

508. The boat drive modelled is a conventional closed loop hydrostatic transmission powered by a governed diesel engine (Figure 5.5). All system parameters including the propeller characteristics are given in appendix 5.3. Values were taken from Huckvale's program, certain parameters however were unavailable, such as pipe diameters, for these realistic estimates were made. A steady state model was used for the crossline relief valve and boost pump circuit. It was assumed that the boost pump could maintain a constant pressure at point A in the circuit irrespective of flow demand. For computing convenience the action of the circuit is represented by two separate but identical components called JUNC, one located in the supply line the other in the return line. Since the circuit is symmetrical this treatment is valid provided both lines are not above relief valve pressure simultaneously.

509. The simulation program included the following component model subroutines (Figure 5.6).

PUMP	Method of characteristics model of a hydrostatic pump
PIPE	Pipe model using the method of characteristics
MOTOR	Method of characteristics model of a hydrostatic motor
JUNC	Method of characteristics model of a crossline relief valve and boost pump circuit
ENGINE	Governed diesel engine model
BOAT	Load model of a propeller driven boat

All of which are fully described in the program report, part II. Fluid friction effects were found to be negligible and the results presented were produced by a program using the 'zero friction' subroutine, ZEROF.

510. The initial steady state of the system was with the pump at zero swash turning at prime mover speed, zero boat velocity and zero

propeller speed. The pipes were at boost pressure (5 bar) and flow was assumed to be zero, the leakage flow through the pump and motor at 5 bar pressure was considered negligible. The system was disturbed by putting the pump to full swash in 0.25 seconds. The results of system simulation are given in Figures (5.7) to (5.11). On each graph the solutions from Huckvale's program are shown for comparison.

511. The pressure trace from the method of characteristics program is slightly underdamped, but in general good agreement is obtained **indicating that the program is performing correctly.** Comparison of computer times is difficult since both programs were run in batch mode. Furthermore Huckvale's program was written specifically to model the boat transmission whereas the method of characteristics program was more general and therefore used different programming techniques. Nevertheless an indication of relative computing times is useful.

512. The lumped parameter program used 15 seconds of computer time per second of system simulation time. The method of characteristics program required 80 seconds of computer time per second of simulation. All computer times are quoted are for programs run on the ICL 470 main frame machine at Exeter. Therefore bearing in mind the conditions mentioned in the preceeding paragraph the lumped parameter program for this system is the faster by a factor of approximately five.

513. The cause of the instability mentioned in the introduction was traced to the load model subroutine. The thrust developed by the propeller is a function of several interrelated parameters. In particular the propeller efficiency and thrust coefficient are both functions of the advance ratio which is the ratio of the boat speed to the speed at which the propeller would move through the water if it was a perfect screw with no slip. Graphs were supplied describing the relationships and these were stored by the computer as data arrays. Each individual array element held the value of the dependent variable (thrust coefficient or efficiency) corresponding to a particular value of the independent variable (advance ratio). There are two ways of handling data supplied in this form. A function may be fitted to the data and values of the dependent variable calculated for any input

value of the independent variable, or interpolation may be used to obtain values lying between the stored data points. The latter method was chosen and to keep the program as simple as possible a linear interpolation method was used, and it was this which caused the instability. With linear interpolation there is a discontinuity of gradient at each data point which was acting as an input to the system causing spurious oscillations. A more sophisticated smoothed interpolation method eliminated the discontinuities of gradient and completely cured the problem.

6. SIMULATION OF A 3 PORT PRESSURE COMPENSATED FLOW CONTROL VALVE

6.1 Introduction

601. This chapter describes the development of a method of characteristics model of a 3 port pressure compensated flow control valve. The model is based on data obtained from a Barmag valve by Baker [6.1]. The purpose of the work was to gain experience in modelling a fast acting component and to investigate methods for solving a set of non linear algebraic and differential equations.

602. A numerical method was used which involves linearising all the equations and solving them at each timestep to predict a solution which is then used to improve the linearisation and produce a set of corrected results. A number of tests was performed to prove the accuracy and stability of the numerical method.

603. The spool-chamber mechanism within the valve was analysed theoretically to identify the effect of the damping restrictor on the spool response. The results of this analysis were useful in understanding the effect of the restrictor on the overall valve performance.

604. An unsuccessful attempt was made to match the results of the method of characteristics program with those of a lumped parameter simulation. The difficulty lay in reproducing the lumped parameter response of a rather large inlet volume using the method of characteristics. Essentially the problem was not in the valve model but in attempting to simulate a system for which the method of characteristics pipe model was not suitable.

6.2 The Valve Model

605. The main features of the Barmag valve are shown schematically in Figure (6.1). The valve tries to maintain a constant flow through the preset orifice by altering the restriction in the return line using a compensating spool. Spool movement is determined by the

balance of forces due to the inlet and outlet pressures and spring compression. Dynamics of the spool and damper are taken into account by the computer model. Inlet and outlet pipelines are modelled using the method of characteristics, the dynamics of the return line are neglected and a constant pressure is assumed to act at the exhaust port. Valve behaviour is described by the following set of equations in seven unknowns: P_I, P_O, v_I, v_O, the inlet and outlet pressures and flow velocities respectively; x, the valve spool position P_C, the spring chamber pressure and Q_T, the bypass flow.

Force balance on compensator spool

$$(P_I - P_C)A = M\ddot{x} + f\dot{x} + Kx + F_C \qquad (6.1)$$

Bypass flow (exhaust flow)

$$Q_T = 0.0514\ (x)^{1.3} \sqrt{\frac{2}{\rho}(P_I - P_T)}.\text{sign}(P_I - P_T) \qquad (6.2)$$

Flow through preset orifice

$$A_{po}v_O = 0.06251(y)^{1.3} \sqrt{\frac{2}{\rho}(P_I - P_O)}.\text{sign}(P_I - P_O) \qquad (6.3)$$

Continuity

$$A_{PI}v_I - A_{po}v_O - Q_T = 0 \qquad (6.4)$$

Spring chamber volume (damper)

$$K_{LIN}(P_O - P_C) + A\dot{x} = \frac{V_{CH}.\dot{P}_C}{\beta_F} \qquad (6.5)$$

Inlet pipeline

$$\frac{1}{\rho c_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad (6.6)$$

Outlet pipeline

$$- \frac{1}{\rho c_o}(P_o - P_s) + (v_o - v_s) + \frac{2f_s v_s |v_s| \Delta t}{d_o} = 0 \qquad (6.7)$$

606. The coefficients of equations 6.1 to 6.3 were determined experimentally by Baker [6.1] and equation 6.5 was derived from data supplied by Baker. A more detailed account of the derivation and the computer solution of these equations is presented in the computer program documentation for subroutine BARMAG (**part** II).

607. The valve equations are a mixture of linear and non linear algebraic equations, plus first and second order differential equations. The solution procedure adopted was to convert the mixed set of equations into a set of seven linear algebraic equations which could then be solved simultaneously. The differential equations 6.1 and 6.5 were directly converted into linear form by writing the differential terms in finite difference form, which is effectively simple Euler integration. The orifice equations 6.2 and 6.3 were linearised using the theory of partial differentiation (small perturbations). Normally a linearisation is only valid for small excursions about a fixed operating point. In a transient situation where the system starts at one steady state, is disturbed and eventually settles down at another steady state a straight forward linearisation is inadequate.

608. Consider the square law pressure-flow relationship at an orifice, Figure 6.2. Taking point 1 as the initial steady state, linearising gives gradient G_1. Solving the straight line equation defining line G_1 simultaneously with the other component equations will give a solution at point 2 lying somewhere along line G_1. As the solution progresses by one timestep a new linearising gradient is calculated, this is the gradient at point $2'$ on the non linear curve. The new gradient is applied at point 2 and the system is solved again to give new values of pressure and flow at point 3 which lies on gradient G_2, and so on. Therefore as the simulation progresses the calculated values diverge from the correct solution which should be somewhere on the non linear curve.

609. Two similar methods may be used to minimise the divergence and therefore improve the accuracy of the linearised solution.

(i) Backward Euler approach.
Solve the linearised set of equations for value P_2 lying on gradient G_1. Calculate a new value of gradient G_2 based on the value of P_2 but this time apply the gradient at point 1 and solve the system of equations again for P_3', Q_3' where hopefully these values are closer to the non linear curve than the first solution.

(ii) Trapezoidal rule approach.
In this case rather than apply gradient G_2 at point 1, a third gradient G_3' is calculated which is the average of G_1 and G_2 and this is applied at point 1 to give an improved solution.

610. Both these methods provide increased accuracy however the solution from the linearised set of equations does not satisfy the non linear set of equations exactly, although the error is very small. From an engineering point of view it is important that flow continuity is maintained at the valve. Therefore as a final adjustment at each timestep the calculated values of pressure and spool position are substituted back into equations (6.2) and (6.3) to recalculate the bypass flow and the outlet flow, these values are then used in the continuity equation (6.4) to recalculate the inlet flow. The adjusted values serve as intial conditions for the next timestep in the system solution. In this way flow values are forced to satisfy continuity and the linearisation errors are distributed amongst the four remaining unknown parameters.

6.3 Accuracy Tests

611. As a first check on the accuracy of this method an error analysis

subroutine, (ERRAN) was written. N.B. Subroutine ERRAN performed a number of very specific diagnostic tasks in addition to the test described below, as such it was considered that the routine would not be of general interest and so has not been included in the computer program report. In general if the solution of a set of simultaneous equations is substitued back into the original equations and if the equations balance then the solution is the correct one. The purpose of the error analysis subroutine was to show that the values obtained from the linearised set of equations also satisfy the non linear set of equations. This was done by evaluating the two sides of each equation and comparing them. However due to linearisation errors an exact balance is not obtained. The error was defined as the difference between the LHS and the RHS expressed as a percentage of the mean value of both sides.

612. Equations (6.2), (6.3) and (6.4) were not included in the analysis since the solution procedure forced these equations to balance exactly. The remaining equations were tested, equations (6.6) and (6.7) had to be recast since the RHS is zero and the comparison of values very close to zero is meaningless. Both sides of equation (6.5) tend to zero as the system approaches steady state, nothing can be done about this, nevertheless the error analysis is valid under transient conditions. The subroutine sampled the calculated results over a number of specified intervals and printed out the percentage errors.

613. In the ranges tested the errors in equations (6.1), (6.6) and (6.7) were of the order 10^{-5}%, and for equation (6.5) generally of the order 10^{-2}%. From this it is concluded that the solution procedure yields a set of values which adequately satisfies the original non linear set of equations chosen to describe the performance of the valve. Both the backward Euler and Trapezoidal rule linearisation corrections were tried and gave satisfactory results, the Trapezoidal method was selected for use because it was marginally more accurate.

614. The error analysis shows that the solution calculated satisfies

the component equations, this does not necessarily mean that the integration of the differential equations is accurate or stable. A rough check is to halve the solution timestep and if the calculated values do not change significantly then the solution at the larger timestep is considered adequate. A more rigorous test can be applied which is sometimes known as Richardson extrapolation. It can be shown (appendix 6) that when a numerical integration method is behaving correctly the following condition is satisfied.

$$\frac{C(\Delta t) - C(2\Delta t)}{C(2\Delta t) - C(4\Delta t)} \simeq \frac{1}{2^P} \qquad (6.8)$$

Where $C(\Delta t)$ is the computed value at a given instant in time using an integration step length Δt, and $C(2\Delta t)$ is a value at the same instant in time using a $2\Delta t$, etc. The index P is the order of the integration method which in this case is one for first order simple Euler.

615. For the Barmag valve the situation is somewhat more complicated since the solution is in terms of seven variables, that is the solution is represented by a point in seven dimensional space. Nevertheless equation 6.8 still applies where the numerator $c(\Delta t) - C(2\Delta t)$ represents the distance between the points which define the solution with timestep Δt and $2\Delta t$ respectively. Similarly the denominator is the distance between points.

616. Equation 6.8 was evaluated at three instants in time, giving values of 0.36, 0.56, and 0.46. Generally values between 0.35 and 0.65 are considered acceptable. So this test gives a reasonable indication that the numerical method used is well behaved and does not introduce any numerical instabilities.

6.4 The System Modelled

617. The Barmag valve subroutine was tested by simulating an idealised system (Figure 6.3). A pump is treated as a constant flow source connected to the valve by 5 m of hose. Flow passes through the valve, through the length of rigid pipe to the load. The load

represents an actuator which is extending at constant speed, producing a constant load pressure, which then runs into an obstruction causing rapid increase in pressure to another constant level. For this system the input was a fast ramp of 21 bar to 40 bar load pressure in 0.001 seconds, true step changes of large amplitude apart from being physically unrealistic can cause numerical problems.

618. The program used to simulate this system is shown as an example in the computer program report, all the subroutines used and the program structure are discussed there in detail. Values of all the system parameters are present in the program listing.

619. A typical set of results is shown in Figures 6.4 to 6.7. The most noticable wave effects are seen in the inlet pressure trace (Fig. 6.4) where the irregularities correspond to wave reflections in pipe 1 arriving back at the valve. The flow (Fig. 6.5) is less sensitive to waves and the spool position graph is very smooth. The spool velocity however (Fig. 6.7) does exhibit high frequency oscillations which coincide with the arrival of waves at the valve. It is difficult to say if these oscillations are a physical phenomenon or a numerical effect. Altering the timestep has little effect on these oscillations, and what differences may be observed are probably due to the graph plotting routine.

620. Apart from the checks on accuracy and stability a number of tests was carried out to investigate the effect of varying certain parameters, with the ultimate intention of tuning the model to fit experimental results. Certain system parameters such as the spool mass, spool area, spring stiffness, etc. could be measured accurately and were therefore considered invariable. Early program runs showed that the inlet and outlet pipes had a significant effect on the system performance, however the pipes are considered as separate components and not a part of the valve. Attention was focused on those parameters which could only be estimated, in particular the linearised flow coefficient K_{LIN} and the viscous friction coefficient f. K_{LIN} is a function of damping restrictor area and discharge coefficient and its derivation is based on a number of assumptions. Preliminary test showed that K_{LIN} had a significant effect on the valve response, however an investigation of these effects by repeated runs of the

simulation program was considered to be too expensive. Instead a theoretical analysis of the spool and damper mechanism was performed, and the results of this analysis gave useful indications of how alterations of K_{LIN} would affect the overall response.

6.5 Analysis of the spool-damper mechanism

621. Taking the spool and damper as a small sub-system (appendix 6.2) it can be shown that the transfer function between P_o (input) and x (output) is third order and therefore has a characteristic equation of the form.

$$(1 + Ts)(s^2 + 2\zeta\omega_n s + \omega_n^2) = 0 \qquad (6.9)$$

Where the time constant T, the damping ratio ζ, and the natural frequency ω_n are functions of flow coefficient K_{LIN}, viscous friction f, spool mass M, etc. etc.

622. Unfortunately the nature of the transfer function is such that direct algebraic expressions cannot be found for T, ζ and ω_n in terms of K_{LIN} and other physical parameters. The problem was circumvented by using a numerical approach. A computer program was written to perform the root locus analysis of the simple third order characteristic equation. The damping ratio, time constant and natural frequency were calculated from the position of the roots. The program looped around incrementing the value of K_{LIN} in fixed steps between preset limits and calculating the corresponding values of T, ζ, and ω_n.

623. K_{LIN} was incremented over the range 10×10^{-13} to 190×10^{-13}. The variation of the time constant and the damping ratio is presented in Figure (6.8), the natural frequency remained constant. As K_{LIN} is decreased, i.e. the damping restrictor is closed off, the time constant increases and the damping ratio decreases. This is as expected because in the limit when the orifice is completely shut off the time constant will be infinite and the enclosed volume will act as a fluid spring with very little energy dissipation to provide damping. The graphs should only be used as indicators of trends since the theoretical analysis is concerned only with a small sub-system, whereas the true response of the spool and the whole system is determined by

a much larger set of parameters.

624. A similar root locus analysis was performed to check out the effects of viscous friction. However later work showed that this parameter is not important in the system described below.

6.6 Matching Results

625. The final stage in the development of the Barmag subroutine was an attempt to match the output of the method of characteristics program to the results of a simulation performed by Baker using a lumped parameter model of the system shown in Figure (6.3). Baker's model neglected the volume of the outlet pipe and assumed that the input step in pressure acted directly at the valve outlet. Furthermore the volume of the inlet pipe was very large, 30 litre. Nevertheless it was felt that by altering parameters in the method of characteristics program the effect of the outlet volume could be minimised and wave effects could be made insignificant by using a very short inlet pipe so that a reasonable comparison could be made with Baker's works. In fact it proved impossible to obtain a good match. Baker's response is shown in Figure (6.9) compared with the best response available from the method of characteristics program.

626. The parameter K_{LIN} was varied and the effect predicted in Figure (6.8) was observed. As K_{LIN} was increased the speed of response was increased (time constant decreased) and the position of the first over shoot was moved back (Figure 6.9). Initially doubling K_{LIN} produced a marked shift, but subsequent doubling and quadrupling had a diminished effect. Varying the viscous friction coefficient and spring stiffness caused little change in the overall response.

627. Baker in his work concludes that the system response is dominated by the inlet pipe volume. To minimise wave effects in the method of characteristics program the inlet pipe was treated as a very short and stubby volume; 0.15 m long and 0.5 m in diameter. It is doubtful if the method of characteristics would produce the same response from a volume of those dimensions as would the simple $dP/dt = (\beta/V)Q_c$ relationship. It appears therefore that the problem lies not

in the valve model but in attempting to model a system for which the method of characteristics is not suitable. When Baker's program is generally available it will be possible to attempt a comparison using a much smaller inlet volume.

7. DEVELOPMENT OF A TIME DOMAIN MODEL OF A HYDROSTATIC PUMP INCLUDING THE EFFECTS OF FLOW RIPPLE AT THE PUMP OUTLET

7.1 Introduction

701. The pulsatile flow produced by positive displacement pumps is a major source of fluid borne noise in hydraulic systems. Increasing awareness of the harmful effects of noise has prompted a great deal of research into the design of quiet hydraulic systems. In most cases impedance methods have been used for analysis and prediction, these methods are very efficient but only deal with steady oscillatory conditions. The purpose of the work described in this section is to develop a simple time domain model of a pump which allows flow ripple effects to be taken into account in the simulation of system transients.

7.2 Pump Model

702. The flow delivered by any positive displacement pump can be thought of as a mean flow level with a superimposed ripple, usually known as the flow source ripple Q_s. The mean level is a function of pump displacement, swash setting and operating speed. Q_s is more difficult to determine, essentially its frequency depends on the operating speed and the number of pumping elements, be they pistons, gear teeth or vanes. Its magnitude depends on the mean system pressure and construction details, such as valve plate timing, internal volume and leakage paths. A program for calculating flow source ripple has been developed at Birmingham University [7.1] . Alternatively Q_s can be measured experimentally, and is usually expressed as a spectrum of ten harmonics of the pumping frequency.

703. A simple model for a piston pump is presented in Figure 7.1, only effects at the pump outlet are considered, conditions at the inlet are taken as constant. All leakage paths are lumped together and represented by one orifice which has a linear p~Q relationship, on the assumption that leakage flows are laminar. The pump internal volume is modelled as a closed ended volume in parallel to the pump outlet. Pump flow is the summation of the outputs from an ideal flow

generator supplying the mean flow and a ripple generator simulating the flow source ripple.

704. To simulate accurately the inductive, capacitive and resistive elements of the pump impedance the internal volume is modelled using the method of characteristics. As far as the computer program is concerned the pump internal volume is treated as a second pipeline connected in parallel to the outlet pipe. Both pipelines are shown on the time distance plane in Figure (7.2). The pump model is based on four equations.

Continuity of flow

$$Q_{ID} + Q_s = Q_R + A_{PE}v_v + A_{po}v_o \tag{7.1}$$

Leakage orifice

$$Q_R = K_R(P_o - P_T) \tag{7.2}$$

Outlet pipeline (backward characteristic)

$$-\frac{1}{\rho c_o}(P_o - P_s) + (v_o - v_s) + \frac{2f_s v_s|v_s|\Delta t}{d_o} = 0 \tag{7.3}$$

Equivalent volume pipeline (backward characteristic)

$$-\frac{1}{\rho c_E}(P_o - P_E) + (v_v - v_E) + \frac{2f_E v_E|v_E|\Delta t}{d_E} = 0 \tag{7.4}$$

705. Equations (7.1) to (7.4) are solved simultaneously to calculate the pump behaviour. Although the equivalent volume pipeline and its termination are integral parts of the pump it is computationally convenient to treat them as separate components. Their behaviour is calculated by calls to the pipe model subroutine and a zero flow pipe termination subroutine. Both calls are made from inside the pump

subroutine so as far as the user is concerned he deals only with one component model although its behaviour is determined by a number of subroutines.

706. The flow ripple term Q_s in the continuity equation is evaluated at each timestep of the solution by summing the contribution of ten harmonics input as data. Another approach would be to store the ripple waveform as a number of data points, the value of Q_s could then be obtained by interpolation. It is felt that the summation of harmonics is more efficient since the other method requires time consuming transfer of control to a smoothed interpolation subroutine

Full details of the pump subroutine are given in the programme report.

7.3 System Simulation

707. The pump model was tested by simulating a simple system as shown in Figure 7.3. Note that the flow paths depicted by single lines are used merely to indicate the direction of flow and may be considered as lossless and volumeless pipes which are not included in the simulation.

708. Extensive experimental data for this system was available from tests carried out by Butler [7.2]. Tests had been performed using a Reyrolle A200 pump with various lengths of the outlet pipe and various settings of the restrictor valve. The pressure ripple waveform produced under a given set of operating conditions was measured by a transducer located near the pump flange, and recorded as the amplitude and phase of ten harmonics. The experimental results were obtained by a transducer located 2.2 cm from the pump flange. The computed results were interpolated from pressure values produced at computation points on either side of the transducer position.

709. The Reyrolle A200 is a seven piston unit, its source flow ripple Q_s was determined experimentally at an operating pressure of 200 bar, a mean flow of 0.7 l/sec and a speed of 153.5 rad/sec

corresponding to a fundamental pumping frequency of 171 Hz. Q_s data is presented as 10 harmonics in table 7.1 and the ripple waveform synthesised by adding the harmonics is presented in Figure 7.4.

710. The system was simulated for two different lengths of the outlet pipe, in both cases the restrictor was adjusted to give the same mean conditions as those at which Q_s was determined. The pipe and fluid properties were such that the wavespeed for the system was 1370 to 1380 m/sec. The pipe lengths chosen were selected to illustrate a severe resonant condition with a 3.944 m length and a non resonant condition with a 2.661 m length. The experimentally measured pressure ripples for these pipes are presented in harmonic form in table (7.2) and graphically in Figure (7.5). The mean level of 200 bar was subtracted from the ripple waveforms and the scales were expanded (Figures (7.6) and (7.7)) to make comparison with computed results easier.

711. The Reyrolle pump has a severe back flow at an operating pressure of 200 bar, the output flow drops very sharply to 40% of the mean level, this disturbance occurs at a frequency of 171 Hz. The 3.944 m pipe has a period corresponding to a frequency of 174 Hz this is very close to resonance and results in large pressure fluctuations with a peak to trough amplitude of 49 bar. The 2.661 m pipe has a frequency of 258 Hz which is just over one and a half times the fundamental pumping frequency, the resulting pressure ripple is much less severe with a peak to trough amplitude of 23 bar. However the third (513 Hz), sixth (1026 Hz) and ninth (1539 Hz) pump harmonics are in resonance with the second (516 Hz), fourth (1032 Hz) and sixth (1548 Hz) harmonics of the pipe period. Consequently the third pump harmonic is particularly strong and the sixth and ninth harmonics are slightly higher than their immediate neighbours. (Table 7.2.)

712. The purpose of the computer simulation was to develop models to match these experimental results as closely as possible. Full details of the restrictor valve and the outlet pipelines were available, the fluid properties and operating conditions were known however certain assumptions had to be made about the pump model. The leakage flow

restrictor and the equivalent volume pipeline were idealisations and had no direct physical counterpart in the pump. It was assumed that the leakage path was a pure resistance giving the pump a volumetric efficiency of about 95%. Butler had estimated the pump volume as 40 cc by examining drawings and Edge [7.3] suggested that a pipe of 14 cm length and 40 cc volume would have an impedance approximating that of the Reyrolle pump. This then formed the starting point for the pump model.

713. The restrictor at the end of the outlet pipe was known to obey a $P = KQ^{1.95}$ law, and furthermore was known to display dynamic effects. However as a first approximation and for ease of computing the restrictor was assumed to obey the usual square law ($P = KQ^2$) and dynamic effects were ignored. The downstream pressure was assumed to be constant.

714. At the early stages of the work it became apparent that although the non resonant waveform was being predicted reasonably accurately, the resonant condition was severely under damped, giving peak to trough amplitudes in excess of 75 bar. In view of this it was felt that the steady state square law restrictor valve model was inadequate and a dynamic valve model was required. A suitable model was developed based on available experimental data. Naturally this more sophisticated model was computationally more complicated and more time consuming. Therefore to try and detect any computational errors and to identify conditions under which the simpler model could be used all tests were carried out on both a program using the steady state valve model and on a program using the dynamic valve model. Thus in effect four cases were considered:-

dynamic valve model	resonant pipe length	Case 1
	non resonant pipe length	Case 2
steady state valve model	resonant pipe length	Case 3
	non resonant pipe length	Case 4

A full description of the dynamic valve model is given below. All system parameters are listed in appendix (7.1).

7.4 Program testing and evaluation

715. The results as presented here are not in chronological order. Initial test runs and program debugging indicated general trends, a 14 cm/40 cc equivalent pipeline was the starting point of the simulation but it was found to produce very large amplitudes for case 3 forcing the graph plotting subroutine to adopt a different scale to other programs. This made the comparison of graphs difficult. For this reason the majority of tests were carried out with systems having a 14 cm/51 cc equivalent pipeline, and a typical set of results is shown in Figures (7.8) and (7.9).

716. Cases 1 and 3 , the resonant pipe length with dynamic and steady state valve models respectively, are compared with experimental results in Figure (7.8). The amplitude of the system with the steady state valve model is about 65 bar, compared with 58 bar when using the dynamic valve. The shape of the wave in both cases is very similar and indicates an excessively strong first harmonic compared with the experimental trace which has an amplitude of 49 bar and exhibits considerably more high frequency components.

717. Comparison between results from programs computing the non resonant system performance, cases 2 and 4 is given in Figure (7.9). The computed traces are much closer to the experimental results and the steady state valve model program gives the closest agreement.

718. Comparison of the traces merely by the overall amplitude and general appearance is in some cases adequate but it can be vague especially when the waveform is complicated as in Figure (7.9). A more satisfactory approach would be to compare the harmonic content of each trace, various deviations from the required shape could then be ascribed to an excess or a deficiency of a particular harmonic or number of harmonics. Giving a precise quantitative indication of the differences in output of various programs.

719. With this object in mind a program was written to perform the Fourier analysis of any waveform defined by a number of data points.

The given waveform is expressed as the amplitude and phase of ten harmonics directly comparable to the ten harmonic representation of the experimental results. Simulation results were processed in this way only in cases where it was felt that harmonic analysis would give more insight into what was happening.

720. In all, two programs were developed to assist in the analysis of simulation results; the Fourier analysis program described above (FAN) and a harmonic synthesis program (HARM) which takes 10 harmonics as data and synthesises the corresponding waveform. HARM was used to generate the experimental waveforms in Figures 7.4 to 7.7. Both these programs are described in the program report.

721. The Fourier analysis of the four computed waveforms (of Figures 7.8 and 7.9) is given in table 7.3 and is presented as frequency spectra in Figures 7.10 and 7.11. As anticipated the amplitude of the first harmonic for the resonant system is very large, the rest of the spectrum compares reasonably well with the experimental results. For the system using the dynamic valve model all the harmonics have slightly smaller amplitudes and greater phase lags than those for the system using the steady state valve model. These are the expected effects of the dynamic valve model. The accuracy of the calculated phase is good for the first four harmonics then it progressively deteriorates with large errors in the 8^{th} and 10^{th} harmonics. However the phase errors of the small amplitude higher harmonics has very little effect on the overall waveform. This can be appreciated by noting that an error of 180^{o} in the phase of the 10^{th} harmonic corresponds to a timing error of $1/20^{th}$ of the fundamental period. So it may be concluded that the main problem lies in the large first harmonic.

722. The amplitude and phase spectra of the non resonant system are also in reasonable agreement with experimental results. In both cases the amplitude of the dominant third harmonic is a little low, but the most significant divergence between experiment and theory lies in the excessively high amplitude of the 4^{th} harmonic computed by the system incorporating the dynamic valve model. Since a

similar effect is not evident in any of the other spectra it seems likely that the effect is due to some numerical interaction between the non resonant pipe length and the dynamic valve model although the nature of such an interaction is not obvious. Earlier program tests at the debugging stage checked for numerical unaccuracies or instabilities by halving the timestep used. Negligible differences were observed between programs using fine and coarse timesteps indicating integration inaccuracies and error accumulations were insignificant.

723. A point worth noting when discussing numerical accuracy is that the results presented here were calculated on a mean level of 200 bar. The total non resonant pressure ripple is only a variation of 23 bar on the mean level, i.e. a variation of 11½%. The average difference between the computed amplitude and experimental amplitude from Figure 19 is about 1½ bar which when compared to the amplitude alone is 6½% error, but when compared to the mean level, which after all is the average magnitude of the numbers manipulated by the computer, the error is only 0.75%. For the resonant case the error compared to amplitude is about 25% and compared to the mean level it is about 5% (for the program using the dynamic valve model). Computational errors of up to 5% are usually considered quite acceptable for engineering purposes therefore the graphs presented in Figures (7.8) and (7.9) are not bad solutions. But, although it is recognised that a resonant condition is difficult to predict accurately one would like to simulate the waveform amplitude more accurately than to within +25%.

724. The Fourier analysis reveals that the main sources of error are one or two rogue harmonics rather than the cumulative effect of errors in all the harmonics so the sequence of tests described below is an attempt to identify the cause of the observed unaccuracies and to obtain a best fit for the experimental results. The following factors were examined:-

1 Pipe friction

2 Pump leakage
3 Effects due to the dimensions of the equivalent volume pipeline
4 Effects due to the dynamic valve time constant

7.5 Frictional effects

725. The mean flow in the outlet pipeline is laminar, consequently Trikha's method for simulating frequency dependent friction (dynamic friction) may be used. This series of tests was intended to check if friction is an important parameter in this system and if the difference between steady state friction and dynamic friction warrants the increased computing effort.

726. Figure (7.12) compares pressure ripple traces from programs using different friction models in a resonant system simulation with the dynamic valve model and a 14 cm/51 cc equivalent pipeline. (Case 1 .) A small difference is observed between traces produced with zero friction and those produced with steady state friction. Using dynamic friction, however, gives a significant reduction in overall amplitude. Nevertheless the shape of the waves indicates that the various friction models do not alter the dominance of the very strong first harmonic.

727. The same set of tests for a non resonant simulation (case 2) is illustrated in Figure (7.13). Again there is an overall reduction in amplitude when using the dynamic valve model. The wave shapes indicate a strong 4^{th} harmonic independant of friction model used.

728. Friction effects for this system are small but the use of a dynamic friction model is justified on the grounds that any mechanism for the reduction of amplitude in the resonant condition is useful. The system simulations using the steady state valve model (cases 3 and 4) produced very similar results to those described above when different friction models were tried.

7.6 Pump leakage

729. A set of programs was run for cases 1 to 4 with the pump leakage term set to zero. The pump flow was adjusted to ensure all initial conditions were correct. It was expected that omitting this term would increase the overall amplitudes since the leakage flow path gives an additional route for high pressures to leak away. As it turned out the results from the non resonant simulations were unaffected, the differences being so small as to be unnoticeable on a graph. The resonant case showed a very slight increase in amplitude of about 0.5 bar. The pump leakage, therefore, has negligible effect on the system simulation. Naturally amplitudes could be reduced to any required level simply by increasing the pump leakage, however the pump model would not be very realistic.

7.7 Effects of the equivalent volume pipeline dimensions

7.7.1 Volume effects

730. The equivalent pipeline was kept at a constant length (14 cm) but the volume was varied, results were calculated for 40 cc, 51 cc and 60 cc volumes. A common method of reducing fluid borne noise is to have an expansion chamber at the pump outlet. It was therefor expected that increasing the pump volume would significantly decrease the magnitude of the pressure ripple. This expectation was confirmed with the resonant system simulation (case 1). A 50 bar peak to trough amplitude was obtained with a 60 cc pipe volume compared with an amplitude of 71 bar with a 40 cc volume. (Figure 7.14.) Once again the general shape of the waves indicates a very strong first harmonic irrespective of the volume.

731. The situation is more complicated when examining waveforms produced by the non resonant system (case 2), Figure (7.15). Little change is seen in the overall amplitude but the actual shapes of the waves indicate significant variation in the harmonic content. Amplitude and phase spectra were generated by Fourier analysis, the most noticeable volume effects were in the 3^{rd}, 4^{th} and 7^{th} harmonics

(Figure 7.16 table 7.4). In the dominant third harmonic the 40 cc volume gave the most accurate amplitude prediction, and increasing the volume to 60 cc decreased the amplitude by 1.7 bar. Again in the 4^{th} harmonic the 40 cc volume gave the most accurate results but this time increasing volume gave increasing amplitude. The 7^{th} harmonic had a suprisingly strong amplitude when using the 40 cc volume. A point worth noting is that in this test the amplitude of each harmonic shows a definite trend with increasing volume. The amplitudes of the first and 4^{th} harmonics increase with increasing volume, the amplitudes of the remainder decrease. None of the other tests where Fourier analysis was used show such clear cut trends indicating that the volume of the equivalent pipeline is a very important and sensitive parameter. Similar trends were not observed in the phase spectrum.

7.7.2 Length effects

732. For these tests the equivalent pipe was kept at a constant volume (40 cc) and the length was varied, results were calculated for 10 cm, 14 cm and 18 cm lengths. The results of the resonant system simulation (case 1) are shown in Figure (7.17). Variation of length does not have as much effect as variation of volume, the peak amplitude is hardly affected indicating little change in the fundamental harmonic, but the depth of the trough des vary considerably implying changes in the higher harmonics. The 18 cm length gives the best fit with the smallest overall amplitude.

733. Once again the output from the non resonant system is more difficult to interpret (Figure 7.18). Fourier analysis was performed and the resulting spectra are shown in Figure (7.19) table (7.5). The 1^{st} and 2^{nd} harmonics are reasonably accurate irrespective of the length used. The 3^{rd} harmonic is predicted accurately by the 14 cm length but is over estimated by the other two lengths. The 14 cm length alone over estimates the 4^{th} harmonic. The only other features of note are the strong 7^{th} harmonic when using 10 cm and 14 cm lengths and the strong 10^{th} harmonic using the 10 cm length. Overall the 18 cm length gives the best fit, no significant trends are evident and the phase spectrum shows no important features.

734. Several conclusions may be drawn from the two tests described above. Firstly, and probably most important, is that for the resonant case variation of length and volume can be used to reduce the overall amplitude of the pressure ripple, but unfortunately in all cases the first harmonic is too strong and swamps any variations in the higher harmonics to give the distinctive waveshape seen in all the figures. For the non resonant system the 40 cc volume produces the best results, unfortunately it produces the worst results for the resonant system. Accurate prediction of a resonant condition is more important from an engineering point of view, therefore a larger volume must be used. The 60 cc volume is too large giving severe distortions of the non resonant waveform due to the very low 3^{rd} harmonic and very large 4^{th}. A reasonable compromise is achieved with the 51 cc volume, even so the 3^{rd} harmonic is still too low and the 4^{th} too high although less so. The 18 cm length gives the best fit for both the resonant and non resonant conditions. It also gives a slightly high 3^{rd} harmonic for the non resonant system; so when combined with a 51 cc volume it should give a good fit.

735. In both tests the system simulation using the steady state valve model showed the same trends.

736. The 14 cm length gives a high 4^{th} harmonic and increasing volume also increases the 4^{th} harmonic amplitude in a non resonant system. Therefore the effect noted in the 14 cm/51 cc simulation discussed earlier appears due to a poor combination of length and volume, rather than a numerical inaccuracy. However it is not clear why a 14 cm length should interact with the dynamic valve model to produce a high 4^{th} harmonic, there is no physical explanation for it, the pipe period of 4910 is not related to the valve break frequency (800 Hz) by an integer number, and even if it was, there is no reason why this should produce an amplitude increase at 684 Hz. Although highly speculative, the most likely explanation is that the particular timesteps and integration constants required to handle the 14 cm pipeline combine in some way to generate an error which manifests itself as a strong 4^{th} harmonic.

7.8 Effects due to the dynamic valve time constant

737. The final set of tests was to investigate the effects of the dynamic valve time constant, to check if the simulation was sensitive to this parameter. The time constant was varied giving break frequencies corresponding to 600 Hz, 800 Hz, 1000 Hz and 1200 Hz. (The break frequency determined by experiment was 800 Hz, see the description of the dynamic valve model.) The results are shown in Figures (7.20) and (7.21) in each figure only two graphs are plotted those for the 600 Hz and 1200 Hz break frequencies, the results for the other two frequencies lie in between the traces given.

738. In the resonant system simulation the first harmonic still dominates. Fourier analysis showed the expected behaviour. As the break frequency was increased the amplitudes of all the harmonics also increased but only by a small amount (table 7.5).

739. The non resonant waveform showed some distortion with changing break frequency. Fourier analysis proved this to be due, almost entirely, to the decreasing amplitude of the 4^{th} harmonic with increasing break frequency. A large break frequency corresponds to a small time constant and therefore a small integration step length in the dynamic valve model. Since a smaller timestep tends to reduce the 4^{th} harmonic amplitude this supports the hypothesis put forward above that the excessive magnitude is due to some integration inaccuracy in the valve model. This point is discussed further in the section describing the dynamic valve model.

740. Overall the effects of the time constant are small and there is no justification for using any other value than the experimentally determined one.

7.9 The best fit solution

741. The above tests have indicated that a system with a 18 cm/51 cc equivalent pipeline would give a good fit to the experimental results. The results of such a simulation are given in Figures (7.22) to (7.27).

742. The resonant system using a dynamic valve model produces an overall amplitude of 52 bar, compared with an experimental one of 49 bar, giving an amplitude error of 6%. Compared to the mean pressure level the error is 1½%. (Figure 7.22.) When using a steady state valve model the amplitude is 58 bar, an error of 4% (Figure 7.23). The Fourier analysis spectra (Figure 7.24) reveal that the first harmonic is still excessively strong and that the reduction of the overall amplitude is due to a decrease in amplitude of all the harmonics.

743. The non resonant system simulation using the dynamic valve model under-estimates the magnitude of the pressure ripple (Figure 7.25) giving an overall amplitude of 20.5 bar compared with an experimental amplitude of 23 bar, an error of 11%, with respect to the mean pressure level this is an error of 1.25%. The shape of the waveform is considerably better than in previous simulations because of the reduced 4^{th} harmonic (Figure 7.27). When using the steady state valve model (Figure 7.26) the overall amplitude is also under-estimated, but the overall shape of the wave is a little better because of the more accurately predicted 3^{rd} harmonic.

744. It cannot be claimed that the above results represent a best fit because in the tests parameters were varied over a wide range but in each test only three or four programs were run. Many more program runs would be required to fine tune the results to give a best fit. However it was felt that any further increase in accuracy would not justify the effort and expense of more tests.

745. Before concluding this section it is appropriate to consider a few computing aspects of the work so far. The initial conditions for each of the simulations was a steady flow in the outlet pipe and zero flow in the equivalent pipeline. The flow ripple was introduced to both pipes after 0.001 sec and the system was simulated for 0.21 seconds to allow the transients to decay and the steady pressure ripple to develop. All the computed traces shown in the figures are approximately three periods of the waveform after

a simulation of 0.21 seconds, the total simulation time was therefore 0.2275 seconds. Earlier tests had shown that with shorter simulation times the transient had not quite settled down. The actual execution times varied depending on the pipe lengths and the models used in each simulation, as well as the timestep used. A summary of parameters and simulation times for important program runs is given in table (7.10). An approximate average computing time for a resonant system with a 14 cm equivalent pipeline using a dynamic valve model and dynamic friction is 2500 ETU's (ETU = Elapsed time unit, 1 ETU = 0.25 cpu. seconds on the Exeter 470's) or approximately 10 minutes of computing time.

746. The sequence of tests provides an opportunity of estimating the cost of various friction models. The most general way of presenting friction costs is the number of ETU's required to model friction at each calculated point in the system for each timestep of the solution. Comparison between programs shows that dynamic friction costs on average 3.4 E-3 ETU/calc point/timestep more than the steady state friction model and 4.7 E-3 ETU/calc point/timestep more than the zero friction model. These figures will allow friction costs to be estimated in future programs.

747. In the present series of programs considering for example a resonant system with a dynamic valve model the following computing times are obtained:-

dynamic friction model	2672 ETU's	program 1
steady state friction model	1802 ETU's	program 5
zero friction	1384 ETU's	program 9

Which clearly shows that modelling dynamic friction in this particular system accounts for half the computing time. It should be noted that the zero friction model sets the friction factor to zero at each calculated point. This approach is convenient allowing the use of the same pipe subroutine irrespective of the friction model. However the pipe subroutine could be rewritten to to model the frictionless case automatically with further savings in computer costs.

748. The work described in this section demonstrates that pump flow ripple can be successfully modelled in the time domain using the method of characteristics. The system chosen for the simulation was in fact a steady oscillatory system which could have been simulated more efficiently using impedance techniques. However the purpose of the method of characteristics solution is not to compete with well established impedance techniques but to provide a basis for an alternative method which may be applied in situations where other methods cannot be used.

749. In systems using actuators the volume between a pump and the actuator piston varies as the actuator extends, therefore the impedance of the system varies making calculation using impedance techniques difficult, requiring repeated calculation at small increments of the piston travel. Also the system is restricted to constant load, constant velocity extension so that the mean conditions at the pump outlet also remain constant.

750. In future work the method of characteristics may be applied to actuator systems, the variation of volume being taken into account by updating at each timestep. The pump model presented above is based on an experimentally determined flow ripple at a given set of mean operating conditions, therefore the application of this model is restricted to steady oscillatory cases or to problems where the transients are small compared to the mean levels and which return to the same initial conditions. The present model could calculate the start up transient for an actuator system provided this transient was not too large, although modelling the ripple over the initial stages would not be particularly accurate.

751. Using an experimentally determined flow source ripple at one mean operating condition places a severe restriction on the usefulness of the model. The computation of large transients settling at a different set of mean conditions is impossible. One possible way of overcoming this problem for systems operating at constant pump speed would be to store a set of flow ripple harmonics determined at a number of mean operating pressures. So as the computation progresses at a given timestep the current value of pressure at the pump could be used to determine which set of flow ripple harmonics should be used to calculate the output from the ripple generator

for the next timestep. The decision process used in the model could be an interpolation between harmonics at pressure levels bracketing the current value of pressure or simply selecting the set of harmonics at a pressure level nearest the current value. More work is required in the future to investigate the feasibility of such an approach. The condition that the pump speed is constant simplifies the problem since speed variations imply frequency variations in the ripple harmonics. However taking account of speed changes should be possible using the same approach as for pressure changes. Although now the flow ripple harmonics would have to be determined at a number of different pump speeds each at a number of different pressures, and the computation of the transient would involve selecting the correct set of harmonics for a given current value of pressure and speed. Which is theoretically feasible but makes for an extremely cumbersome model. Nevertheless the method of characteristics based approach outlined here does provide the starting point for the analysis of various problems which cannot be handled using classical analytical techniques.

7.10 Restrictor valve model including dynamic effects

752. Experiments have shown that the impedance of the restrictor valve is not constant but obeys a transfer function such that; (Ref. 7.4)

$$\frac{Z_T}{Z_{TSS}} = G(s) \qquad (7.5)$$

Where Z_T is the instantaneous valve impedance at any frequency and Z_{TSS} is the steady state impedance. G(s) depends on the mean operating pressure and at 200 bar it approximates to a first order lag with a break frequency of approximately 800 Hz (see Figure 7.28 and table (7.9)). Equation (7.5) may therefore be written as

$$\frac{Z_T}{Z_{TSS}} = \frac{1}{1 + Ts} \qquad (7.6)$$

The steady state flow equation for this valve is given as

$$Q = K_{NL} P^{\frac{1}{n}} \qquad (7.7)$$

Where n = 1.95, K_{NL} is a constant and P is the pressure drop across the valve.

753. Impedance is defined as the ratio of pressure to flow (Z = P/Q). Therefore from equation (7.7) the steady state impedance is:

$$Z_{TSS} = \frac{P}{K_{NL} P^{\frac{1}{n}}}$$

Which when substituted in equation (7.6) gives:

$$Z_T = \frac{P}{K_{NL} P^{\frac{1}{n}} (1 + Ts)}$$

but Z_T is the instantaneous relationship between pressure and flow therefore the above equation becomes

$$\frac{P}{Q} = \frac{P}{K_{NL} P^{\frac{1}{n}} (1 + Ts)}$$

Rearranging and substituting d/dt for the Laplace operator s, yields the dynamic flow equation for the restrictor valve.

$$Q = K_{NL} P^{\frac{1}{n}} + \frac{K_{NL} T (P^{\frac{1}{n-1}})}{n} \frac{dP}{dt} \qquad (7.8)$$

754. This expression reverts to the steady state flow equation when dp/dt is zero. The physical implication of equation (7.8) is that with a rapidly increasing pressure drop across the valve, the flow,

at a given instantaneous pressure, is greater than one would expect from the steady state equation. In other words the pressure lags the flow.

755. Funk et al (7.5) developed an unsteady flow equation for an orifice of the form

$$P = \underbrace{\frac{\rho Q^2}{2(CdAo)^2}}_{\text{S. State effect}} + \underbrace{\frac{\rho}{(CdAo\pi/2)^{\frac{1}{2}}} \frac{dQ}{dt}}_{\text{Dynamic effect}} \qquad (7.9)$$

This implies that with a high rate of change of flow, the flow through the valve at a given instantaneous pressure is smaller than one would expect from the steady state equation, therefore the pressure leads the flow.

756. An equation of the same form as (7.9) can be generated following the procedure described above but with the first order lag in equation (7.6) replaced by a first order lead. Experimental evidence shows that for this particular valve Funk's equation is not suitable.

757. The computer model for the restrictor is based on the solution of two equations, the valve equation derived above and the forward characteristic equation of the pipe connected to the valve. Downstream pressure is considered constant therefore only one pipe equation is required. The time distance plane representation of the valve model is given in Figure (7.29)

valve equation

$$A_{pI} v_I = K_{NL} P^{\frac{1}{n}} + \frac{K_{NL}}{n} T P^{(\frac{1}{n} - 1)} \frac{dp}{dt} \qquad (7.10)$$

forward characteristic

$$\frac{1}{\rho C_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad (7.11)$$

where $P = (P_I - P_T)$

758. One solution is to express the dp/dt term as a finite difference, linearise equation (7.10) and solve the two equations simultaneously for P_I and v_I. The Barmag valve model illustrated the difficulties that can arise from linearisation under transient conditions. Although this particular system oscillates about a mean value and therefore linarisation would be adequate it was felt that for the sake of generality an alternative approach should be tried.

Equation (7.10) can be rearranged as follows:-

$$\frac{dp}{dt} = \frac{A_{PI} v_I - K_{NL} P^{\frac{1}{n}}}{\frac{K_{NL}}{n} P^{(\frac{1}{n} - 1)}} \qquad (7.12)$$

Assuming the solution has reached a particular time level t and the values of $P_{I(t)}$ and $v_{I(t)}$ are known, equation (7.12) is used to calculate the value of dp/dt. Integration then yields a value of P a timestep Δt_o later ($P_{(t + \Delta t_o)}$) and consequently a value of $P_{I(t + \Delta t_o)}$. This new value is now substituted in equation (7.11) to give a value of $v_{I(t + \Delta t_o)}$. Thus the solution has progressed one timestep.

759. A wide variety of techniques is available to perform the integration of dp/dt, the simplest and easiest to apply being the Simple Euler method. However to maintain accuracy the timestep Δt_o must be sufficiently small. For a first order system it should be less than one hundreth of the time constant. The method of characteristics by its very nature imposes a timestep Δt on the

system simulation, if Δt is less than the required Δt_o all is well and the valve equation may be integrated directly in one step. If however Δt is greater than Δt_o one step integration would be inaccurate, consequently the integration scheme has to be suitably modified.

760. A break frequency of 800 Hz corresponds to a time constant of 2×10^{-4} seconds, hence for the valve being considered an integration timestep of $\Delta t_o = 2 \times 10^{-6}$ seconds is required. The method of characteristics timestep is approximately 5×10^{-5} seconds so direct one step integration is not possible. The procedure used is similar to the one described in chapter 2.

761. The method of characteristic timestep is divided into an integer number of timesteps Δt_o each less than $\frac{1}{100}$ th of the time constant. Similarly Δx the distance between pipe calculation points is divided into an integer number of intervals Δx_o. The forward characteristics which apply at the nodes between each Δx_o interval intersect the time axis at the nodes of the Δt_o intervals (Figure 7.30). Each of the new forward characteristic equations is based on values of $P_{R(i)}$ and $v_{R(i)}$ which are obtained by linear interpolation between the known values of P_R and v_R at the calculation point located a distance Δx from the valve and the values of $P_{I(t)}$ and $v_{I(t)}$ at the valve itself. The integration procedure is the same as described above, equation (7.12) is used to calculate dp/dt, thus it is integrated over one timestep Δt_o giving a value of $P_{I(t + \Delta t_o)}$, which in turn is substituted in equation (7.11) (based on suitable values of $P_{R(I)}$ and $v_{R(i)}$) to calculate the flow velocity $vI(t + \Delta t_o)$, the procedure is repeated until the complete interval Δt has been integrated.

762. The integration is automatic and is set up by an initialisation subroutine called before the start of the simulation. The restrictor model decides whether or not a multiple timestep integration is required and behaves accordingly. The numerical accuracy of the model

was checked by substituting computed values back into equations (7.10) and (7.11), the errors detected were acceptably small. However the sequence of tests carried out to find the best fit system simulation indicated that there may be some numerical problems in the dynamic valve model.

763. The tests investigating the effect of the dynamic valve time constant in a non resonant system, showed a marked reduction in the 4^{th} harmonic amplitude as the time constant was decreased, (break frequency increased). The amplitudes of all the other harmonics were affected to a much lesser degree (table 7.7). Decreasing the valve time constant increases the number of Δt_o timesteps required to integrate between the time levels used in the method of characteristic solution of the pipes. The increased number of timesteps implies an increase in numerical accuracy. On the basis of tests carried out it is impossible to state categorically that this is the reason for the increased accuracy in predicting the 4^{th} harmonic amplitude, too many questions are unanswered, in particular why should numerical inaccuracies manifest themselves as an inaccurate 4^{th} harmonic. Nevertheless the tests point to an area of uncertainty and future applications of the dynamic valve model should include tests which vary the number of integration steps used to check the effect on the integration scheme described above.

8. DISCUSSION AND CONCLUSIONS

8.1 Introduction

801. The more general aspect of this work has been an extensive literature review on dynamic system simulation techniques and related topics with particular emphasis on their application to fluid power systems. The method of characteristics emerged as the most suitable technique for modelling wave propagation in liquid pipelines in the time domain and it proved to be a convenient mechanism for linking component model subroutines. Other techniques although in some cases more accurate or more efficient are restricted in their application. The method of characteristics is extremely versatile and even in its most simplified form is capable of giving good results.

802. From the outset this project was intended to complement the simulation program HGSP being developed at Bath University. (N.B. the latest developments in HGSP, now renamed GHSP, are described in Ref. 8.1.) Some general points may be made about the respective advantages and disadvantages of lumped parameter and distributed parameter hydraulic system simulation programs.

803. The fundamental difference between the two approaches has been pointed out in chapter 1, lumped parameter simulation models compressibility; friction and fluid inertia can only be dealt with in a crude way. The method of characteristics models wave propagation and handles sophisticated friction models with ease.

804. More specific differences arise when considering the solution procedures adopted for the two approaches. Lumped parameter programs can use centralised integration with the potential of using very sophisticated integration routines such as Gear's algorithm for stiff systems. The number of state variables which may be integrated simultaneously imposes a certain limit on the size of a system model. This however is not usually a great restriction since the ratio between time constants of hydraulic components can be very

large and it is possible to eliminate state variables by assuming these components act very quickly and are therefore instantaneous, or act very slowly and are therefore constant. The elimination of extreme state variables also reduces the system stiffness, leaving only state variables in the range which the system designer feels may affect a particular aspect of system performance. The situation is somewhat complicated by the fact that some systems are inherantly stiff and the fast and slow time constants cannot be eliminated. (Ref. 8.2.) Variable multiple step integrators tend to be sensitive to discontinuities which must be detected and negotiated by interval halving and restarting (Ref. 8.1), this all complicates the integration process. Problems may be avoided by replacing an instantaneous change in the gradient or in the value of a parameter by a rapid but continuous function, at the expense however of introducing stiffness. Due to the centralised integrator the construction of a system program requires careful ordering of the component model subroutines. For a general purpose program a complex sorting routine is required (program generator). Storage requirements increase as the square of the number of state variables since the integrator handles data in matrix form.

805. Method of characteristics programs are limited to simple, usually explicit integration methods if the integrator is included in the component model. More advanced integrators may be used in the form an external subroutine, but increased integration efficiency and capability must be balanced against time lost due to transfers of control and any interfacing required between the integration method and the method of characteristics. If integration is distributed amongst the component models there is in principal no limit to the system size which may be simulated, however there are practical restrictions relating to computing time and computer storage. Storage requirements increase linearly with the number of components and because of the sequential nature of the solution, sorting of component models is not necessary.

8.2 Operational status of the method of characteristics program developed

806. The program consists of a general structure which may be used to model a variety of systems by using the appropriate component model subroutines. At present the program is not user orientated, to model a given system the user has to write data input and output coding and a specific main program following the format specified in the program report (part II). No alterations are required to the component model subroutines, except possibly changes to the array dimension statements if the user desires to minimise storage requirements. Standard fortran and single precision arithmetic is used throughout, and the use of facilities specific to a given installation or operating system has been avoided. The aim was to produce a structure which could easily be converted into an interactive user orientated program. The data storage and manipulation is organised with this in mind and since a sorting routine is not necessary the task would be relatively easy although a considerable volume of interactive coding would be required.

807. The programs produced are intended for use on small computers, however the logistics of designing and testing a large complicated program precluded the use of a mini-computer for program development. Initially all computing work was performed on the South Western Universities Computer Network, primarily on the ICL 470's at Cardiff and Exeter, all programs were run in batch mode. Later work was carried out on the Avon Universities Multics system, an interactive facility. However, except for Fourier analysis, the programs were too large and time consuming to be run economically in the interactive mode.

808. No general quidelines can be given as to the computer times required by the method of characteristics program, since too many factors affect execution time. The time requirements for each specific simulation are discussed in the relevant chapters.

809. The storage required for the program plus the component model subroutines but excluding data arrays averaged at 20K words, which

on a two byte per word machine such as the PDPII, translates to 40K bytes of store. The development programs used very large (24K word) data arrays, the majority of this was used for diagnostic purposes and was therefore redundant. A generous estimate of data storage for an average 10 pipe (100 calculation point per pipe), and 10 component system is 7500 words, that is 15K bytes. Thus a reasonably sized system simulation would require approximately 55K bytes of store, well within the capacity of most computer installations. An additional 6K bytes is required if the dynamic fluid friction model is used. If necessary savings can be made by overlaying. In particular the initialising routines are called only once and may be written out to disc without incurring any penalty in terms of execution time.

810. The library of tested component model subroutines is presented in the program report (**part** II). Other subroutines have been prepared but have not been tested in system simulations. These include; a branch point where up to 10 pipes meet, a check valve, a two port flow control valve, an actuator plus load model and a linearised orifice model.

8.3 Discussion of the method of characteristic program

811. The program is based on a simple form of the method of characteristics using a constant timestep and a fixed regular grid with no interpolation. The characteristic equations are integrated using a first order finite difference method. Fluid friction can be simulated by steady state or frequency dependent models. All dynamic components are integrated using Simple Euler, where necessary an interpolation is used to interface the method of characteristics timestep and the component timestep.

812. The integration techniques used were found to be adequate for the systems simulated. Simple Euler integration plus interpolation proved easy to program, test and debug, however for more complex components such as the Barmag valve the method was found to be intractable. The addition and deletion of equations was awkward

and time consuming. From the point of view of component model development where frequent changes of component equations is anticipated a centralised integrator may be more practical. Since the largest timestep is defined by the method of characteristics there is little advantage in using higher order variable timestep integration methods which can accurately solve the boundary conditions using larger timesteps. One is only concerned with integration accuracy when components require a smaller timestep than that imposed by the method of characteristics, and for this purpose the integration techniques discussed above are adequate. However, the recommendations for future work, namely the interfacing of HGSP with the method of characteristics, include a scheme where a centralised integrator uses timesteps greater than the method of characteristics timestep. This is primarily to avoid tampering with the existing integrator set up for HGSP, but it may prove advantageous to impose a limit on the maximum integration step length.

813. The method of characteristics provides an excellent framework for a system simulation program. The pipelines isolate each component model and allow the use of a wide choice of solution methods for each boundary. It is possible therefore to custom build a system model with different solution methods employed at each boundary, each method matched specifically to the requirements of its component model. To a large extent this is what has been done in this project, the problems encountered indicate that a general approach to modelling boundary conditions would be more practical in the long term. Lumped parameter modelling techniques as used in the HGSP package have advantages which are complementary to the method of characteristics program. The centralised integrator provides a general way of solving boundary conditions and the large library of component models is a useful resource for setting up boundaries consisting of one or more components. Rather than developing a separate user orientated package based on the method of

characteristics with its own component model library it is more practical to incorporate the method of characteristics into HGSP to make use of all the facilities already available. Two schemes for such a marriage are proposed in chapter 9.

8.4 Summary of conclusions

814. The transmission system simulations (chapter 5) served as a test of the general program structure and gave good agreement with theoretical analysis and lumped parameter simulation. The program was sensitive to numerically generated gradient discontinuities in the load model and a 'smoothed' interpolation routine was required to prevent spurious oscillations. Computing times were approximately five times greater than those of the lumped parameter program.

815. The Barmag flow control valve simulation investigated the modelling of fast acting components and involved developing a linearised integration method for the set of non linear, discontinuous component equations.

816. The pump model with flow ripple examined the feasibility of using the method of characteristics to model high frequency pressure ripple, i.e. fluid borne noise in hydraulic systems. The study was based on a semi-empirical pump model which was tuned to fit experimental results. Experimental correlation was good and the model behaviour was consistent with real physical effects. The simulation also provided an opportunity for evaluating the performance of the frequency dependent friction subroutine and testing the effect of a dynamic restrictor valve model.

817. This project has shown that a simple form of the method of characteristics is adequate for the simulation of fluid power systems even at high frequencies. The limiting factor on program performance is not the pipe model or the friction model but the boundary conditions imposed by components. Simple integration of components is perfectly

adequate but can be time consuming when developing models. The method of characteristics is suitable for system simulation on small computers but is not ideal for a general purpose simulation package and a marriage with the lumped parameter technique as used in HGSP is the more practical solution.

818. A promising area of study is the modelling of fluid borne noise in the time domain by the method of characteristics. Future work could include the development of a specific program and library of components orientated towards this application.

9. RECOMMENDATIONS FOR FUTURE WORK

9.1 Introduction

901. Two possible ways of incorporating distributed parameter pipeline models into the HGSP package are discussed in this chapter. In both instances the pipeline model is based on the method of characteristics.

(i) Development of a pipe model which fits into the existing HGSP program structure as simply another component model subroutine. Except in this case the pipe performance is not determined by integrating the compressibility equation using Gear's algorithm. Instead, as far as HGSP is concerned, the pipe is treated as a steady state model without any need of the centralised integrator. The solution of the pipeline compatibility equations, i.e. finite difference integration, is performed inside the pipe model subroutine.

(ii) Development of a new program structure where the HGSP package is used to determine the boundary conditions for a network of pipes modelled using the method of characteristics. Essentially this would be a method of characteristics program structure, similar to that described in the program report (part II) with the HGSP package acting as a sophisticated mechanism for creating and solving component models.

Approach II is considerably more ambitious as it would involve stripping down the HGSP package, making it subordinate to a new program structure and rewriting large portions of the interactive input and output coding. Approach I has more potential in the short term since major changes to the HGSP structure are not necessary.

9.2 APPROACH I: Development of a distributed parameter subroutine for use in HGSP

902. The practical feasibility of the procedures outlined here cannot be fully appreciated until actual attempts have been made at programming the subroutine. Therefore the scheme proposed below should only be considered as a description of the problems anticipated and possible solutions to those problems.

903. The basic principles of interfacing the method of characteristics with component models using more complicated integration have been discussed in chapter two and applied in chapter seven to the dynamic valve model. To summarise; the pipe behaviour is defined by the finite difference form of the compatibility equations, with two unknowns, pressure and flow. At all internal calculation points the two compatibility equations apply simultaneously and may be solved directly for both unknowns. However, at the boundaries only one equation applies and so the boundary component must specify one of the unknowns to allow the other to be calculated.

904. The solution scheme for HGSP is illustrated by considering as an example a pump connected via a long pipeline to a relief valve (Figure 9.1). The outputs from the pump and relief valve models are flows which act as inputs to the pipe model which returns values of pressure to both components. The situation is shown on the time distance plane in Figure (9.2).

905. Assuming the solution is complete at time level 1, conditions at all internal points may be calculated directly for the next time level. (time $= \Delta t + \Delta t_c$, where t_c is the fixed timestep due to the method of characteristics.) The boundary conditions cannot be calculated until the pump and relief valve pressures are known at time $t + \Delta t_c$. Gear's algorithm proceeds with a variable timestep which may be larger or smaller than that used for the method of characteristics. When smaller timesteps are used the pipe model outputs (P_p and P_{RV}) are calculated using the procedure explained in chapter two. Finally when the solution exceeds time level 2 the output from the component models

(Q_{RV}, Q_p) may be interpolated to give exact values at time $t + \Delta t_c$. Now the compatibility equations at the boundary may be used to calculate P_{RV2}, P_{p2}, hence completely specifying the solution at time level 2. If the Gear's algorithm timestep is greater than that used by the method of characteristics boundary flow values at all intermediate time levels may be calculated directly by interpolation.

906. The computing problems and requirements may be listed as follows:-

(a) The HGSP program structure is such that the pipe subroutine is called every time the integrator calls the system description subroutine, which may be several times per timestep. This is because Gear's algorithm is a variable order predictor - corrector method. The pipe model on the other hand has to call subroutines for calculating all internal conditions and friction factors, but only at fixed intervals. Therefore the current value of time, according to the integrator, is required as an input each time the pipe routine is called, plus a system of flags and conditional statements to ensure that transfers of control only occur at the correct times.

(b) HGSP uses interactive data input subroutines to specify component parameters. The corresponding routine for the pipe model must include facilities for calculating the pipe wavespeed, for setting up a suitable timestep and for working out the necessary number of calculation points and the initial conditions at each point. In HGSP individual component data which does not change as the solution progresses is stored in an array named CON. Time varying data is transferred via the subroutine argument list. The pipe model however will contain time varying data which is of no interest outside the routine, this data must be stored in arrays and suitable coding will have to be inserted in the main program to set up the necessary common block storage.

(c) Since the pipe appears to be a steady state model the output pressure (P_p, P_{RV}) are not considered as state variables. Consequently the program generator will assume that their values are not known at the start of the simulation and will work out the calling sequence in the system description routine accordingly. This may result in problems so the presentation of the model should be carefully examined to ensure that the program generator is not confused.

9.3 APPROACH II: Development of a new program structure incorporating HGSP

907. The design of a completely new program structure based on the method of characteristics and incorporating HGSP probably represents a logical next step in the development of a versatile user orientated simulation language. The task is considerable and it would be impossible at this stage to propose a scheme for such a program. However some advantages are immediately apparent.

908. The HGSP component model library is very comprehensive. Individual components could be used directly and more complex boundaries consisting of several components could be created using the program generator. Since the individual components or clusters of components are separated by distributed parameter pipelines, the integration is performed sequentially rather than simultaneously. The maximum system size which may be handled is no longer limited by the capacity of the integration algorithm or the core storage of the computer. Provided the largest cluster of components and the adjoining pipes is within capacity, the rest of the system may be temporarily written out to disc. The potential for overlaying would not be so great when using a distributed parameter pipeline simply as a component model. Naturally overlaying is paid for by increased computing time, however often this is not the most important consideration. The ability to handle large and complex system simulations on a relatively modest

computer is a definite advantage.

BIBLIOGRAPHY

1 FOX, J.A., 'Hydraulic analysis of unsteady flow in pipe networks'. MACMILLAN PRESS LTD. 1977

2 WYLIE, E.B., STREETER, V.L., 'Fluid transients'. MCGRAW HILL 1978

3 KAYE, G.W., LABY, T.H., 'Tables of physical and chemical constants'. LONGMANS

4 PARMAKIAN, J., 'Water hammer analysis'. DOVER PUBLICATIONS NEW YORK 1963

5 CONSTANTINESCO, G., 'Theory of wave transmission, a treatise on the transmission of power by vibration'. WALTER HADDON 1922

6 BLACKBURN, J.F., REETHOF, G., SHEARER, J.L., 'Fluid Power Control'. THE M.I.T. PRESS 1960

7 ABBOTT, M.B., 'An introduction to the method of characteristics'. THAMES & HUDSON 1966

8 STEVEN, G.P., On the application of the finite element method to problems in fluid flow in - 'Numerical Simulation of fluid motion'. Proceedings of an international conference on the numerical simulation of fluid dynamics systems - held at Monash University MELBOURNE 1976. Edited by JOHN NOYE NORTH-HOLLAND PUBLISHING COMPANY 1978 (Bath University library ref. No. 532.5:518.1).

9 FENNER, R.T., 'Finite element methods for engineers'. MACMILLAN PRESS 1975

10 IRONS, B., SOHRAB, A., 'Techniques of finite elements'. ELLIS HORWOOD (J. WILEY & SONS) 1980

11 ZIENKIEWICZ, O.C., 'The finite element method'. MCGRAW HILL (U.K.) 1977

12 MCCLOY, D., MARTIN, H.R., 'Control of fluid power'. 1973

REFERENCES

1.1 CLANCY, J.J., FINEBERG, M.S., 'Digital simulation languages: a critique and a guide'. AFIPS CONFERENCE 27.1.1965 pg 23-36 (1965)

1.2 TIECHROEW, D., LUBIN, J.F., TRUITT, T.D., 'Discussion of computer simulation techniques and comparison of languages'. SIMULATION OCTOBER, 1967 pg 181, 190

1.3 BRENNAN, R.D., SILBERBERG, M.Y., 'The System/360 continuous system modelling program'. SIMULATION DECEMBER 1968 pg 301, 307

1.4 BOWNS, D.E., ROLFE, A.C., 'Computer Simulation as a first step towards computer aided design of fluid power systems'. 5th INTERNATIONAL FLUID POWER SYMPOSIUM SEPTEMBER, 13th, 15th 1978 A3. B.H.R.A.

1.5 AMIES, G., LINDQUIST, 'Computer simulation of hydraulic systems under dynamic conditions'. NATIONAL CONFERENCE ON FLUID POWER 1976

1.6 BOWNS, D.E., BONSON, L.A., RICHARDS, C.W., CANEY, K., 'The simulation of hydraulic systems'. WARSAW CONFERENCE 1980. International federation of automatic control - Symposium on Pneumatic and Hydraulic Systems.

2.1 YOUNGDAHL, C.K., KOT, C.A., VALENTIN, R.A., 'Pressure transient analysis in piping systems including the effects of plastic deformation and cavitation'. ASME PAPER 78-PVP-56 1978

2.2 LONGMORE, D.K., 'The transmission and attenuation of fluid borne noise in hydraulic hoses'. PAPER C267/77 - Research project seminar on quiet oil hydraulic systems I.MECH.E. NOVEMBER, 1977

2.3 HAYWARD, A.J.T., 'Generalisations for isentropic and isothermal compressibility of hydraulic mineral oils'. N.E.L. REPORT NO. 443 JUNE 1970

2.4 DOW, FINK, (Empirical formulae for density changes with temperature and pressure). 'JOURNAL OF APPLIED PHYSICS'. 1940 VOLUME 11 pg 393

2.5 ISDATE, J.D., BRONTON, W.C., SPENCE, C.M., 'Bulk modulus measurement and prediction'. N.E.L. REPORT 591

2.6 PEARSALL, I.S., 'The velocity of water hammer waves'. Symposium on surges in pipelines. PROC I.MECH.E. VOLUME 80 pt 3E 1965-66

2.7 KELLER, G.R., 'Hydraulic system analysis'. Published by the editors of Hydraulic and Pneumatic magazine

2.8 JOUKOWSKY, N., 'Water hammer'. PROC AM WATER WORKS ASSOCIATION 1904 pg 341. (An English translation of the original work which appeared in the Memoires de L'Academie des Sciences de St. Petersburg, 1898 Vol 1X No. 5 8 e Serie

2.9 ALLIEVI, L., 'General theory of peturbed flow of water in conduits'. MILAN 1903. (Translated into English by E.E. Halmos 1925 as 'Theory of Water hammer' included in Symposium on Water hammer ASME 1933 second printing 1949)

2.10 ANGUS, R.W., 'Simple graphical solution for pressure rise in pipes and pump discharge lines'. J. ENG. INST. CANADA pp 72-81 FEBRUARY 1935

2.11 BERGERON, L., 'Etude des variation de regime dans le conduits d'eau: Solution graphique generale'. Revue generale d l'hydraulic vol 1 pp 12-35, PARIS 1935 and - BERGERON, L.,: 'Water hammer in hydraulics and wave surges in electricty'. (Translated by ASME.) JOHN WILEY & SONS INC N. YORK 1961

2.12 SCHNYDER, O., 'Drukstosse in Pumpensteigheitungen'. Schweiz Bauztg 94 Nos. 22 and 23 1929

2.13 MARPLES, E.I.B., 'The significance of surge diagrams'. Symposium on Surge. PROC I.MECH.E. VOL 180 pt 3E 1965-66

2.14 PEARSALL, I.S., 'Water hammer effects due to branched and stepped pipes'. Symposium on Surge. PROC I.MECH.E. VOL 180 pt 3E 1965-66

2.15 HARDING, D.A., 'A method of programming graphical surge analysis for medium speed computers'. PROC I.MECH.E. VOL 180 pt 3E 1965-66. Symposium on Surge.

2.16 GOODSON, R.E., LEONARD, R.G., 'A survey of modelling techniques for fluid line transients'. TRANS ASME J. BASIC. ENG. JUNE 1972

2.17 EZEKIEL, F.D., PAYNTER, H.M., 'Computer representation of engineering systems involving fluid transients'. TRANS ASME VOL 79 NOS. NOVEMBER 1957 p 1840-1850

2.18 CODRINGTON, J.B., WITHERALL, R.G., 'The use of impedance concepts and digital modelling techniques in the simulation of pipeline transients'. PAPER A2 INTERNATIONAL CONFERENCE ON PRESSURE SURGES SEPTEMBER 1976 B.H.R.A. LONDON

2.19 BOWNS, D.E., MCCANDLISH, D., 'Pressure ripple propagation'. Research project seminar on quiet hydraulic systems. I.MECH.E. 1977

2.20 MCCANDLISH, D., EDGE, K.A., TILLEY, D.G., 'Fluid borne noise generated by positive displacement pumps'. Research project seminar on quiet hydraulic systems. I.MECH.E. 1977

2.21 BOWNS, D.E., EDGE, K.A., TILLEY, D.G., 'The assessment of pump fluid borne noise'. Research project seminar on quiet hydraulic systems. I.MECH.E. 1977

2.22 GREY, C.A.M., 'Analysis of water hammer by characteristics'. TRANS AM SOC. CIV. ENG. 1954 paper No. 2717

2.23 STREETER, V.L., 'Computer Solution of surge problems'. Symposium on Surge. PROC I.MECH.E. VOL 180 pt 3E 1965-66

2.24 KRANE, R.J., REIFF, A., 'A method of characteristics solution for the equations governing the unsteady flow of liquids in closed systems'. NASA technical report No. 388 30.8.66 Under Contract NAS 8-5321

2.25 STREETER, V.L., 'Unsteady flow calculation by numerical methods'. TRANS ASME J. BASIC. ENG. JUNE 1972 p 457-466

2.26 TRIKHA, A.K., 'Variable timesteps for simulating transient liquid flow by the method of characteristics'. TRANS ASME J. FLUIDS. ENG. MARCH 1977 pg 259

2.27 WIGGERT, D.C., SUNDQUIST, M.J., 'Fixed grid characteristics for pipeline transients'. PROC ASCE J. HYD. DIV. HY12 DECEMBER 1977 pg 1403

2.28 DOYLE, T.J., SWAFFIELD, J.A., 'Evaluation of the method of characteristics applied to a pressure transient analysis of the BAC/SNIAS Concorde refuelling system. PROC I.MECH.E. 1972 VOL 186 40/72.

2.29 INTERNATIONAL CONFERENCE ON PRESSURE SURGES. B.H.R.A. CANTERBURY ENGLAND 6th-8th SEPTEMBER 1972

3.1 ZIELKE, W., 'Frequency dependent friction in transient pipe flow'. J. BASIC. ENG. TRANS ASME MARCH 1968

3.2 BROWN, F.T., 'The transient response of fluid lines'. J. BASIC. ENG. TRANS ASME DECEMBER 1962

3.3 HOLMBOE, E.L., ROULEAU, W.T., 'The effects of viscous shear on transients in liquid lines'. J. BASIC. ENG. TRANS ASME MARCH 1967

3.4 BROWN, F.T., NELSON, S.E., 'Step responses of liquid lines with frequency-dependent effects of viscosity'. J. BASIC. ENG. TRANS ASME JUNE 1965

3.5 OLDENBURGER, R., GOODSON, R.E., 'Simplification of hydraulic line dynamics by use of infinite products'. J. BASIC. ENG. TRANS ASME MARCH 1964

3.6 D'SOUZA, A.F., OLDENBURGER, R., 'Dynamic response of fluid lines'. J. BASIC. ENG. TRANS ASME SEPTEMBER 1964

3.7 FOSTER, K., PARKER, G.A., 'Transmission of power by sinusoidal wave motion through hydraulic oil in a uniform pipe'. PROC I.MECH.E. 1964-65 VOL 179 pt 1 No. 19

3.8 KARAM, J R., J.T. 'A simple but complete solution for the step response of a semi-infinite, circular fluid transmission line'. J. BASIC. ENG. TRANS ASME JUNE 1972

3.9 KARAM, J R., J.T., LEONARD, R.G., 'A simple yet theoretically based time domain model for fluid transmission line systems'. J. FLUIDS. ENG. TRANS ASME DECEMBER 1973

3.10 KARAM, J R., J.T., TINDALL, R.R., 'A refinement of a simplified model for transients in fluid lines'. J. FLUIDS. ENG. TRANS ASME SEPTEMBER 1975

3.11 BROWN, F.T., 'A quasi method of characteristics with application to fluid lines with frequency dependent wall shear and heat transfer'. J. BASIC. ENG. TRANS ASME JUNE 1969

3.12 TRIKHA, A.K., 'An efficient method for simulating frequency dependent friction in transient liquid flow'. J. FLUIDS. ENG. TRANS ASME MARCH 1975

3.13 BROWN, F.T., MARGOLIS, D.L., SHAH, R.P., 'Small amplitude frequency behaviour of fluid lines with turbulent flow'. J. BASIC. ENG. TRANS ASME DECEMBER 1969

3.14 WOOD, D.J., FUNK, J.E., 'A boundary layer theory for transient viscous losses in turbulent flow'. J. BASIC. ENG. TRANS ASME DECEMBER 1970

3.15 FUNK, J.E., WOOD, D.J., 'Frequency response of fluid lines with turbulent flow'. J. FLUIDS. ENG. TRANS ASME DECEMBER 1974

3.16 BALTZER, R.A., 'Column separation accompanying liquid transients in pipes'. J. BASIC. ENG. TRANS ASME DECEMBER 1967

3.17 WEYLER, M.E., STREETER, V.L., LARSEN, P.S., 'An investigation of the effect of cavitation bubbles on the momentum loss in transient pipe flow'. J. BASIC. ENG. TRANS ASME MARCH 1971

3.18 SWAFFIELD, J.A., 'A study of the influence of air release on column separation in an aviation kerosene pipeline'. PROC I.MECH.E. 1972 VOL 186 56/72 pg 693

3.19 DRIELS, M.R., 'An investigation of pressure transients in a system containing a liquid capable of air absorption'. TRANS. ASME. J. FLUIDS. ENG. SEPTEMBER 1973

3.20 WIGGERT, D.C., SUNDQUIST, M.J., 'The effect of gaseous cavitation on fluid transients'. TRANS. ASME. J. FLUIDS. ENG. MARCH 1979

3.21 KARAM, J.T., 'Two port models of column separation in line systems containing an aerated liquid'. TRANS. ASME. J. FLUIDS. ENG. SEPTEMBER 1974

3.22 EDGE, K., 'The characteristics of engine - hydrostatic transmission systems'. PHd Thesis, Bath University 1975

3.23 YAMAGUCHI, K., TSUNEO, I., SUZUKI, S., 'The transient characteristics of oil pipelines with column separation'. BULL. JSME. VOL. 20, No. 143 MAY 1977

3.24 LICHTAROWICZ, A., PEARCE, I.D., 'Cavitation and aeration effects in long orifices'. Cavitation Symposium I.MECH.E. 1974

3.25 MACLELLAN, G.D.S., MITCHELL, A.E., TURNBULL, D.E., 'Flow characteristics of piston type control valves'. PROC Symposium recent mech. eng. dev. in auto control. I.MECH.E. 1960

3.26 STONE, J.A., 'Discharge coefficients and steady state flow forces for hydraulic poppet valves'. TRANS. ASME. J. BASIC. ENG. MARCH 1960

3.27 HALL, G.W., 'Analytical determination of the discharge characteristics of cylindrical tube orifices'. J. MECH. ENG. SCIENCE VOL. 5 NO. 1 1963

3.28 ISHIHARA, T., OUCHI, M., KOBAYASHI, T., TAMURA, N., 'An experimental study on cavitation in unsteady oil flow'. BULL. JSME. VOL. 22 NO. 170 AUGUST 1979

3.29 NUMACHI, F., YAMABE, M., OBA,R., 'Cavitation effects on the discharge coefficient of the sharp edged orifice plate'. TRANS. ASME. J. BASIC. ENG. MARCH 1960

3.30 CONTRACTOR, D.N., 'The reflection of water hammer pressure waves from minor losses'. TRANS. ASME. J. BASIC. ENG. JUNE 1965

3.31 SWAFFIELD, J.A., 'The influence of bends on fluid transients propagated in incompressible pipe flow'. PROC. I.MECH.E. VOL. 183 pt 1 NO. 29 1968-69

3.32 THORLEY, A.R.D., 'Pressure transients in hydraulic pipelines'. TRANS. ASME. J. BASIC. ENG. SEPTEMBER 1969

3.33 WASHIO, S., KONISHI, T., SONODA, T., 'Research on wave phenomena in hydraulic lines'. 4th report, coupling between oil pulsations and longditudinal pipe vibration BULL. JSME., VOL. 22 NO. 166 APRIL 1979

4.1 HYTRAN, documentation **(see below).

4.2 ZIELKE, W., 'Digital simulation of airplane hydraulic systems'. ASME paper 71-WA

4.3 ROSENBERG, R.C., KARNOPP, D.C., 'A definition of the bong graph language'. J. DYN SYS. SEPTEMBER 1972 179-182 TRANS ASME.

4.4 GEBBEN, V.D., 'Bond Graph bibliography for 1961-1976'. J. DYN SYS. JUNE 1977 pg 143-145 TRANS ASME.

4.5 IYENGAR, S.K.R., FITCH, E.C., 'A systematic approach to the analysis of complex fluid power systems'. 4th INTERNATIONAL FLUID POWER SYMPOSIUM, CRANFIELD 1975

4.6 IYENGAR, S.K.R., 'Application of large scale system analysis techniques to the analysis and design of hydraulic servomechanism'. National conference on fluid power 1975.

5.1 BOWNS, D.E., HUCKVALE, S.A., 'A semi-analytic approach to the dynamic simulation of shaft coupled power transmission systems'. PROC. I.MECH.E. 1978 VOL. 192 NO. 8

6.1 BAKER, O., MSc. Thesis SCH.MECH.ENG. BATH UNIVERSITY 1981

7.1 HANNAN, D.M., 'Flow pressure and force pulsations in a hydraulic axial piston pump'. MSc. Thesis DEPT. MECH. ENG. UNIVERSITY OF BIRMINGHAM 1976

7.2 PRIVATE COMMUNICATION: MALCOLM BUTLER, BATH UNIVERSITY

7.3 PRIVATE COMMUNICATION: KEVIN EDGE, BATH UNIVERSITY

7.4 'Noise in hydraulic systems'. University of Bath. School of engineering report No. BaU4 April 1978

7.5 FUNK, J.E., WOOD, D.J., CHAO, S.P., 'The transient response of orifices and very short lines'. J. BASIC. ENG. ASME. JUNE 1972

8.1 BOWNS, D.E., TOMLINSON, S.P., DUGDALE, S.K., 'Progress towards a general purpose hydraulic system simulation language'. B.H.R.A. SYMPOSIUM MANUAL 1981

8.2 BOWNS, D.E., ROLFE, A.C., 'The digital computation of pressure and flows in interconnected fluid volumes, using lumped parameter theory'. 4th INTERNATIONAL FLUID POWER SYMPOSIUM 1975 (SHEFFIELD ENGLAND).

** HYTRAN is available from a software firm:-
National Technical Information Service,
5285 Port Royal Road,
Springfield,
Virginia 22151,
U.S.A.

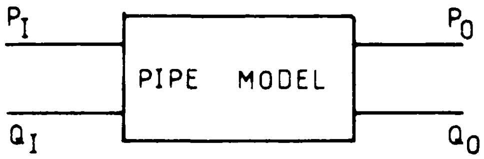

FIGURE 2·1a COMPONENT BLOCK REPRESENTATION OF A PIPE

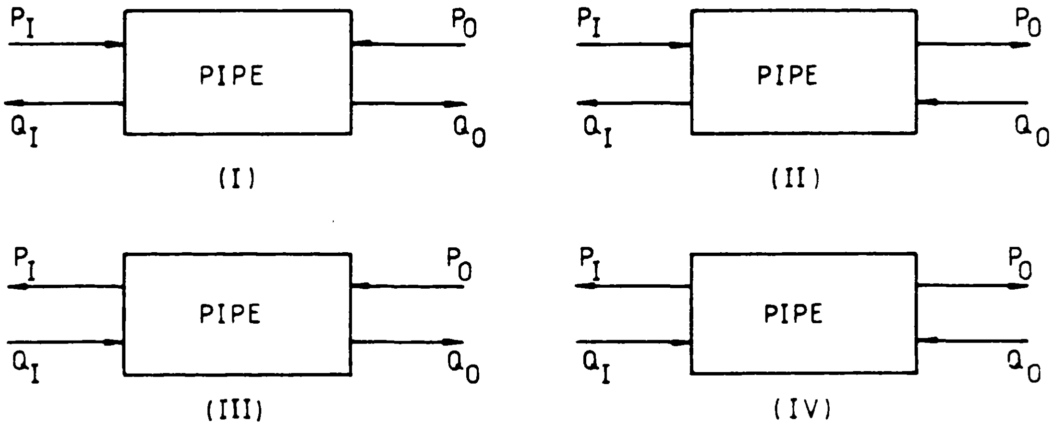

FIGURE 2·1b POSSIBLE CONFIGURATIONS OF THE TWO PORT-FOUR TERMINAL NETWORK

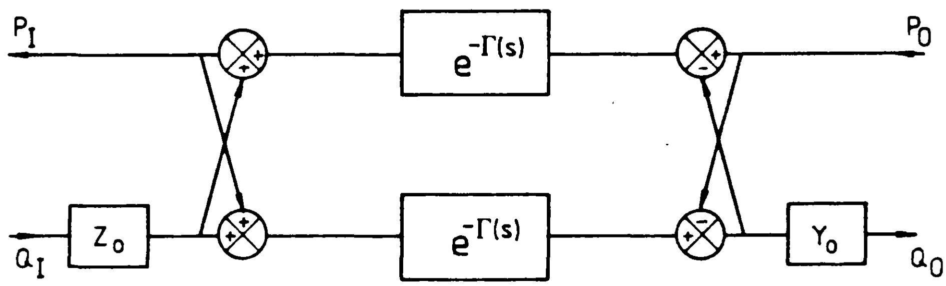

FIGURE 2·1c BLOCK DIAGRAM FOR CONFIGURATION (III)

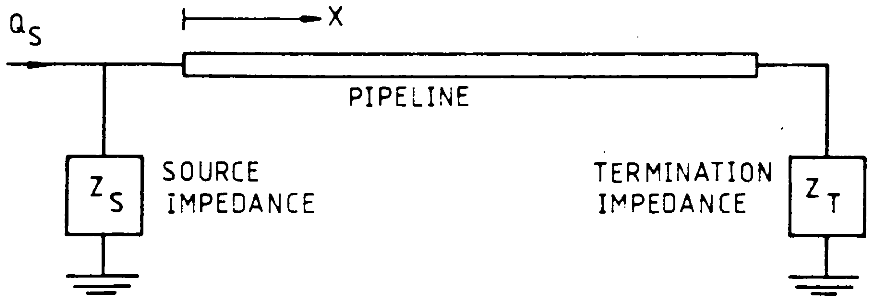

FIGURE 2·2 SOURCE-LINE-TERMINATION SYSTEM

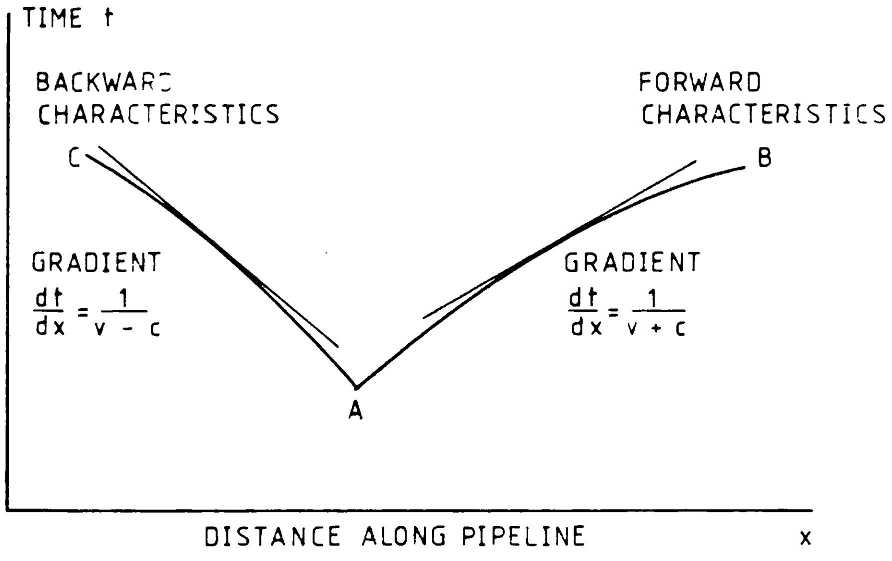

FIGURE 2·3 CHARACTERISTICS IN THE TIME DISTANCE PLANE

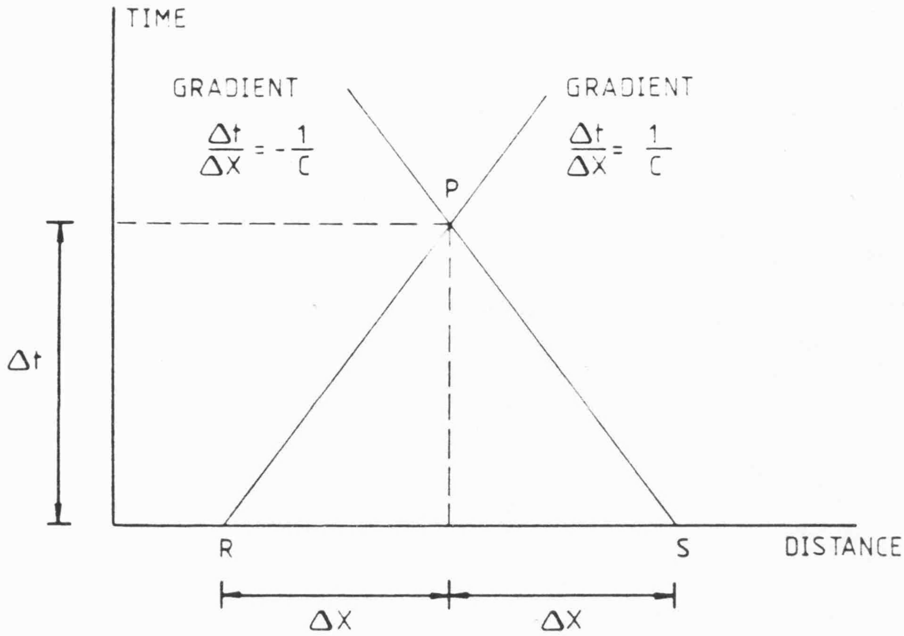

FIGURE 2·4 WAVE PROPAGATION FROM TWO POINTS IN A PIPE

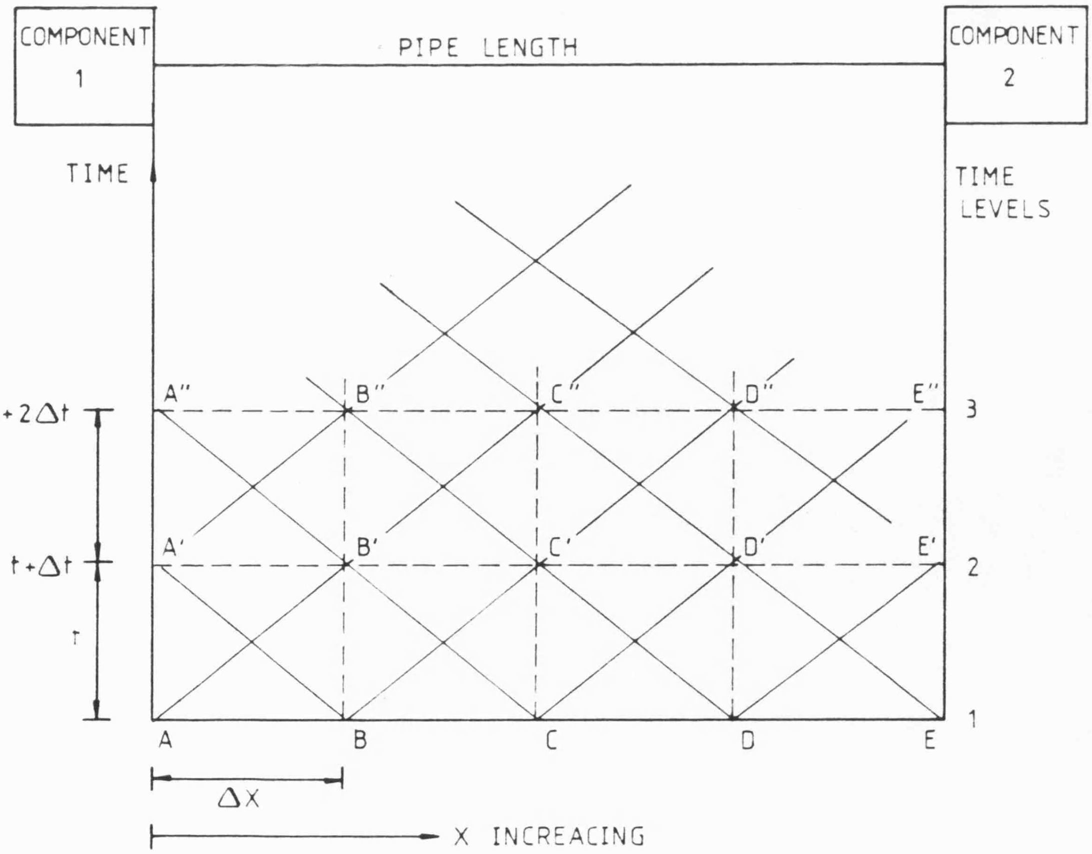

FIGURE 2·5 PIPELINE SHOWN ON THE TIME-DISTANCE PLANE

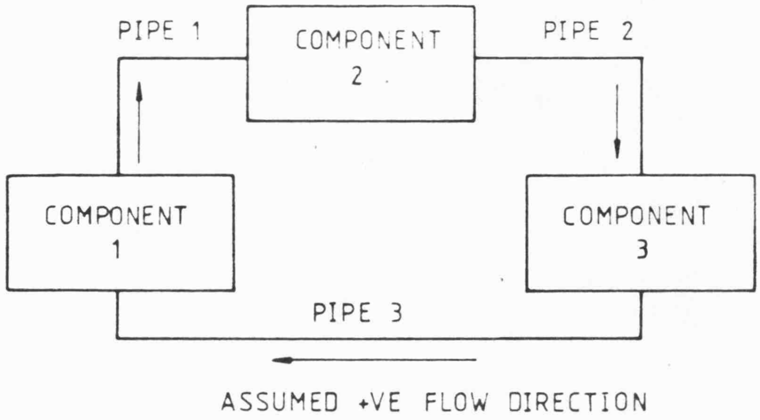

FIGURE 2·6a SIMPLE 3 COMPONENT SYSTEM

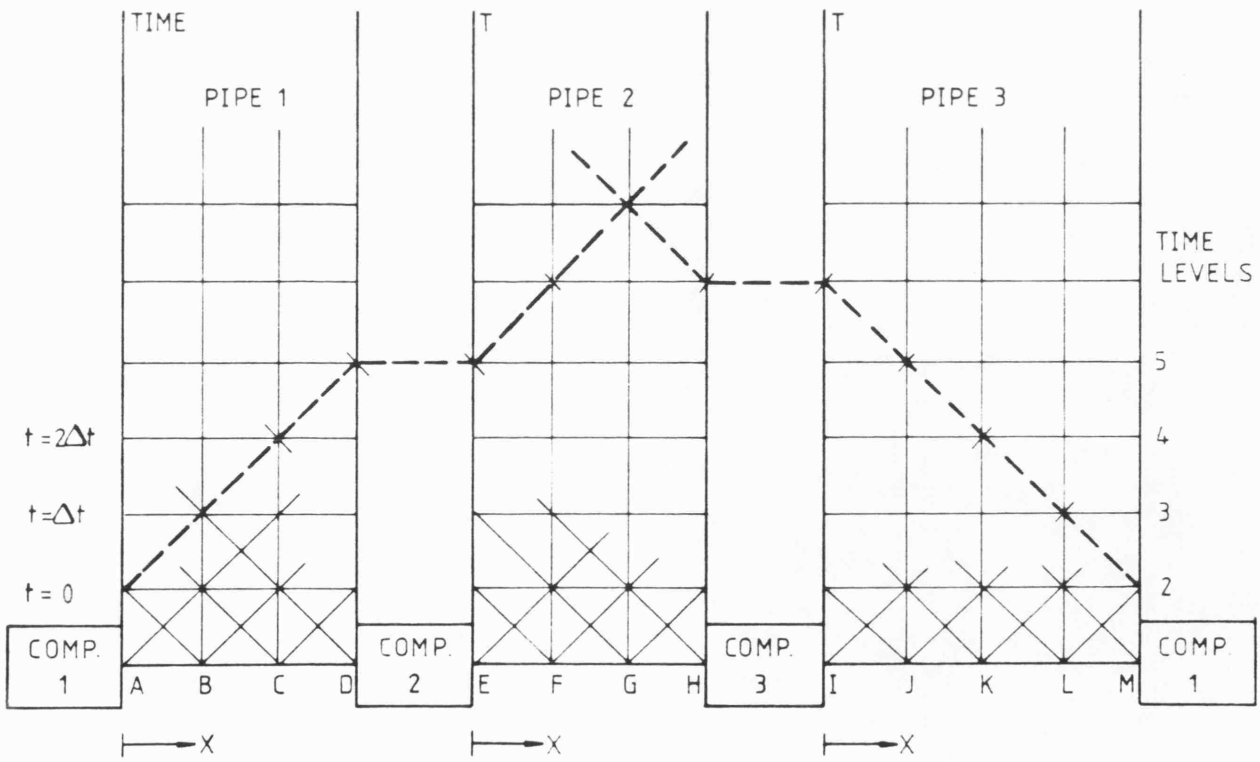

FIGURE 2·6b 3 COMPONENT SYSTEM ON THE TIME-DISTANCE PLANE

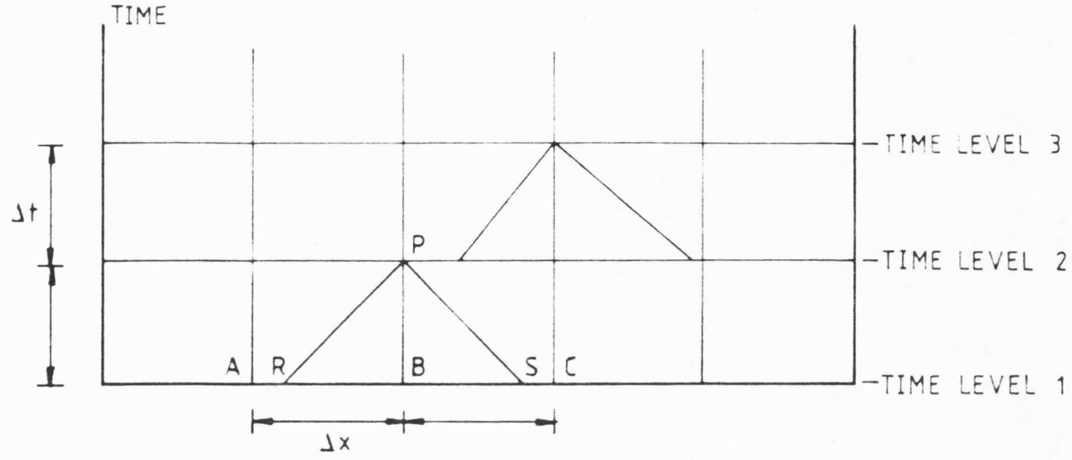

FIGURE 2·7 METHOD OF SPECIFIED INTERVALS

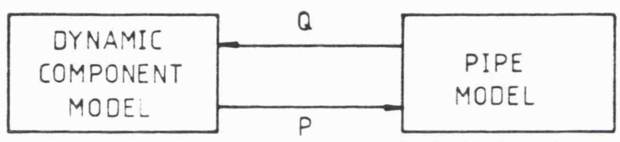

FIGURE 2·8a INFORMATION TRANSFER BETWEEN COMPONENT MODEL AND PIPE MODEL

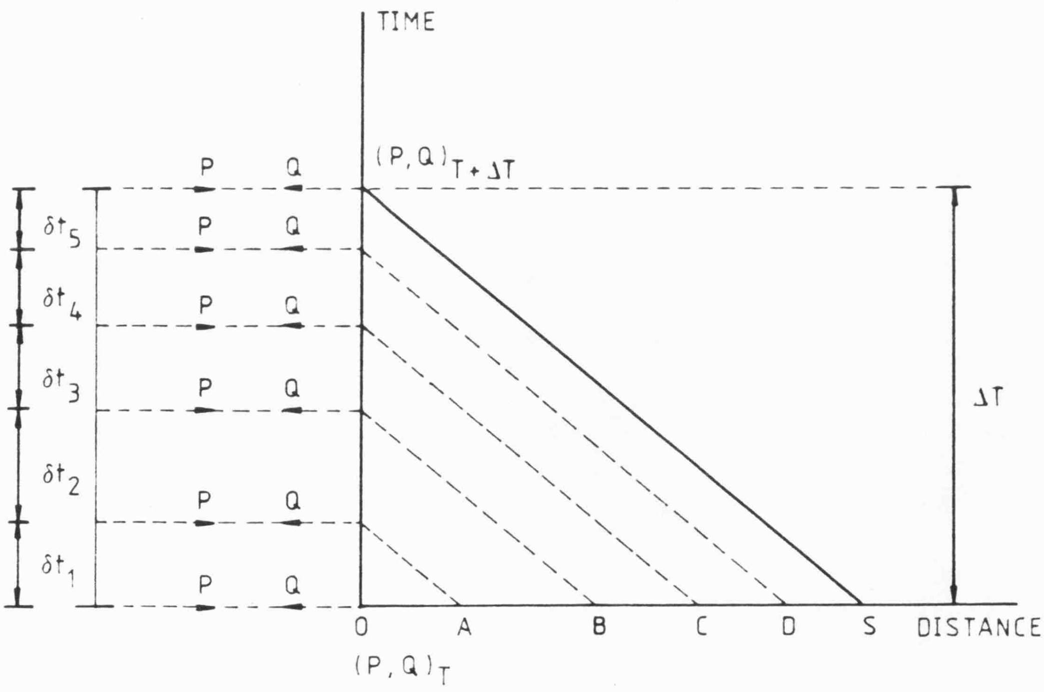

FIGURE 2·8b INTERFACING INTEGRATION THE PIPE AND COMPONENT MODELS

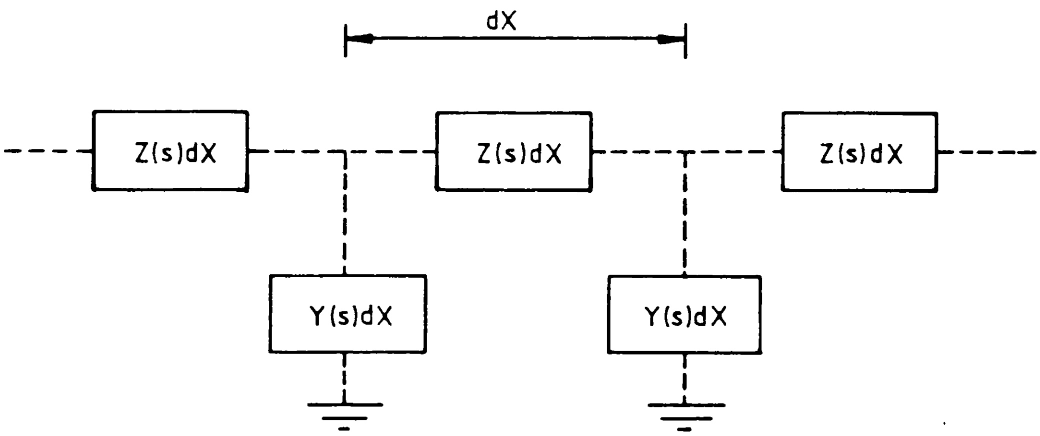

FIGURE 3·1 SERIES IMPEDANCE / SHUNT ADMITTANCE REPRESENTATION OF A FLUID PIPELINE

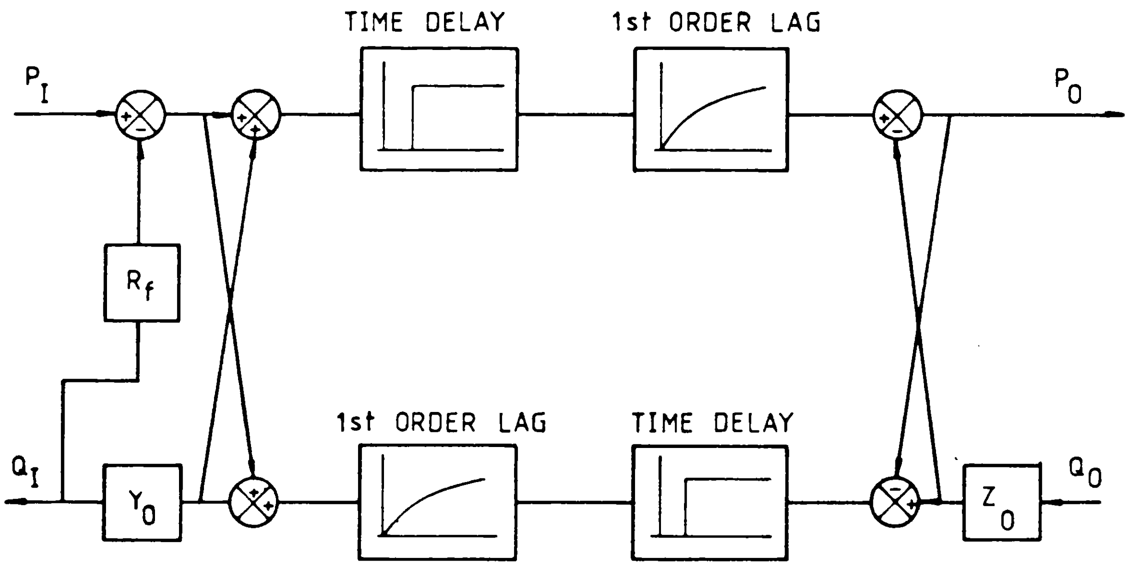

FIGURE 3·2 LEONARDS SIMPLIFIED TWO PORT MODEL

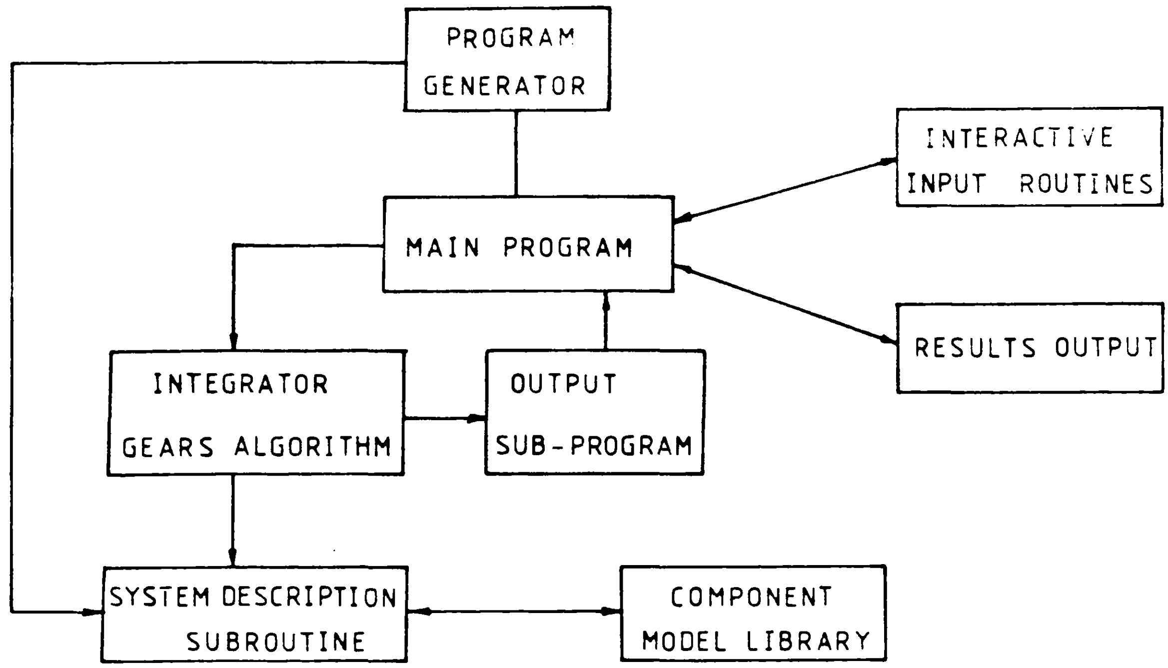

FIGURE 4.1 STRUCTURE OF THE HGSP SIMULATION PROGRAM

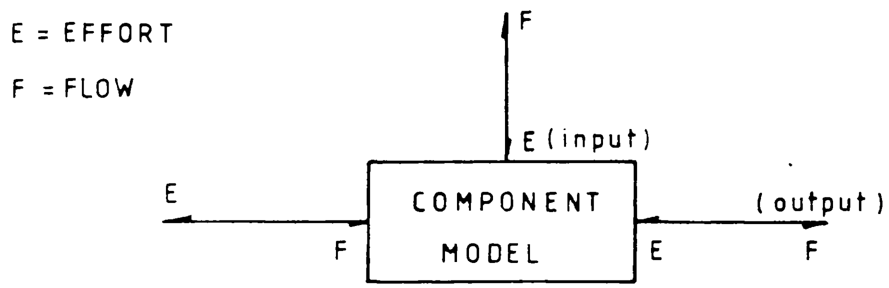

FIGURE 4.2 HGSP COMPONENT MODEL FORMAT

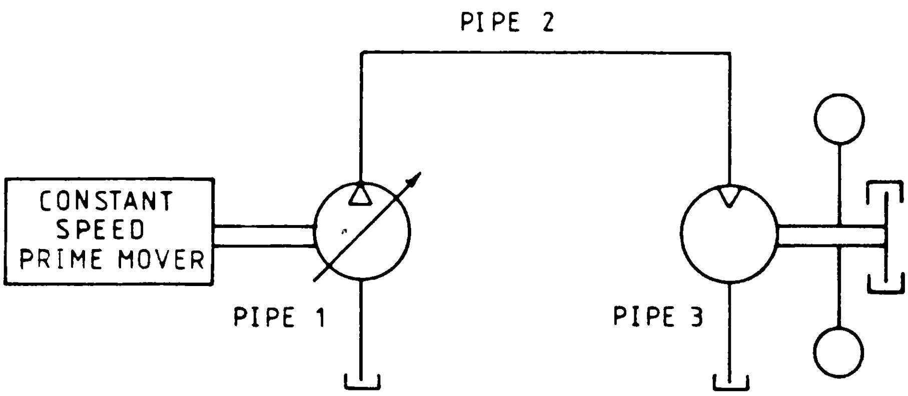

FIGURE 5·1 SIMPLE TRANSMISSION SYSTEM

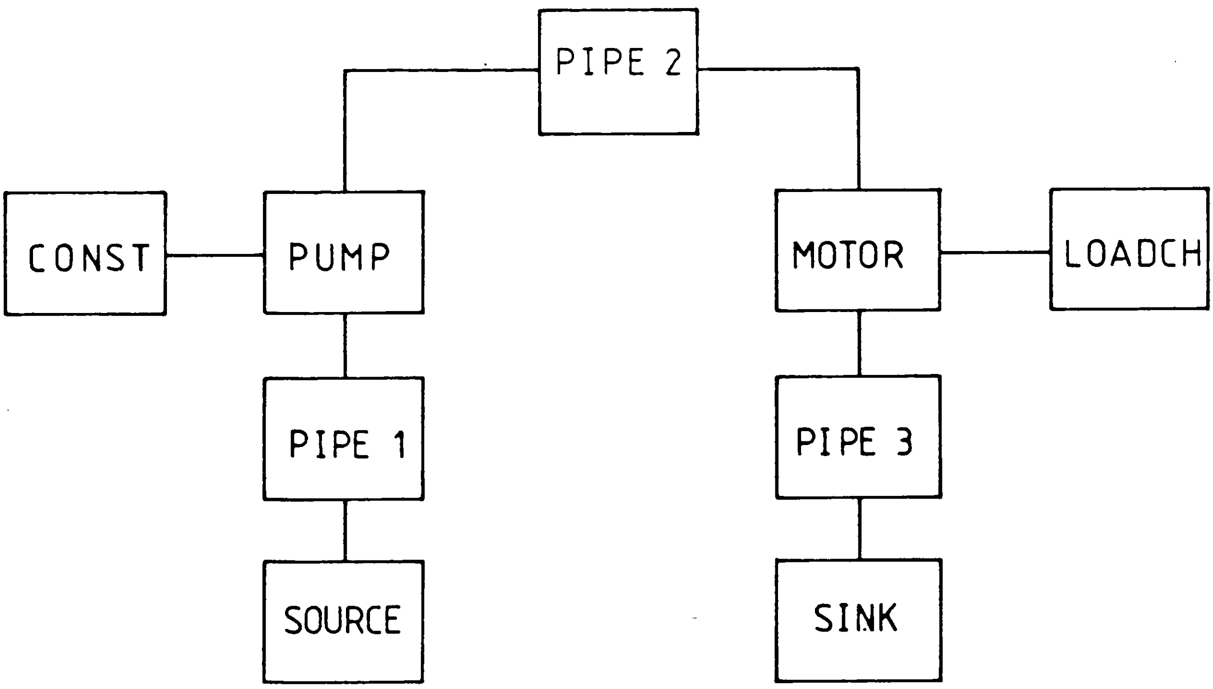

FIGURE 5·2 COMPONENT BLOCK DIAGRAM FOR TRANSMISSION SYSTEM

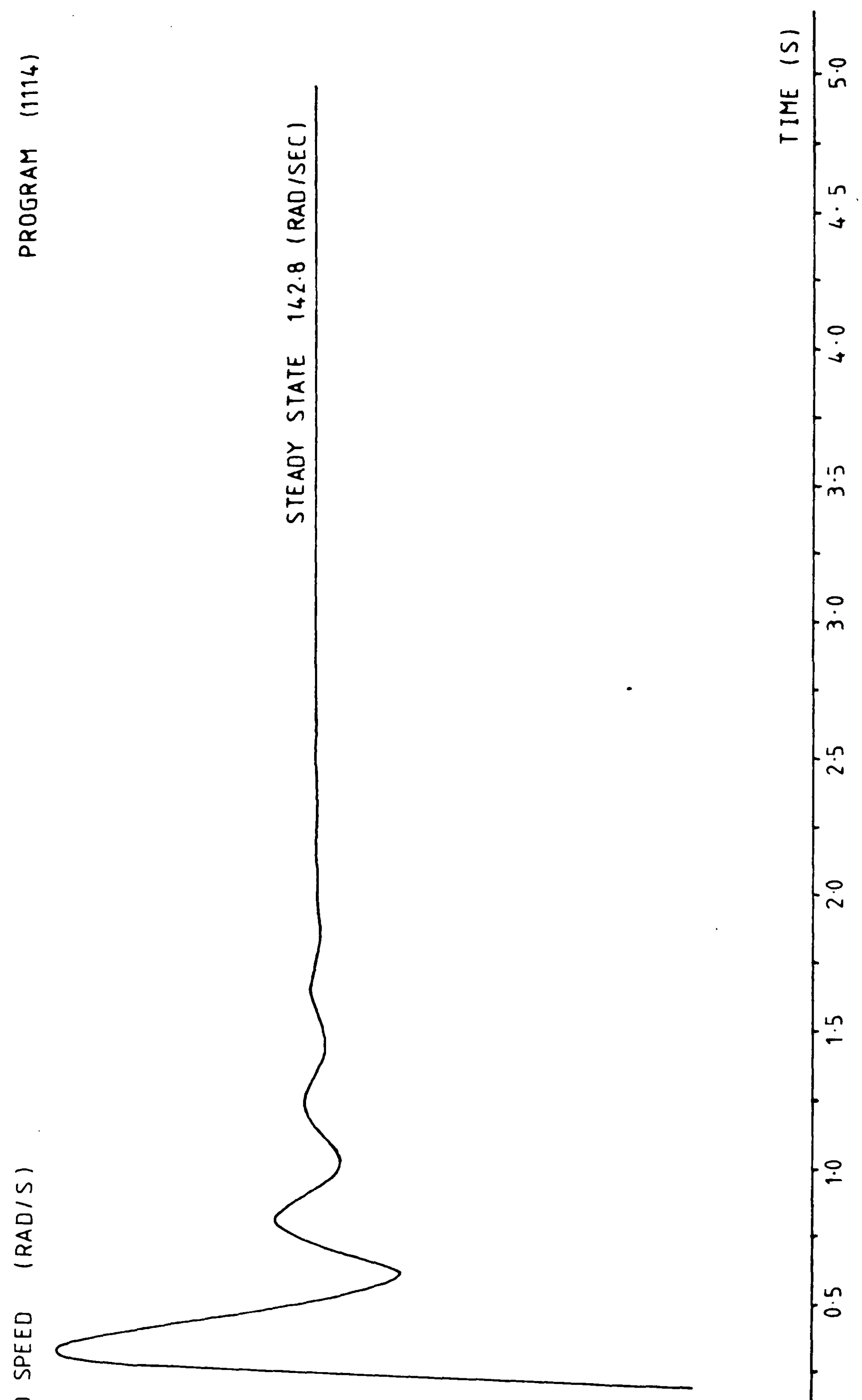

FIGURE 5.3 LOAD SPEED FLUCTUATION

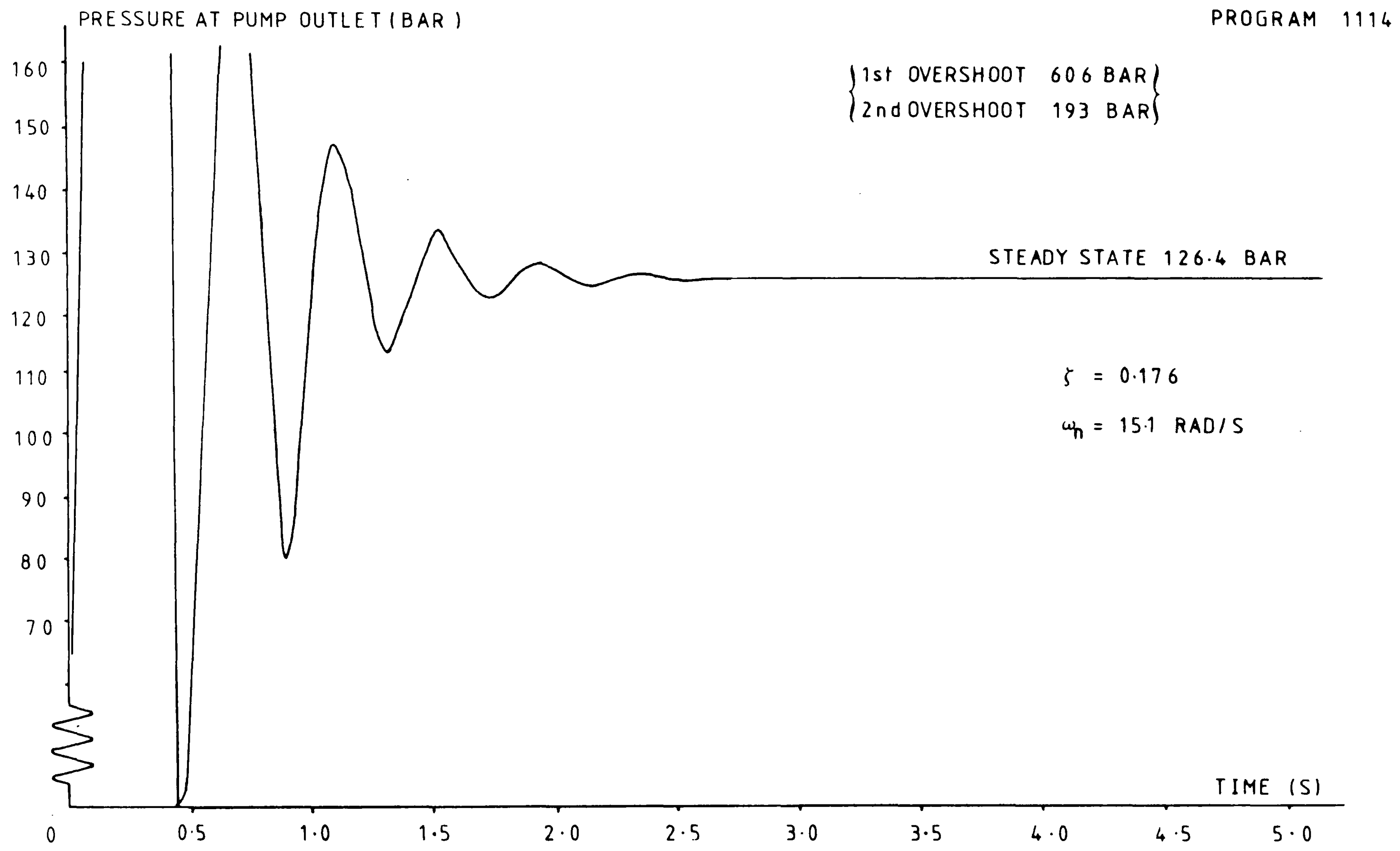

FIGURE 5·4 PRESSURE FLUCTUATION AT PUMP OUTLET

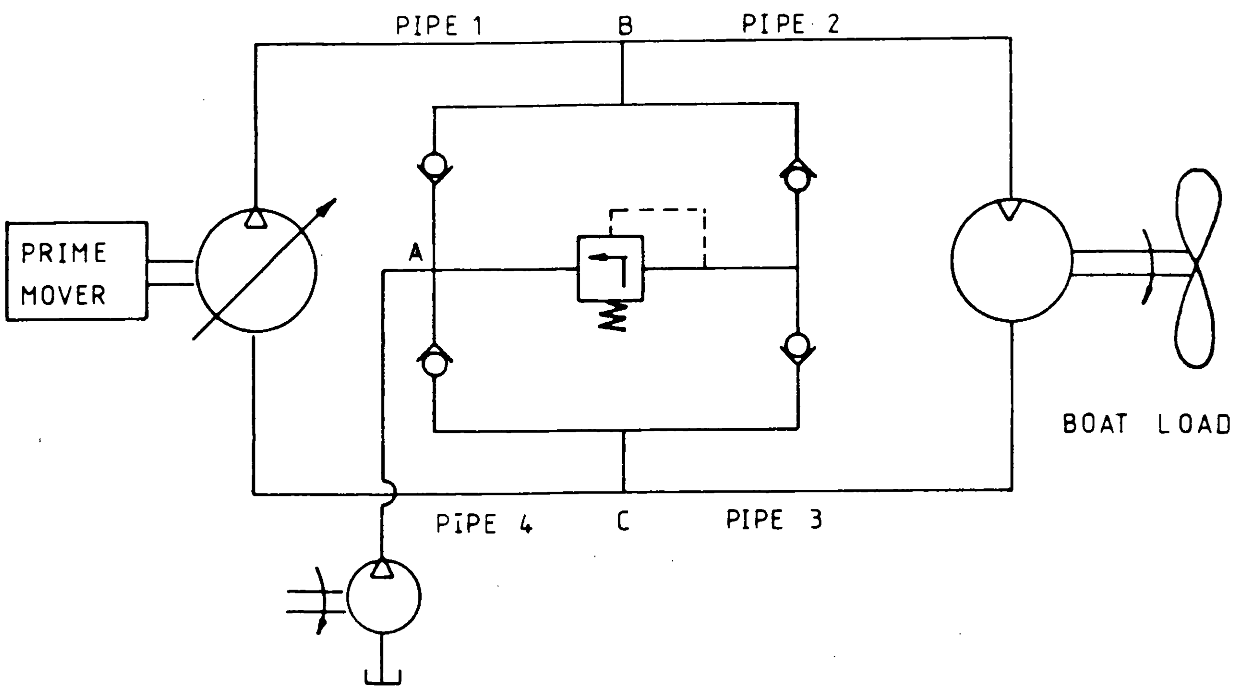

FIGURE 5·5 BOAT TRANSMISSION

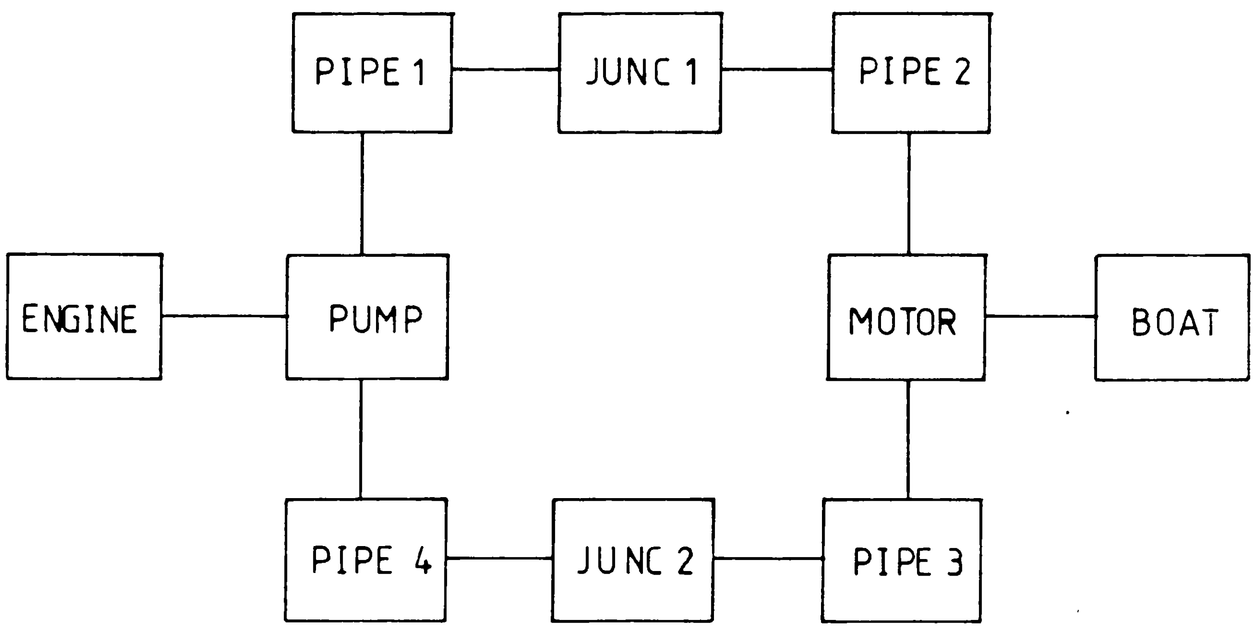

FIGURE 5·6 COMPONENT BLOCK DIAGRAM FOR BOAT TRANSMISSION

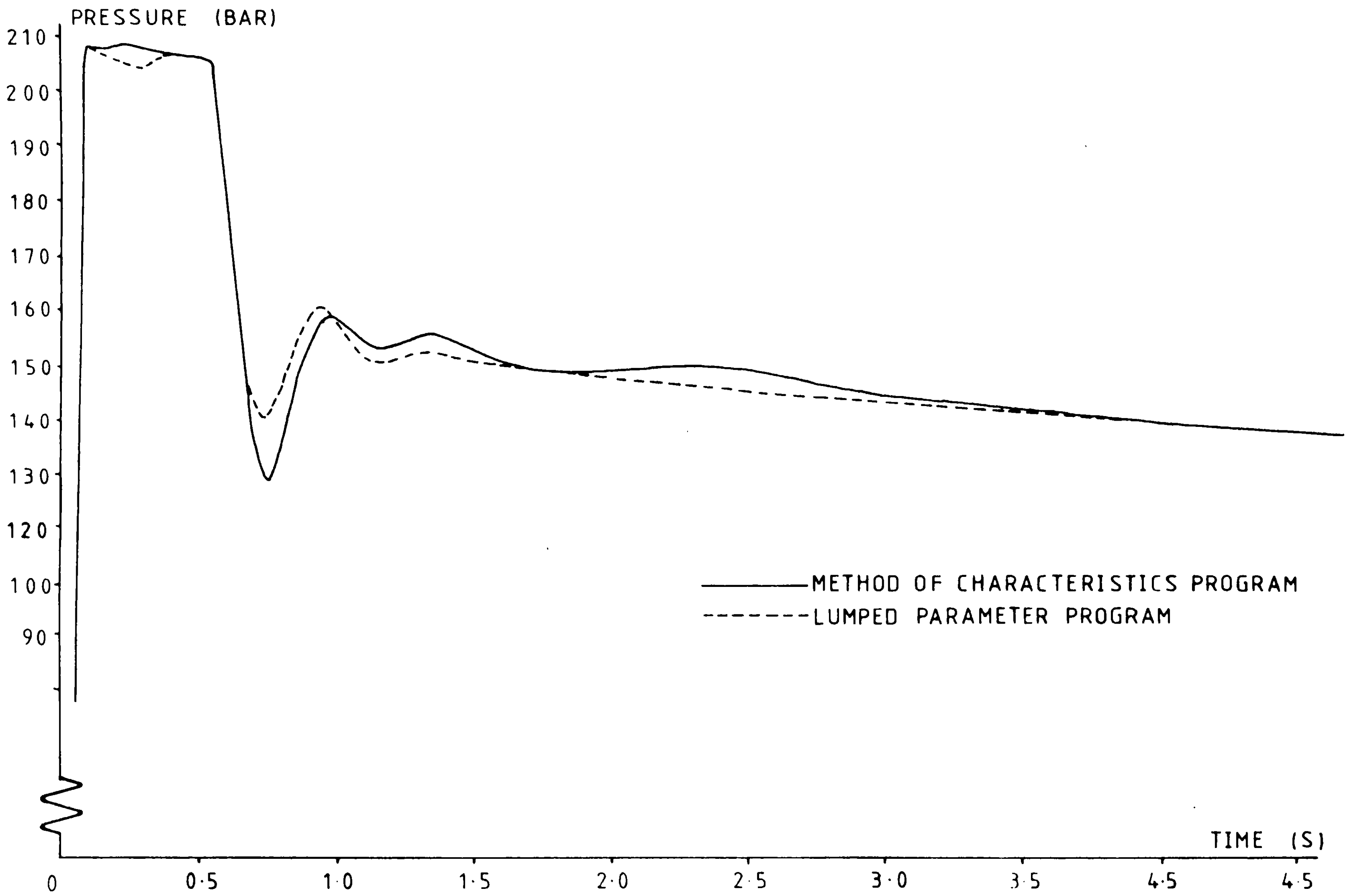

FIGURE 5·7 BOAT TRANSMISSION - PRESSURE AT PUMP OUTLET

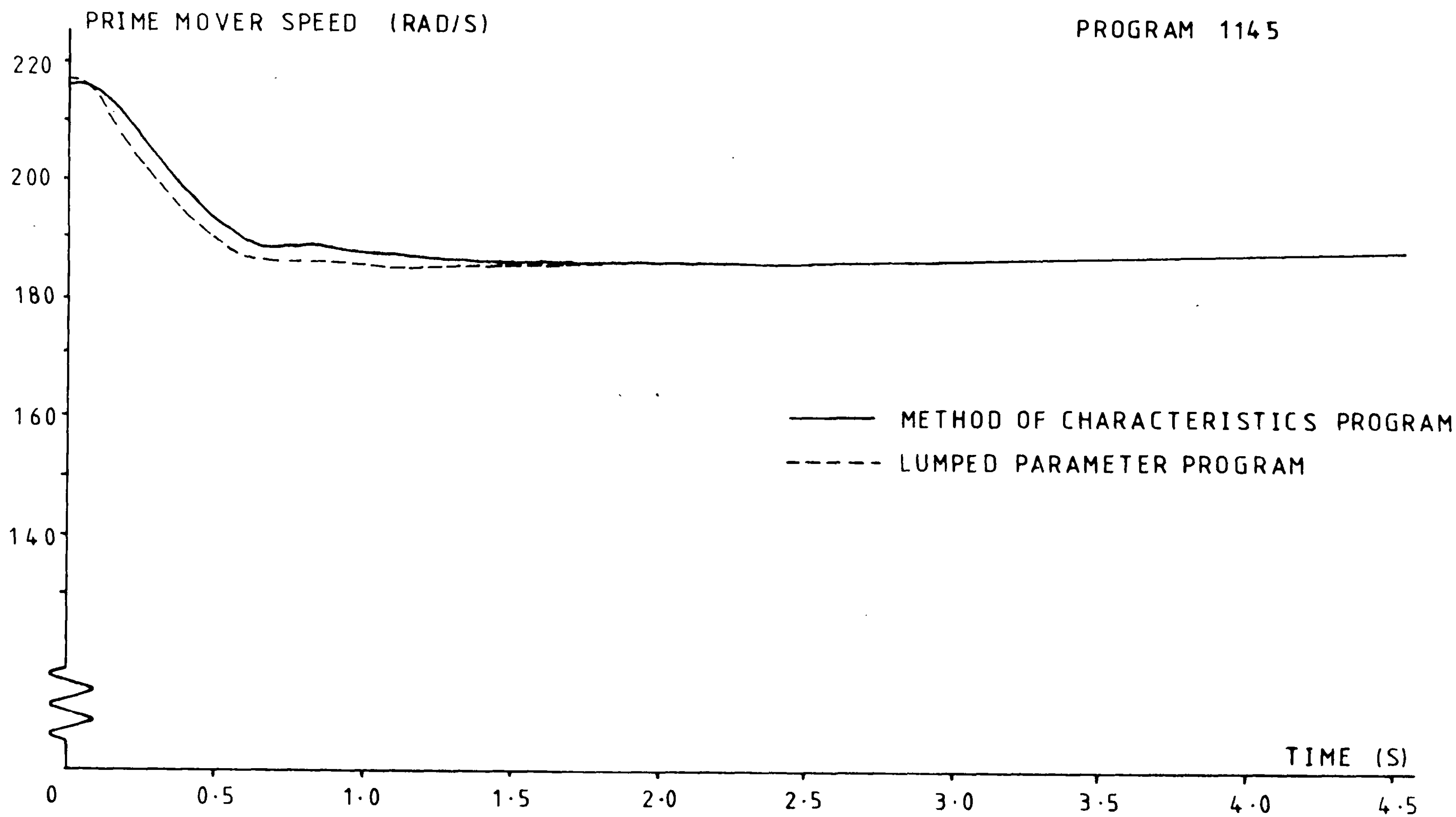

FIGURE 5·8 BOAT TRANSMISSION-FLUCTUATION IN PRIME MOVER SPEED

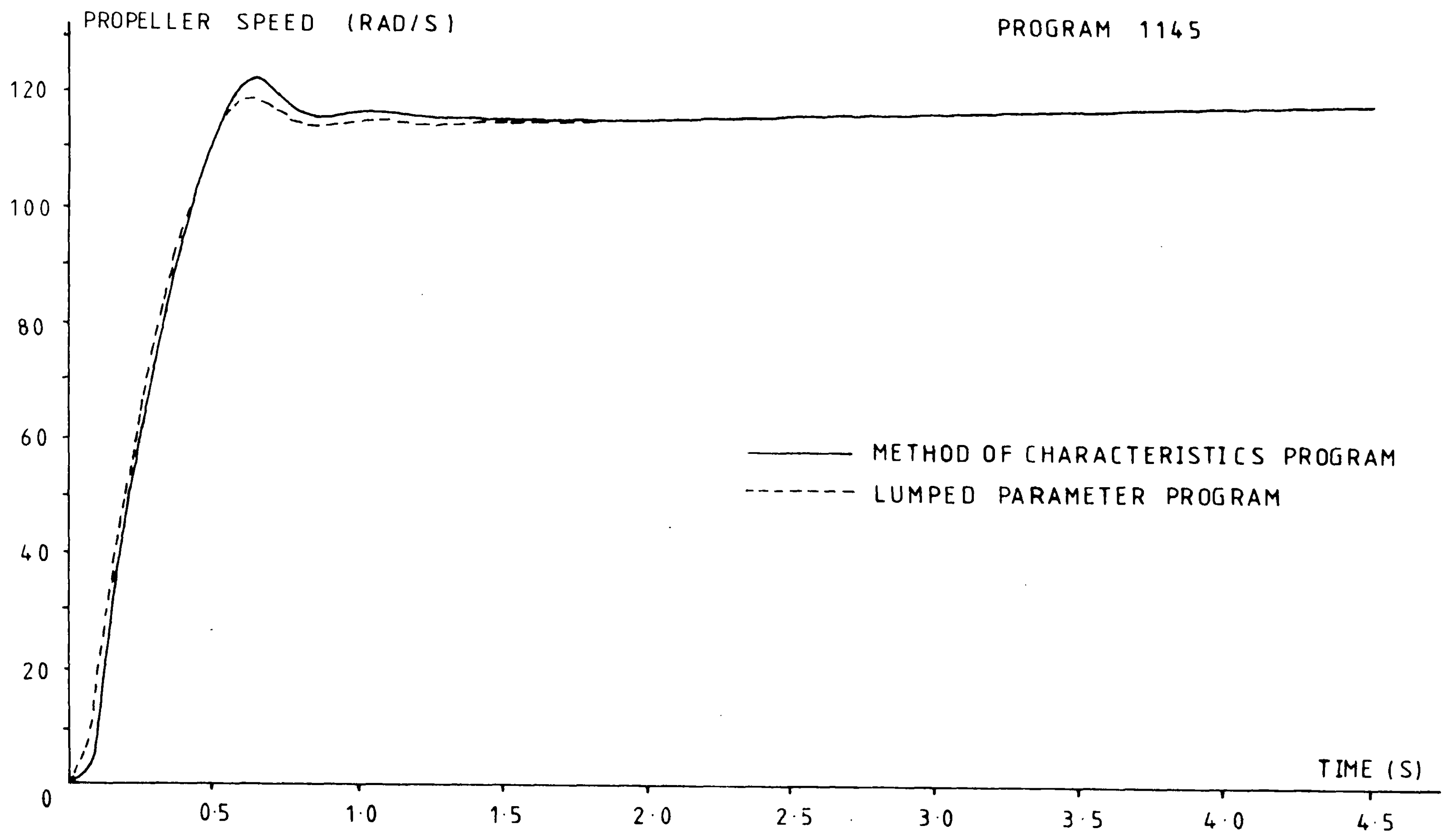

FIGURE 5·9 BOAT TRANSMISSION - FLUCTUATIONS IN PROPELLER SPEED

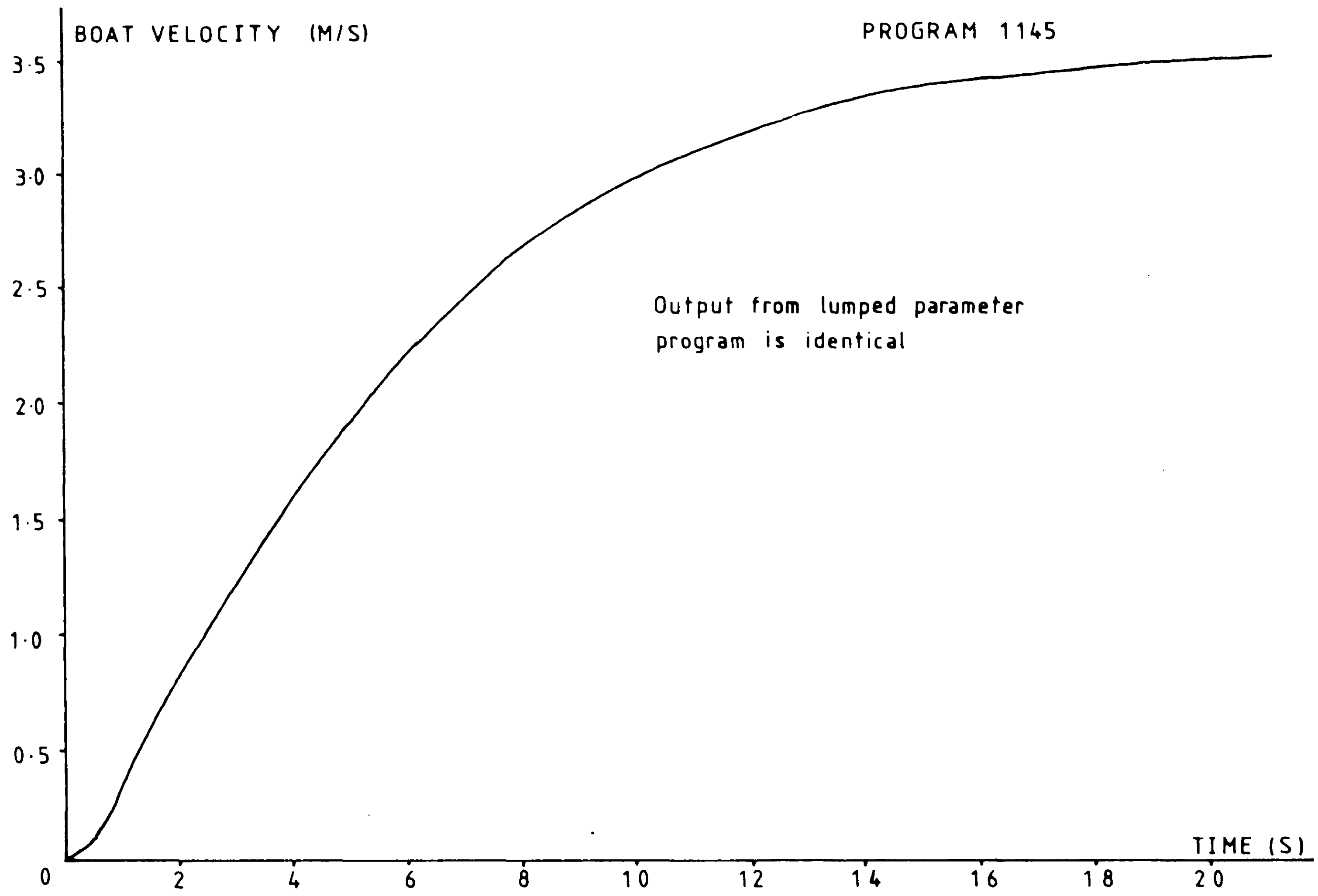

FIGURE 5.10 BOAT TRANSMISSION - BOAT VELOCITY

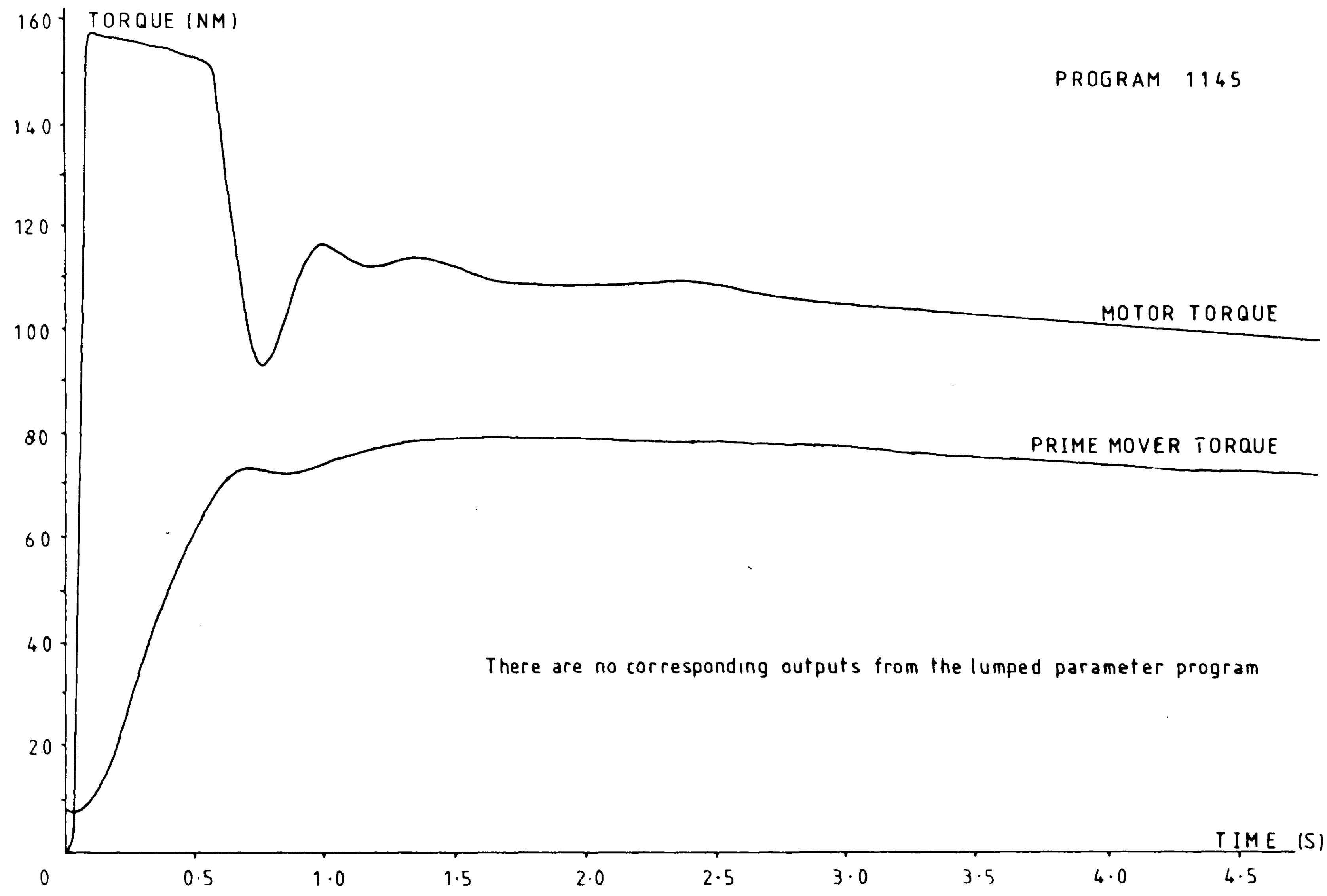

FIGURE 5·11 BOAT TRANSMISSION - TORQUE FLUCTUATION AT MOTOR & PRIME MOVER

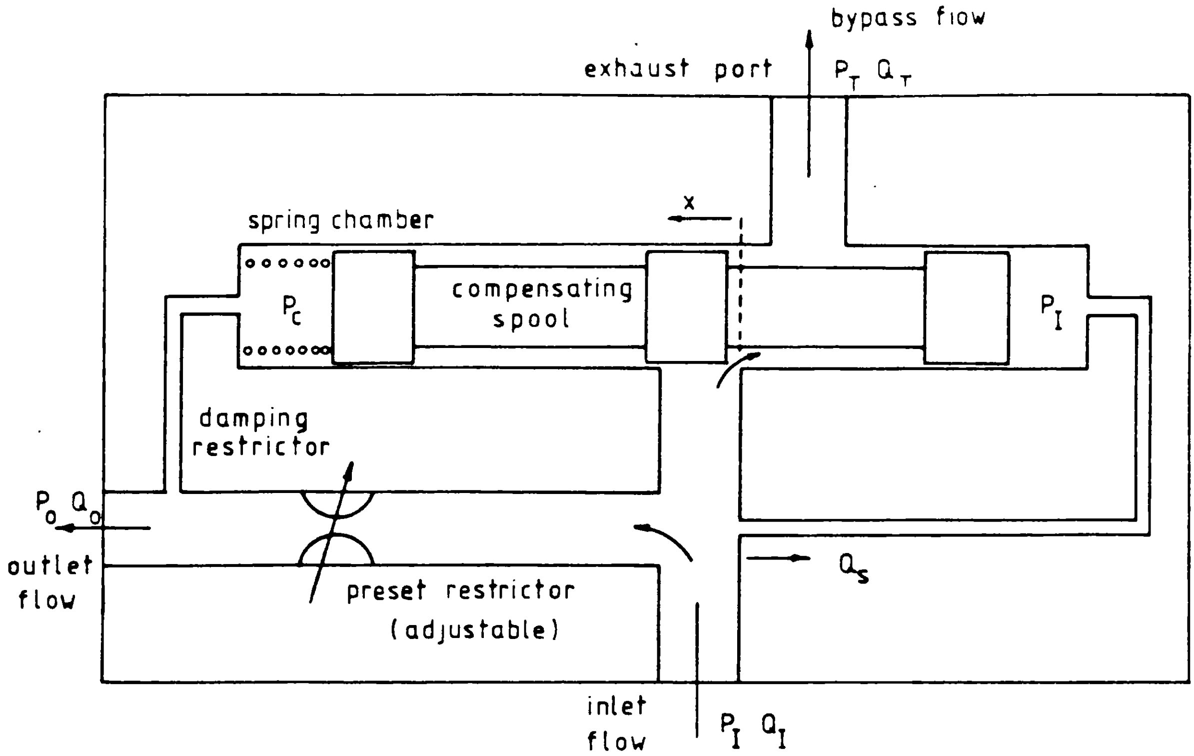

FIGURE 6·1 SCHEMATIC DIAGRAM OF BARMAG VALVE

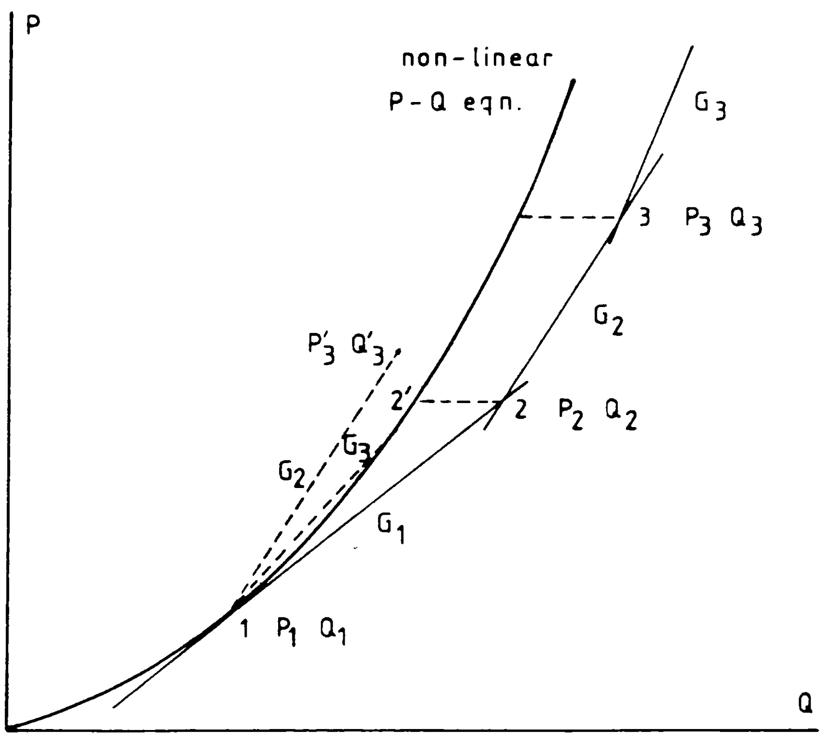

FIGURE 6·2 DIAGRAM ILLUSTRATING LINEARISING PROCEDURE

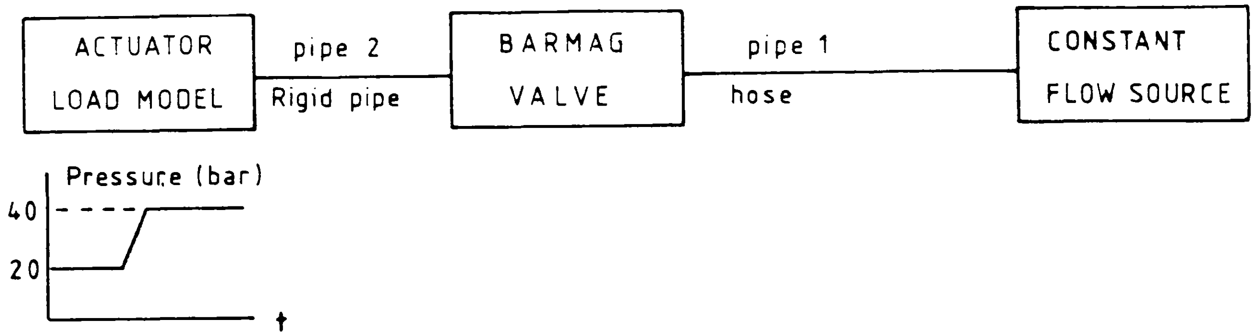

FIGURE 6·3 IDEALISED SYSTEM

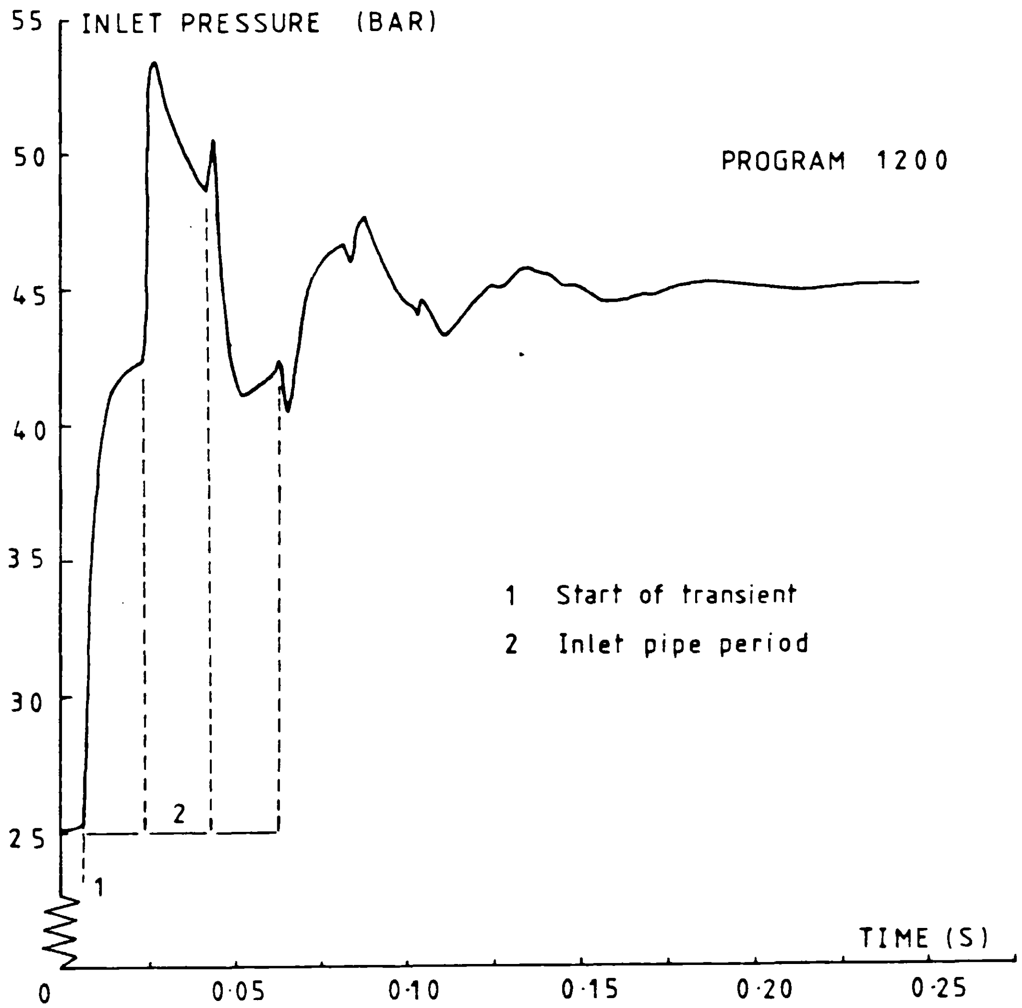

FIGURE 6·4 BARMAG VALVE INLET PRESSURE (P_I)

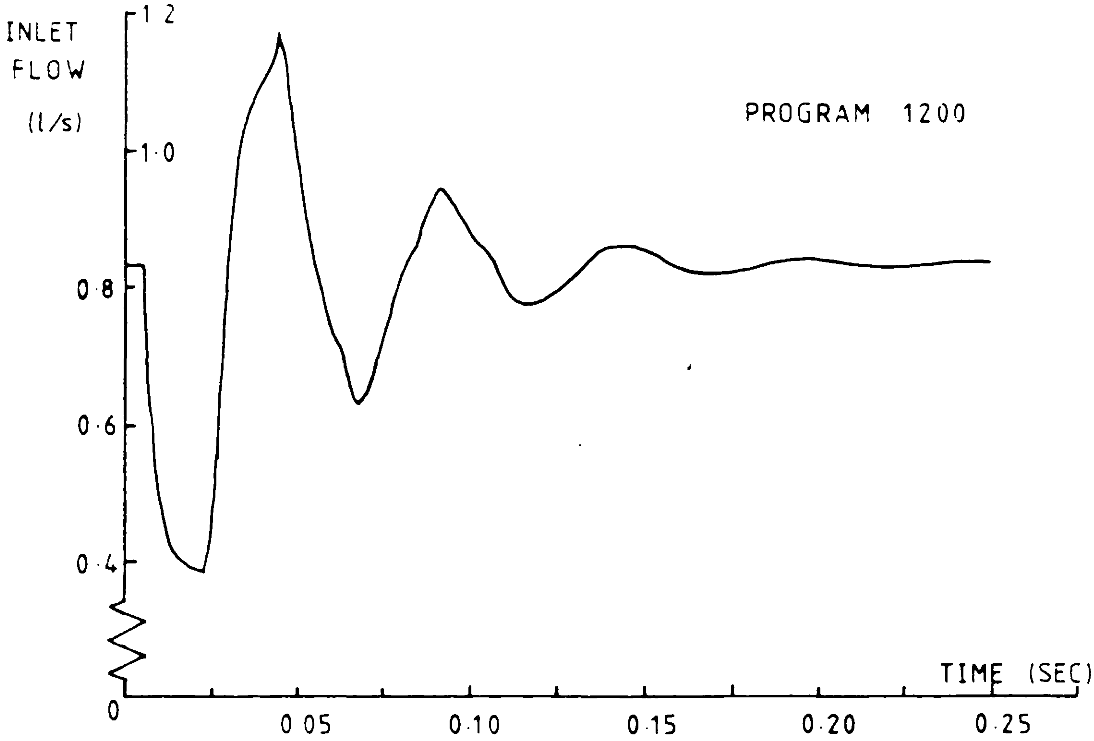

FIGURE 6.5 BARMAG VALVE INLET FLOW (Q_I)

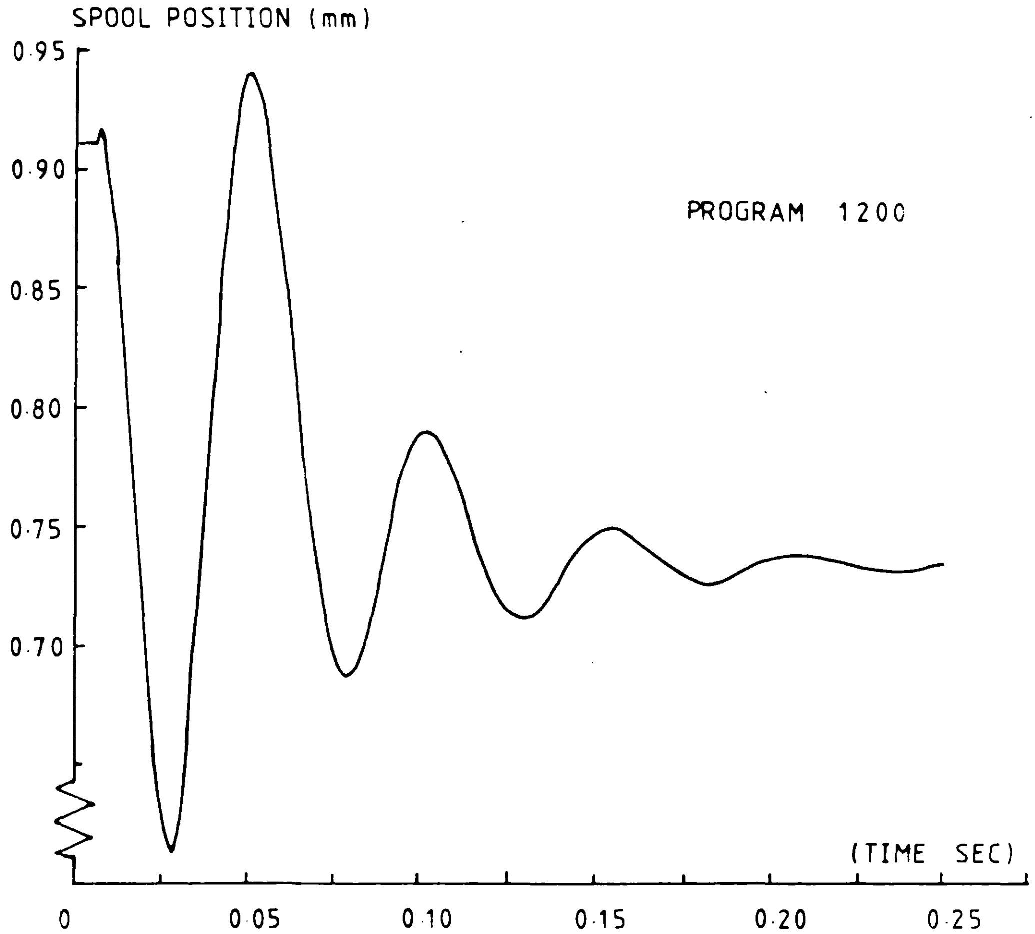

FIGURE 6.6 BARMAG VALVE SPOOL POSITION (x)

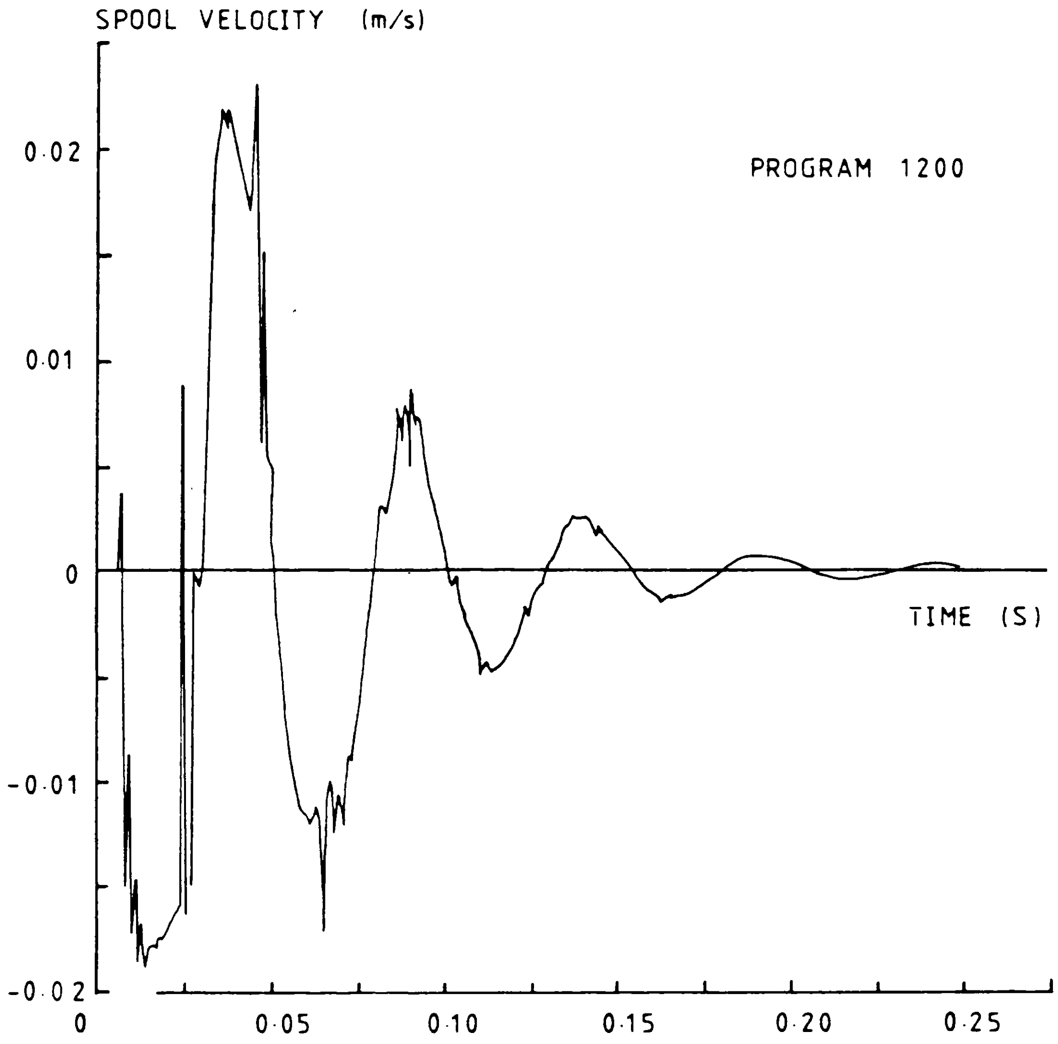

FIGURE 6·7 BARMAG VALVE SPOOL VELOCITY ($\dot{x}$)

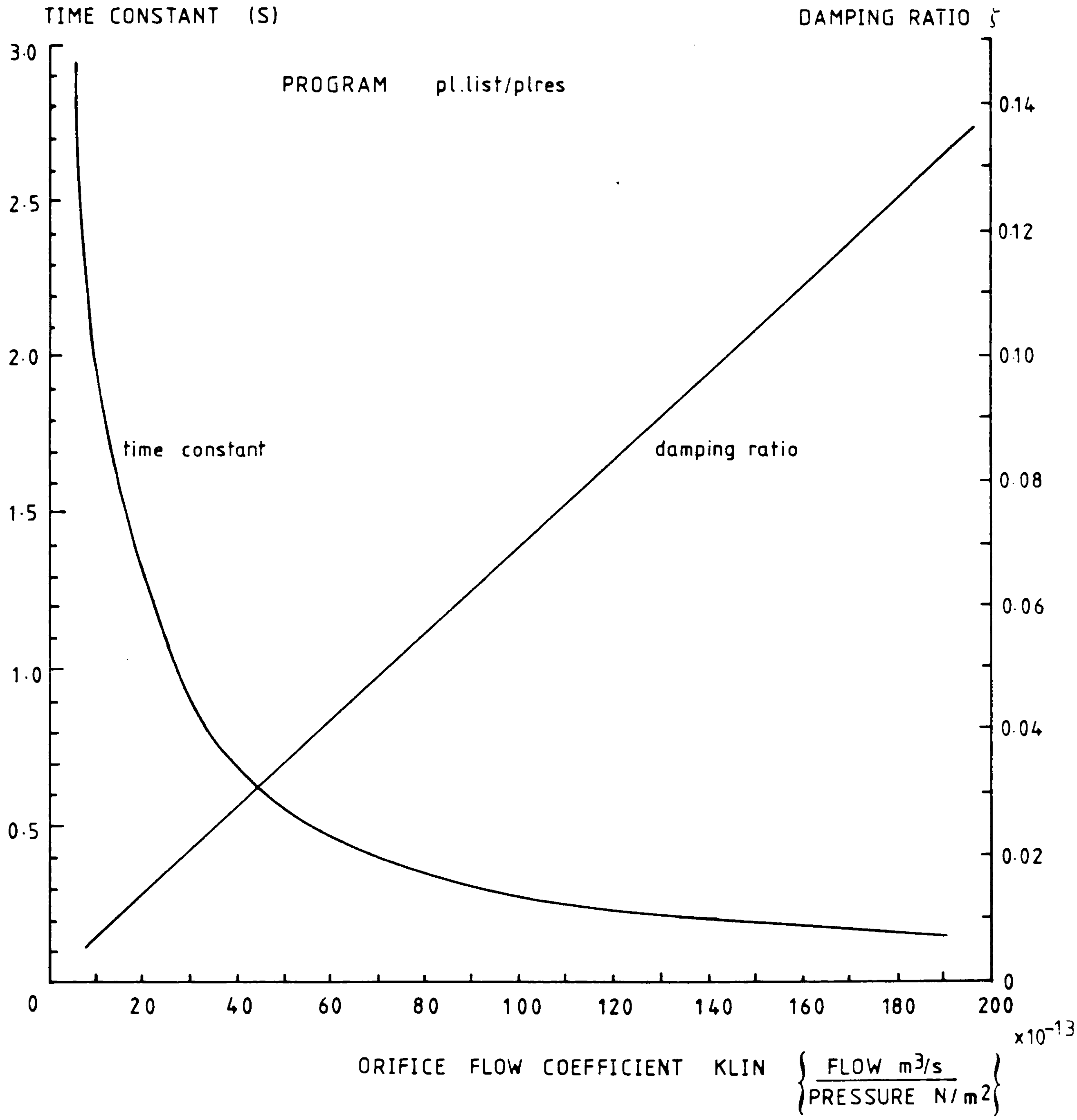

FIGURE 6.8 TIME CONSTANT & DAMPING RATIO FOR SPOOL DAMPER MECHANISM

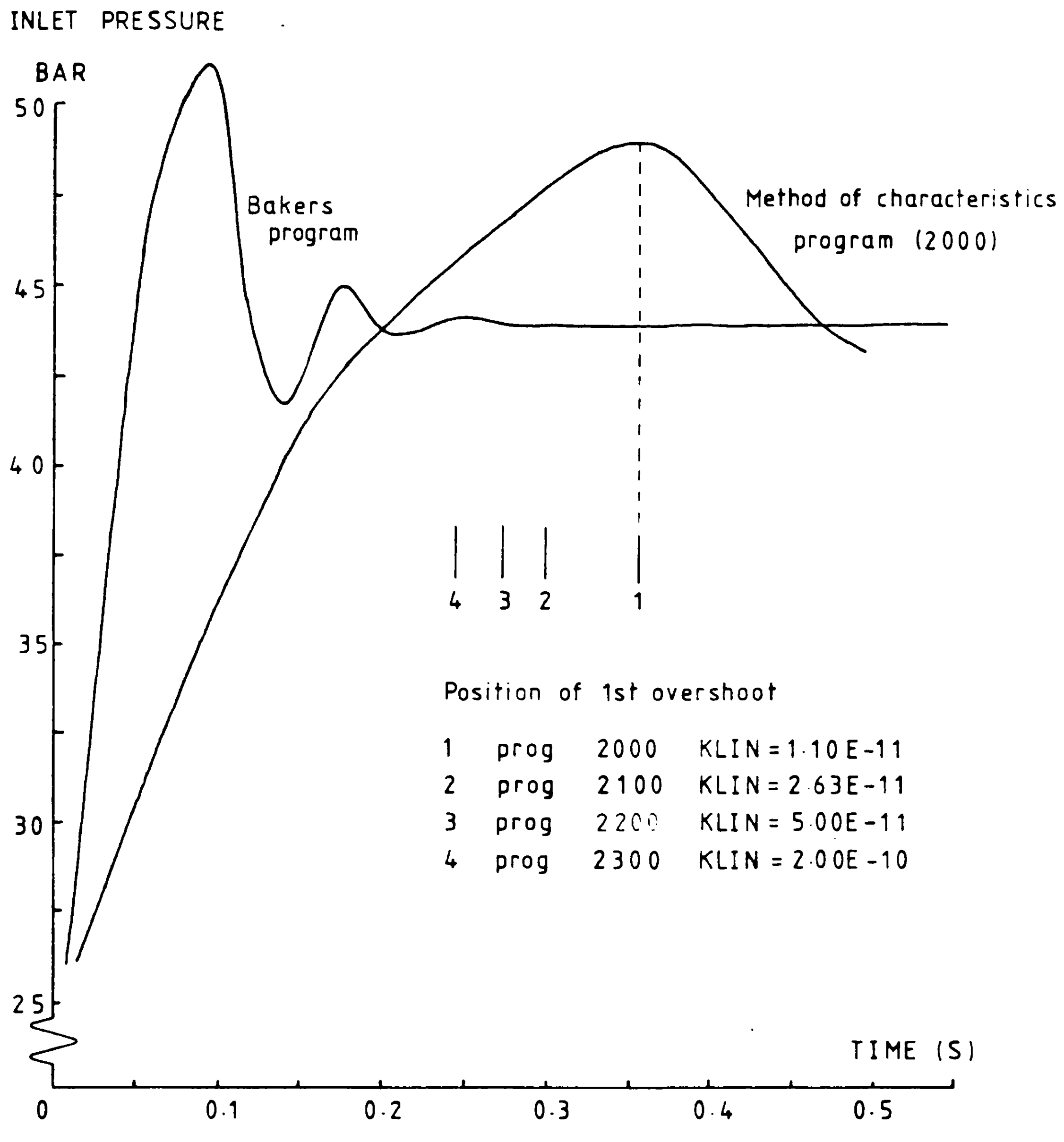

FIGURE 6·9 COMPARISON BETWEEN LUMPED PARAMETER AND METHOD OF CHARACTERISTICS PROGRAMS

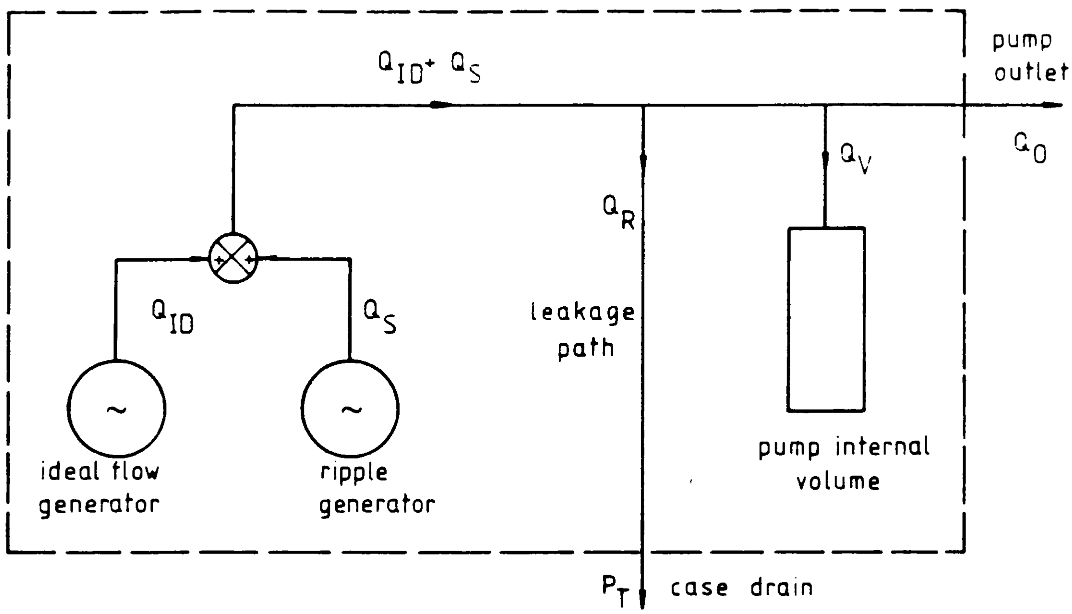

FIGURE 7·1 SIMPLE MODEL FOR A PUMP INCLUDING FLOW RIPPLE

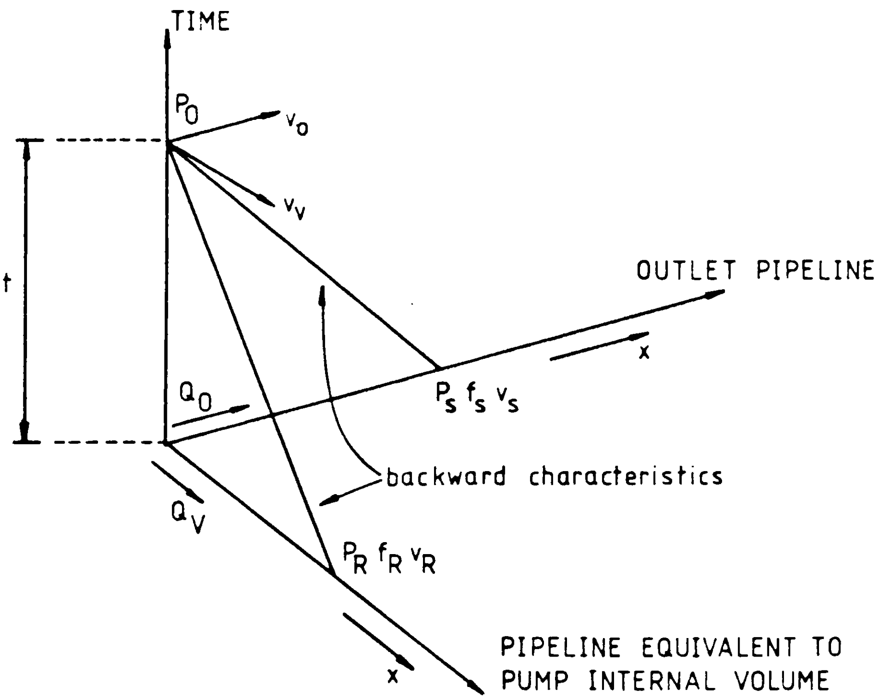

FIGURE 7·2 TIME-DISTANCE PLANE DIAGRAM FOR THE PUMP OUTLET

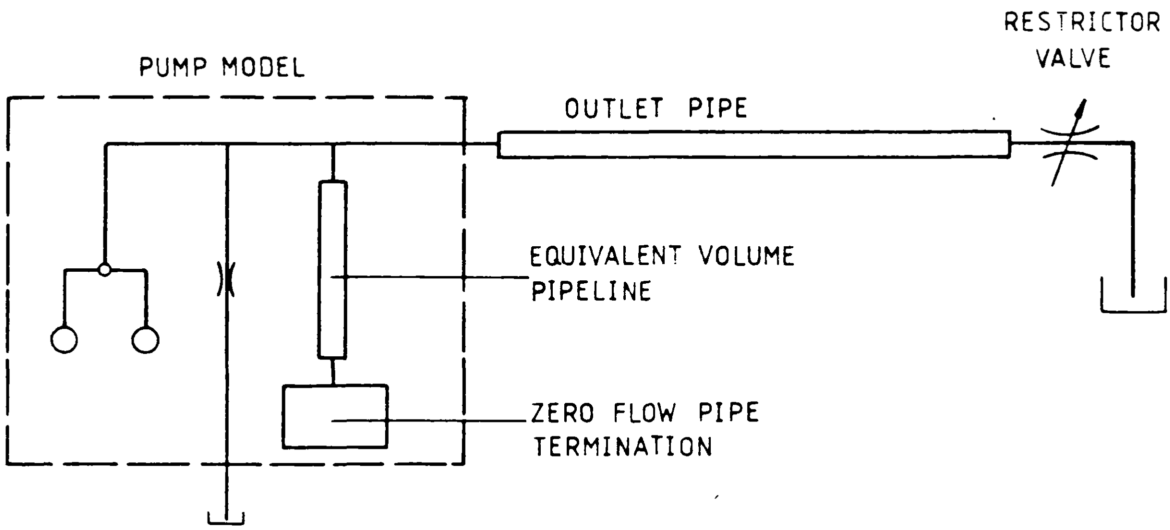

FIGURE 7.3 SYSTEM SIMULATED TO TEST THE PUMP MODEL

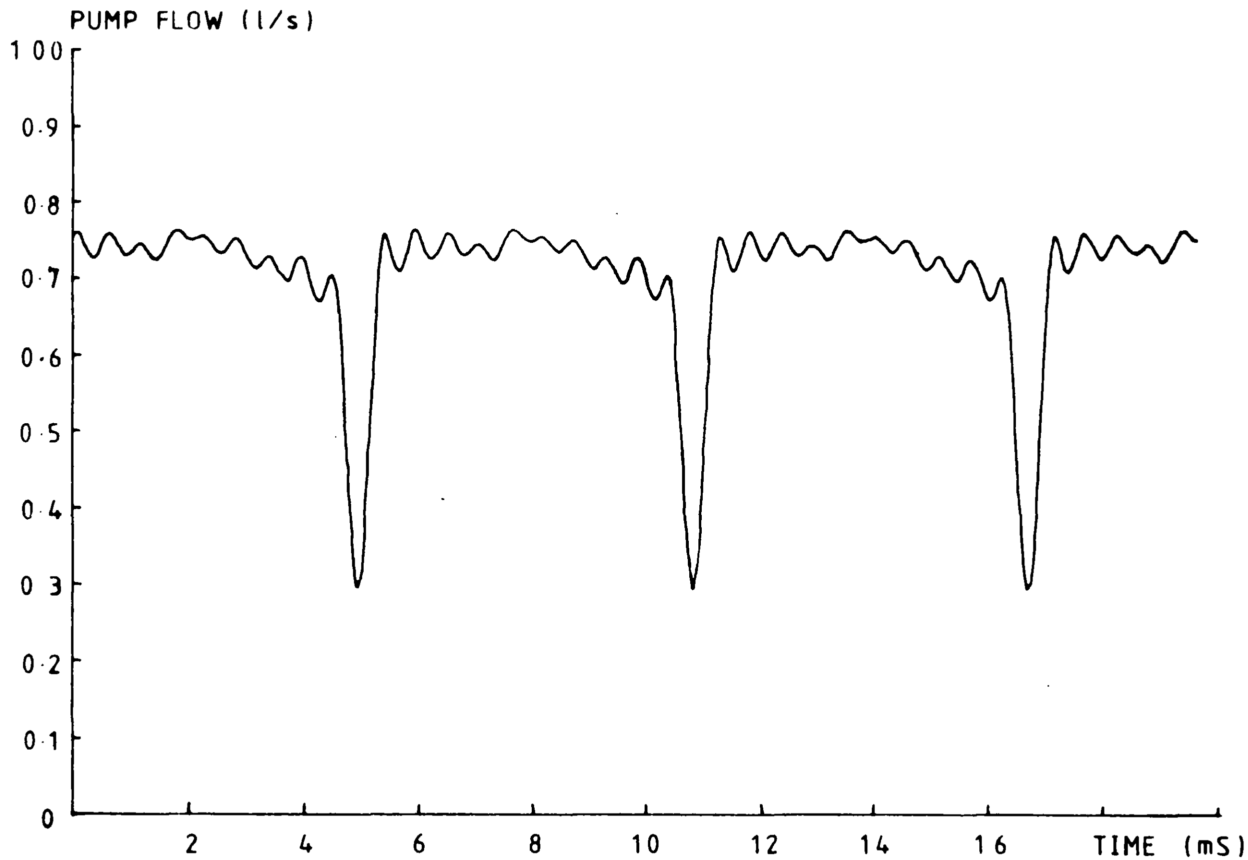

FIGURE 7·4 FLOW RIPPLE FROM REYROLLE A200 AT 200 BAR

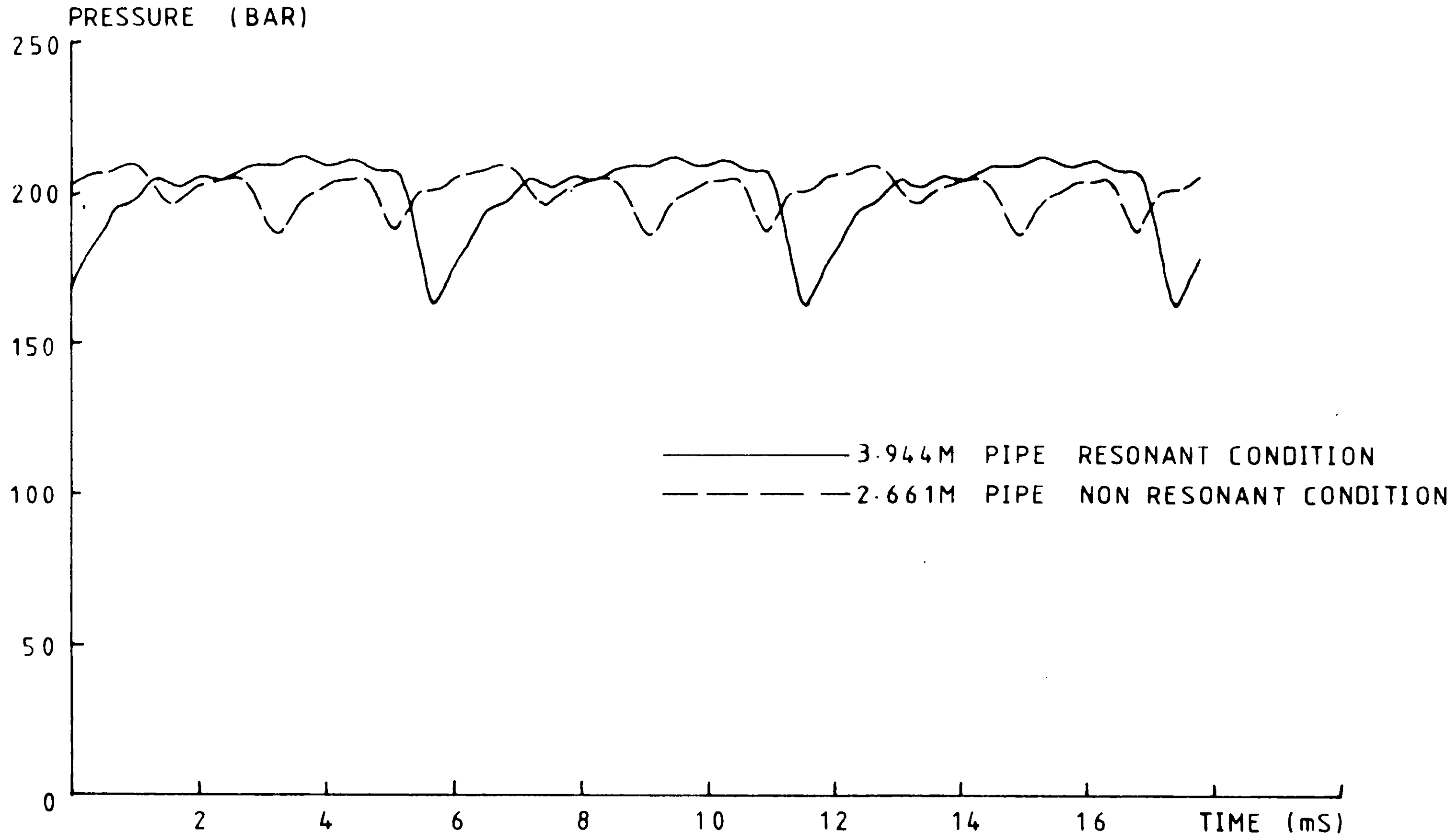

FIGURE 7·5 PRESSURE RIPPLE AT PUMP FOR TWO PIPE LENGTHS

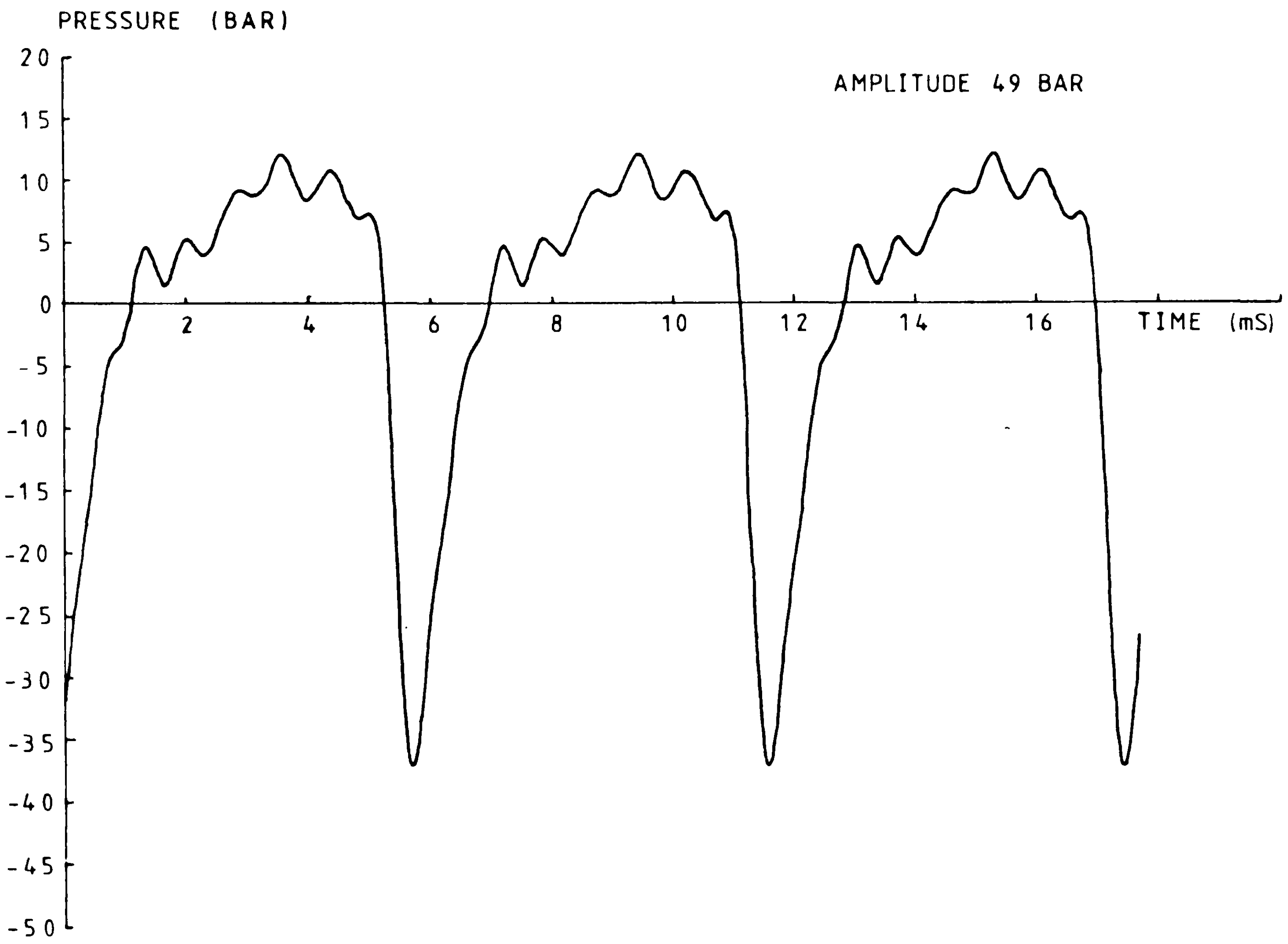

FIGURE 7·6 PRESSURE RIPPLE FOR 3·994M PIPE-EXPANDED SCALE

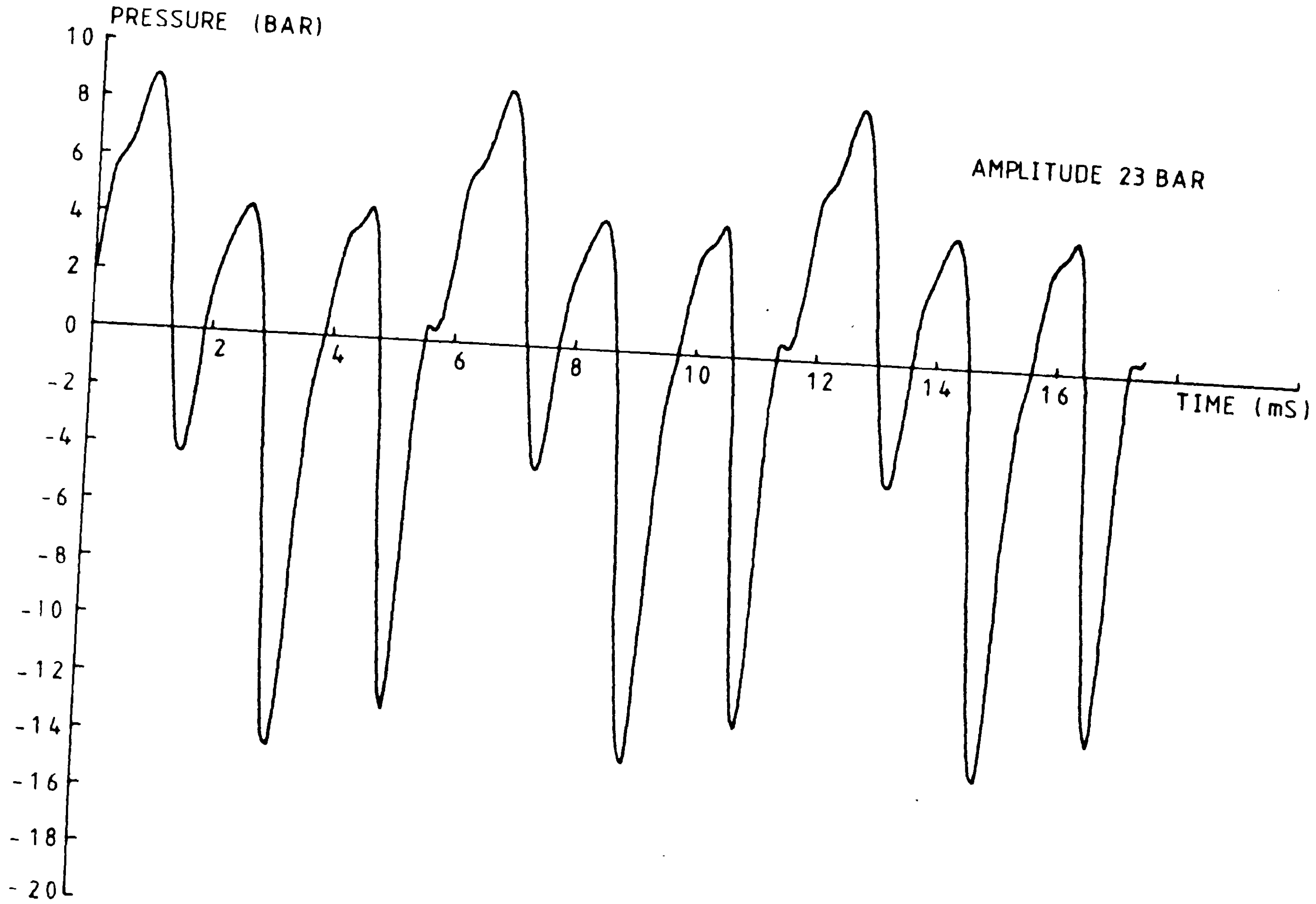

FIGURE 7.7 PRESSURE RIPPLE 2·661M PIPE - EXPANDED SCALE

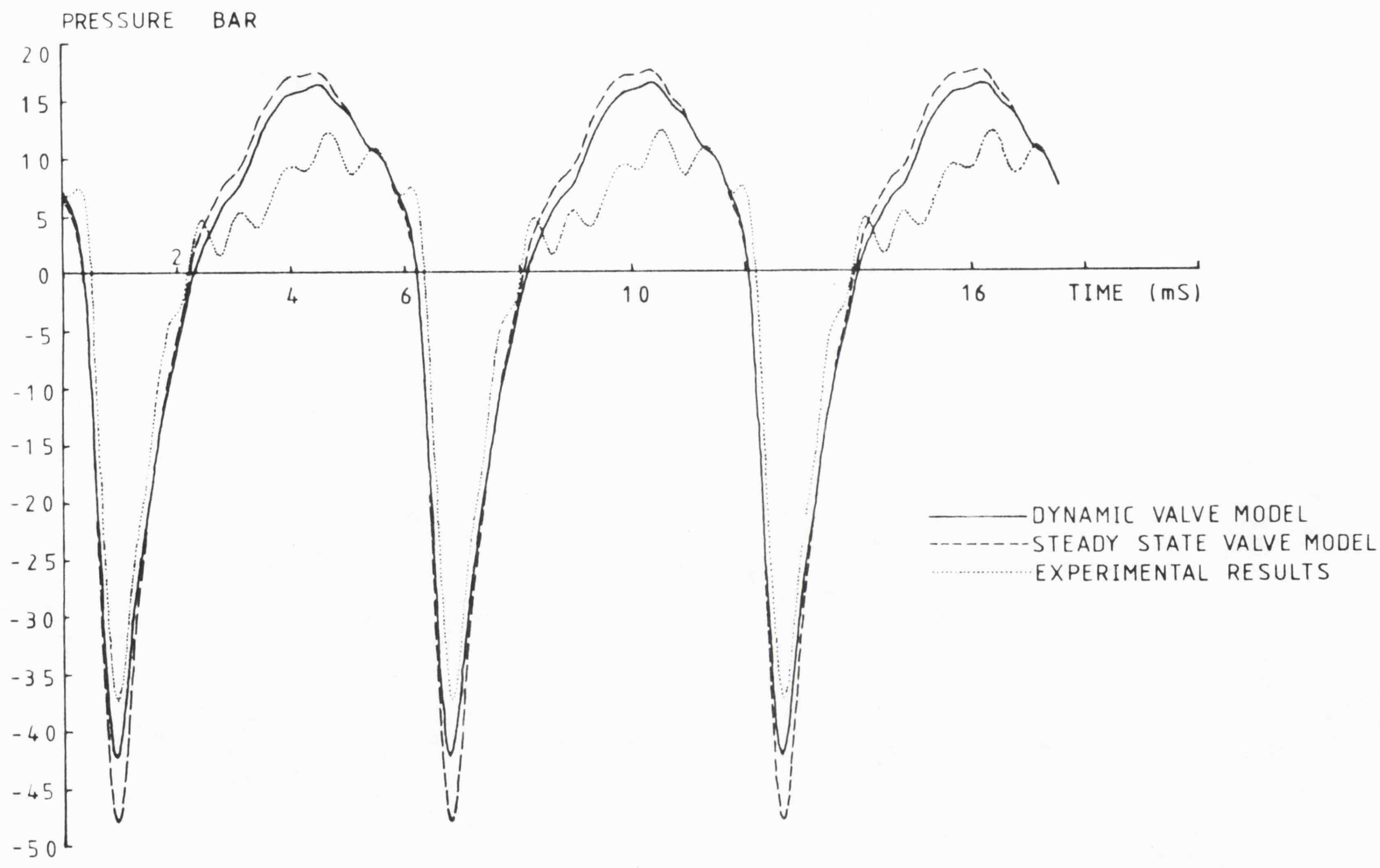

FIGURE 7.8 COMPARISON OF COMPUTED RESULTS - RESONANT SYSTEM (14cm/51cc EQ. PIPE)

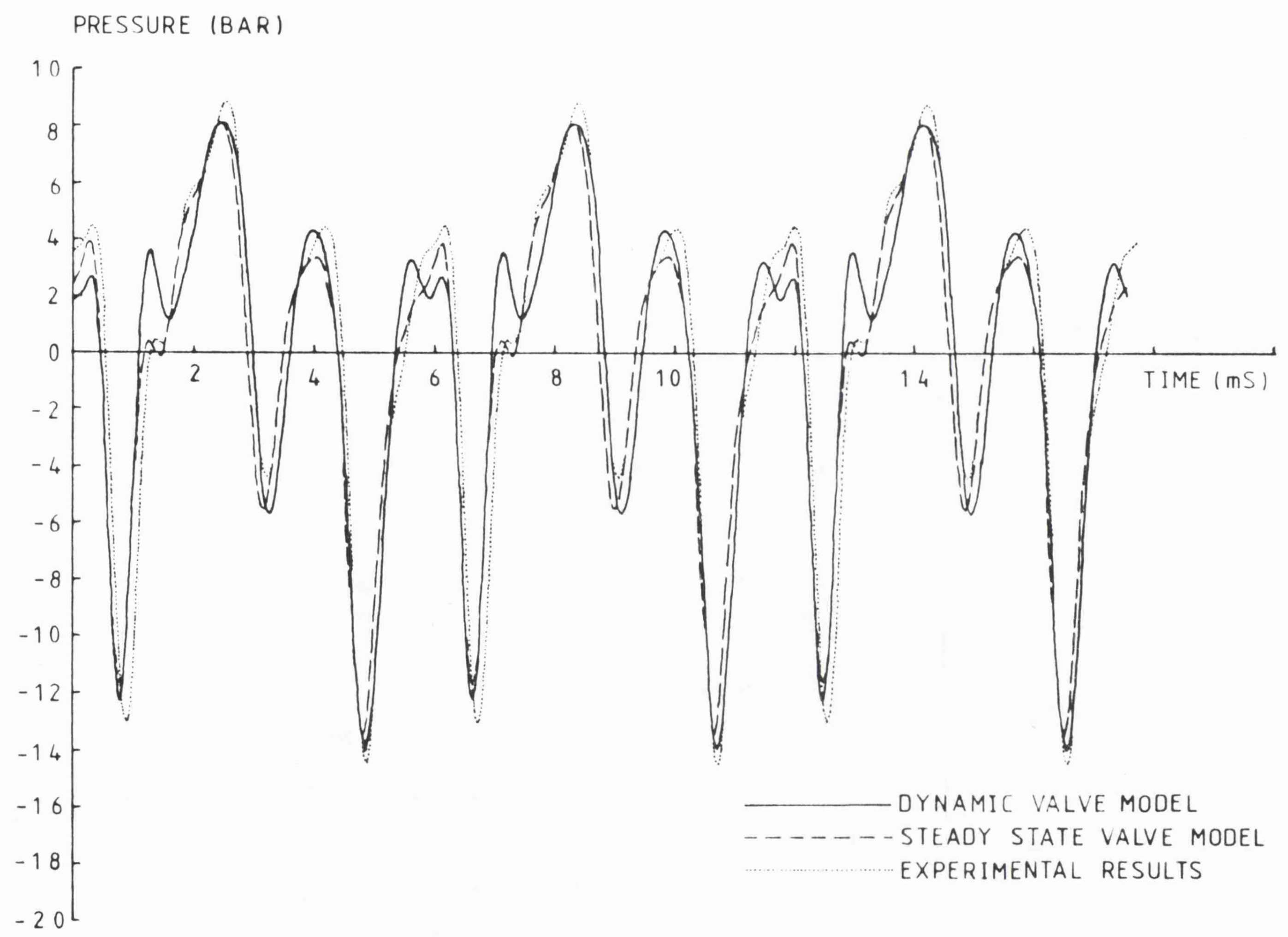

FIGURE 7·9 COMPARISON OF COMPUTED RESULTS FOR NON RESONANT SYSTEM
(14cm/51cc EQU. PIPE)

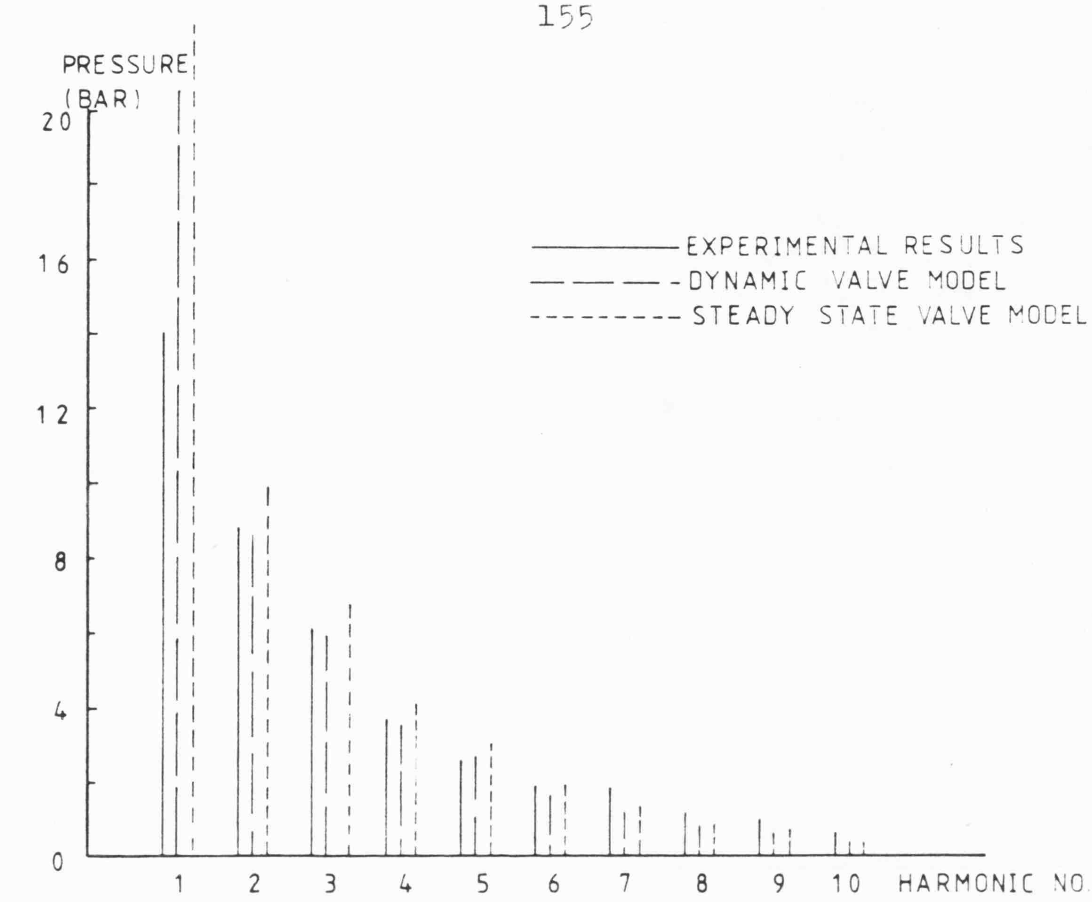

FIG. 7·10a AMPLITUDE SPECTRUM RESONANT SYSTEM
(14cm/51cc EQU. PIPE)

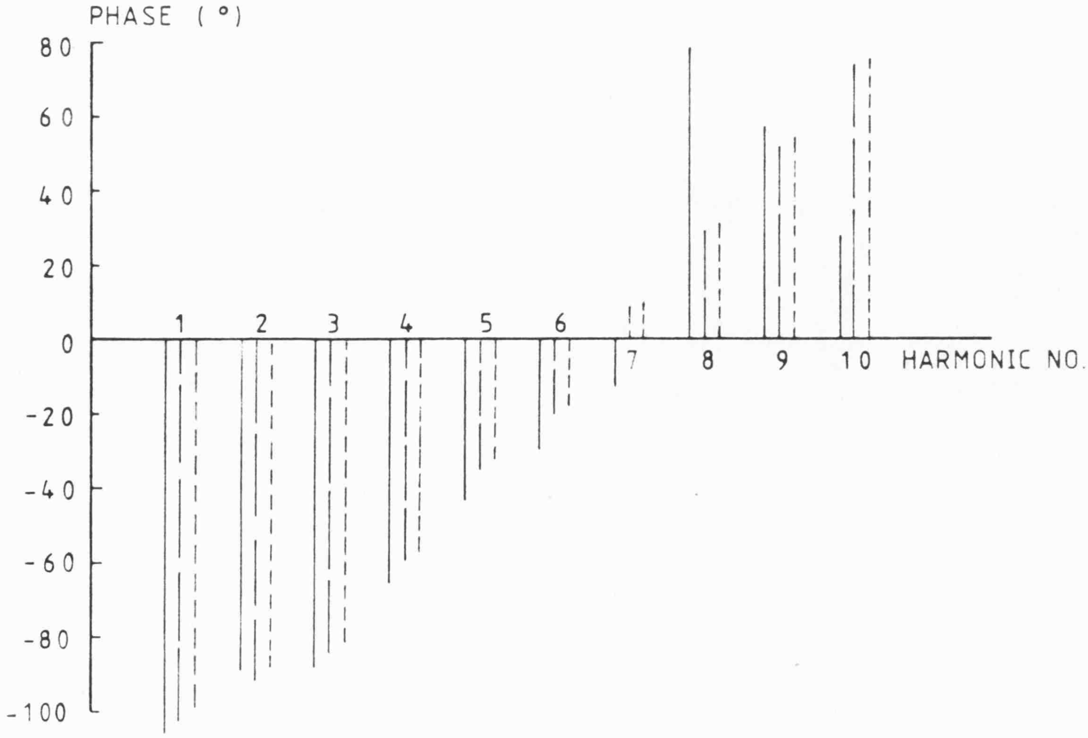

FIG. 7·10b PHASE SPECTRUM

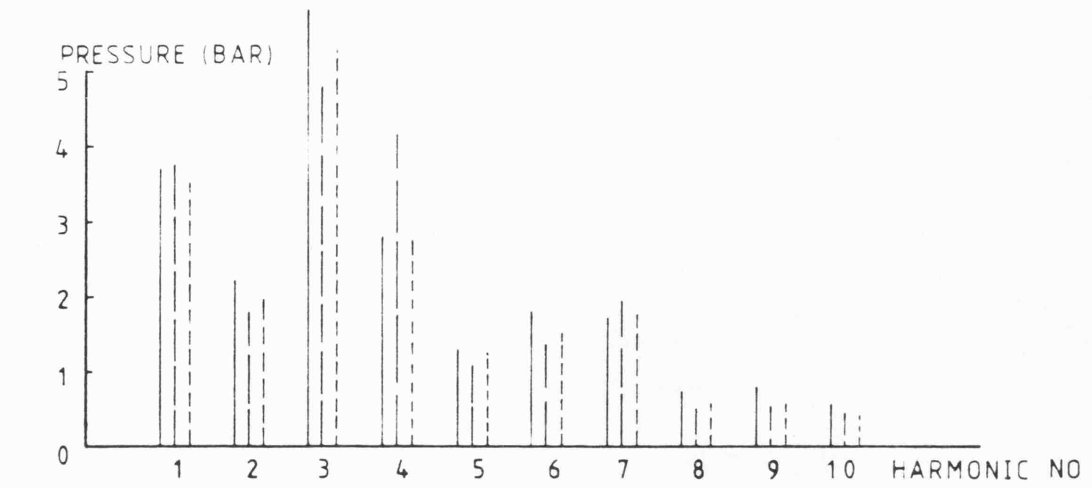

FIG. 7·11a AMPLITUDE SPECTRUM NON RESONANT SYSTEM (14cm/51cc EQU. PIPE)

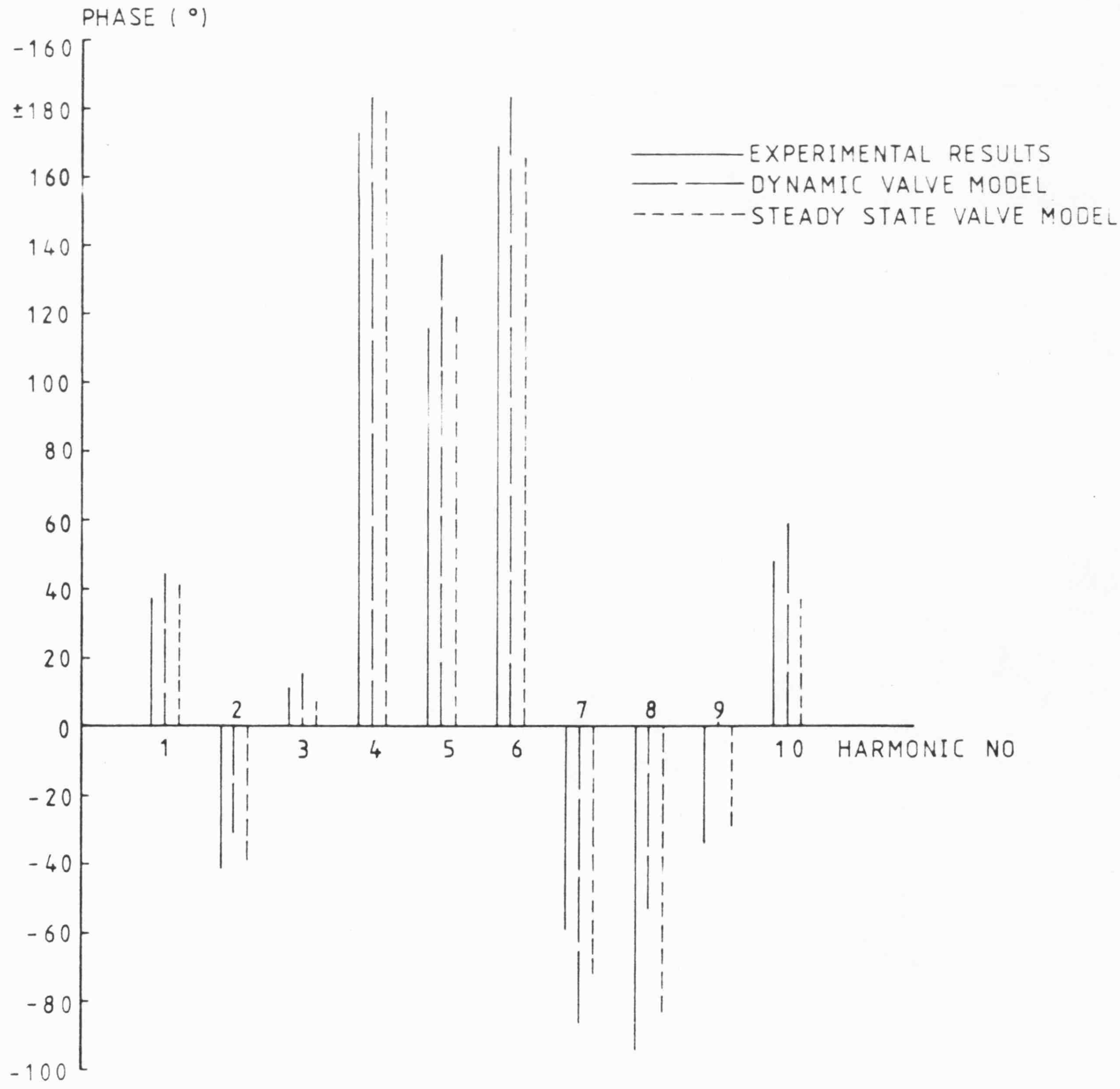

FIG. 7·11b PHASE SPECTRUM

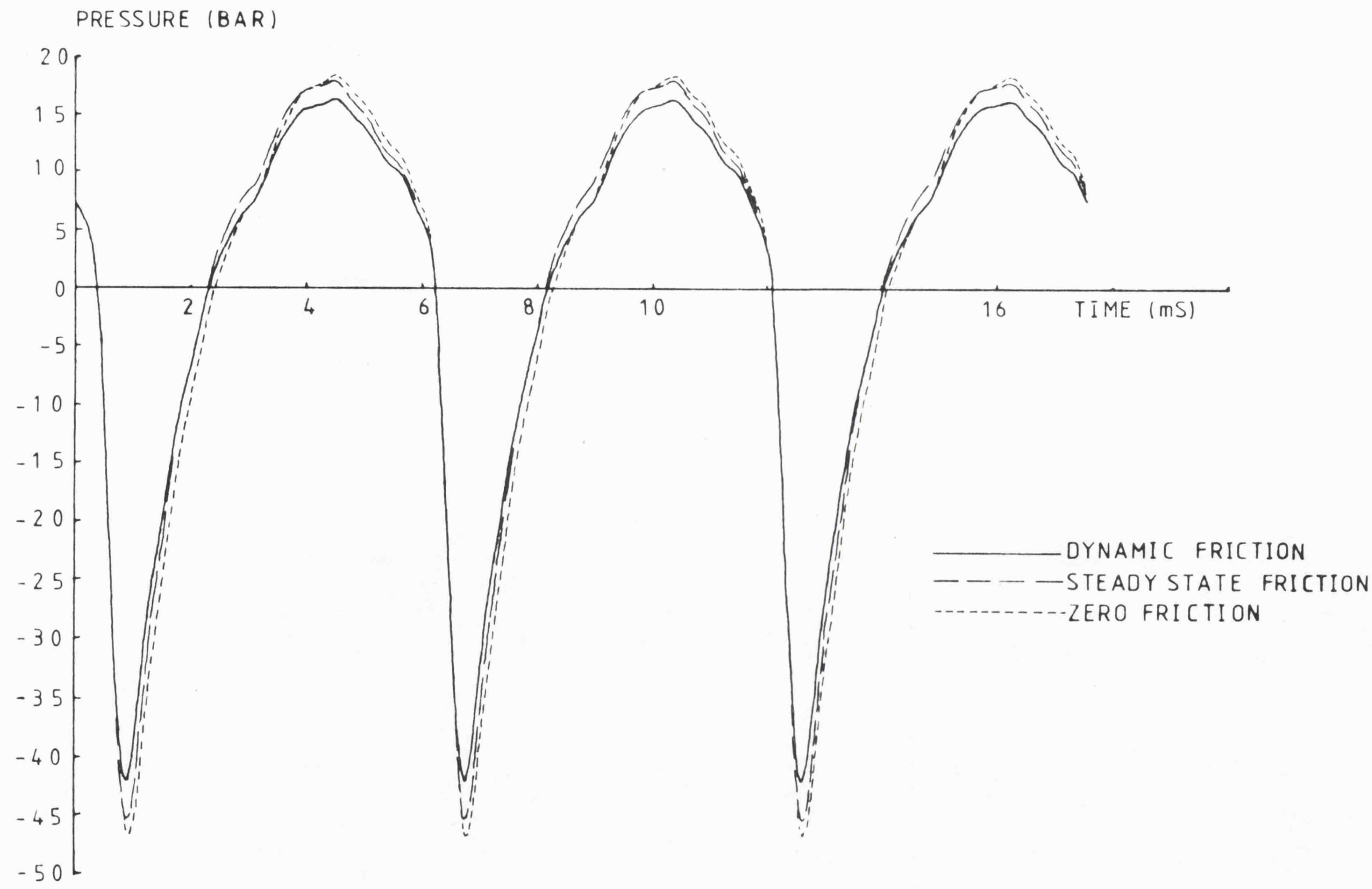

FIGURE 7·12 COMPARISON OF FRICTION MODELS–RESONANT SYSTEM
(DYNAMIC VALVE, 14cm/51cc EQU. PIPE)

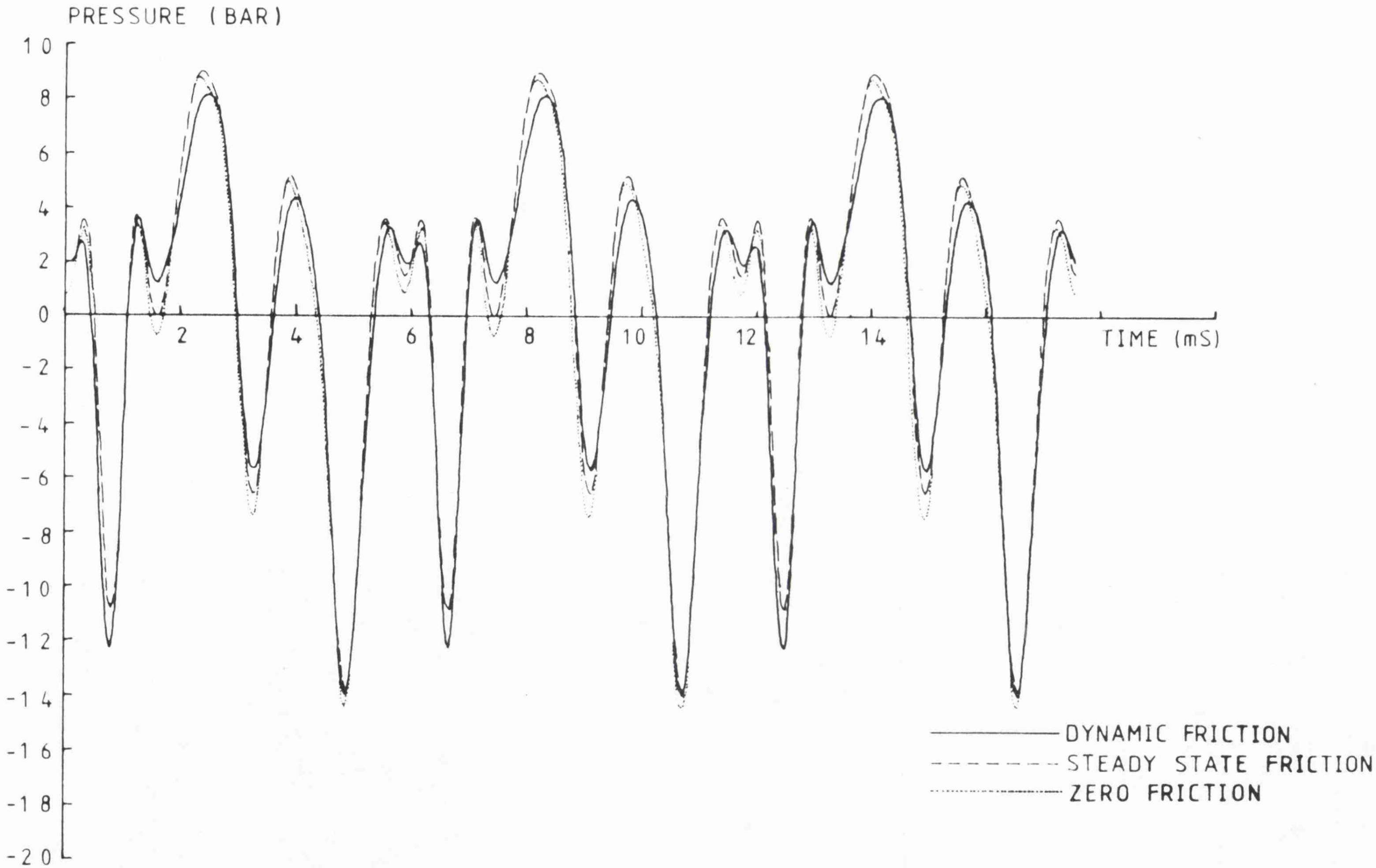

FIGURE 7.13 COMPARISON OF FRICTION MODELS - NON RESONANT SYSTEM (DYNAMIC VALVE MODEL, 14cm/51cc EQU. PIPE)

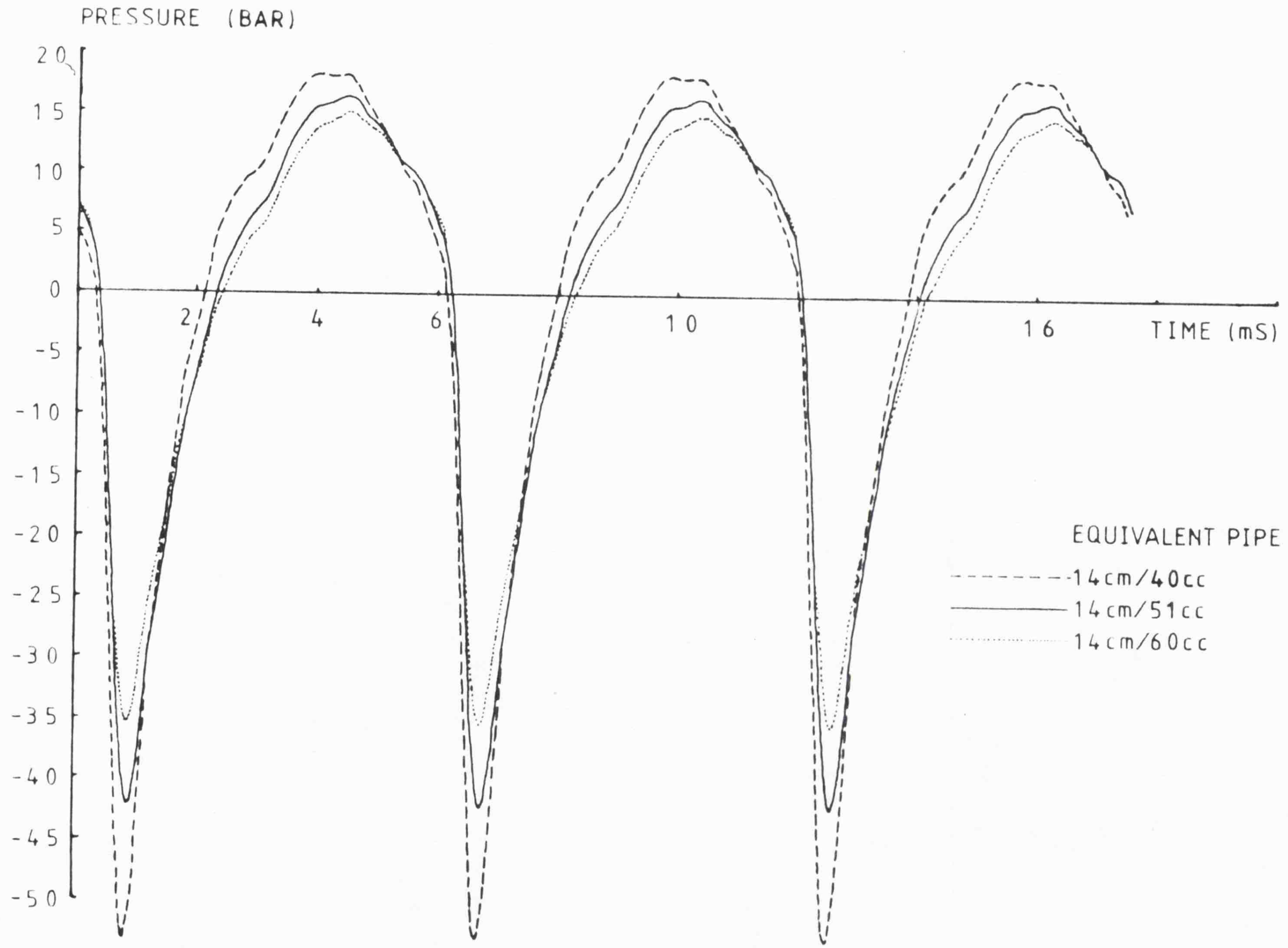

FIGURE 7.14 VOLUME EFFECTS (DYNAMIC VALVE MODEL, RESONANT SYSTEM)

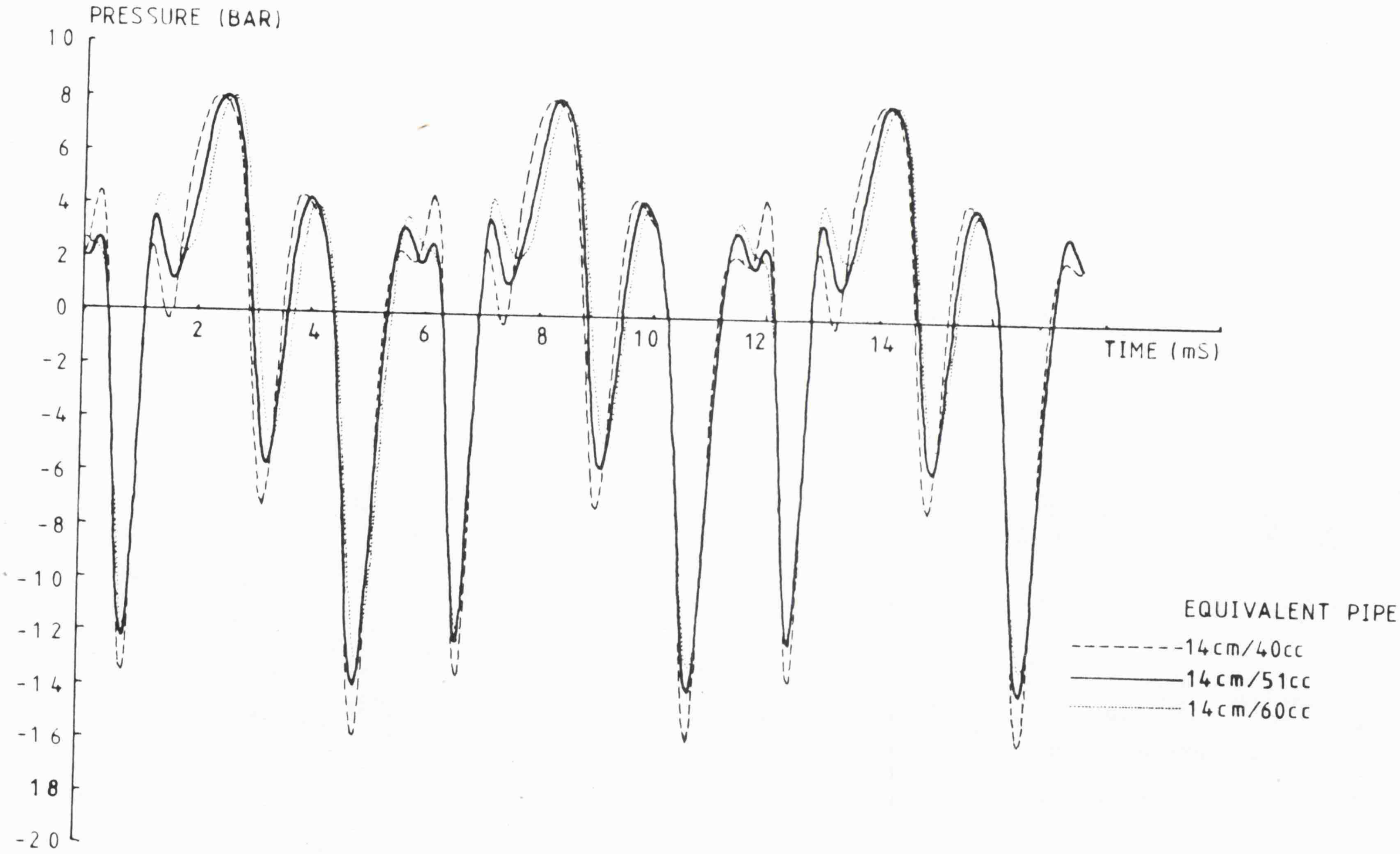

FIGURE 7.15 VOLUME EFFECTS (DYNAMIC VALVE MODEL, NON RESONANT SYSTEM)

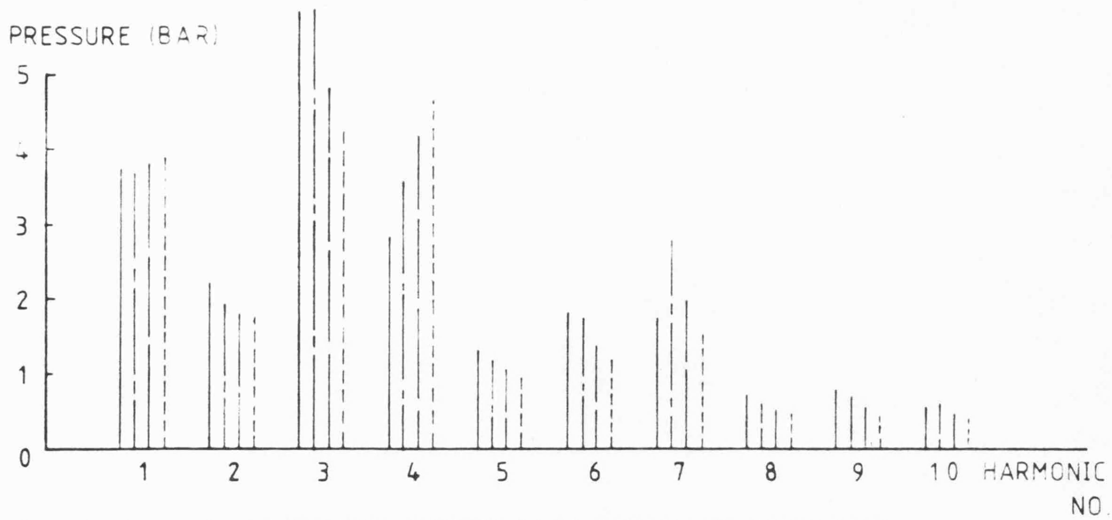

FIG. 7·16a AMPLITUDE SPECTRUM VOLUME EFFECTS NON RESONANT SYSTEM

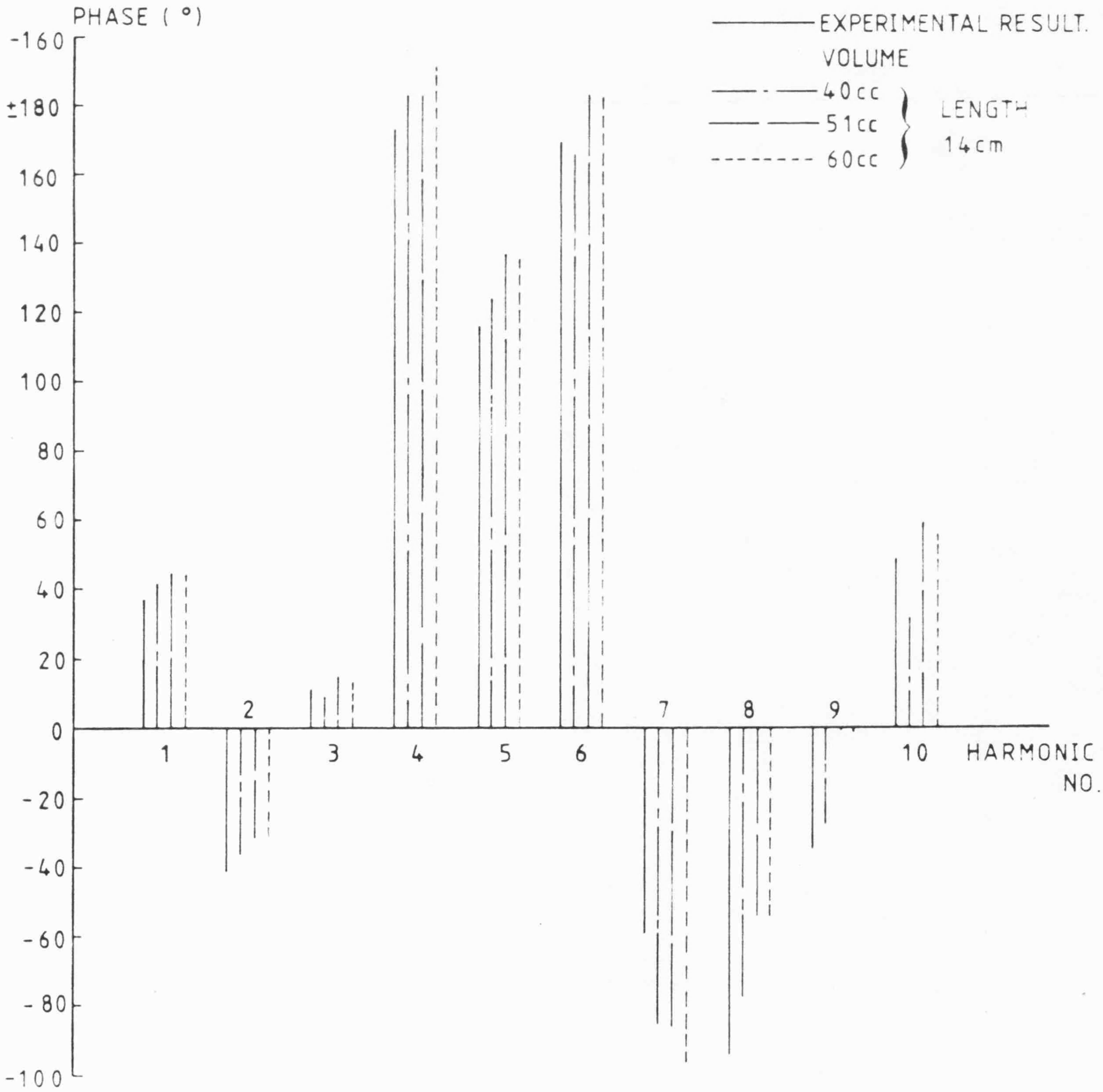

FIG. 7·16b PHASE SPECTRUM

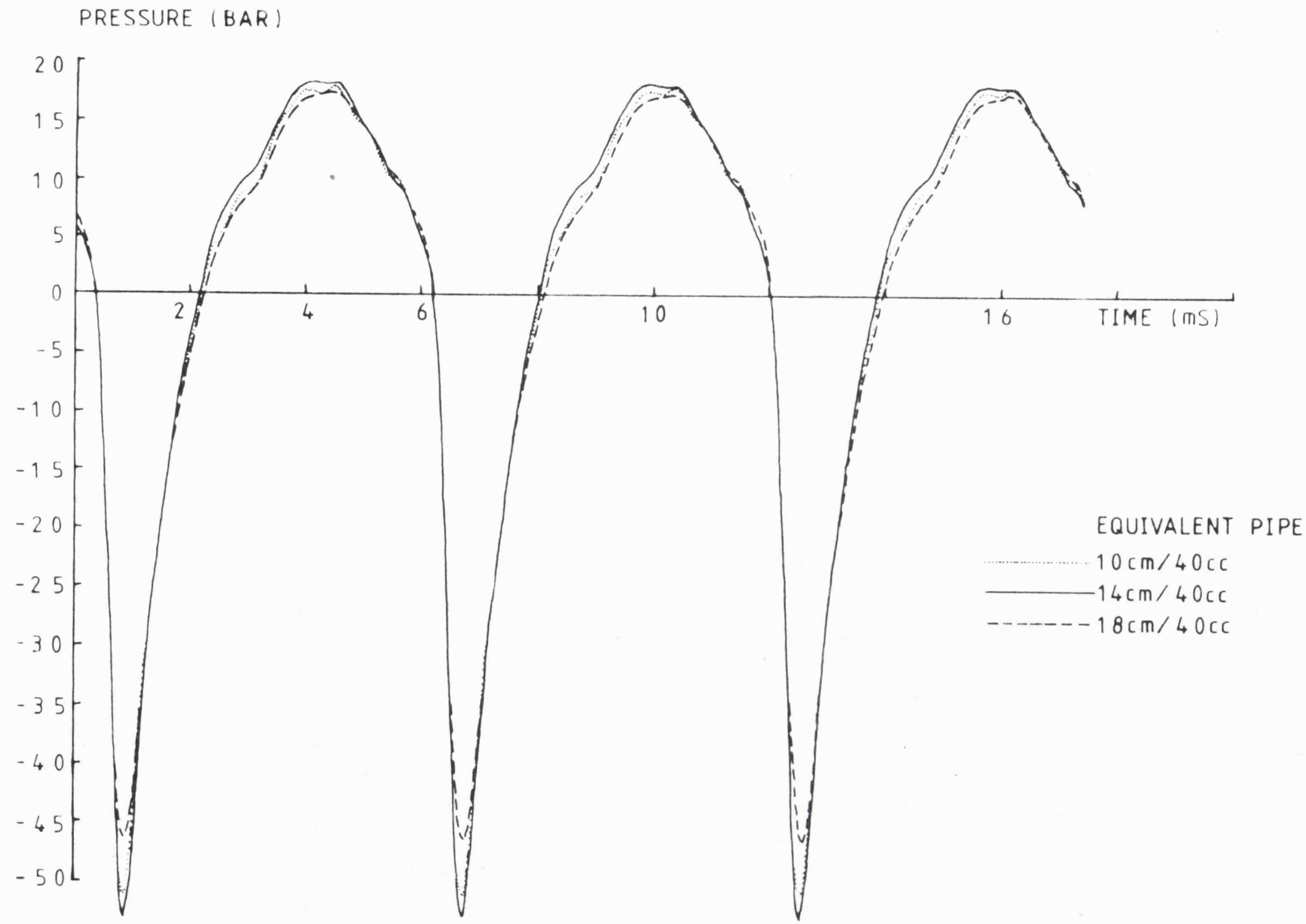

FIGURE 7.17 LENGTH EFFECTS (DYNAMIC VALVE MODEL, RESONANT SYSTEM)

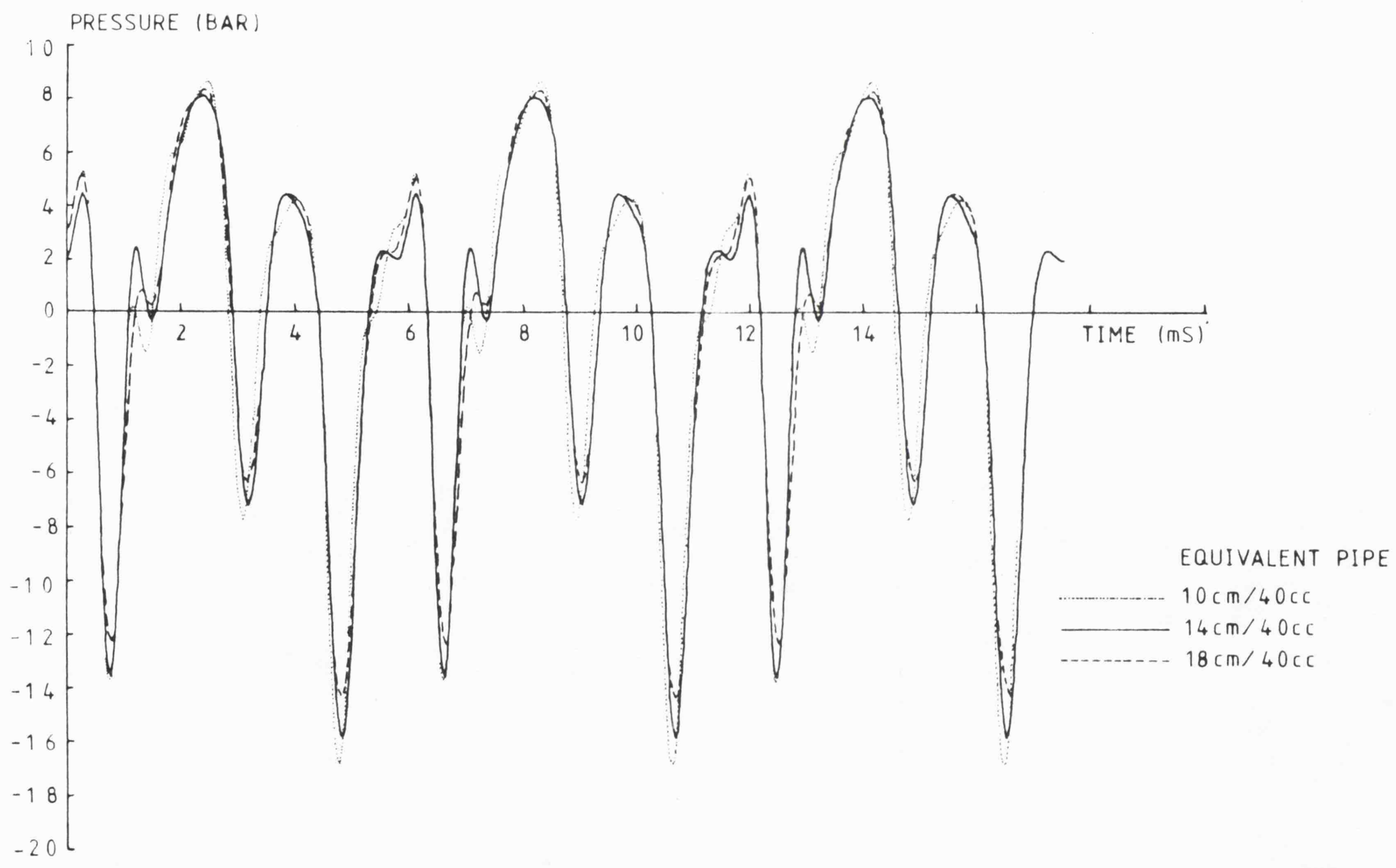

FIGURE 7.18 LENGTH EFFECTS (DYNAMIC VALVE MODEL, NON RESONANT SYSTEM)

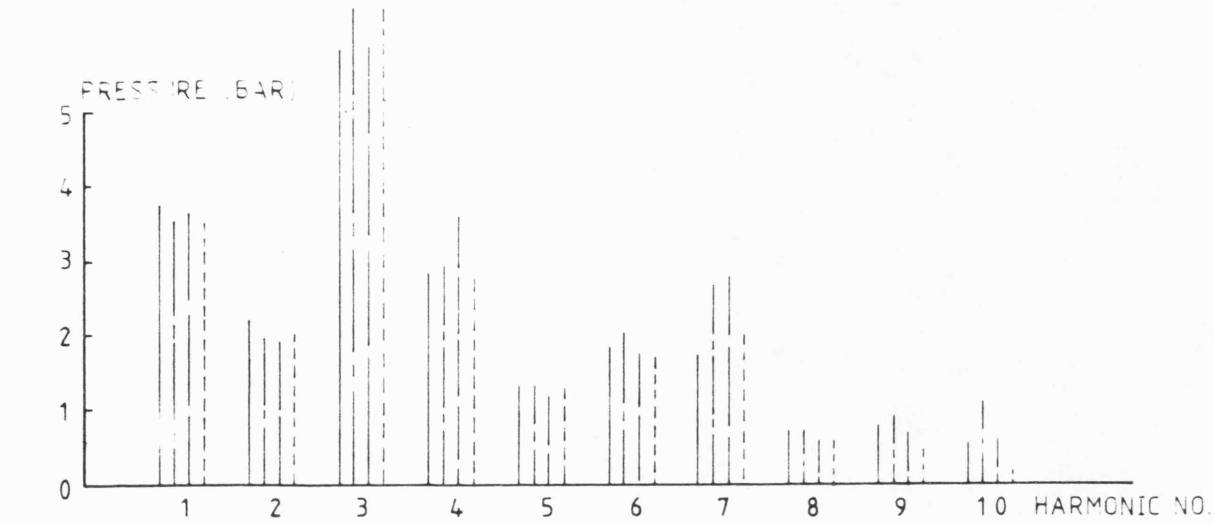

FIG. 7·19a AMPLITUDE SPECTRUM VOLUME EFFECTS
NON RESONANT SYSTEM

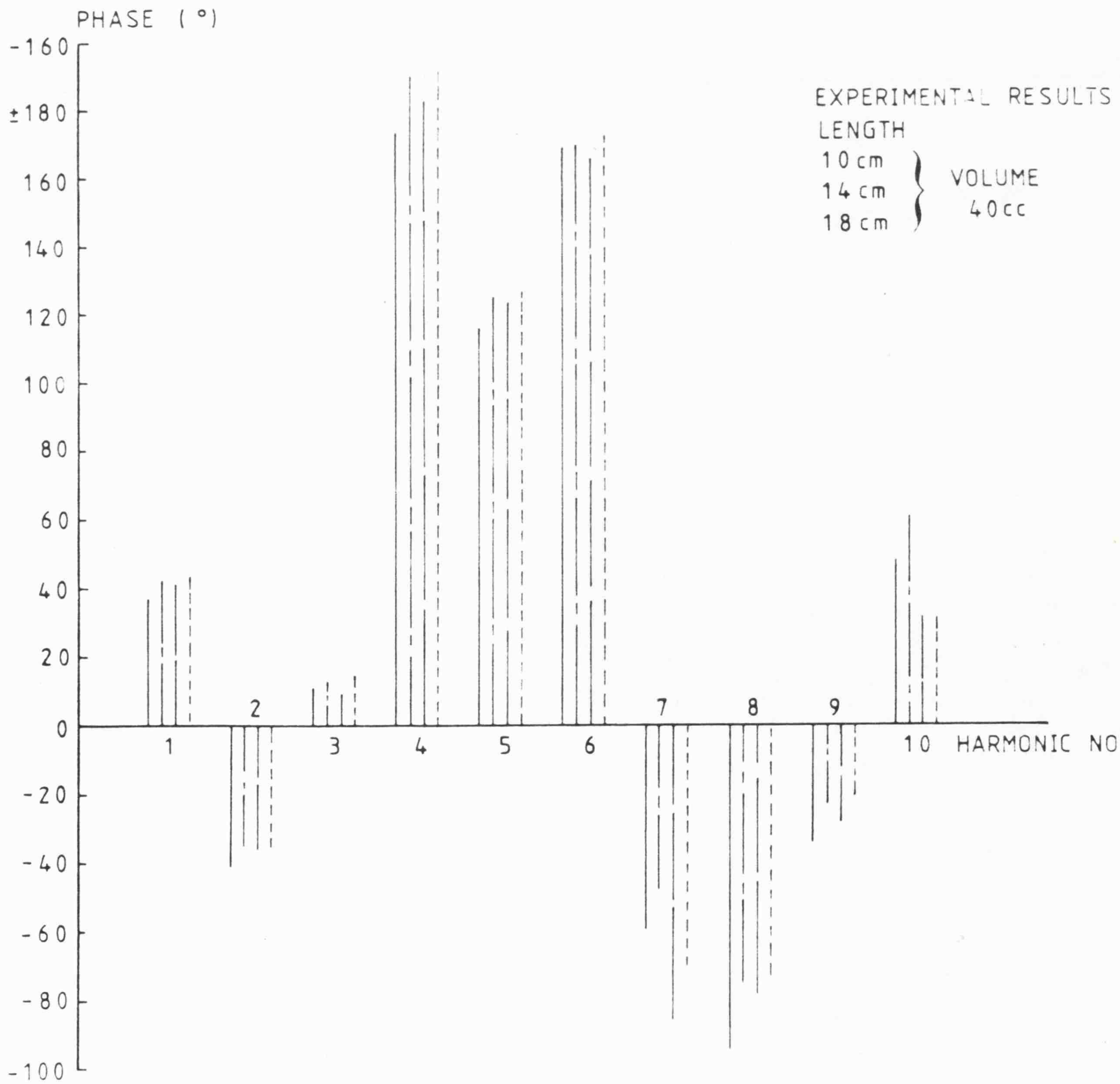

FIG. 7·19b PHASE SPECTRUM

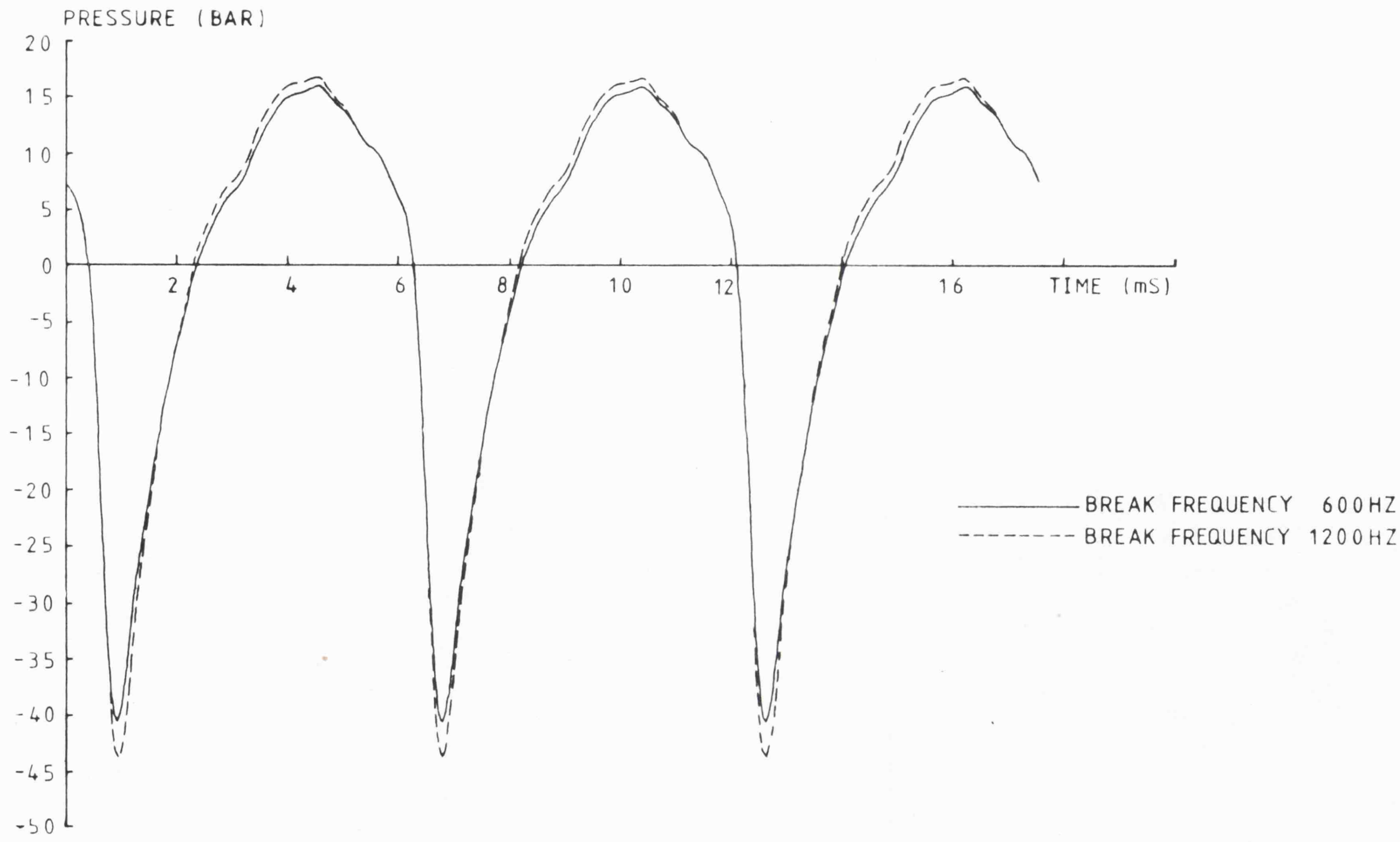

FIGURE 7.20 EFFECTS OF THE DYNAMIC VALVE TIME CONSTANT-RESONANT SYSTEM

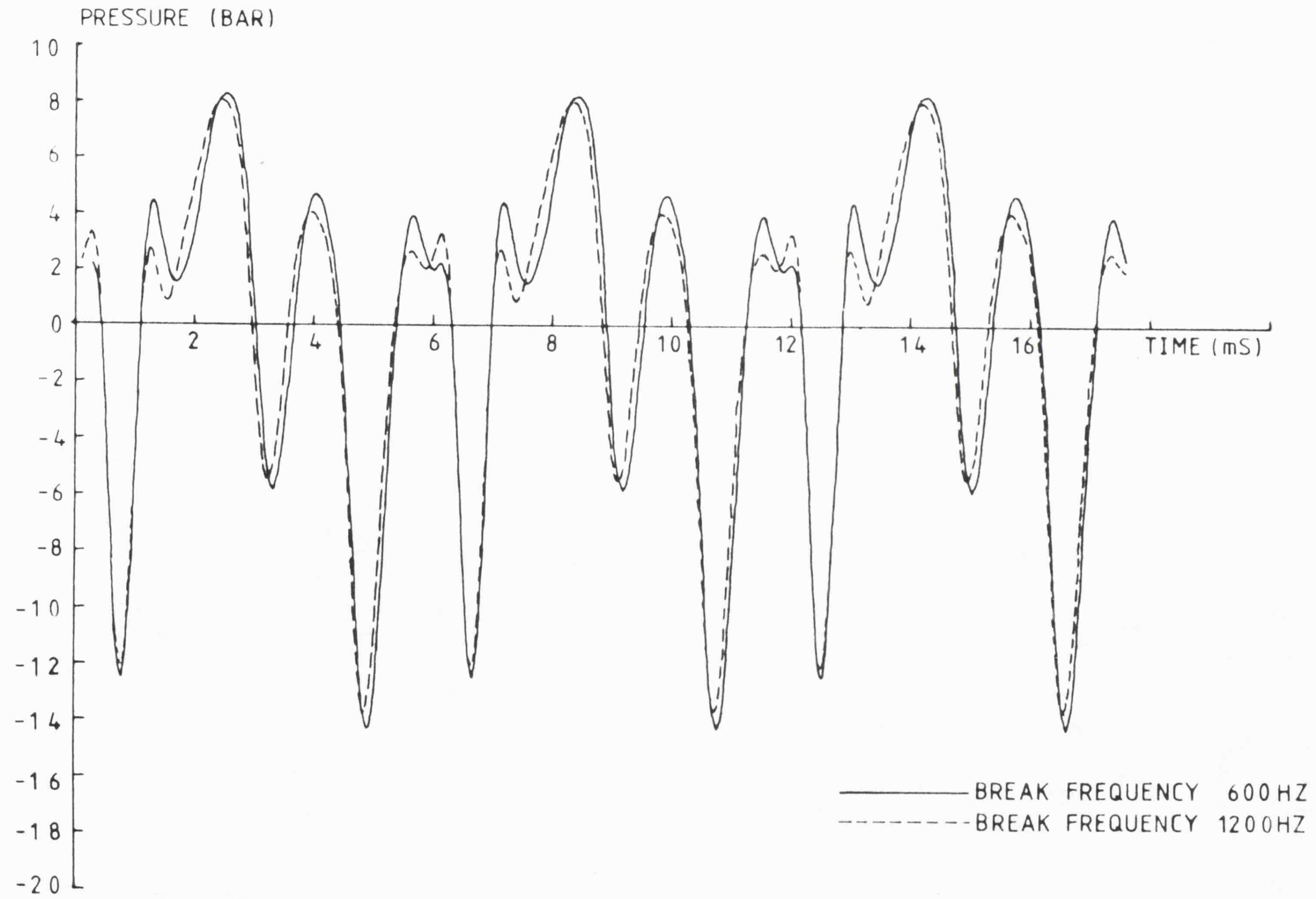

FIGURE 7.21 EFFECTS OF THE DYNAMIC VALVE TIME CONSTANT - NON RESONANT SYSTEM

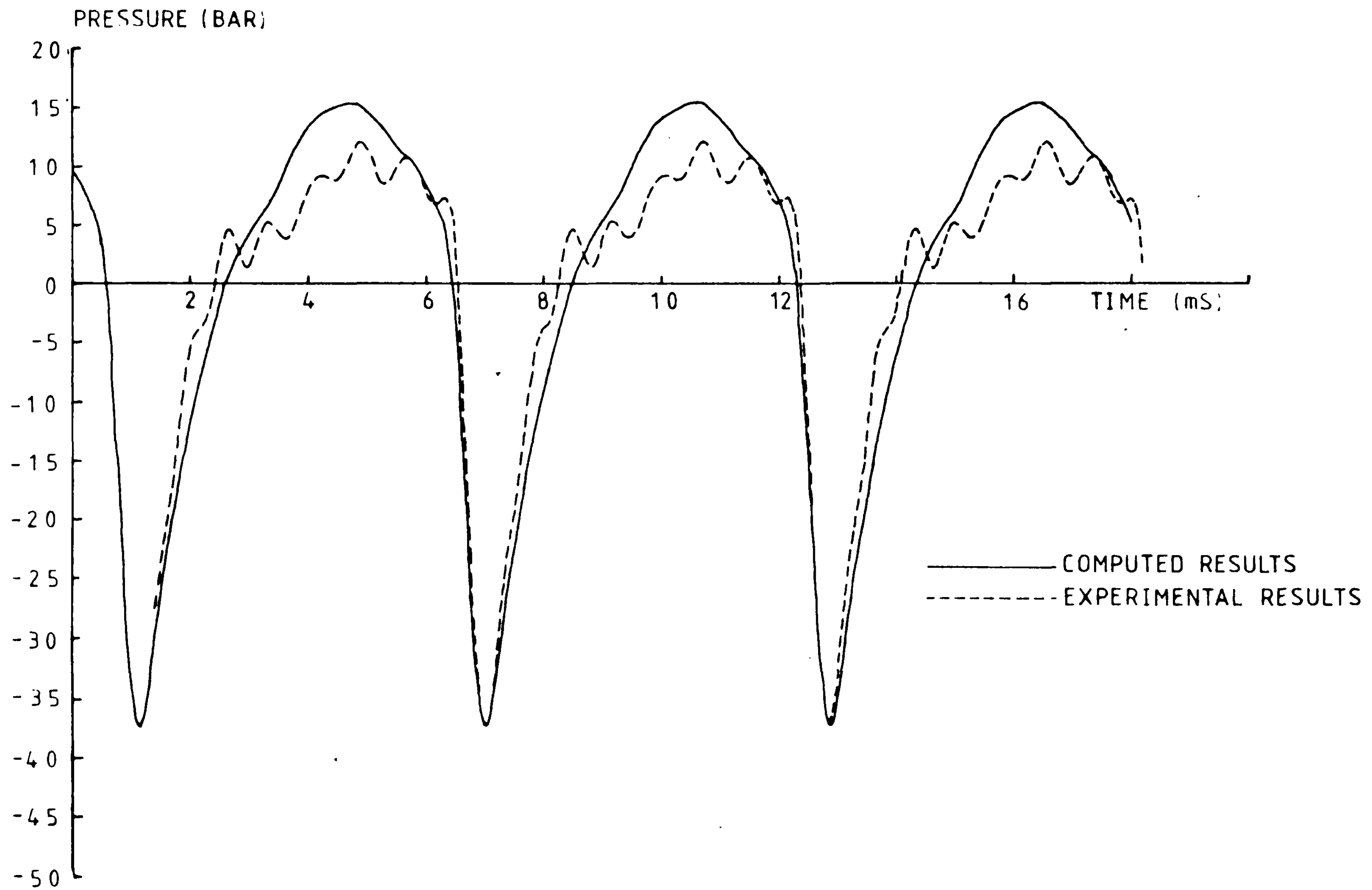

FIGURE 7·22 RESONANT SYSTEM WITH 18cm/51cc EQU. PIPE & DYNAMIC VALVE

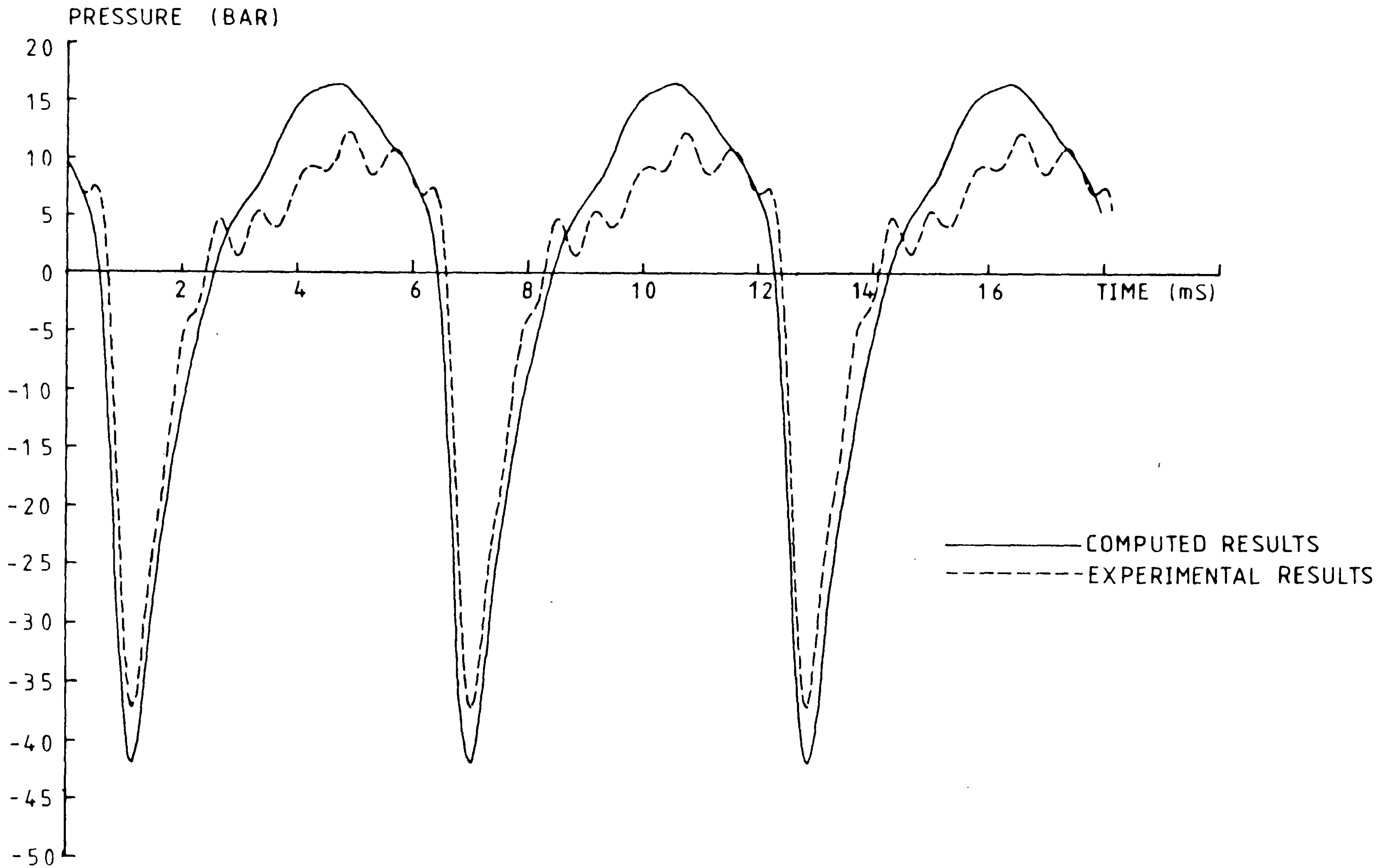

FIGURE 7·23 RESONANT SYSTEM WITH 18cm/51cc EQU. PIPE & STEADY STATE VALVE

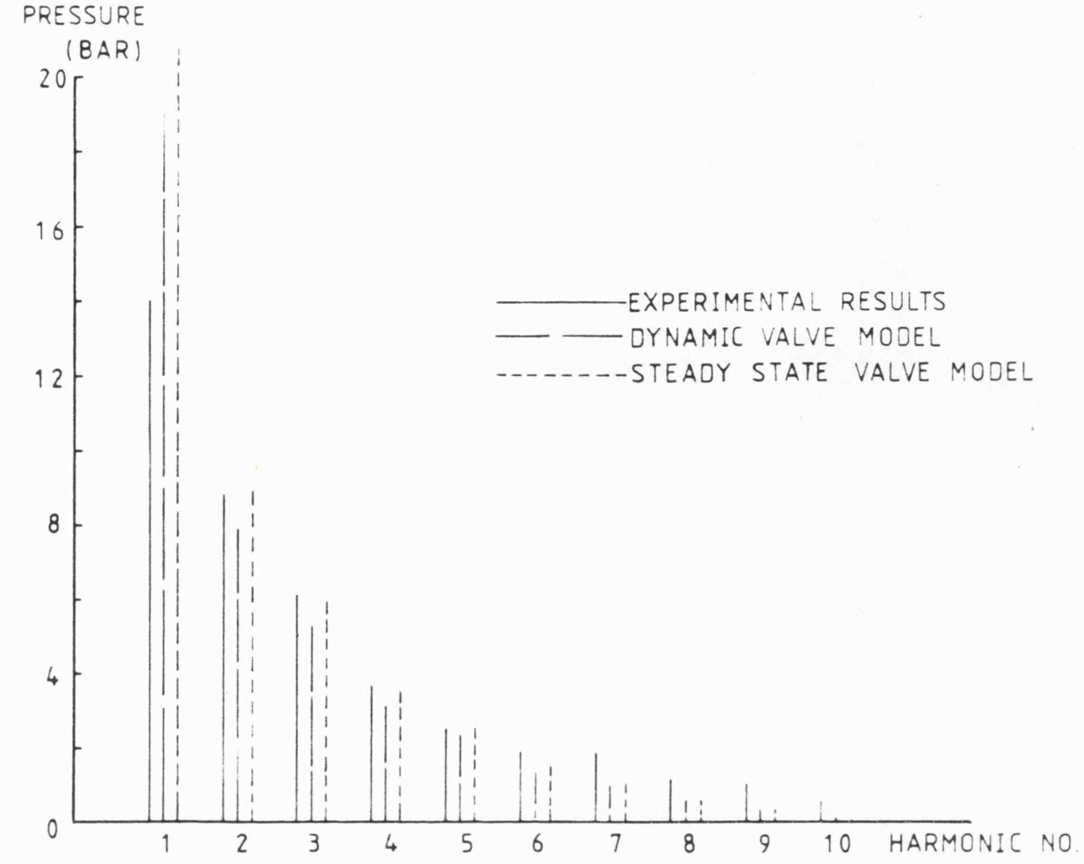

FIG. 7·24a AMPLITUDE SPECTRUM RESONANT SYSTEM (18cm/51cc EQU. PIPE)

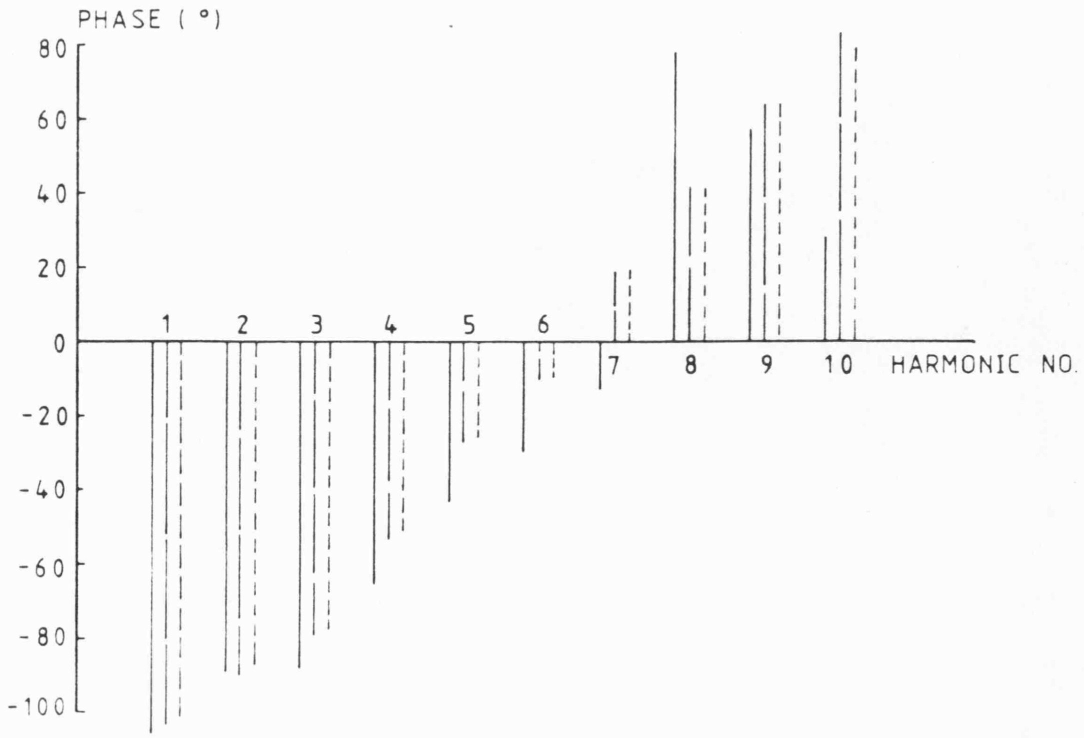

FIG. 7·24b PHASE SPECTRUM

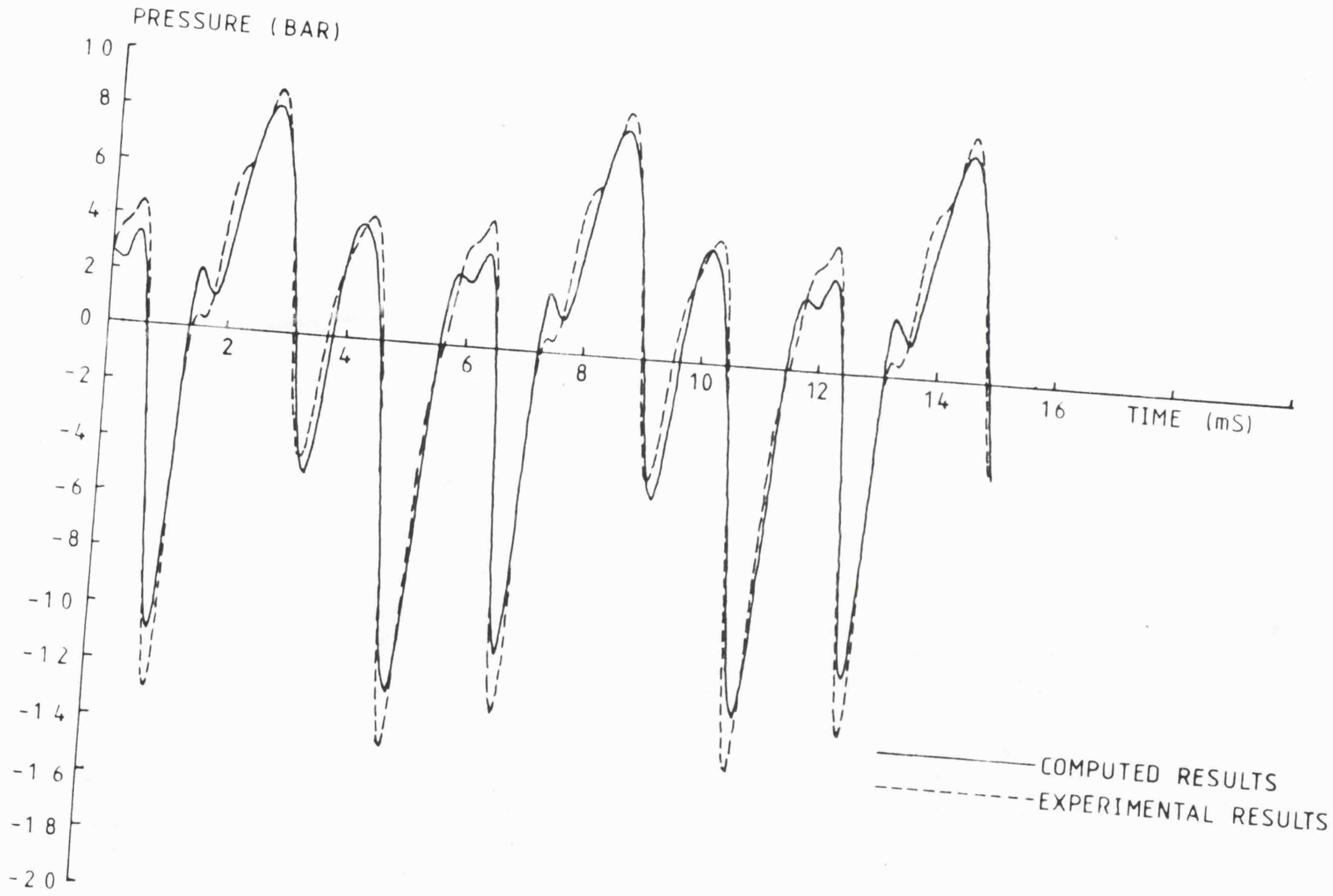

FIGURE 7.25 NON RESONANT SYSTEM WITH 18cm/51cc EQU. PIPE & DYNAMIC VALVE

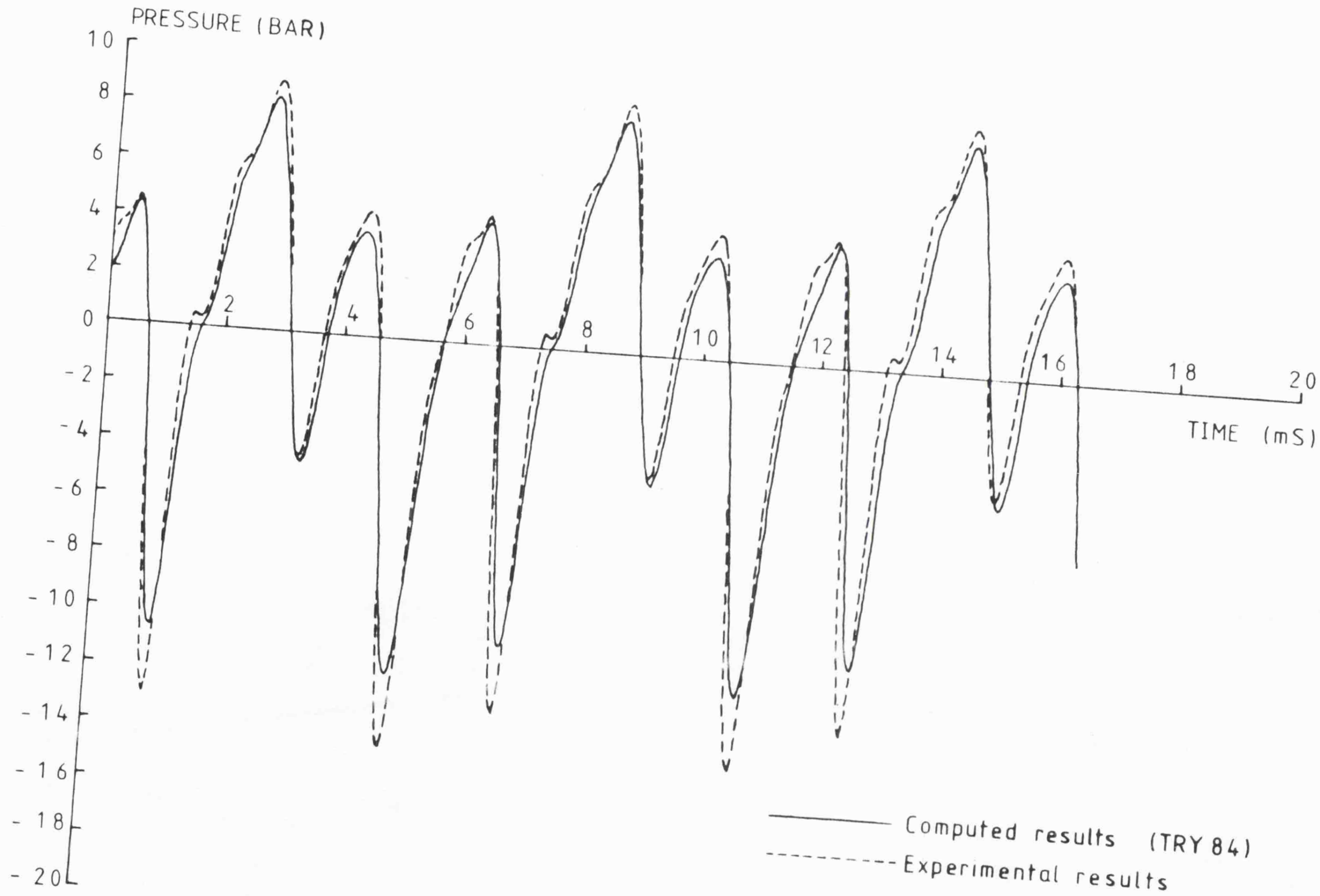

FIGURE 7·26 NON RESONANT SYSTEM SIMULATION
STEADY STATE VALVE MODEL, 18cm/51cc EQUIV. PIPE

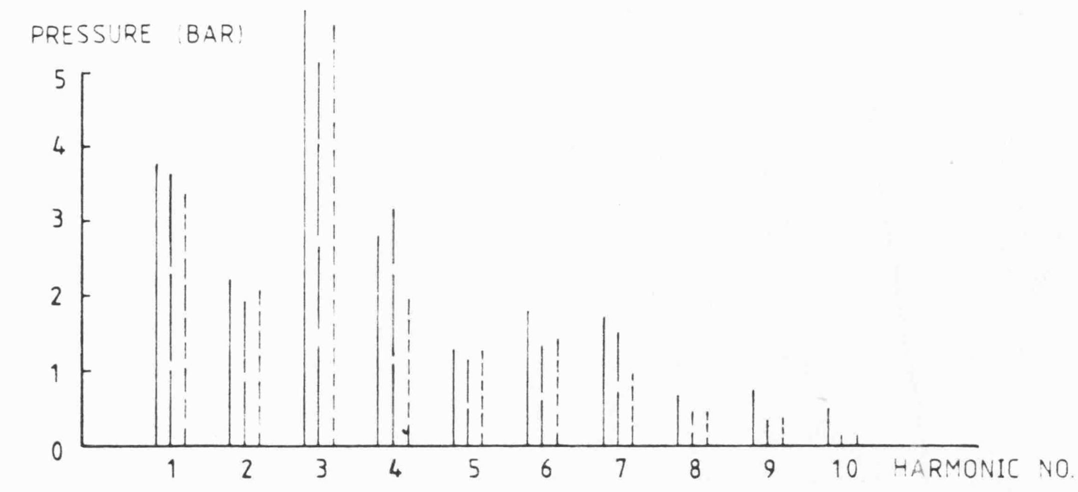

FIG. 7·27a AMPLITUDE SPECTRUM NON RESONANT SYSTEM (18cm/51cc EQU. PIPE)

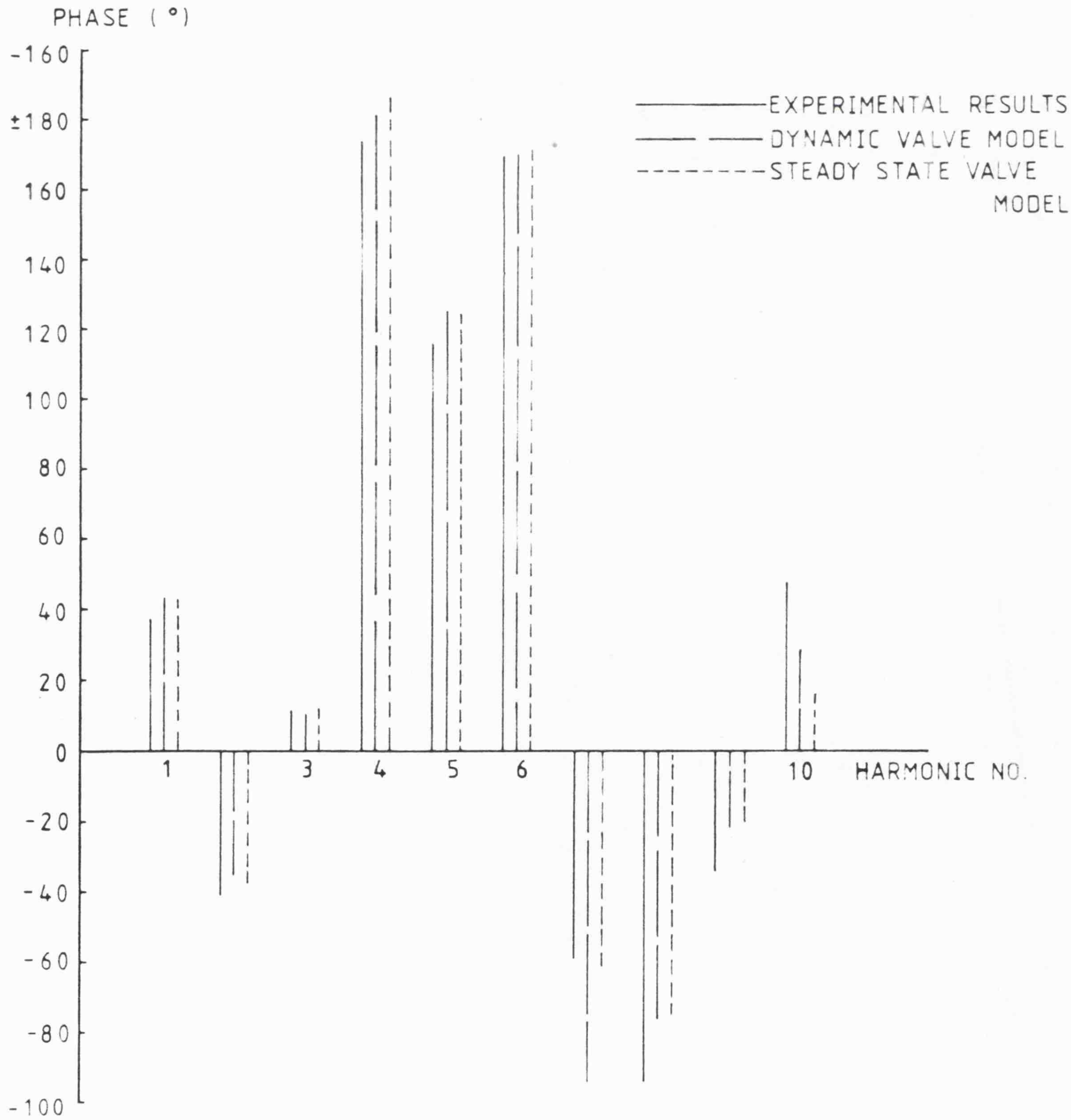

FIG. 7·27b PHASE SPECTRUM

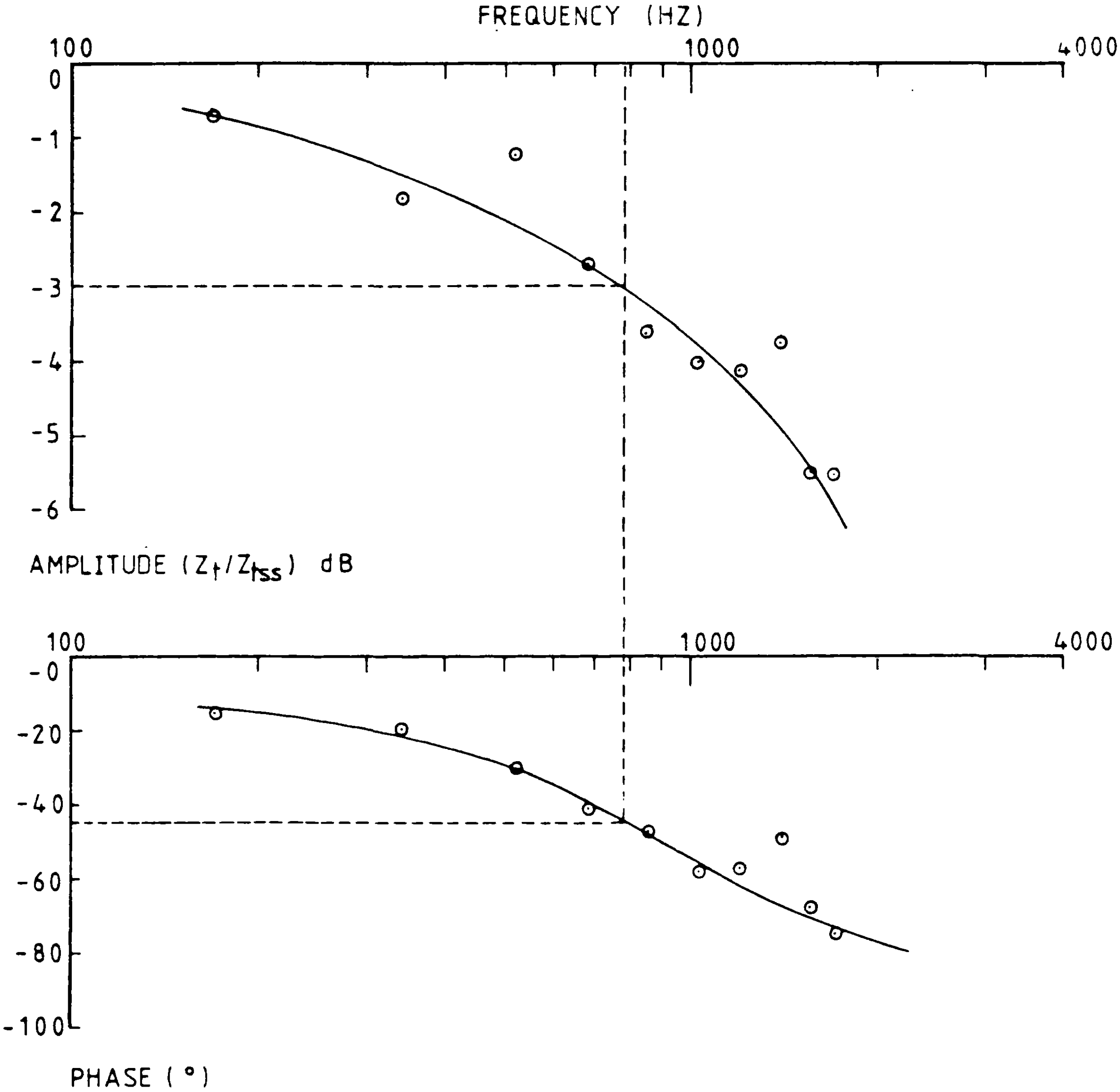

FIGURE 7·28 BODE PLOT OF RESTRICTOR VALVE IMPEDANCE AT 200 BAR

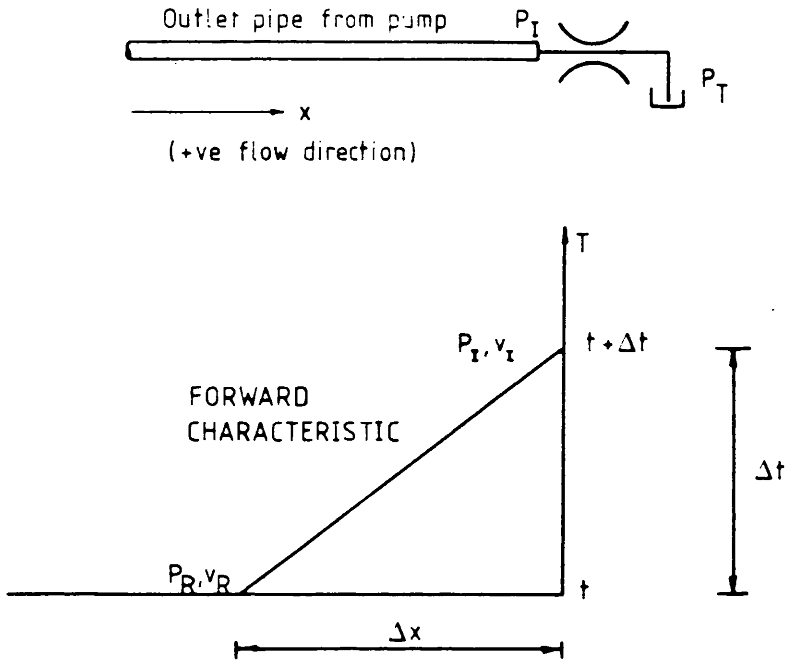

FIGURE 7.29 RESTRICTOR VALVE ON TIME-DISTANCE PLANE

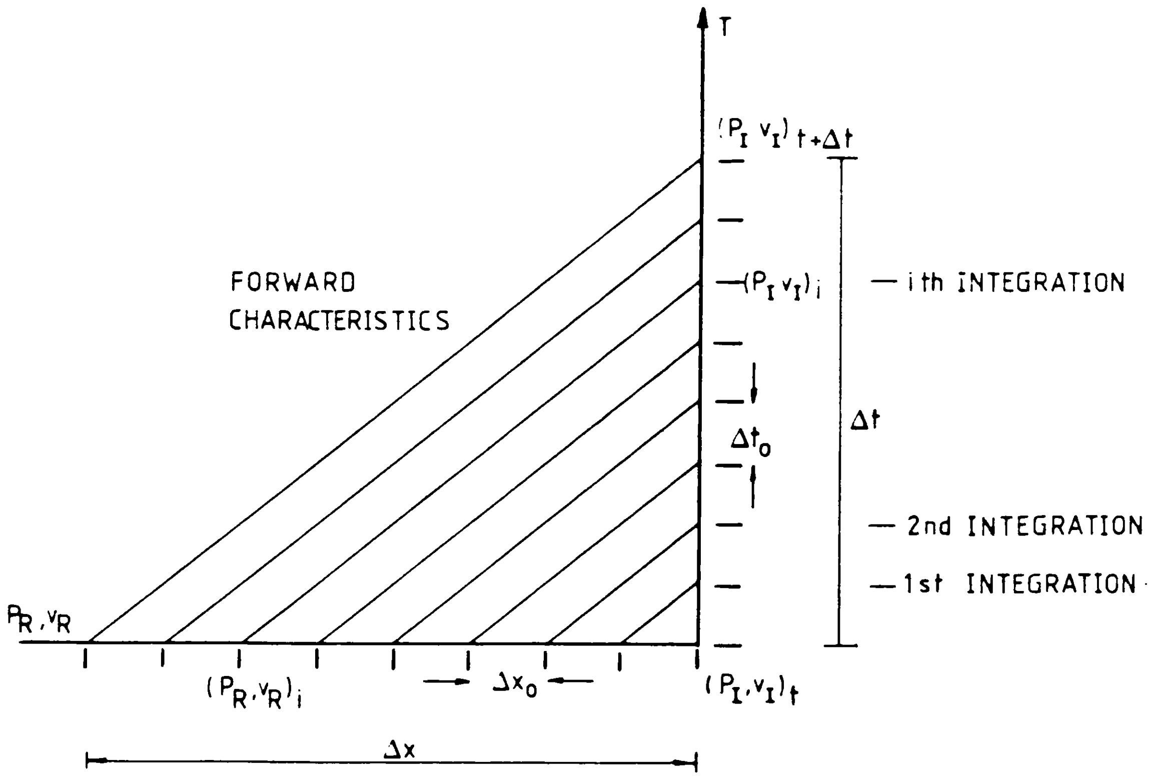

FIGURE 7.30 MULTIPLE STEP INTEGRATION SCHEME

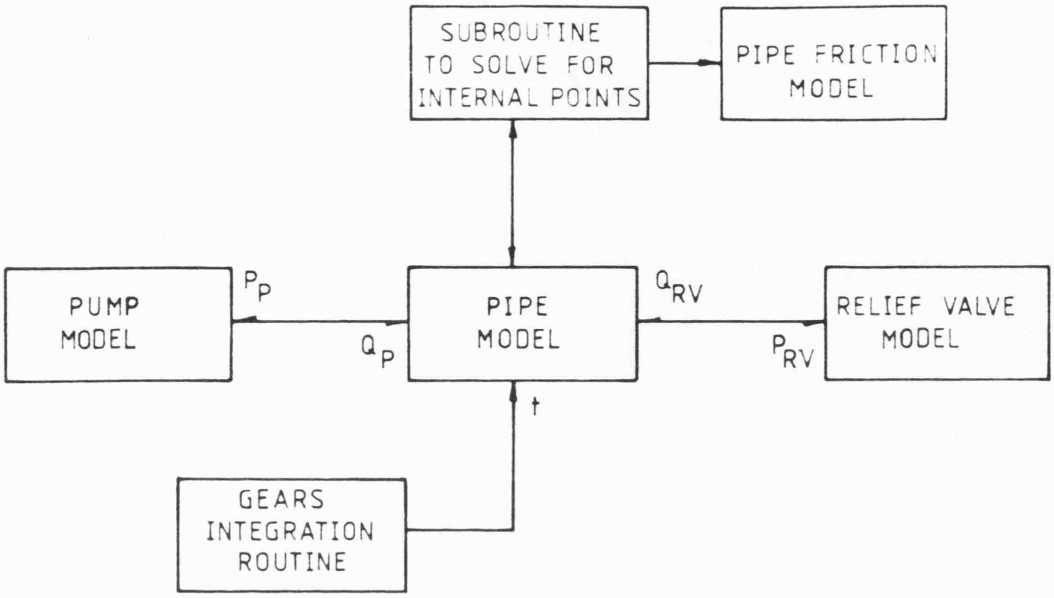

FIGURE 9.1 SCHEME FOR INTERFACING THE METHOD OF CHARACTERISTICS WITH HGSP.

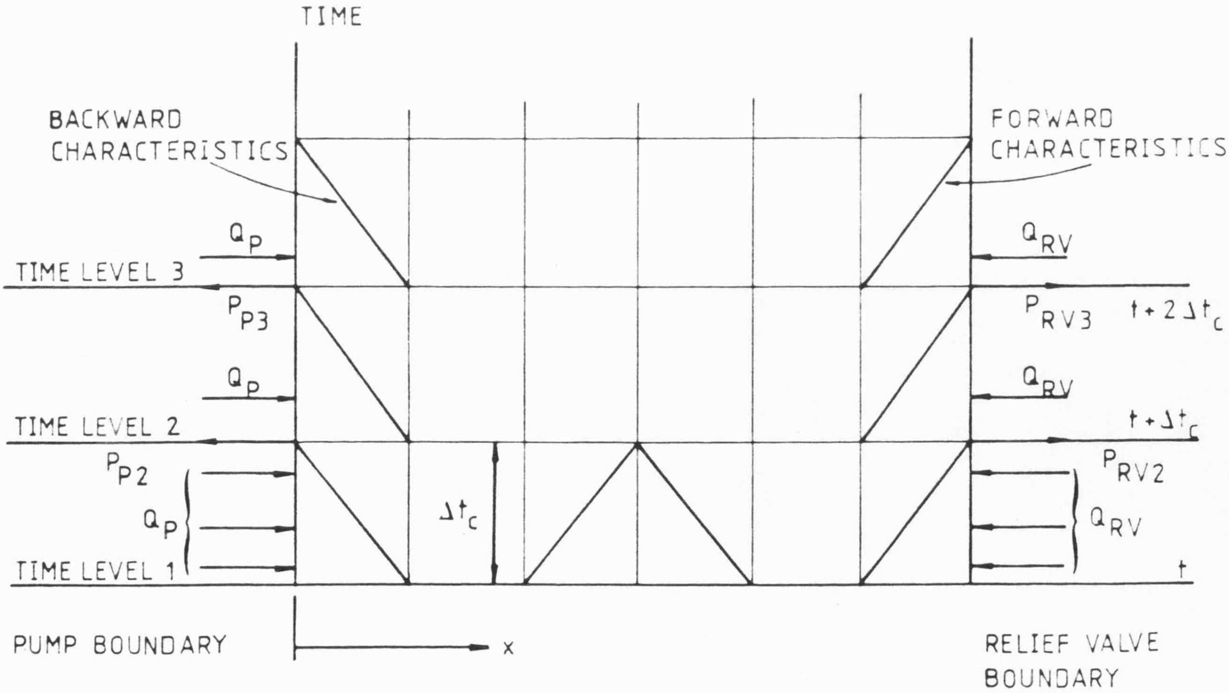

FIGURE 9.2 SYSTEM SHOWN ON THE TIME-DISTANCE PLANE

Qs DATA REYROLLE A200 (7 PISTON)			
MEAN PRESSURE	200 bar		
MEAN FLOW	0.7 l/sec		
PUMP SPEED	153.5 rad/sec (1466 rpm)		
HARMONIC NO	FREQUENCY (Hz)	AMPLITUDE (l/sec)	PHASE (degrees)
1	171	0.0758	-24
2	342	0.0550	+33
3	513	0.0538	+78
4	684	0.0439	+136
5	855	0.0424	-167
6	1026	0.0337	-120
7	1197	0.0314	-60
8	1368	0.0270	-8
9	1539	0.0253	+46
10	1710	0.0224	+99

TABLE 7.1 EXPERIMENTALLY DETERMINED Qs DATA FOR THE REYROLLE A200 PUMP

HARMONIC NO	FREQUENCY (Hz)	3.944 M PIPE (RESONANT)		2.661 M PIPE (NON RESONANT)	
		AMPLITUDE (BAR)	PHASE (DEGREES)	AMPLITUDE (BAR)	PHASE (DEGREES)
1	171	14.02	-106	3.73	37
2	342	8.81	-89	2.21	-41
3	513	6.15	-88	5.83	11
4	684	3.67	-65	2.80	173
5	855	2.48	-43	1.29	116
6	1026	1.90	-29	1.81	169
7	1197	1.84	-12	1.71	-59
8	1368	1.17	78	0.68	-94
9	1539	1.03	57	0.77	-34
10	1710	0.57	28	0.52	48

TABLE 7.2 PRESSURE RIPPLE DATA FOR 3.944 M AND 2.661 M PIPE LENGTHS OBTAINED BY EXPERIMENT

	RESONANT PIPE LENGTH 3.944 M			NON RESONANT PIPE LENGTH 2.661 M		
	EXPERIMENTAL RESULTS	DYNAMIC VALVE PROG. NO. 1	STEADY STATE VALVE PROG. NO. 3	EXPERIMENTAL RESULTS	DYNAMIC VALVE PROG. NO. 2	STEADY STATE VALVE PROG. NO. 4
AMPLITUDE (BAR)	14.02	20.46	22.33	3.73	3.75	3.51
	8.81	8.78	9.93	2.21	1.81	1.98
	6.15	5.95	6.80	5.83	4.80	5.30
	3.67	3.56	4.07	2.80	4.19	2.73
	2.48	2.67	3.04	1.29	1.07	1.23
	1.90	1.64	1.86	1.81	1.37	1.50
	1.84	1.21	1.35	1.71	1.96	1.76
	1.17	0.81	0.89	0.68	0.50	0.57
	1.03	0.59	0.63	0.77	0.54	0.56
	0.57	0.39	0.40	0.52	0.42	0.44
PHASE (DEGREES)	-106	-103.1	-99.2	37	44.2	41.1
	-89	-92.0	-88.5	-41	-30.8	-38.6
	-88	-83.9	-80.8	11	15.2	7.3
	-65	-59.3	-56.5	173	-177.2	179.4
	-43	-35.0	-32.4	116	137.2	119.3
	-29	-20.2	-17.7	169	-177.0	164.6
	-12	8.5	11.0	-59	-86.0	-72.1
	78	29.0	31.3	-94	-53.8	-83.3
	57	51.7	53.6	-34	-0.6	-28.9
	28	73.4	74.8	48	58.6	36.9

TABLE 7.3 FOURIER ANALYSIS OF SYSTEM SIMULATION WITH 14 cm/51 cc EQUIVALENT PIPELINE

	NON RESONANT PIPE LENGTH 2.661 M (DYNAMIC VALVE MODEL)			
	EXPERIMENTAL RESULTS	14 cm/40 cc PROG. NO. 14	14 cm/51 cc PROG. NO. 2	14 cm/60 cc PROG. NO. 18
AMPLITUDE (BAR)	3.73	3.67	3.76	3.86
	2.21	1.89	1.81	1.75
	5.83	5.88	4.80	4.17
	2.80	3.56	4.19	4.66
	1.29	1.17	1.07	0.99
	1.81	1.72	1.37	1.18
	1.71	2.76	1.96	1.50
	0.68	0.57	0.50	0.46
	0.77	0.67	0.54	0.46
	0.52	0.59	0.42	0.34
PHASE (DEG)	37	41.5	44.2	43.7
	-41	-35.9	-30.8	-31.4
	11	9.2	15.2	13.2
	173	-177.1	-177.2	169.0
	116	123.9	137.2	135.9
	169	165.9	-177.0	-178.0
	-59	-85.6	-86.0	-96.1
	-94	-77.8	-53.8	-54.5
	-34	-27.8	-0.6	-1.1
	48	31.7	58.6	56.3

TABLE 7.4 FOURIER ANALYSIS NON RESONANT SYSTEM, EQUIVALENT PIPELINE VOLUME EFFECTS

	NON RESONANT PIPE LENGTH 2.661 M (DYNAMIC VALVE MODEL)			
	EXPERIMENTAL RESULTS	10 cm/40 cc PROG. NO. 22	14 cm/40 cc PROG. NO. 14	18 cm/40 cc PROG. NO. 26
AMPLITUDE (BAR)	3.73	3.55	3.67	3.49
	2.21	1.97	1.89	2.02
	5.83	6.38	5.88	6.37
	2.80	2.89	3.56	2.71
	1.29	1.29	1.17	1.28
	1.81	2.01	1.72	1.68
	1.71	2.64	2.76	2.00
	0.68	0.70	0.57	0.56
	0.77	0.89	0.67	0.44
	0.52	1.09	0.59	0.19
PHASE (DEG)	37	42.3	41.5	43.2
	-41	-35.5	-35.9	-34.7
	11	12.6	9.2	14.4
	173	-169.9	-177.1	-169.3
	116	125.5	123.9	126.9
	169	170.0	165.9	171.9
	-59	-47.5	-85.6	-70.0
	-94	-74.6	-77.8	-72.9
	-34	-22.6	-27.8	-20.1
	48	61.0	31.7	31.1

TABLE 7.5 FOURIER ANALYSIS - NON RESONANT SYSTEM, EQUIVALENT PIPELINE LENGTH EFFECTS

RESONANT PIPE LENGTH 3.944 M					
	EXPERIMENTAL RESULTS	BREAK FREQ. 600 HZ	BREAK FREQ. 800 HZ	BREAK FREQ. 1000 HZ	BREAK FREQ. 1200 HZ
		PROG. NO. 29	PROG. NO. 1	PROG. NO. 31	PROG. NO. 33
AMPLITUDE (BAR)	14.02	19.94	20.46	20.78	20.98
	8.81	8.47	8.78	8.98	9.12
	6.15	5.72	5.95	6.10	6.19
	3.67	3.43	3.56	3.65	3.71
	2.48	2.57	2.67	2.73	2.77
	1.90	1.59	1.64	1.68	1.70
	1.84	1.18	1.21	1.23	1.25
	1.17	0.79	0.81	0.82	0.83
	1.03	0.58	0.59	0.59	0.60
	0.57	0.38	0.39	0.39	0.39
PHASE (DEGREES)	-106	-104.4	-103.1	-102.3	-101.7
	-89	-93.1	-92.0	-91.3	-90.8
	-88	-84.8	-83.9	-83.3	-82.9
	-65	-60.1	-59.3	-58.8	-58.4
	-43	-35.8	-35.0	-34.5	-34.2
	-29	-20.9	-20.2	-19.7	-19.4
	-12	7.8	8.5	9.0	9.3
	78	28.4	29.0	29.4	29.7
	57	51.0	51.7	52.0	52.2
	28	73.0	73.4	73.7	73.9

TABLE 7.6 FOURIER ANALYSIS - RESONANT SYSTEM, DYNAMIC VALVE TIME CONSTANT EFFECTS

NON RESONANT PIPE LENGTH 2.661 M					
	EXPERIMENTAL RESULTS	BREAK FREQ. 600 HZ PROG. NO. 30	BREAK FREQ. 800 HZ PROG. NO. 2	BREAK FREQ. 1000 HZ PROG. NO. 32	BREAK FREQ. 1200 HZ PROG. NO. 34
AMPLITUDE (BAR)	3.73	3.87	3.76	3.71	3.68
	2.21	1.76	1.81	1.84	1.87
	5.83	4.65	4.80	4.89	4.95
	2.80	4.81	4.19	3.85	3.64
	1.29	1.01	1.07	1.11	1.13
	1.81	1.34	1.37	1.40	1.42
	1.71	1.88	1.96	1.99	1.99
	0.68	0.48	0.50	0.52	0.52
	0.77	0.53	0.54	0.54	0.54
	0.52	0.41	0.42	0.42	0.43
PHASE (DEGREES)	37	44.2	44.2	44.1	44.2
	-41	-30.3	-30.8	-31.1	-31.2
	11	14.6	15.2	15.5	15.8
	173	178.5	-177.2	-175.1	-173.7
	116	138.2	137.2	136.9	136.5
	169	-177.1	-177.0	-176.9	-176.8
	-59	-93.7	-86.0	-80.2	-76.0
	-94	-51.9	-53.8	-55.0	-55.5
	-34	-0.7	-0.6	-0.7	-0.6
	48	57.3	58.6	59.6	60.4

TABLE 7.7 FOURIER ANALYSIS - NON RESONANT SYSTEM DYNAMIC VALVE TIME CONSTANT EFFECTS

	RESONANT PIPE LENGTH 3.944 M			NON RESONANT PIPE LENGTH 2.661 M		
	EXPERIMENTAL RESULTS	DYNAMIC VALVE PROG. NO. 35	STEADY STATE VALVE PROG. NO. 37	EXPERIMENTAL RESULTS	DYNAMIC VALVE PROG. NO. 36	STEADY STATE VALVE PROG. NO. 38
AMPLITUDE (BAR)	14.02	19.01	20.70	3.73	3.60	3.35
	8.81	7.90	8.85	2.21	1.93	2.09
	6.15	5.28	5.94	5.83	5.12	5.66
	3.67	3.12	3.49	2.80	3.16	1.97
	2.48	2.27	2.54	1.29	1.15	1.28
	1.90	1.37	1.49	1.81	1.31	1.43
	1.84	0.95	1.04	1.71	1.51	0.98
	1.17	0.58	0.59	0.68	0.46	0.48
	1.03	0.35	0.38	0.77	0.34	0.35
	0.57	0.16	0.14	0.52	0.14	0.15
PHASE (DEGREES)	-106	-104.1	-100.9	37	42.7	42.7
	-89	-90.1	-87.3	-41	-35.5	-37.5
	-88	-79.4	-77.5	11	10.5	12.1
	-65	-52.9	-50.9	173	-179.5	-174.2
	-43	-26.6	-25.6	116	125.5	123.6
	-29	-10.2	- 8.7	169	170.0	171.4
	-12	19.4	19.7	-59	-94.2	-61.4
	78	40.7	41.4	-94	-75.6	-74.9
	57	63.7	63.7	-34	-21.4	-19.9
	28	83.4	78.7	48	29.3	16.0

TABLE 7.8 FOURIER ANALYSIS - SYSTEM SIMULATION WITH 18 cm 51 cc EQUIVALENT PIPELINE

HARMONIC NO.	FREQUENCY HZ	AMPLITUDE $\frac{Z_T}{Z_{TSS}}$ (dB)	PHASE (DEG)
1	171	-0.72	-15
2	342	-1.83	-20
3	513	-1.21	-30
4	684	-2.73	-41
5	855	-3.61	-47
6	1026	-4.01	-58
7	1197	-4.15	-57
8	1368	-3.74	-49
9	1539	-5.51	-68
10	1710	-5.51	-75

TABLE 7.9 DYNAMIC CHARACTERISTICS OF RESTRICTOR VALVE

PROG. NO.	TASK/RUN NO.	OUTLET PIPE LENGTH (M)	EQUIVALENT PIPE		VALVE MODEL		FRICTION MODEL	TIMESTEP (SEC)	RUN TIME (E.T.U'S)	COMMENTS
			LENGTH	VOLUME	TYPE	BREAK FREQ. (HZ)				
1	TRY1A (5504)	3.944	14 cm	51 cc	DYN	800	DYNAMIC	5.0909E-5	2672	Fourier analysis res 1a/data 1a
2	TRY2A (5505)	2.661	14 cm	51 cc	DYN	800	DYNAMIC	5.0909E-5	2029	Fourier analysis res 2a/data 2a
3	TRY3A (5506)	3.944	14 cm	51 cc	S.S	-	DYNAMIC	5.0909E-5	2076	Fourier analysis res 3a/data 3a
4	TRY4A (5507)	2.661	14 cm	51 cc	S.S	-	DYNAMIC	5.0909E-5	1475	Fourier analysis res 4a/data 4a
5	TRY1B (5493)	3.944	14 cm	51 cc	DYN	800	STEADY STATE	5.0909E-5	1802	
6	TRY2B (5494)	2.661	14 cm	51 cc	DYN	800	STEADY STATE	5.0909E-5	1371	
7	TRY3B (5495)	3.944	14 cm	51 cc	S.S	-	STEADY STATE	5.0909E-5	1222	
8	TRY4B (5496)	2.661	14 cm	51 cc	S.S	-	STEADY STATE	5.0909E-5	810	
9	TRY1C (5458)	3.944	14 cm	51 cc	DYN	800	ZERO FRICTION	5.0909E-5	1384	
10	TRY2C (5459)	2.661	14 cm	51 cc	DYN	800	ZERO FRICTION	5.0909E-5	1145	
11	TRY3C (5460)	3.944	14 cm	51 cc	S.S	-	ZERO FRICTION	5.0909E-5	813	

TABLE 7.10 SUMMARY OF PROGRAMS RUN

PROG. NO.	TASK/RUN NO.	OUTLET PIPE LENGTH (M)	EQUIVALENT PIPE		VALVE MODEL		FRICTION MODEL	TIMESTEP (SEC)	RUN TIME (E.T.U'S)	COMMENTS
			LENGTH	VOLUME	TYPE	BREAK FREQ. (HZ)				
12	TRY4C (5461)	2.661	14 cm	51 cc	S.S	-	ZERO FRICTION	5.0909E-5	608	
13	TRY51 (5512)	3.944	14 cm	40 cc	DYN	800	DYNAMIC	5.0909E-5	2671	
14	TRY42 (5513)	2.661	14 cm	40 cc	DYN	800	DYNAMIC	5.0909E-5		Fourier analysis res 42/data 42
15	TRY43 (5514)	3.944	14 cm	40 cc	S.S	-	DYNAMIC	5.0909E-5	2067	
16	TRY44 (5515)	2.661	14 cm	40 cc	S.S	-	DYNAMIC	5.0909E-5	1548	
17	TRY61 (5516)	3.944	14 cm	60 cc	DYN	800	DYNAMIC	5.0909E-5	2687	
18	TRY62 (5517)	2.661	14 cm	60 cc	DYN	800	DYNAMIC	5.0909E-5	2138	Fourier analysis res 62/data 62
19	TRY63 (5518)	3.944	14 cm	60 cc	S.S	-	DYNAMIC	5.0909E-5	2087	
20	TRY64 (5519)	2.661	14 cm	60 cc	S.S	-	DYNAMIC	5.0909E-5	1462	
21	TRY11 (5520)	3.944	10 cm	40 cc	DYN	800	DYNAMIC	3.6364E-5	4413	
22	TRY12 (5521)	2.661	10 cm	40 cc	DYN	800	DYNAMIC	3.6364E-5	3128	Fourier analysis res 12/data 12

TABLE 7.10 SUMMARY OF PROGRAMS RUN

PROG. NO.	TASK/RUN NO.	OUTLET PIPE LENGTH (M)	EQUIVALENT PIPE		VALVE MODEL		FRICTION MODEL	TIMESTEP (SEC)	RUN TIME (E.T.U'S)	COMMENTS
			LENGTH	VOLUME	TYPE	BREAK FREQ. (HZ)				
23	TRY13 (5522)	3.944	10 cm	40 cc	S.S.	-	DYNAMIC	3.6364E-5	3599	
24	TRY14 (5523)	2.661	10 cm	40 cc	S.S	-	DYNAMIC	3.6364E-5	2583	
25	TRY81 (5534)	3.944	18 cm	40 cc	DYN	800	DYNAMIC	4.3636E-5	3131	
26	TRY82 (5535)	2.661	18 cm	40 cc	DYN	800	DYNAMIC	4.3636E-5	2431	Fourier analysis res 82/data 82q
27	TRY83 (5536)	3.944	18 cm	40 cc	S.S	-	DYNAMIC	4.3636E-5	2816	
28	TRY84 (5537)	2.661	18 cm	40 cc	S.S	-	DYNAMIC	4.3636E-5	2048	
29	TRYF1 (5538)	3.944	14 cm	51 cc	DYN	600	DYNAMIC	5.0909E-5	2339	Fourier analysis res f1/data f1
30	TRYF2 (5539)	2.661	14 cm	51 cc	DYN	600	DYNAMIC	5.0909E-5	1891	Fourier analysis res f2/data f2
31	TRYF3 (5540)	3.944	14 cm	51 cc	DYN	1000	DYNAMIC	5.0909E-5	2588	Fourier analysis res f3/data f3
32	TRYF4 (5541)	2.661	14 cm	51 cc	DYN	1000	DYNAMIC	5.0909E-5	2101	Fourier analysis res f4/data f4
33	TRYF5 (5542)	3.944	14 cm	51 cc	DYN	1200	DYNAMIC	5.0909E-5	2735	Fourier analysis res f5/data f5

TABLE 7.10 SUMMARY OF PROGRAMS RUN

PROG. NO.	TASK/RUN NO.	OUTLET PIPE LENGTH (M)	EQUIVALENT PIPE		VALVE MODEL		FRICTION MODEL	TIMESTEP (SEC)	RUN TIME (E.T.U'S)	COMMENTS
			LENGTH	VOLUME	TYPE	BREAK FREQ. (HZ)				
34	TRYF6 (5543)	2.661	14 cm	51 cc	DYN	1200	DYNAMIC	5.0909E-5	2250	Fourier analysis res f6/data f6
35	TRY81 (5544)	3.944	18 cm	51 cc	DYN	800	DYNAMIC	4.3636E-5	3223	Fourier analysis res 81/data 81
36	TRY82 (5545)	2.661	18 cm	51 cc	DYN	800	DYNAMIC	4.3636E-5	2442	Fourier analysis res 82/data 82
37	TRY83 (5546)	3.944	18 cm	51 cc	S.S	-	DYNAMIC	4.3636E-5	2670	Fourier analysis res 83/data 83
38	TRY84 (5547)	2.661	18 cm	51 cc	S.S	-	DYNAMIC	4.3636E-5	1955	Fourier analysis res 84/data 84
39	TRYF1 (5528)	3.944	14 cm	40 cc	DYN	600	DYNAMIC	5.0909E-5	2516	
40	TRYF2 (5529)	2.661	14 cm	40 cc	DYN	600	DYNAMIC	5.0909E-5	1845	
41	TRYF3 (5530)	3.944	14 cm	40 cc	DYN	1000	DYNAMIC	5.0909E-5	2585	
42	TRYF4 (5531)	2.661	14 cm	40 cc	DYN	1000	DYNAMIC	5.0909E-5	2130	
43	TRYF5 (5532)	3.944	14 cm	40 cc	DYN	1200	DYNAMIC	5.0909E-5	2763	
44	TRYF6 (5533)	2.661	14 cm	40 cc	DYN	1200	DYNAMIC	5.0909E-5	2286	

TABLE 7.10 SUMMARY OF PROGRAMS RUN

APPENDIX 5.1 OPEN LOOP HYDROSTATIC TRANSMISSION SYSTEM DATA (14 KW APPROX.)

	DIA (mm)	LENGTH (m)	WALL THICKNESS (mm)	YOUNG'S MODULUS (N/m^2)
PIPE 1	33	1.0	2.5	1.165×10^{11}
PIPE 2	17	4.0	1.5	1.165×10^{11}
PIPE 3	17	1.0	1.5	1.165×10^{11}

PUMP DATA

displacement	DP	$= 5.16 \times 10^{-3}$	l/rad
slip loss coefficients	KLP	$= 3.25 \times 10^{-13}$	m^5/Ns
	KTP	$= 6.50 \times 10^{-13}$	m^5/Ns
torque loss coefficient	KVP	$= 3.60 \times 10^{-2}$	Nm/(rad/s)
inertia	IPM	= 0.1	$Kg\ m^2$

MOTOR DATA

displacement	DM	$= 7.73 \times 10^{-3}$	l/rad
slip loss coefficients	KLM	$= 4.87 \times 10^{-13}$	m^5/Ns
	KTM	$= 9.75 \times 10^{-13}$	m^5/Ns
torque loss coefficient	KVM	$= 5.39 \times 10^{-2}$	Nm/(rad/s)
inertia	IM	= 0.14	$Kg\ m^2$

LOAD DATA

load inertia = 0.44 Kg/m^2

viscous friction = 0.63 Nm/(rad/sec)

INITIAL CONDITIONS

All pipe pressures 5 bar   Pump speed = 2100 rpm

load speed = 0

APPENDIX 5.2 DYNAMIC ANALYSIS OF SIMPLE TRANSMISSION SYSTEM

System equations

Pump flow $$Q_{o\ PUMP} = \omega_p D_p x_p - (K_{LP} + K_{TP})P \qquad 1$$

Motor flow $$Q_{I\ MOTOR} = \omega_m D_m x_m + (K_{LM} + K_{TM})P \qquad 2$$

Compressibility $$Q_c = \frac{V}{\beta_E}\frac{dp}{dt} \qquad 3$$

where P is the system pressure and V is the volume of the system pipeline. All other parameters are defined in appendix 5.1.

The above set of equations can be represented as a block diagram (Figure 1) and a transfer function can be obtained between x_p and ω_m.

$$\frac{\omega_m}{x_m} = \frac{K}{(s^2 + 2\zeta\omega_n s + \omega_n 2)} \qquad 4$$

where

$$K = \frac{\omega_p \beta_E D_p D_m x_m}{V I_L} \qquad 5$$

$$2\zeta\omega_n = \frac{\beta_E K_{TS} I_L + f_L V}{V I_L} \qquad 6$$

$$\omega n^2 = \frac{f_L \beta_E K_{TS} + (D_m x_m)^2 \beta_E}{V I_L} \qquad 7$$

where

$$K_{TS} = K_{LP} + K_{TP} + K_{LM} + K_{TM} \qquad 8$$

Substituting values from appendix 5.1 into expressions 6 , 7 and 8 allows values of damping ratio ζ and natural frequency ω_n to be calculated.

The damped natural frequency is given by the expression

$$\omega = \omega_n \sqrt{1 - \zeta^2} \qquad 9$$

For the transmission system in question the following values were obtained

damping ratio ζ = 0.186

damped natural frequency ω = 15.3 rad/sec

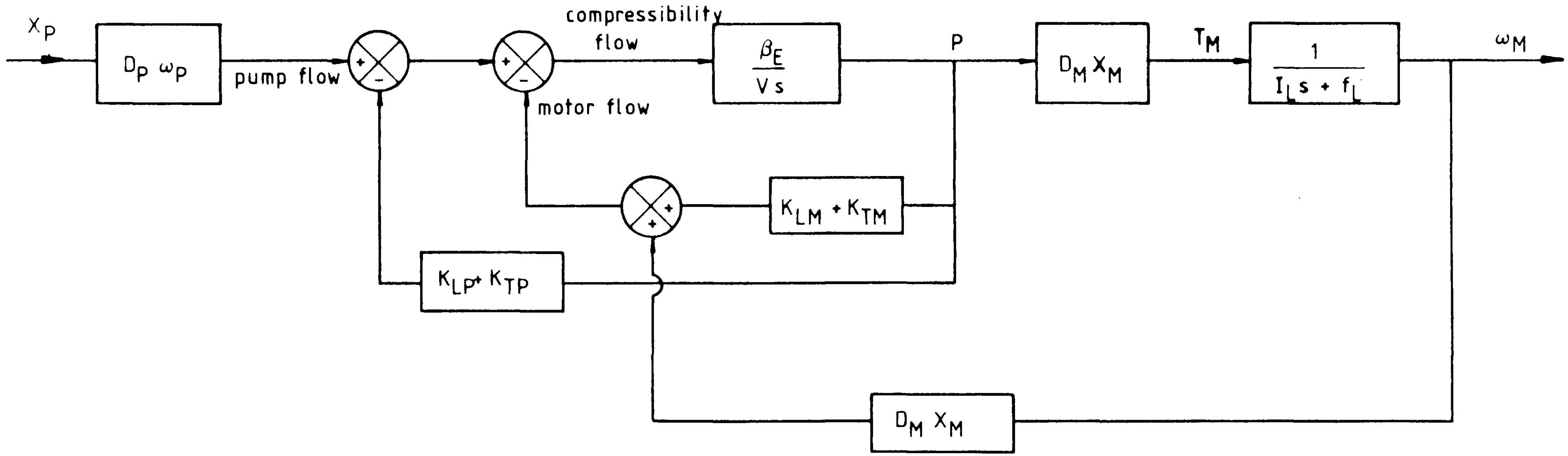

FIGURE 1 BLOCK DIAGRAM FOR SIMPLE TRANSMISSION SYSTEM

APPENDIX 5.3 CLOSED LOOP HYDROSTATIC BOAT TRANSMISSION DATA

PIPE DATA

	DIA (mm)	LENGTH (m)	WALL THICKNESS (mm)	YOUNG'S MODULUS (N/m^2)
PIPE 1	17.0	1.1	1.5	5.98×10^{10}
PIPE 2	17.0	3.3	1.5	5.98×10^{10}
PIPE 3	17.0	3.3	1.5	5.98×10^{10}
PIPE 4	17.0	1.1	1.5	5.98×10^{10}

PUMP DATA

displacement	DP	$= 5.16 \times 10^{-3}$	l/rad
slip loss coefficients	KLP	$= 6.07 \times 10^{-13}$	m^5/Ns
	KTP	$= 1.21 \times 10^{-12}$	m^5/Ns
torque loss coefficient	KVM	$= 1.92 \times 10^{-2}$	Nm/(rad/sec)
inertia	IPM	= 0.1	Kg m^2

MOTOR DATA

displacement	DM	$= 7.73 \times 10^{-3}$	l/rad
slip loss coefficients	KLM	$= 9.09 \times 10^{-13}$	m^5/Ns
	KTM	$= 1.82 \times 10^{-12}$	m^5/Ns
torque loss coefficient	KVM	$= 2.89 \times 10^{-2}$	Nm/(rad/sec)
inertia	IM	= 0.14	Kg m^2

BOOST PUMP AND RELIEF VALVE DATA

Boost pressure	PB	= 5.0	bar
Relief valve cracking pressure	PCRV	=200	bar
Check valve cracking pressure	PCNR	= 1.0	bar
Relief valve gradient	GRV	$= 4.0 \times 10^{-9}$	$(M^3/s)/(N/m^2)$
Check valve gradient	GNRV	$= 4.0 \times 10^{-9}$	$(M^3/s)/(N/m^2)$

PRIME MOVER AND LOAD DATA

Prime mover and load characteristics are given in Figures 1 and 2

Prime mover inertia	IE = 1.0	Kg m^2
Propeller diameter	PDIA = 0.411	m
Frontal area of boat	AB = 0.85	m^2
Drag coefficient	CD = 0.3	-
Propeller inertia	IP = 0.3	Kg m^2
Mass of boat	BMS = 5000	Kg

INITIAL CONDITIONS

All pipe pressures 5 bar

Initial pump speed 2068 rpm (2100 nominal) Initial pump torque = 7.79 Nm

Initial load speed 0 m/s

Initial propellor speed 0 rad/sec

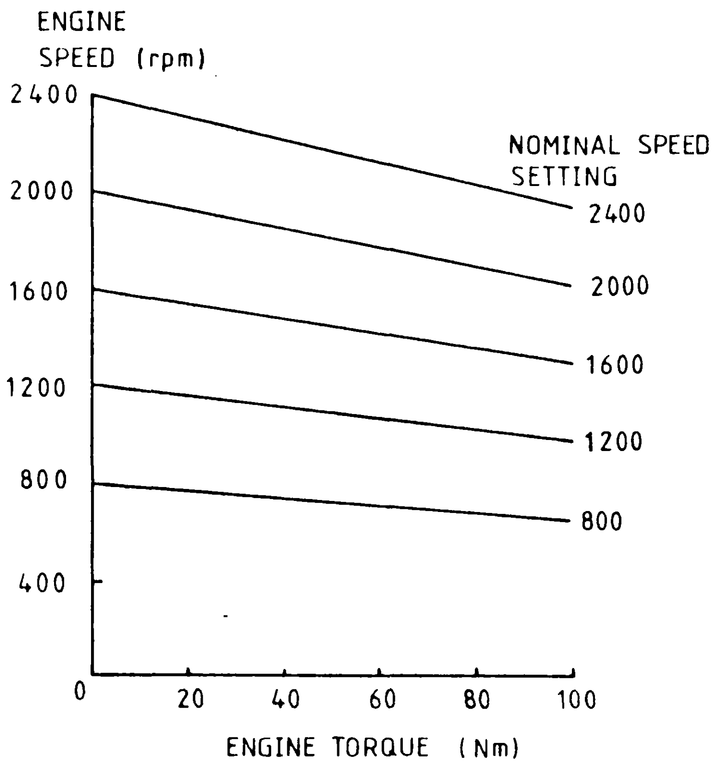

FIGURE 1 ENGINE CHARACTERISTICS

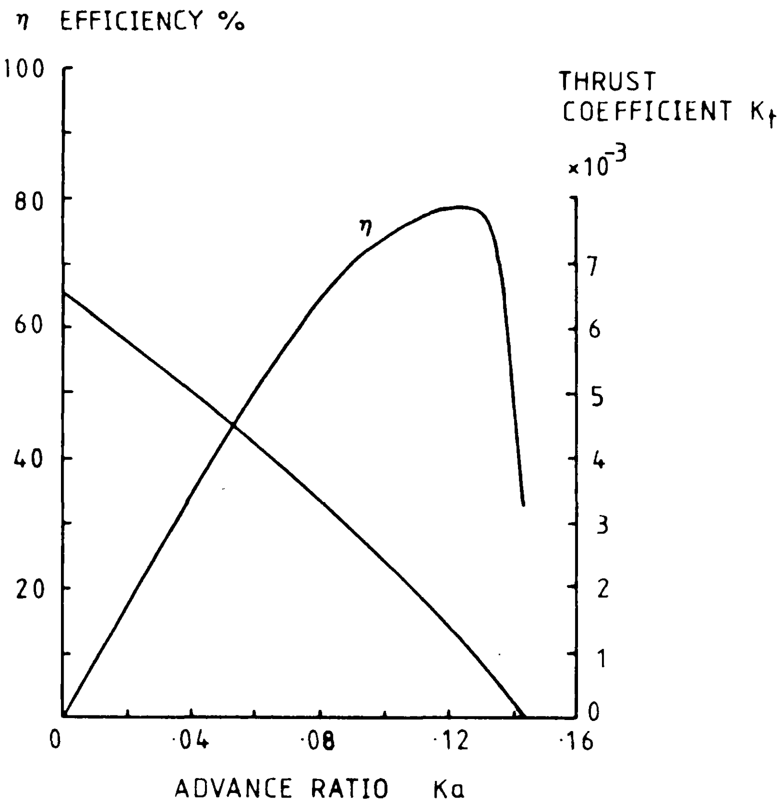

FIGURE 2 PROPELLER CHARACTERISTICS

APPENDIX 6.1 TEST FOR INTEGRATION ACCURACY AND STABILITY

RICHARDSON EXTRAPOLATION

In general for a p^{th} order numerical integration method the following relationship applies, and is a measure of the numerical errors produced.

$$T - c(h) \simeq Kh^p \qquad 1$$

where T is the true solution, c(h) the computed solution with step length h and K is some constant.

Therefore $$T - c(2h_o) \simeq 2^p\, h_o{}^p K \qquad 2$$

$$T - c(h_o) \simeq h_o{}^p K \qquad 3$$

$$T - c(h_o/2) \simeq \frac{1}{2^p}\, h_o{}^p K \qquad 4$$

thus

$$c(h_{o/2}) - c(h_o) \simeq h_o{}^p K - \frac{h_o{}^p K}{2^p} \simeq h_o{}^p K\left(1 - \frac{1}{2^p}\right) \qquad 5$$

$$c(h_o) - c(2h_o) \simeq 2^p h_o K - h_o{}^p K \simeq 2^p h_o{}^p K\left(1 - \frac{1}{2^p}\right) \qquad 6$$

thus

$$\frac{c(^{ho}/2) - c(h_o)}{c(h_o) - c(2h_o)} \simeq \frac{1}{2^p} \equiv \frac{c(\Delta t) - c(2\Delta t)}{c(2\Delta t) - c(4\Delta t)} \simeq \frac{1}{2^p} \qquad 7$$

In a system where the solution is represented by i variables ($y_1, y_2, y_3 \ldots\ldots y_i$) equation 7 can be written as

$$\frac{\sqrt{\sum_{j=1}^{i} (y_j^{\Delta t} - y_j^{2\Delta t})^2}}{\sqrt{\sum_{j=1}^{i} (y_j^{2\Delta t} - y_j^{4\Delta t})^2}} = \frac{\left\| \underline{y}^{\Delta t} - \underline{y}^{2\Delta t} \right\|_2}{\left\| \underline{y}^{2\Delta t} - \underline{y}^{4\Delta t} \right\|_2} \simeq \frac{1}{2^p} \qquad 8$$

The term $\left\| \underline{y}^{\Delta t} - \underline{y}^{2\Delta t} \right\|_2$ is called a 2 norm and represents a distance between points in a multidimensional space.

The Barmag valve solution is in seven variables however four of these are of the order 10^{-4} (x, Q_T, Q_I, Q_O) and three are of order 10^5 (P_I, P_O, P_C). The small variables would be negligible in evaluating the 2 norm. Equation 8 is evaluated at three instants in time and the results are presented in table 1 .

TIME (s)	TIMESTEP = Δt			TIMESTEP = $2\Delta t$			TIMESTEP = $4\Delta t$			$\frac{c(\Delta t) - c(2\Delta t)}{c(2\Delta t) - c(4\Delta t)}$
	PROGRAM NO. 1200			PROGRAM NO. 1600			PROGRAM NO. 1700			
	P_I bar	P_o bar	P_c bar	P_I bar	P_o bar	P_c bar	P_I bar	P_o bar	P_c bar	
3.1050E-2	52.109	45.238	48.259	52.191	45.346	48.348	51.956	45.057	48.098	0.36
9.4999E-2	45.340	39.983	41.358	45.719	40.077	41.747	45.054	39.874	41.060	0.56
2.4803E-2	44.981	40.982	41.031	45.004	41.001	41.054	44.952	40.968	41.001	0.46

TABLE 1 INTEGRATION ACCURACY TEST RESULTS

APPENDIX 6.2 THEORETICAL ANALYSIS OF SPOOL-DAMPER MECHANISM

The spool and damper mechanism was reduced to the system shown in Figure 1. Flow through the damping restrictor was assumed to obey a linear pressure flow relationship.

$$Q_R = K_L(P_o - P_c) \qquad 1$$

Considering the fluid compressibility in the spring chamber volume and a force balance on the spool yields the following equations

$$K_L(P_o - P_c) + A\dot{x} = \frac{Vc}{\beta}\dot{P}_c \qquad 2$$

$$- P_c A = M\ddot{x} + f\dot{x} + kx \qquad 3$$

Equations 2 and 3 are shown as a block diagram in Figure 2. The closed loop transfer function between P_o (input) and x (output) is;

$$\frac{x}{-P_o} = \frac{AK_L}{(K_L + T_c s)(Ms^2 + fs + k) + A^2 s} \qquad 4$$

Note T_c is the time constant of the spring chamber volume only.
$T_c = \frac{V_c}{\beta}$ The characteristic equation of the transfer function is;

$$s^3 + \left(\frac{K_L M + T_c f}{T_c M}\right)s^2 + \left(\frac{K_L f + T_c K + A^2}{T_c M}\right)s + \frac{K_L K}{T_c M} = 0 \qquad 5$$

Any third order system can be treated as the product of a first and second order system.

$$(1 + Ts)(s^2 + 2\zeta\omega_n s + \omega_n^2) \qquad 6$$

The purpose of this analysis was to investigate how the time constant T, the damping ratio ζ and the natural frequency ω_n were affected by varying certain parameters defining the spool-damping mechanism. Unfortunately direct algebraic expressions could not be found for T, ζ and ω_n, and a numerical approach was required.

A third order equation such as equation 5 has three roots which may be plotted on an Argand diagram (Figure 3). T, ζ and ω_n are defined by the position of the roots. Using a numerical polynominal equation solver the roots of the characteristic equation may be found in terms of their real and imaginary parts and hence T, ζ and ω_n may be calculated.

Computer program pl.list used the above procedure to investigate the effect of flow coefficient K_L on the spool performance. K_L was incremented thus altering the coefficients of the characteristic equation, at each step the roots were found and the corresponding values of T, ζ and ω_n were calculated.

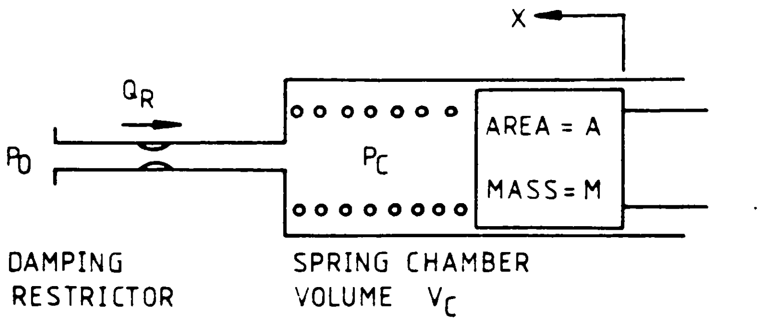

FIGURE 1 SPOOL-DAMPER MECHANISM

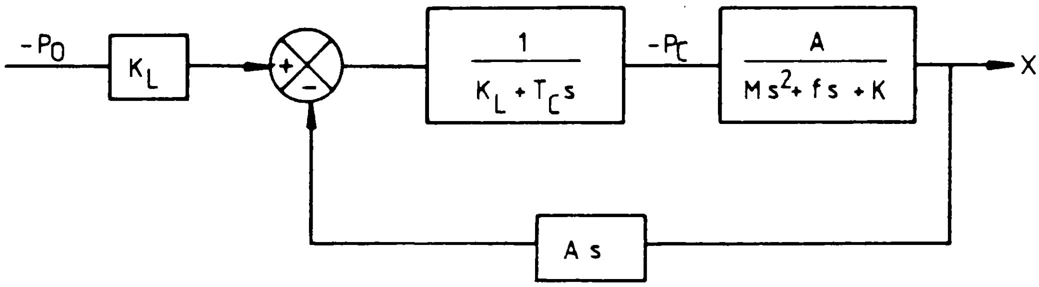

FIGURE 2 BLOCK DIAGRAM FOR SPOOL-DAMPER

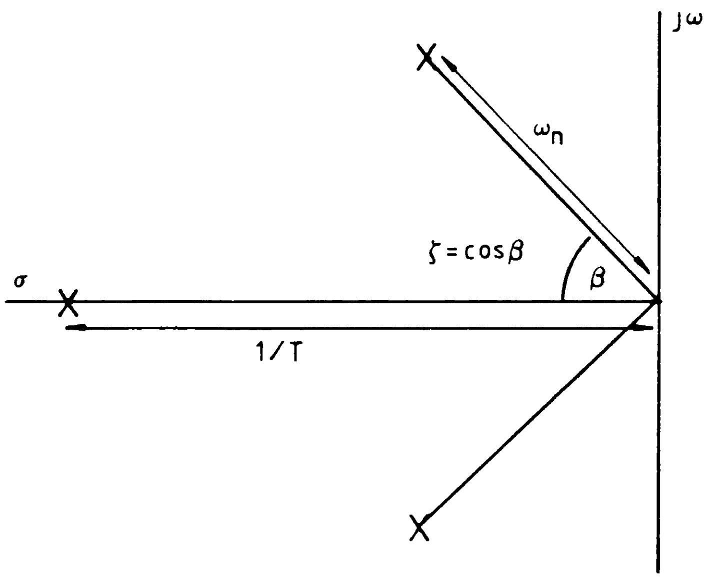

FIGURE 3 ROOTS ON THE S PLANE

APPENDIX 7.1 GENERAL DATA FOR SYSTEM SIMULATION USING THE PUMP MODEL INCORPORATING FLOW RIPPLE

PIPE DATA

	Outlet pipeline	Equivalent pipeline
diameter	15 mm	variable
wall thickness	1.5 mm	1.5 mm
Young's modulus	1.165 x 10" N/m^2	1.165 x 10" N/m^2
length	3.944 m or 2.661 m	10 cm/14 cm/18 cm

PUMP DATA

ideal flow	0.735 l/sec
number of pistons	7
leakage constant	$1.7561 \times 10^{-12} (M^3/s)/N/m^2$
speed	153.5 rad/sec

RESTRICTOR VALVE DATA

Steady state model		Dynamic valve model
discharge coefficient	0.7	0.7
area	$4.7605 \times 10^{-6}\ m^2$	$4.1485 \times 10^{-6}\ m^2$
downstream pressure	1	1
index	——	1.95
time constant	——	1.989E-4s

P A R T I I

COMPUTER PROGRAM REPORT

CONTENTS

PAGE NO.

MAIN PROGRAM STRUCTURE

INTRODUCTION

This document describes the structure and execution of a complete method of characteristics program. The Barmag valve simulation is used as an example of a typical application. Component model subroutines are documented separately. However, the data storage and overall format of component models is discussed, to enable a user with some knowledge of Fortran programming to write a specific simulation program and to develop further component model subroutines.

Certain sections of the main program are essential for the execution of the method of characteristics and so must be included in all programs irrespective of the application. Other sections such as those dealing with the output of data or the setting of common blocks are not critical and may therefore be tailored to suit the users specific requirements.

Data Storage

Two types of component data are defined; parameters which remain constant throughout a simulation, called constant data, and parameters which vary with each timestep, called dynamic data. For example the data storage arrays for the Barmag valve subroutine are:

constant data array FCV(15,NV)   held in named common block BLK35
dynamic data array FCVD(5,IT,NV) held in named common block BLK36

The majority of data transfer between the main program and various subroutines is via named common blocks, this is faster and more convenient than transfer through subroutine argument lists.

The argument NV is a component number which distinguishes between components of the same type. In a system with two Barmag valves the array elements FCV(9,1), FCV(9,2) contain the value of the preset orifice setting for valves 1 and 2 respectively. The argument IT is a relative time level indicator which shows the time at which a particular parameter was calculated. For example array element FCVD(1, 5,1) contains the compensator spool position at time level 5 (say time

= t) for valve number 1. Element FCVD(1,4,1) contains the value of spool position at the previous time level (time = t-Δt). A system simulation may take several thousand timesteps so it is impractical to store every item of dynamic data. The dynamic data arrays are over-written with new data as the solution progresses. At present the maximum value of IT is 10, which means that at any instant the dynamic data arrays contain the current value of every dynamic parameter plus the values calculated at 9 previous timesteps. In fact only the current value and one previous value are required for the successful execution of the program. IT maximum was set at 10 simply to store sufficient previous values for diagnostic purposes, array sizes can be minimised by setting IT maximum as 2.

Generally the user must specify the values of all constant array elements and the initial values of all dynamic parameters. (See below the read data section of the main program.) For some components initialising subroutines are called to calculate the values of constant parameters or to initialise the dynamic data arrays. The program documentation for each component gives details of the data arrays used and which array elements must be assigned values by the user at the start of the program.

Main Program

The main program structure is described in two parts; first a general discussion of how the program works outlining the essential features and the options open to the user, followed by a description of the Barmag valve simulation as an example of a specific application. The operation of the program is quite simple since the majority of computing is executed inside subroutines. Figure 1 is a flowchart showing the important features.

The computing environment is set by declaration statements at the head of the program. The data input includes a number of essential parameters, a number of optional parameters controlling the output of results, and all component data as specified in the documentation. Initialising subroutines are called to calculate any data which was not explicitly read in.

The simulation starts by setting the indicators IT = 2, ITM = 1, JT = 2. The function of IT has been explained above, ITM is used in accessing data values one timestep previous to the current time level. JT is used to calculate the current value of time at each step and indicates the number of timesteps taken by the solution. A sequence of call statements to component model subroutines calculates the system performance at the current time level. The sequence of call statements is called the system description. A check is made to see if results printout is required, if so the program branches, performs the output operations and increments the printout counters. The indicators IT, ITM, and JT are incremented. Conditional statements determine if IT has exceeded the maximum value (usually 10) and reset it to 1. If IT does equal 1, ITM is set to IT maximum (usually 10). If the simulation time limit has been reached the program stops, otherwise it loops back, the value of time is recalculated and the sequence of call statements is repeated. The program continues to loop until the simulation time limit is reached.

The declaration statements at the start of the program must include; dimension statements for all arrays used in the main program, labelled common statements as specified in the documentation for the system component models, any EXTERNAL statements required by plotting routines and any REAL or INTEGER statements. The dimensions in the labelled common statements must be the same as those of the corresponding statements in the subroutine listings. Otherwise the program may fail to compile, or if it does compile values will be assigned to the wrong data locations and the program will crash. The user may have to edit the component model subroutines to ensure that this requirement is satisfied. This is an easy task using a standard context editor.

The following parameters control program execution and must be assigned values at the start of the program.

TLIM - the time limit specifying the number of seconds of simulation time required. This is not a computer time limit, default computer time limits are set outside the program and depend on the operating system.

MAXP - a number specifying the total number of pipes in the system.

NPS - the number of the shortest pipe in the system.

FCTR - a factor used for artificially setting the simulation timestep. Its function is described in the documentation for subroutine CLPNT.

Parameters for controlling the output of results are all defined by the user and depend on the particular output scheme required.

The initialising subroutines complete the task of setting all the system data necessary for a simulation. The routines most commonly used are:-

FLUID - to calculate the properties of the hydraulic fluid. The user may omit this subroutine provided values are assigned to the variables:-

RHO	- fluid density	Kg/M^3
AVISC	- absolute viscosity	Ns/M^2
BM	- isentropic bulk modulus	N/M^2

WAVSPD - calculates the wavespeed based on the effective bulk modulus in each pipeline. This routine may be omitted if the user specifies wavespeeds directly. (See pipe model documentation.)

CLPNT - sets up the entire method of characteristics integration scheme. The computation performed is quite complex therefore CLPNT is included in all system simulations.

PQIN - initialises all the elements of the pipe data array (PDYN) with values of the initial system steady state. This function can be performed by reading data directly into the array elements, however, the number of pipe calculation points has to be known and this information is not available until subroutine CLPNT has been called. Subroutine PQIN is very convenient especially in applications where the user wishes to alter the integration timestep or the initial conditions frequently.

These subroutines must be called in the sequence given above. Certain component models have associated initialising routines, details are given in the individual component model documentation.

<u>Example program</u> - The simulation of a Barmag pressure compensated flow control valve

Before beginning to write a simulation program the user must have made all the engineering decisions relating to the simulation, selected the necessary models and prepared the necessary data by consulting the component model documentation.' The Barmag system consists of 5 components (Ref. 1 Main body of thesis) (Fig. 3) numbers are assigned to components which distinguish between components of the same type. The system description may now be written:-

```
CALL CFS (args, 1)     - constant flow source model (idealised pump)
CALL BARMAG (args, 1)  - Barmag valve model
CALL CPT (args, 1)     - constant pressure pipe termination (idealised
                         actuator load model)
CALL PIPE (1, args)    - } pipe model, one call statement required
CALL PIPE (2, args)    - } for each pipe in the system
CALL ZEROF (args)      - Zero friction routine which sets the
                         friction factor at all pipe calculation
                         points to zero
```

(where 'args' = the program defined arguments specified in the documentation.)

The order in which call statements are arranged is not important and does not affect the solution.

The rest of the program can be built up around the system description by consulting the individual component model documentation.

The initialising subroutines required are WAVSPD, CLPNT, PQIN, BARIN. Subroutine fluid in not used, the fluid data is entered directly. Routine BARIN is associated with the Barmag valve component model

The output of results is the only section left entirely to the user who must decide on the results required and the frequency and format of output. The output from the Barmag program consists of printout and graph plotting. A further feature is the inclusion of a subroutine called ERRAN which calculates the computing errors in the Barmag model. ERRAN is called at specific intervals and prints out additional data. Certain switches (NER, NPLOT) are input at the start of the program and used to suppress certain sections of the output. For instance if NPLOT is set to zero no graphs are plotted. The program includes a section which writes out for verification all the input data and all data calculated by the initialising routines. The entire program structure including details of the input and output is presented as a flowchart in Figure 2 . Further details may be found in the program listing, (Figure 5). Examples are given of a typical input data file and a results file in Figures 4 and 6 respectively.

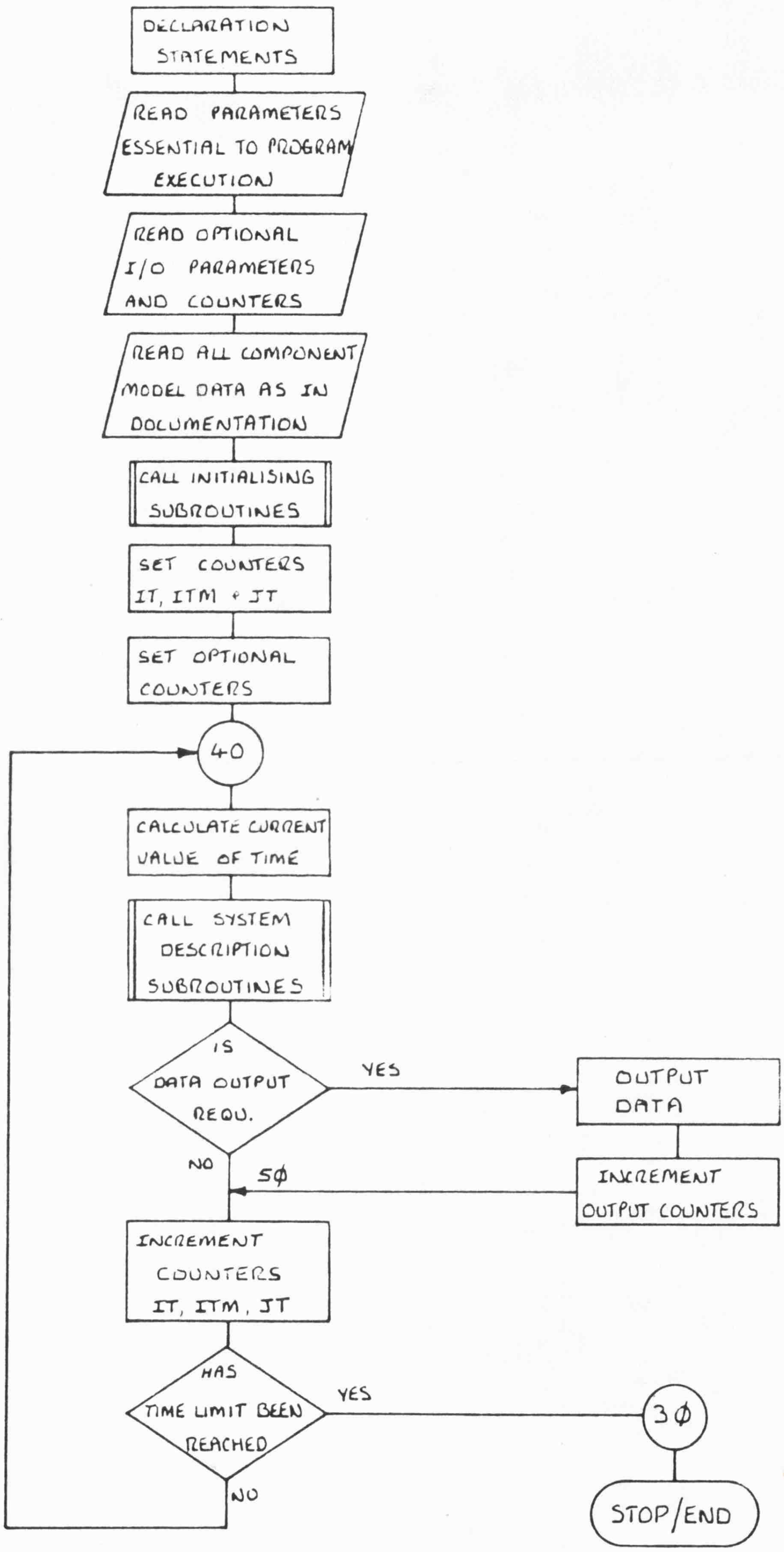

FIGURE 1 BASIC STRUCTURE OF THE MAIN PROGRAM

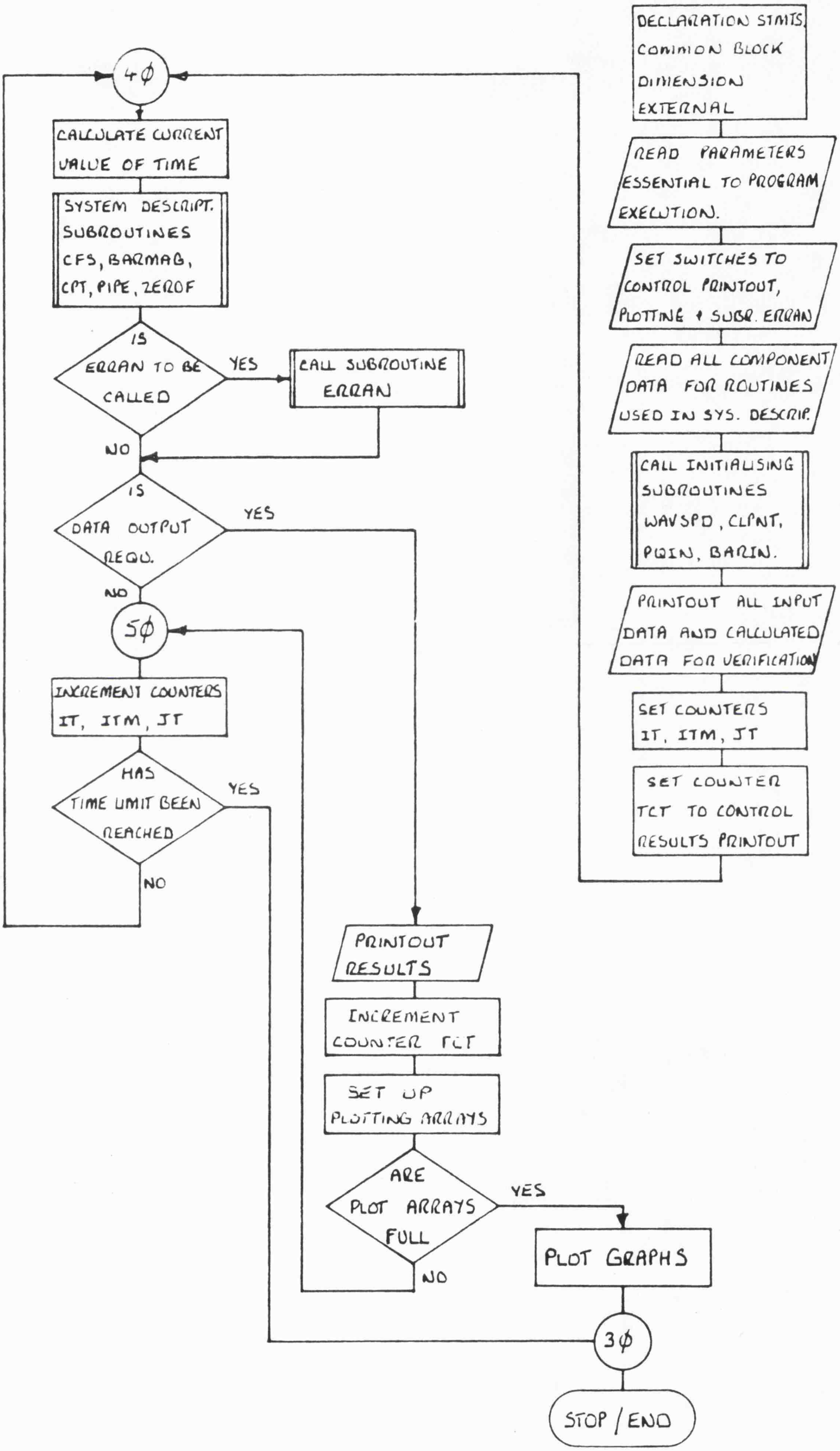

FIGURE 2 MAIN PROGRAM FOR BARMAG VALVE SIMULATION

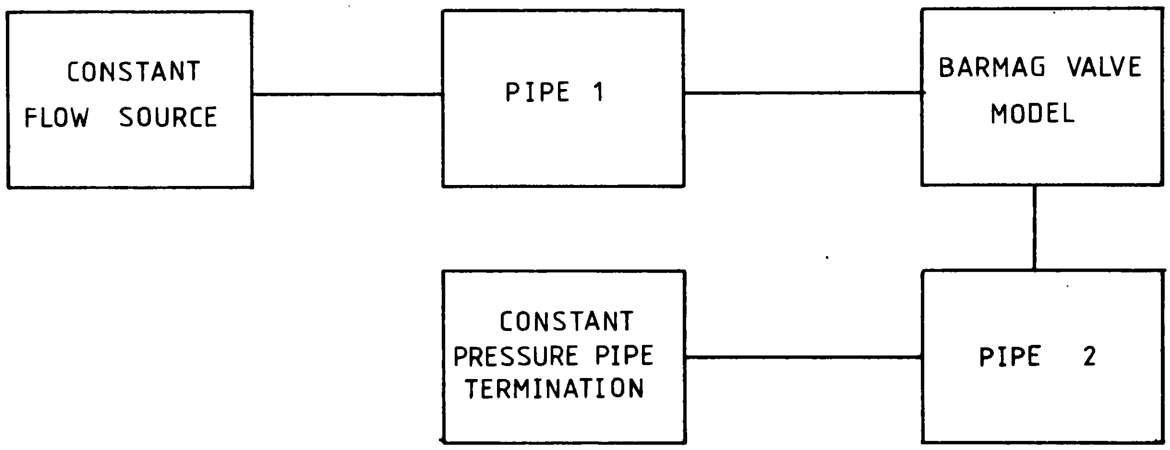

FIGURE 3 BARMAG VALVE SYSTEM SIMULATION

```
0.05 2 2 0.33333333
0 250 0.001
0 0.0060 0.0070 0.0290 0.0310 0.0875 0.0950
0.0127 1.26676e-4 0.006 6.4035e+8 5.1114
0.0127 1.26676e-4 0.0015 116.522e+9 1.5
25.12e+5 8.3294e-4
21.00e+5 4.0000e-4
2.56e-4 0.0653 2.78 16000.0 8520.0 1.239e-3
9.8e-6 2.8274e-7 1.471e-3 8.5188e-3 1.0 2.0
1.0e+5 1.10e-11
0.9109e-3 0.0 21.0e+5 0.0 4.3294e-4
8.3294e-4 1.0
21.0e+5 41.0e+5 5.0e-3 1.0e-3 3.25 2.0
870.0 6.973e-2 1.8e+9
```

FIGURE 4 INPUT DATA FILE FOR BARMAG SYSTEM

```
c     program name  bar1.fortran
c
c     library classification
c
c     title   simulation of barmag pressure compensated flow control
c             valve using the method of characteristics
c
c     fortran iv          honeywell multics                    20 nov 1980
c
c     no special hardware
c
c     purpose    calculation of transient response of the barmag
c                valve to a rapid change of load pressure
c
c     associated subroutines
c        component models      -  cfs,barmag,cpt,pipe,zerof
c        initialising routines-  wavspd,clpnt,pqin,barin
c        multics subroutines   -  plot_$setup
c                                 plot_
c                                 plot-$scale
c
c     variables
c
c     arpi    plotting array (inlet pressure)              n/m2
c     arpo    plotting array (outlet pressure)             n/m2
c     arqi    plotting array (inlet flow)                  m3/s
c     arqo    plotting array (outlet flow)                 m3/s
c     arqt    plotting array (bypass flow)                 m3/s
c     artm    plotting array (time)                        s
c     arxd    plotting array (spool velocity)              m/s
c     arpc    plotting array (spring chamber pressure)     n/m2
c     arxx    plotting array (spool position)              m
c     avisc   fluid viscosity (absolute)                   ns/m2
c     base    plotting routine argument                    -
c     bm      bulk modulus                                 n/m2
c     cf      constant flow source model data array        --
c     cp      pipe termination model data array            --
c     dt      timestep                                     s
c     fctr    factor for artificially altering the timestep --
c     fcv     barmag valve data (constant)                 --
c     fcvd    barmag value data (dynamic)                  --
c     i       counter                                      --
c     iplot   counter                                      --
c     it      time level indicator                         --
c     itm     time level indicator                         --
c     j       counter                                      --
c     jt      time level indicator                         --
c     k       counter                                      --
c     maxp    total number of pipes in system              --
c     ncp     number of calculation points                 --
c     ncp1    number of calculation points (pipe 1)        --
c     ncp2    number of calculation points (pipe 2)        --
c     ner     error analysis switch                        --
c     nplcr   number of point to be plotted                --
c     nplot   graph plotting switch                        --
c     nps     number of shortest pipe in system            --
c     pd      pipe data array (constant)                   --
c     pdyn    pipe data array (dynamic)                    --
```

FIGURE 5 PROGRAM LISTING

```
c       pin      initial condition data array                     --
c       pmin     laminar flow threshold pressure                  n/m2
c       rho      fluid density                                    kg/m3
c       t11      erran printout time                              s
c       t12      erran printout time                              s
c       t21      erran printout time                              s
c       t22      erran printout time                              s
c       t31      erran printout time                              s
c       t32      erran printout time                              s
c       tct      results printout time                            s
c       time     time                                             s
c       tinc     printout increment                               s
c       tlim     simulation time limit                            s
c
c       declaration statements
        common /blk1/pd(8,2) /blk2/pin(2,2) /blk3/pdyn(3,400,10,2)
        common /blk35/fcv(15,2) /blk36/fcvd(5,10,2)
        common /blk37/cf(2,2) /blk38/cp(6,2)
        dimension arpi(500),arpo(500),arqi(500),arqo(500)
        dimension arqt(500),arxx(500),arpc(500),arxd(500),artm(500)
        external plot_$setup (descriptors)
        external plot_ (descriptors)
        external plot_$scale (descriptors)
c
c       input parameters essential to program execution
        read(5,500)tlim,maxp,nps,fctr
c
c       set switches to control results printout
        iplot=0
        read(5,500)nplot,nplcr,tinc
        read(5,500)ner,t11,t12,t21,t22,t31,t32
c
c       input all component data
c
c       pipe data
        read(5,500)((pd(i,j),i=1,5),j=1,2)
c
c       initial pressure and flow condition data
        read(5,500)((pin(i,j),i=1,2),j=1,2)
c
c       flow control valve data
        read(5,500)(fcv(i,1),i=1,14)
        read(5,500)(fcvd(i,1,1),i=1,5)
c
c       pipe termination data
        read(5,500)(cf(i,1),i=1,2)
        read(5,500)(cp(i,1),i=1,6)
c
c       fluid data
        read(5,500)rho,avisc,bm
c
c       call initialising subroutines
        call wavspd(maxp,rho,bm)
        call clpnt(nps,dt,maxp,fctr)
        call pqin(maxp,rho,avisc,0,dt)
        call barin(rho,avisc,pmin,1)
c
c       write out input data for verification
        write(6,600)
        write(6,601)tlim,tinc,maxp,nps,fctr
        write(6,615)ner,t11,t12,t21,t22,t31,t32
```

```
      write(6,602)((pd(i,j),i=1,8),j=1,2)
      write(6,603)((pin(i,j),i=1,2),j=1,2)
      write(6,604)(fcv(i,1),i=1,15)
      write(6,605)(fcvd(i,1,1),i=1,5)
      write(6,606)(cf(i,1),i=1,2)
      write(6,607)(cp(i,1),i=1,6)
      write(6,609)dt,avisc,bm,rho
      write(6,616)pmin
      write(6,610)
      do 300 i=1,maxp
      write(6,611)i
      ncp=pd(6,i)
      do 301 j=1,ncp
      write(6,612)j,(pdyn(k,j,1,i),k=1,3)
  301 continue
  300 continue
      write(6,613)
c
c     set counters
      it=2
      jt=2
      itm=1
c
c     set printout counter
      tct=tinc
c
c     calculate current value of time
   40 time=(jt-1)*dt
c
c     system description
      call cfs(it,itm,dt,rho,1)
      call barmag(it,itm,dt,rho,bm,pmin,1)
      call cpt(it,itm,dt,rho,time,1)
      call pipe(1,it,itm,rho,dt)
      call pipe(2,it,itm,rho,dt)
      call zerof(maxp,it)
c
c     check if subroutine erran is to be called
      if(ner.eq.0)go to 999
      if(time.ge.t11.and.time.le.t12)go to 20
      if(time.ge.t21.and.time.le.t22)go to 20
      if(time.ge.t31.and.time.le.t32)go to 20
      go to 999
   20 call erran(it,dt,rho,time,tct,tinc,bm,pmin)
  999 continue
c
c     check if data output is required
      if(time.ge.tct)go to 25
c
c     increment counters
   50 it=it+1
      jt=jt+1
      itm=it-1
      if(it.gt.10)it=1
      if(it.eq.1)itm=10
c
c     has time limit been reached
      if((tlim-time).lt.dt)go to 30
      go to 40
c
```

```
c     printout results
   25 continue
      ncp1=pd(6,1)
      ncp2=pd(6,2)
      write(6,614)time,pdyn(1,ncp1,it,1),pdyn(1,1,it,2),
     1pdyn(2,ncp1,it,1),pdyn(2,1,it,2),fcvd(5,it,1),fcvd(1,it,1),
     1fcvd(2,it,1),fcvd(3,it,1),fcvd(4,it,1),pdyn(1,ncp2,it,2)
c
c     increment counter tct
      tct=tct+tinc
c
c     set up plotting arrays
      if(nplot.eq.0)go to 60
      iplot =iplot+1
      arpi(iplot)=pdyn(1,ncp1,it,1)
      arpo(iplot)=pdyn(1,1,it,2)
      arqi(iplot)=pdyn(2,ncp1,it,1)
      arqo(iplot)=pdyn(2,1,it,2)
      arqt(iplot)=fcvd(5,it,1)
      arxx(iplot)=fcvd(1,it,1)
      arpc(iplot)=fcvd(3,it,1)
      arxd(iplot)=fcvd(2,it,1)
      artm(iplot)=time
      if(iplot.eq.nplcr)go to 70
   60 continue
c
      go to 50
c
c     plot graphs
   70 continue
      call plot_$setup("inlet pressure","time (s)","pressure bar"
     1,1,base,2,0)
      call plot_$scale(0.0,0.275,2.0e6,5.5e6)
      call plot_ (artm,arpi,nplcr,1,'.')
      call plot_$setup("outlet pressure","time (s)","pressure bar"
     1,1,base,2,0)
      call plot_$scale(0.0,0.275,2.0e6,5.5e6)
      call plot_ (artm,arpo,nplcr,1,'.')
      call plot_$setup("inlet flow","time(s)","flow m3/s",1,base,2,0)
      call plot_$scale(0.0,0.275,-2.0e-4,12.0e-4)
      call plot_ (artm,arqi,nplcr,1,'.')
      call plot_$setup("outlet flow","time (s)","flow m3/s",1,base,2,0)
      call plot_$scale(0.0,0.275,-2.0e-4,12.0e-4)
      call plot_ (artm,arqo,nplcr,1,'.')
      call plot_$setup("bypass flow","time (s)","flow m3/s",1,base,2,0)
      call plot_$scale(0.0,0.275,-2.0e-4,12.0e-4)
      call plot_ (artm,arqt,nplcr,1,'.')
      call plot_$setup("spool position","time (s)","x  m",1,base,2,0)
      call plot_$scale(0.0,0.275,0.6e-3,0.96e-3)
      call plot_ (artm,arxx,nplcr,1,'.')
      call plot_$setup("spring chamber pressure","time (s)","pc bar"
     1,1,base,2,0)
      call plot_$scale(0.0,0.275,2.0e6,5.5e6)
      call plot_ (artm,arpc,nplcr,1,'.')
      call plot_$setup("spool velocity ","time (s)","velocity m/s"
     1,1,base,2,0)
      call plot_$scale(0.0,0.275,-0.02,0.025)
      call plot_ (artm,arxd,nplcr,1,'.')
c
   30 continue
c
```

```
c     formats
  500 format(v)
c
c
  600 format(1h0,20x,'simulation of a pressure compensated flow control
     1valve using the method of characteristics'/1h ,20x,91('*')
     1/1h0,'input data')
  601 format(1h0,'system constants'/1h0,'tlim =',1pe11.4
     1/1h ,'tinc =',1pe11.4/1h ,'maxp =',i2/1h ,'nps  =',i2/
     11h ,'fctr =',1pe11.4)
  615 format(1h0,'error analysis printout constants'/1h0,
     1'ner =',i2/1h ,'t11 =',1pe11.4/1h ,'t12 =',1pe11.4
     1/1h ,'t21 =',1pe11.4/1h ,'t22 =',1pe11.4/1h ,'t31 =',
     11pe11.4/1h ,'t32 =',1pe11.4)
  602 format(1h0,'pipe data'/1h0,15x,'dia',13x,'ap',13x,'wt',13x,
     1'ym',12x,'alp',12x,'rncp',12x,'ws',12x,'err'/1h ,'pipe 1 ',
     18(4x,1pe11.4)/1h ,'pipe 2 ',8(4x,1pe11.4))
  603 format(1h0,'initial pressure and flow conditions',
     1/1h0,14x,'pressure',8x,'flow'
     1/1h0,'pipe 1  ',2(4x,1pe11.4)/1h ,'pipe 2  ',2(4x,1pe11.4))
  604 format(1h0,'barmag flow control valve data'
     1/1h0,4x,'a',12x,'m',12x,'f',11x,'kff',10x,'ks',12x,'l'
     1,11x,'vsc',9x,'ad',/1h ,1pe11.4,7(2x,1pe11.4)
     1/1h0,4x,'y',12x,'isc',10x,'ipn',10x,'opn',10x,'prt',10x,'klin',
     19x,'kor'/1h ,1pe11.4,7(2x,1pe11.4))
  605 format(1h0,'flow control valve initial conditions'
     1/1h0,5x,'x',12x,'xd',13x,'pc',13x,'pcd',12x,'qt'
     1/1h ,1pe11.4,4(4x,1pe11.4))
  606 format(1h0,'pipe termination data'
     1/1h0,'constant flow source'/1h0,5x,'q',12x,'opn'/1h ,
     11pe11.4,4x,1pe11.4)
  607 format(1h0,'constant pressure load'/1h0,5x,'p1',12x,'p2',
     114x,'t1',13x,'t2',13x,'t3',13x,'ipn'/1h ,1pe11.4,5(4x,1pe11.4))
  609 format(1h0,'results',/1h0,'time step (dt)=',1pe11.4,' secs'
     1/1h ,'abs viscosity =',1pe11.4,' ns/m2'
     1/1h ,'bulk modulus  =',1pe11.4,' n/m2'
     1/1h ,'density       =',1pe11.4,' kg/m2')
  610 format(1h0,'initial conditions')
  611 format(1h0,'pipe  ',i2/1h ,24x,'pressure',7x,'flow',8x,
     1'fric. fac.')
  612 format(1h ,'calculation point',i2,1p3e14.4)
  613 format(1h0,'  time (s)    pi (n/m2)  po (n/m2)   qi (m3/s)    ',
     1'qo (m3/s)   qt (m3/s)   x (m)       xd (m/s)   pc (n/m2)    ',
     1'   pcd      pl (n/m2)')
  614 format(1h0,1pe11.4,10(1x,1pe11.4))
  616 format(1h0,'pmin = ',2x,1pe11.4)
c
c
      close(5)
      close(6)
      stop
      end
c
```

simulation of a pressure compensated flow control valve using the method of characteristics
****

input data

system constants

tlim = 5.0000E-02
tinc = 1.0000E-03
maxp = 2
nps = 2
fctr = 3.3333E-01

error analysis printout constants

ner = 0
t11 = 6.0000E-03
t12 = 7.0000E-03
t21 = 2.9000E-02
t22 = 3.1000E-02
t31 = 8.7500E-02
t32 = 9.5000E-02

pipe data

	dia	ap	wt	ym	alp	rncp	ws	err
pipe 1	1.2700E-02	1.2668E-04	6.0000E-03	6.4035E+08	5.1114E+00	5.2000E+01	5.4227E+02	-6.1360E-01
pipe 2	1.2700E-02	1.2668E-04	1.5000E-03	1.1652E+11	1.5000E+00	7.0000E+00	1.3527E+03	1.4541E-06

initial pressure and flow conditions

	pressure	flow
pipe 1	2.5120E+06	8.3294E-04
pipe 2	2.1000E+06	4.0000E-04

barmag flow control valve data

a	m	f	kff	ks	l	vsc	ad
2.5600E-04	6.5300E-02	2.7800E+00	1.6000E+04	8.5200E+03	1.2390E-03	9.8000E-06	2.8274E-07

y	isc	ipn	opn	prt	klin	kor
1.4710E-03	8.5188E-03	1.0000E+00	2.0000E+00	1.0000E+05	1.1000E-11	1.9319E-09

flow control valve initial conditions

x	xd	pc	pcd	qt
9.1090E-04	0.0000E+00	2.1000E+06	0.0000E+00	4.3294E-04

pipe termination data

constant flow source

q	opn
8.3294E-04	1.0000E+00

constant pressure load

p1	p2	t1	t2	t3	ipn
2.1000E+06	4.1000E+06	5.0000E-03	1.0000E-03	3.2500E+00	2.0000E+00

results

time step (dt)= 1.8482E-04 secs
abs viscosity = 6.9730E-02 ns/m2
bulk modulus = 1.8000E+09 n/m2
density = 8.7000E+02 kg/m2

pmin = 9.5865E+04

initial conditions

pipe 1

	pressure	flow	fric. fac.
calculation point 1	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 2	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 3	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 4	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 5	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 6	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 7	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 8	2.5120E+06	8.3294E-04	0.0000E+00
calculation point 9	2.5120E+06	8.3294E-04	0.0000E+00
calculation point10	2.5120E+06	8.3294E-04	0.0000E+00
calculation point11	2.5120E+06	8.3294E-04	0.0000E+00
calculation point12	2.5120E+06	8.3294E-04	0.0000E+00
calculation point13	2.5120E+06	8.3294E-04	0.0000E+00
calculation point14	2.5120E+06	8.3294E-04	0.0000E+00
calculation point15	2.5120E+06	8.3294E-04	0.0000E+00
calculation point16	2.5120E+06	8.3294E-04	0.0000E+00
calculation point17	2.5120E+06	8.3294E-04	0.0000E+00
calculation point18	2.5120E+06	8.3294E-04	0.0000E+00
calculation point19	2.5120E+06	8.3294E-04	0.0000E+00

FIGURE 6 RESULTS OUTPUT FILE

calculation point20	2.5120E+06	8.3294E-04	0.0000E+00
calculation point21	2.5120E+06	8.3294E-04	0.0000E+00
calculation point22	2.5120E+06	8.3294E-04	0.0000E+00
calculation point23	2.5120E+06	8.3294E-04	0.0000E+00
calculation point24	2.5120E+06	8.3294E-04	0.0000E+00
calculation point25	2.5120E+06	8.3294E-04	0.0000E+00
calculation point26	2.5120E+06	8.3294E-04	0.0000E+00
calculation point27	2.5120E+06	8.3294E-04	0.0000E+00
calculation point28	2.5120E+06	8.3294E-04	0.0000E+00
calculation point29	2.5120E+06	8.3294E-04	0.0000E+00
calculation point30	2.5120E+06	8.3294E-04	0.0000E+00
calculation point31	2.5120E+06	8.3294E-04	0.0000E+00
calculation point32	2.5120E+06	8.3294E-04	0.0000E+00
calculation point33	2.5120E+06	8.3294E-04	0.0000E+00
calculation point34	2.5120E+06	8.3294E-04	0.0000E+00
calculation point35	2.5120E+06	8.3294E-04	0.0000E+00
calculation point36	2.5120E+06	8.3294E-04	0.0000E+00
calculation point37	2.5120E+06	8.3294E-04	0.0000E+00
calculation point38	2.5120E+06	8.3294E-04	0.0000E+00
calculation point39	2.5120E+06	8.3294E-04	0.0000E+00
calculation point40	2.5120E+06	8.3294E-04	0.0000E+00
calculation point41	2.5120E+06	8.3294E-04	0.0000E+00
calculation point42	2.5120E+06	8.3294E-04	0.0000E+00
calculation point43	2.5120E+06	8.3294E-04	0.0000E+00
calculation point44	2.5120E+06	8.3294E-04	0.0000E+00
calculation point45	2.5120E+06	8.3294E-04	0.0000E+00
calculation point46	2.5120E+06	8.3294E-04	0.0000E+00
calculation point47	2.5120E+06	8.3294E-04	0.0000E+00
calculation point48	2.5120E+06	8.3294E-04	0.0000E+00
calculation point49	2.5120E+06	8.3294E-04	0.0000E+00
calculation point50	2.5120E+06	8.3294E-04	0.0000E+00
calculation point51	2.5120E+06	8.3294E-04	0.0000E+00
calculation point52	2.5120E+06	8.3294E-04	0.0000E+00

pipe 2

	pressure	flow	fric. fac.
calculation point 1	2.1000E+06	4.0000E-04	0.0000E+00
calculation point 2	2.1000E+06	4.0000E-04	0.0000E+00
calculation point 3	2.1000E+06	4.0000E-04	0.0000E+00
calculation point 4	2.1000E+06	4.0000E-04	0.0000E+00
calculation point 5	2.1000E+06	4.0000E-04	0.0000E+00
calculation point 6	2.1000E+06	4.0000E-04	0.0000E+00
calculation point 7	2.1000E+06	4.0000E-04	0.0000E+00

time (s)	pi (n/m2)	po (n/m2)	qi (m3/s)	qo (m3/s)	qt (m3/s)	x (m)	xd (m/s)	pc (n/m2)	pcd
1.1089E-03	2.5120E+06	2.1000E+06	8.3294E-04	4.0000E-04	4.3294E-04	9.1090E-04	7.0861E-07	2.1000E+06	-2.5362E+03
2.0330E-03	2.5120E+06	2.1000E+06	8.3294E-04	4.0000E-04	4.3294E-04	9.1090E-04	7.4798E-07	2.1000E+06	-5.0724E+02
3.1420E-03	2.5120E+06	2.1000E+06	8.3294E-04	4.0000E-04	4.3294E-04	9.1090E-04	5.9051E-07	2.1000E+06	3.3816E+02
4.0661E-03	2.5120E+06	2.1000E+06	8.3294E-04	4.0000E-04	4.3294E-04	9.1090E-04	5.1177E-07	2.1000E+06	-1.6908E+03
5.1750E-03	2.5120E+06	2.1000E+06	8.3294E-04	4.0000E-04	4.3294E-04	9.1090E-04	5.1177E-07	2.1000E+06	5.0724E+02
6.0991E-03	2.5120E+06	2.1000E+06	8.3294E-04	4.0000E-04	4.3294E-04	9.1090E-04	4.3304E-07	2.1000E+06	-1.5217E+03
7.0232E-03	3.2775E+06	3.2122E+06	6.2740E-04	1.2604E-04	5.0136E-04	9.1717E-04	3.5321E-03	2.8659E+06	8.8704E+08
8.1322E-03	3.3791E+06	3.3266E+06	6.0011E-04	1.0146E-04	4.9865E-04	9.0235E-04	-1.4826E-02	2.9680E+06	2.8009E+07
9.0563E-03	3.6442E+06	3.6350E+06	5.2894E-04	1.7676E-05	5.1127E-04	8.9278E-04	-8.6501E-03	3.2336E+06	4.1403E+08
1.0165E-02	3.7812E+06	3.7896E+06	4.9214E-04	-1.6107E-05	5.0825E-04	8.7584E-04	-1.7467E-02	3.3724E+06	2.1976E+07
1.1089E-02	3.8806E+06	3.9005E+06	4.6546E-04	-3.8470E-05	5.0393E-04	8.6124E-04	-1.4477E-02	3.4735E+06	1.8612E+08
1.2013E-02	3.9845E+06	4.0164E+06	4.3755E-04	-6.1628E-05	4.9918E-04	8.4612E-04	-1.8724E-02	3.5795E+06	2.4855E+06
1.3122E-02	4.0247E+06	4.0556E+06	4.2676E-04	-5.9737E-05	4.8650E-04	8.2626E-04	-1.6808E-02	3.6204E+06	9.0929E+07
1.4046E-02	4.0917E+06	4.1277E+06	4.0878E-04	-6.9610E-05	4.7839E-04	8.1035E-04	-1.8498E-02	3.6909E+06	1.3072E+07
1.5155E-02	4.1080E+06	4.1387E+06	4.0440E-04	-5.9383E-05	4.6379E-04	7.9002E-04	-1.7780E-02	3.7071E+06	3.6896E+07
1.6080E-02	4.1483E+06	4.1795E+06	3.9357E-04	-6.0161E-05	4.5373E-04	7.7382E-04	-1.7749E-02	3.7502E+06	3.3448E+07
1.7004E-02	4.1549E+06	4.1805E+06	3.9182E-04	-4.9554E-05	4.4137E-04	7.5709E-04	-1.7982E-02	3.7577E+06	8.8977E+06
1.8113E-02	4.1777E+06	4.1996E+06	3.8568E-04	-4.2204E-05	4.2788E-04	7.3762E-04	-1.7339E-02	3.7824E+06	2.8067E+07
1.9037E-02	4.1844E+06	4.2011E+06	3.8389E-04	-3.2134E-05	4.1602E-04	7.2139E-04	-1.7488E-02	3.7907E+06	6.9974E+06
2.0146E-02	4.1953E+06	4.2064E+06	3.8095E-04	-2.1363E-05	4.0231E-04	7.0232E-04	-1.7004E-02	3.8034E+06	1.4930E+07
2.1070E-02	4.2018E+06	4.2078E+06	3.7923E-04	-1.1755E-05	3.9098E-04	6.8663E-04	-1.6924E-02	3.8113E+06	5.4353E+06

FIGURE 6 (cont.) RESULTS OUTPUT FILE

INTRODUCTION TO COMPONENT MODEL DOCUMENTATION

The program documentation for each component model subroutine provides all the information required to include the model in a simulation program. Also described is the mathematical basis of the model and the method of solution. The documentation specifies the format of the call statement, a list of arrays used in the subroutine which are held in named common blocks and all the user defined information which must be read in at the start of the main program. The underscored arguments in the call statement are defined by the main program and should be coded exactly as presented. The remaining arguments are user defined component numbers or switches. Named common block arrays in each subroutine must have corresponding declaration statements in the main program. Array dimensions depend on the number of components in the system and in the case of pipe data on the number of calculation points.

For example the pipe data arrays are:-
constant data PD(8,NP)
dynamic data PDYN(3,NCP,IT,NP)
where NP = number of pipes
NCP = number of calculation points
IT = time level indicator

The maximum number of calculation points has to be estimated by considering the wavespeed and timestep in the longest pipe. IT maximum can be set at any value from 2 upwards provided the necessary alterations are made in the main program. Currently the maximum value is 10. For a system of 12 pipes with an estimated maximum number of calculation points of 250 the minimum array dimensions are specified by the declaration statements.

```
COMMON/BLK1/PD(8,12)
COMMON/BLK3/PDYN(3,250,10,12)
```

The user defined information is presented as a list of array elements with their associated program variable names. The physical significance of each element is described plus the units assumed in the calculations (where applicable). Values must be read into these array elements at the start of the program. For example the user defined information for subroutine BARMAG consists of elements 1 - 14 of constant data array FCV(15,NV) and elements 1 - 5 of dynamic data array FCVD(5,IT,NV). Only initial conditions are specified in array FCVD, therefore IT = 1. The input coding for a system of 3 valves could be:-

```
READ(5,500)((FCV(I,J),I = 1,14),J = 1,3)
READ(5,500)((FCVD(I,1,J),I = 1,5),J = 1,3)
500 FORMAT (V)  -  (free format in MULTICS)
```

Some subroutines do not require user defined information, meaning that no READ statements are necessary in the main program and that all input information is available from inputs performed for other subroutines. This is usually obvious when the purpose of the subroutine is considered.

Results calculated by the subroutine are presented as output information. The array subscripts given define elements containing values of the calculated results at the current time level. Values at previous time levels may be obtained by altering the subscript IT.

The subroutine listings were produced by the Honeywell Multics system which uses lower case characters for all standard coding. However, upper case characters are used in the program desciption to distinguish between coding and ordinary text. Standard Fortran is used throughout and the subroutines may be used on MULTICS or the SYSTEM 4 at Cardiff or Exeter, no special hardware is required.

Archiving details are not included in the individual component model documentation but are available in a separate document.

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'FLUID'

Calculation of hydraulic fluid properties

Purpose

This subroutine calculates the density, the dynamic viscosity and the isentropic tangent bulk modulus of a mineral oil hydraulic fluid at a given operating temperature and pressure. The equations are set up for oils similar to Shell Tellus 25 and Lorco fourfold.

No associated subroutines

```
CALL FLUID(PM,TEMP,RHOP,AMUP,AK)
```

Input via argument list

PM	mean system pressure	N/M^2	R
TEMP	mean fluid temperature	^oC	R

Output via argument list

AK	isentropic tangent bulk modulus at system pressure	N/M^2	R
AMUP	dynamic viscosity at system pressure	Ns/M^2	R
RHOP	fluid density at system pressure	Kg/M^3	R

Program action and algorithm

The subroutine performs the straight forward calculation of a number of parameters, there is no branching or looping and the program action is best appreciated by examining the listing. The equations used are described in references 1 and 2 and are set up for mineral oil hydraulic fluids similar to Shell Tellus 27 or Lorco four-fold.

References

1 EIRICH
'Rheology Vol. 1'. ACADEMIC PRESS

2 D.G. TILLEY

'Phd THESIS'. UNIVERSITY OF BATH 1976

```
      subroutine fluid(pm,temp,rhop,amup,ak)
c
c     subroutine name  fluid
c
c     title  calculation of hydraulic fluid properties
c
c     library classification
c
c     author  c.m. skarbek-wazynski
c
c     purpose   this subroutine calculates the density,the dynamic
c               viscosity and the isentropic tangent bulk modulus for
c               a mineral oil hydraulic fluid at a given operating
c               temperature and pressure the equations are set up for
c               oils similar to shell tellus and lorco fourfold
c
c     no associated subroutines
c
c     no common block data
c
c     input information
c     input via argument list
c     pm    mean system pressure                                    n/m2
c     temp  mean fluid temperature                                  degc
c
c     output information
c     output via argument list
c     ak    isentropic tangent bulk modulus at sys pressure         n/m2
c     amup  dynamic viscosity at system pressure                    ns/m2
c     rhop  fluid density at system pressure                        kg/m3
c
c     variables (excluding i/o variables)
c     a     dow-fink density factor                                 --
c     ako   isentropic secant bulk modulus at athmos pressure       bar
c     aks   isentropic secant bulk modulus at system pressure       bar
c     amu   dynamic viscosity at athmospheric pressure              ns/m2
c     amuh  dynamic viscosity at 345 bar                            ns/m2
c     anu   kinematic viscosity at athmospheric pressure            m2/s
c     anuh  kinematic viscosity at 345 bar                          m2/s
c     b      dow-fink density factor                                --
c     const variable used to simplify calculation of amup           --
c     rho   density at athmospheric pressure                        kg/m3
c     rhoh  density at 345 bar                                      kg/m3
c
      a=4.4
      b=5.7
      rho=(870.0-0.625*(temp-20.0))
      rhop=rho*(1.0+((a*1.0e-4/6.895)*pm*1.0e-5)-(b*1.0e-7/(6.895**2.0))
     1*(pm*1.0e-5)**2.0)
      anu=(10.0**(10.0**(10.67-4.213*alog10(temp+273.0))))+0.6
      amu=anu*rho*1.0e-3
      anuh=(10.0**(10.0**(8.245-3.218*alog10(temp+273.0))))+0.6
      rhoh=rho*(1.0+((a*1.0e-4/6.895)*345.0)-(b*1.0e-7/(6.895**2.0))
     1*345.0**2.0)
```

```
      amuh=anuh*rhoh*1.0e-3
      const=((alog10(amuh))**2.0-(alog10(amu))**2.0)/345.0
      amup=(10.0**sqrt((alog10(amu))**2.0+const*(pm*1.0e-5)))
c
c     convert amup from cp to ns/m2
      amup=amup*1.0e-3
c
c     determine isentropic tangent bulk modulus of fluid
      aks=(1.85/(10.0**(0.0024*(temp-20.0)))*1.0e+4+5.6*(pm*1.0e-5))
      ako=(1.85/(10.0**(0.0024*(temp-20.0)))*1.0e+4)
      ak=(aks*(aks-(pm*1.0e-5))/ako)*1.0e+5
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'WAVSPD'

LIBRARY CALSSIFICATION

Subroutine to calculate the wavespeed in each system pipeline

Purpose

Subroutine WAVSPD is an initialisation subroutine which calculates the wavespeed in each system pipeline, and assigns the value to the appropriate element in the pipe data array PD.

No associated subroutines

CALL WAVSPD(MAXP,RHO,BM)

Common block data arrays

COMMON/BLK1/PD   pipe data (constant)

No user defined information required

OUTPUT INFORMATION (via common block)

WS	wavespeed	PD(7,I)	M/S	R

Programe action and algorithm

The wavespeed in a distensible pipeline is given by the equation

$$c = 1/\sqrt{\rho\left(\frac{1}{\beta} + \frac{d}{TE}\right)} \qquad - 1$$

The subroutine WAVSPD calculates the wavespeed in each pipe in turn by solving the above equation using data held in array PD (see program listing).

LIST OF VARIABLES USED

BM	fluid bulk modulus	β	N/M^2	R
DIA	pipe diameter	d	M	R

I	counter	–	–	I
MAXP	total number of pipes in the system	–	–	I
RHO	fluid density	ρ	Kg/M^3	R
WS	wavespeed	c	M/S	R
WT	pipe wall thickness	T	M	R
YM	Youngs modulus of pipe wall material	E	N	R

```fortran
      subroutine wavspd(maxp,rho,bm)
c
c     subroutine name   wavspd
c
c     library classification
c
c     title  subroutine to calculate the wavespeed in each
c            system pipeline
c
c     author  c.m. skarbek-wazynski
c
c     purpose   wavspd is an initialising subroutine which calculates
c               the wavespeed in each system pipeline and assigns the
c               value to the appropriate element in the pipe data array
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/pd       pipe data (constant)
c
c     input information
c     input via argument list
c     bm     fluid bulk modulus         n/m2
c     maxp   total number of pipes      --
c     rho    fluid density              kg/m3
c
c     input via common block
c     dia    pipe diameter              m          pd(1,-)
c     wt     pipe wall thickness        m          pd(3,-)
c     ym     pipe material youngs mod.  n          pd(4,-)
c
c     output information
c     output via common block
c     ws     wavespeed                  m/s        pd(7,-)
c
c     variables (excluding i/o variables)
c     i      counter                    --
c
      common /blk1/ pd(8,2)
      do 10 i=1,maxp
c     input data
      dia=pd(1,i)
      wt =pd(3,i)
      ym =pd(4,i)
c
c     calculate wavespeed
      ws=1.0/sqrt(rho*(1.0/bm+dia/(wt*ym)))
c
c     output data
      pd(7,i)=ws
   10 continue
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE CLPNT

Subroutine to set the number of calculation points in a system of pipes

Purpose

This subroutine calculates the timestep to be used for a given system simulation and sets the number of calculation points in each pipe accordingly. Where necessary the pipe wavespeeds are adjusted slightly to allow an integer number of ΔX intervals.

No associated subroutines

CALL CLPNT(NPS,DT,MAXP,FCTR)

NPS	Pipe number of shortest pipe in the system (integer)
FCTR	Factor for artificially setting the timestep (integer)

Common block data arrays

COMMON/BLK1/PD   Pipe data (constant)

No user defined information required

Output information via common block

RNCP	Number of calculation points along the pipeline	PD(6,-)	-	R
ERR	% error between true and adjusted wavespeeds	PD(9,-)	-	R
WSN	Wavespeed after adjustment	PD(7,-)	M/S	R

Program action and algorithm

The subroutine usually sets the system timestep as that timestep which gives three calculation points for the shortest pipe in the system. The user inputs the number of the shortest pipe and sets the variable FCTR = 1.0. The timestep thus calculated is the coarsest possible for that particular system. Under certain circumstances however the user may wish to employ a finer timestep in which case the variable FCTR must be set at values less than 1.0. For maximum accuracy FCTR should only be the decimal eqivalent of proper fractions which when divided by two and inverted yield an integer number (e.g. 2/3, 1/4, 2/9, are acceptable, whereas 5/6, 4/5 would be unaccurate).

The subroutine first of all calculates the system timestep based on the shortest pipe and the variable FCTR. Each pipe is considered in turn inside a do loop (do 10 I = 1,MAXP) (Figure 1). The ΔX (DX) interval for a given pipe is calculated from the wavespeed in that pipe and the timestep. It is very unlikely that ΔX will be an integer division of the pipe length as is required by the method of characteristics. Therefore the number of ΔX intervals is rounded to the nearest integer value. The ΔX interval is recalculated (DXN) and the necessary wavespeed is adjusted to produce the new value of ΔX. The percentage error between the adjusted wavespeed and the true wave-speed is calculated to enable the user to judge if the approximation is sufficiently accurate for his purposes.

The values of the new wavespeed, the error, and the number of calculation points are assigned to the pipe data array PD.

<u>LIST OF VARIABLES USED</u>

ALP	pipe length	M	R
DIFF	variable used in rounding up	-	R
DT	timestep	S	R
DX	ΔX division of pipe length	M	R
DXN	ΔX division after adjustment	M	R
ERR	% error between true and adjusted wavespeeds	-	R
FCTR	Factor for artificially setting the timestep	-	R
I	do loop counter	-	I
INPP	truncation variable	-	I
MAXP	Maximum number of pipes in the system	-	I
NPS	Pipe number of shortest pipe in system	-	I
RNCP	Number of calculation points along pipeline	-	R

RNP	Number of DXN divisions along pipeline	-	R
SPL	Shortest pipe length	M	R
WS	Wavespeed	M/S	R
WSN	Wavespeed after adjustment	M/S	R
XNP	Number of ΔX divisions of pipe length	-	R

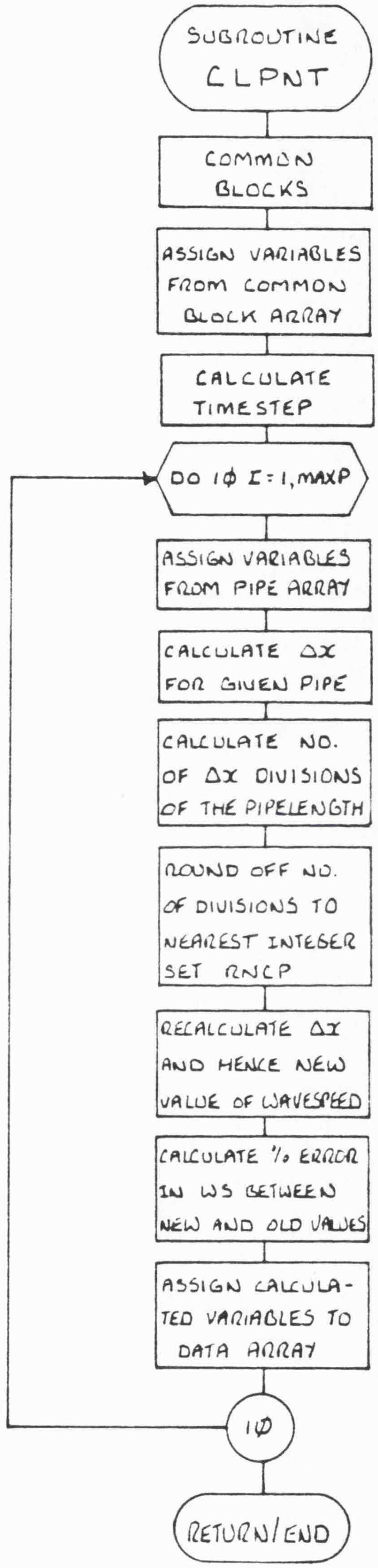

FIGURE 1 FLOWCHART FOR SUBROUTINE CLPNT

```
      subroutine clpnt( nps,dt,maxp,fctr)
c
c     subroutine name     clpnt
c
c     library classification
c
c     title   subroutine to set the number of calculation points for a
c             pipe system simulated using the method of characteristics
c
c     author  c.m. skarbek-wazynski
c
c     purpose    this subroutine calculates the timestep to be used for
c                a given system simulation and sets the number of
c                calculation points in each pipe accordingly. where
c                necessary the pipe wavespeeds are adjusted slightly to
c                allow an integer number of delta x intervals.
c
c     no associated subroutines
c
c     common blocks
c     common /blk1/ pd     pipe data (constant)
c
c     input information
c     input via argument list
c     nps      pipe number of shortest pipe
c     maxp     maximum number of pipes
c     fctr     factor for artifically setting the timestep
c
c     input via common block
c     alp      pipe length  (m)                                 pd(5,-)
c     spl      shortest pipe length (m)                         pd(5,-)
c     ws       wavespeed  (m/s)                                 pd(7,-)
c
c     output information
c     output via the argument list
c     dt       timestep (s)
c
c     output via the common block
c     rncp     number of calculation points                     pd(6,-)
c     err      percentage error in adjusted wavespeed           pd(5,-)
c     wsn      adjusted wavespeed                               pd(7,-)
c
c     variables excluding i/o variables
c     diff     variable used in rounding up
c     dx       delta x division of pipe length
c     dxn      delta x division after adjustment
c     i        do loop counter
c     inpp     truncation variable
c     rnp      number of delta x divisions after rounding
c     xnp      number of delta x divisions
c
```

```
c
      common /blk1/ pd(8,2)
c
c     shortest pipe data
      spl=pd(5, nps)
      ws =pd(7, nps)
c
c     calculate timestep
      spl=spl*fctr
      dt=spl/(2.0*ws)
c
c     calculate  delta x and wavespeed for each pipe
      do 10 i=1,maxp
c
c     pipe data
      ws=pd(7,i)
      alp=pd(5,i)
c
c     calculate delta x
      dx=dt*ws
      xnp=alp/dx
c
c     round off
      inpp=xnp
      diff=xnp-inpp
      rnp=aint(xnp)
      if(diff.lt.0.5) go to 20
      rnp=rnp+1.0
   20 continue
c
c     set number of calculation points
      rncp=rnp+1.0
c
c     recalculate deltax and wavespeed
      dxn=alp/rnp
      wsn=dxn/dt
c
c     calculate error
      err=((wsn-ws)/ws)*100.0
c
c     output
      pd(6,i)=rncp
      pd(7,i)=wsn
      pd(8,i)=err
   10 continue
      return
      end
c
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'PQIN'

Subroutine to set initial pressure and flow conditions in all system pipelines

Purpose

This subroutine is used to initialise the pipe pressure and flow data array PDYN with the steady state condition existing at the start of the system simulation.

No associated subroutines

CALL PQIN(MAXP,RHO,AVISC,IFTR,DT)
IFTR = zero friction switch (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK2/PIN	initial condition data
COMMON/BLK3/PDYN	pipe data (dynamic)

User defined information

ARRAY PIN(2,NO)

PIN(1,NO)	= SSP	steady state pressure in pipe	N/M^2	R
PIN(2,NO)	= QS	steady state flow in pipe	M^3/S	R

Output information via common block

P	pressure at calculation point	PDYN(1,-,1,I)	N/M^2	R
QS	steady state flow	PDYN(2,-,1,I)	M^3/S	R
F	friction factor at calculation point	PDYN(3,-,1,I)	-	R

Program action and algorithm

The purpose of this subroutine is to facilitate the setting of initial conditions for a system simulation. The values of pressure, flow and friction factor must be set at all the pipe calculation points in the system, this data is stored in array PDYN.

The initial steady state conditions for each pipe are expressed as a mean pressure and a steady flow. The user is required to set the switch IFCTR = 0 if the system is being modelled without the effects of pipe friction, otherwise IFCTR may be set to any integer value.

For each pipe in turn (do loop 10) the subroutine checks if IFCTR = 0 and if so the array PDYN is initialised directly with the values of steady state pressure and flow. Two further checks are carried out; one for zero pipe pressure i.e. a cavitating pipe, the other for zero pipe flow, in both cases the array PDYN is initialised accordingly. (Figure 1). Once it is determined that a pipe is being modelled with frictional effects and is operating at a finite pressure and flow the subroutine calculates the frictional pressure drop thereby establishing the pressure gradient in the pipe and an interpolation procedure is used to calculate the value of pressure at each calculation point along the length of the pipe.

Operational Status

A second version of this subroutine is available under the name PQINU. It only differs from the description above in that the steady state pressure is taken to be the pressure at the upstream end of the pipe rather than the mean pressure. Only the interpolation procedure is affected by this change.

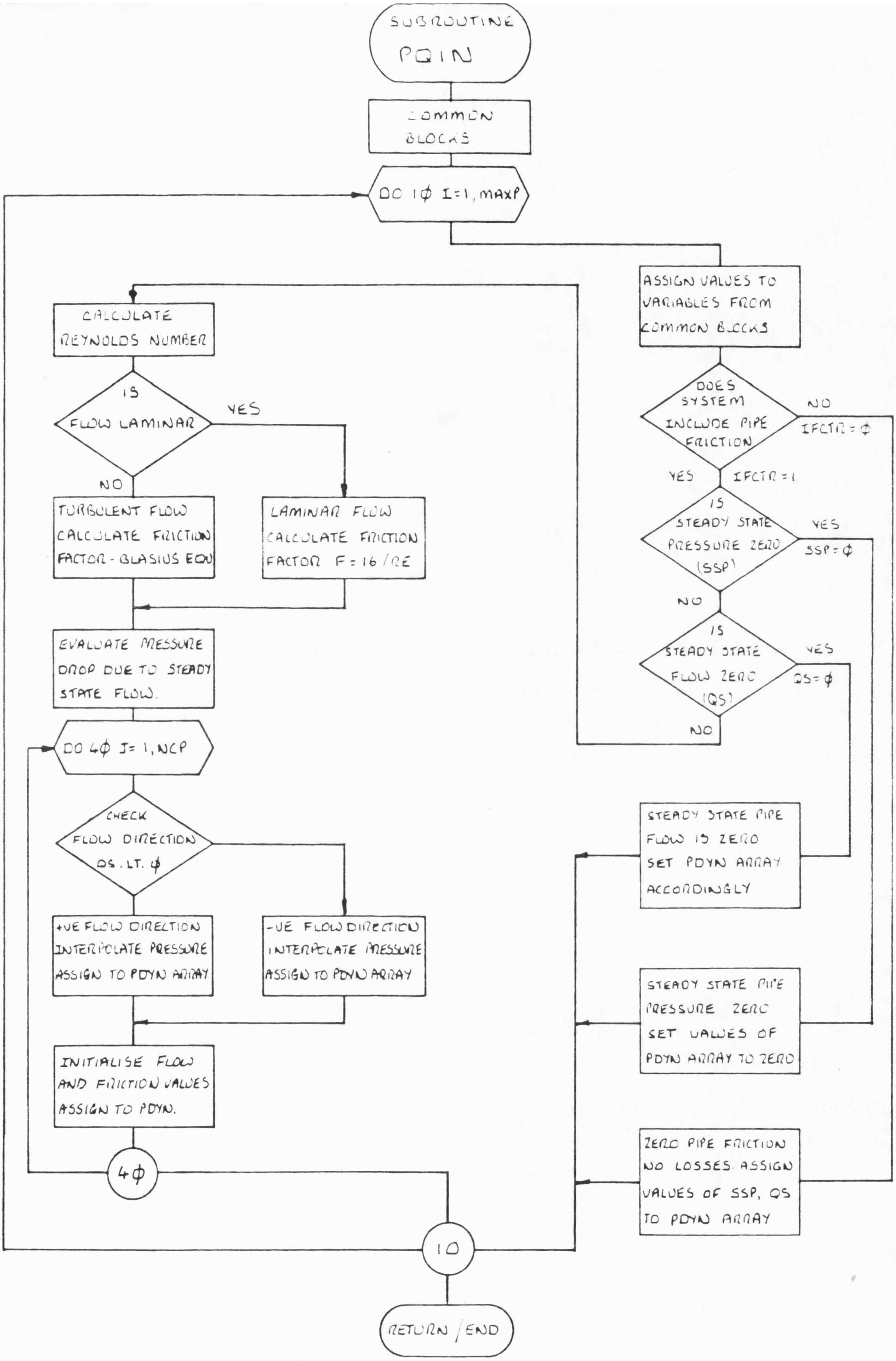

FIGURE FLOWCHART FOR SUBROUTINE PQIN

```
      subroutine pqin(maxp,rho,avisc,iftr,dt)
c
c     subroutine name  pqin
c
c     library classification
c
c     title   subroutine to set initial pressure and flow conditions
c             in all system pipelines
c
c     author  c.m. skarbek-wazynski
c
c     purpose   this subroutine is used to initialise the pipe pressure
c               and flow data array pdyn with the steady state
c               conditions existing at the start of the simulation
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/ pd        pipe data (constant)
c     common/blk2/ pin       initial condition data
c     common/blk3/ pdyn      pipe data (dynamic)
c
c     input information
c     input via argument list
c     avisc    fluid viscosity (absolute ns/m2)
c     dt       timestep (s)
c     iftr     zero friction switch
c     maxp     total number of pipes
c     rho      fluid density (kg/m3)
c
c     input via common block
c     ap       pipe area      (m2)                        pd(2,-)
c     alp      pipe length    (m)                         pd(5,-)
c     dia      pipe diameter  (m)                         pd(1,-)
c     ncp      number of calculation points               pd(6,-)
c     ws       wavespeed (m/s)                            pd(7,-)
c     qs       steady state flow     (m3/s)               pin(2,-)
c     ssp      steady state pressure (n/m2)               pin(1,-)
c
c     output information
c     output via common block
c     f        friction factor                          pdyn(3,-,-,-)
c     p        pressure  (n/m2)                         pdyn(1,-,-,-)
c     qs       steady state flow  (m3/s)                pdyn(2,-,-,-)
c
c     variables (excluding i/o variables)
c     dx       delta x division of pipe length (m)
c     i        do loop counter
c     j        do loop counter
c     k        do loop counter
c     l        do loop counter
c     mm       do loop counter
c     pdr      pressure drop  (n/m2)
c     re       reynolds number
c     x        distance along the pipe
c
```

```
      common /blk1/ pd(8,2)
      common /blk2/ pin(2,2)
      common /blk3/ pdyn(3,400,10,2)
c
c     set initial conditions for each pipe
      do 10 i=1,maxp
c
c     pipe and initial condition data
      dia=pd(1,i)
      ap=pd(2,i)
      alp=pd(5,i)
      ncp=pd(6,i)
      ws=pd(7,i)
      ssp=pin(1,i)
      qs=pin(2,i)
c
c     check for zero friction
      if(iftr.eq.0)go to 300
c
c     check for zero pressure
      if(ssp.eq.0.0)go to 70
c
c     check for zero flow
      if(qs .eq.0.0)go to 80
c
c     calculate reynolds number
      re=(abs(qs)/ap)*dia*rho/avisc
c
c     check if flow is laminar or turbulent
      if(re.le.2000.0)go to 20
c
c     flow is turbulent use blausius formula
      f=0.079*(re)**(-0.25)
      go to 30
c
c     flow is laminar
   20 f=16.0/re
   30 continue
c
c     evaluate pressure drop in pipe due to steady state flow
      pdr=4.0*f*alp*((abs(qs)/ap)**2.0)*rho/(2.0*dia)
c
c     evaluate pressure at each calculation point
      dx=ws*dt
      do 40 j=1,ncp
c
c     check flow direction
      if(qs.lt.0.0)go to 50
c
c     positive flow direction
c     interpolate on mean pressure
      x=dx*(j-1)
      p=(ssp+pdr/2.0)-pdr*x/alp
      pdyn(1,j,1,i)=p
      go to 60
```

```
c
c     negative flow direction
c     interpolate on mean pressure
   50 continue
      x=dx*(j-1)
      p=(ssp-pdr/2.0)+pdr*x/alp
      pdyn(1,j,1,i)=p
   60 continue
c
c     initialise flow data
      pdyn(2,j,1,i)=qs
c
c     initialise friction factor data
      pdyn(3,j,1,i)=f
   40 continue
      go to 10
c
c     zero pressure
c     initialise pdyn array
   70 continue
      do 100 k=1,ncp
      pdyn(1,k,1,i)=0.0
      pdyn(2,k,1,i)=0.0
      pdyn(3,k,1,i)=0.0
  100 continue
      go to 10
c
c     zero flow
c     initialise pdyn array
   80 continue
      do 200 l=1,ncp
      pdyn(1,l,1,i)=ssp
      pdyn(2,l,1,i)=0.0
      pdyn(3,l,1,i)=0.0
  200 continue
      go to 10
c
c     zero friction
c     initialise pdyn array
  300 continue
      do 310 mm=1,ncp
      pdyn(1,mm,1,i)=ssp
      pdyn(2,mm,1,i)=qs
      pdyn(3,mm,1,i)=0.0
  310 continue
   10 continue
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'PIPE'

Pipe model using the method of characteristics

Purpose

PIPE is a distributed parameter model of a pipeline carrying a liquid. The method of characteristics is used to calculate the pressure and flow at a number of internal crossections (calculation points), the effects of fluid friction are taken into account.

Associated subroutines

WAVSPD	Subroutine to calculate the wavespeed in each system pipeline
CLPNT	Subroutine to set the number of calculation points in a system of pipes
PQIN	Subroutine to set initial pressure and flow conditions in all system pipelines
FRIC	Calculation of friction factor using Trikha's method
FRICSS	Calculation of steady state friction factors
ZEROF	Zero friction model

CALL PIPE(NP,IT,ITM,RHO,DT)

NP = pipe number (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)

User defined information

Array PD(8,NP)

PD(1,NP)	DIA	Pipe diameter	M	R
PD(2,NP)	AP	pipe area	M^2	R
PD(3,NP)	WT	pipe wall thickness	M	R
PD(4,NP)	YM	Youngs modulus of pipe wall material	N	R

PD(5,NP)	ALP	pipe length	M	R
PD(7,NP)	WS	wavespeed	M/S	R

The Youngs modulus of the pipe wall material and the pipe wall thickness must be specified if subroutine WAVSPD is used, otherwise they may be omitted and wavespeed specified directly. The array PDYN must be initialised, this is most conveniently done by calling subroutine PQIN.

Output information via common block

PNEW	new value of pressure	PDYN(1,K,IT,NP)	N/M^2	R
QNEW	new value of flow	PDYN(2,K,IT,NP)	M^3/S	R

Program action and algorithm

Mathematical model

The basic equations for the method of characteristics, when expressed in finite difference form are:-

$$\frac{1}{\rho C}(Pp-Pr) + (vp - vr) + \frac{2Frvr|vr|\Delta t}{d} = 0 \qquad 1$$

$$-\frac{1}{\rho C}(Pp - Ps) + (vp - vs) + \frac{2Fsvs|vs|\Delta t}{d} = 0 \qquad 2$$

$$\frac{\Delta X}{\Delta t} = +C \qquad 3$$

$$\frac{\Delta X}{\Delta t} = -C \qquad 4$$

Equations 1 and 2 are only valid on the characteristic lines defined by equations 3 and 4 respectively. (Ref 1)(Figure 1). At the intersections of two characteristics lines equations 1 and 2 apply simultaneously and may be solved for the two unknown Pp and vp.

$$Pp = \frac{(Ps + Pr)}{2} + \frac{\rho C}{2}(Vr - vs) + \Delta t.\rho.C\,(Fs\,vs|vs| - Fr.vr.|vr|) \qquad 5$$

$$vp = -2Fr\,vr|vr|\Delta t + vr - \frac{1}{\rho C}(Pp - Pr) \qquad 6$$

The complete modelling of a pipeline involves solving equations 5 and 6 at all internal calculation points.

Computing procedure

The pressure, flow and friction factor data is copied from array PDYN to temporary one dimensional arrays, this is to speed up execution and help keep the program coding readable. Equations 5 and 6 are repeatedly solved inside do loop 20 to calculate new values of pressure and flow at all internal calculation points. The equations are set up using previous values of pressure, flow velocity and friction factor (Pr, Ps, Vr, vs, Fr, Fs) these values are stored in the temporary data arrays and counters J and K are set to ensure data from the correct calculation point is accessed. A cavitation check is performed to ensure pressure values calculated are not negative and the new values of pressure and flow are assigned to array PDYN. (Figure 2.)

LIST OF VARIABLES USED

AP	pipe area	-	M^2	R
C	Wavespeed	C	M/S	R
DIA	pipe diameter	d	M	R
DT	timestep	Δt	S	R
FF	friction factor (forward characteristic)	Fr	-	R
FR	Temporary array for friction factor data	-	-	R
FS	friction factor (backward characteristics)	Fs	-	R
I	do loop counter	-	-	I
IT	time level indicator (time = t)	-	-	I
ITM	time level indicator (time = t-Δt)	-	-	I
J	counter	-	-	I
K	counter	-	-	I
M	counter	-	-	I
NCP	number of calculation points	-	-	I
NP	pipe number	-	-	I

PF	pressure (forward characteristic)	Pr	N/M^2	R
PNEW	new value of pressure	Pp	N/M^2	R
PR	temporary array for pressure data	-	N/M^2	R
PS	pressure (backward characteristic)	Ps	N/M^2	R
QNEW	new value of flow	-	M^3/S	R
QR	flow value used in temporary array assignment	-	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
VF	flow velocity (forward characteristic)	vr	M/S	R
VR	temporary array for flow velocity data	-	M/S	R
VS	flow velocity (backward characteristic)	vs	M/S	R
VNEW	new value of flow velocity	v_p	M/S	R

References

1 J.A. FOX
'Hydraulic analysis of unsteady flow in pipe networks' pg 80
MACMILLAN PRESS LIMITED

FIGURES

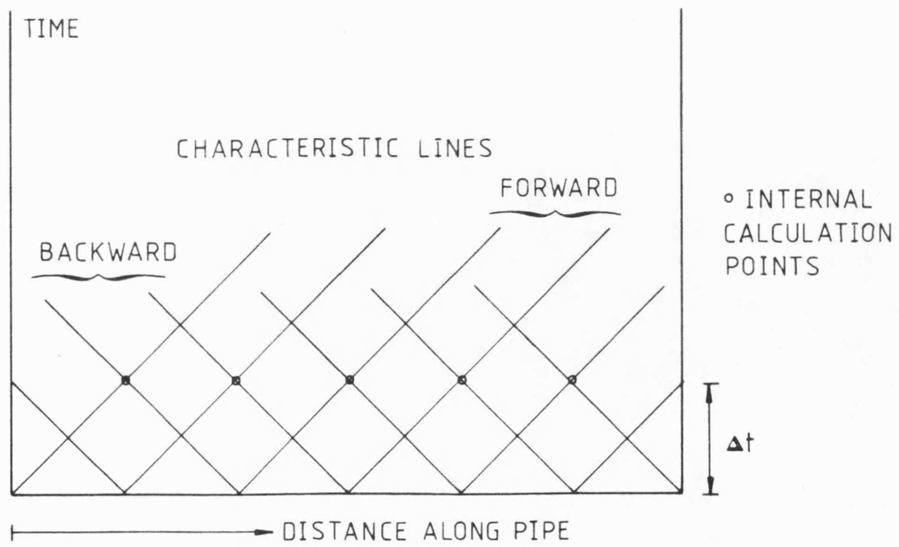

FIGURE 1 METHOD OF CHARACTERISTICS GRID

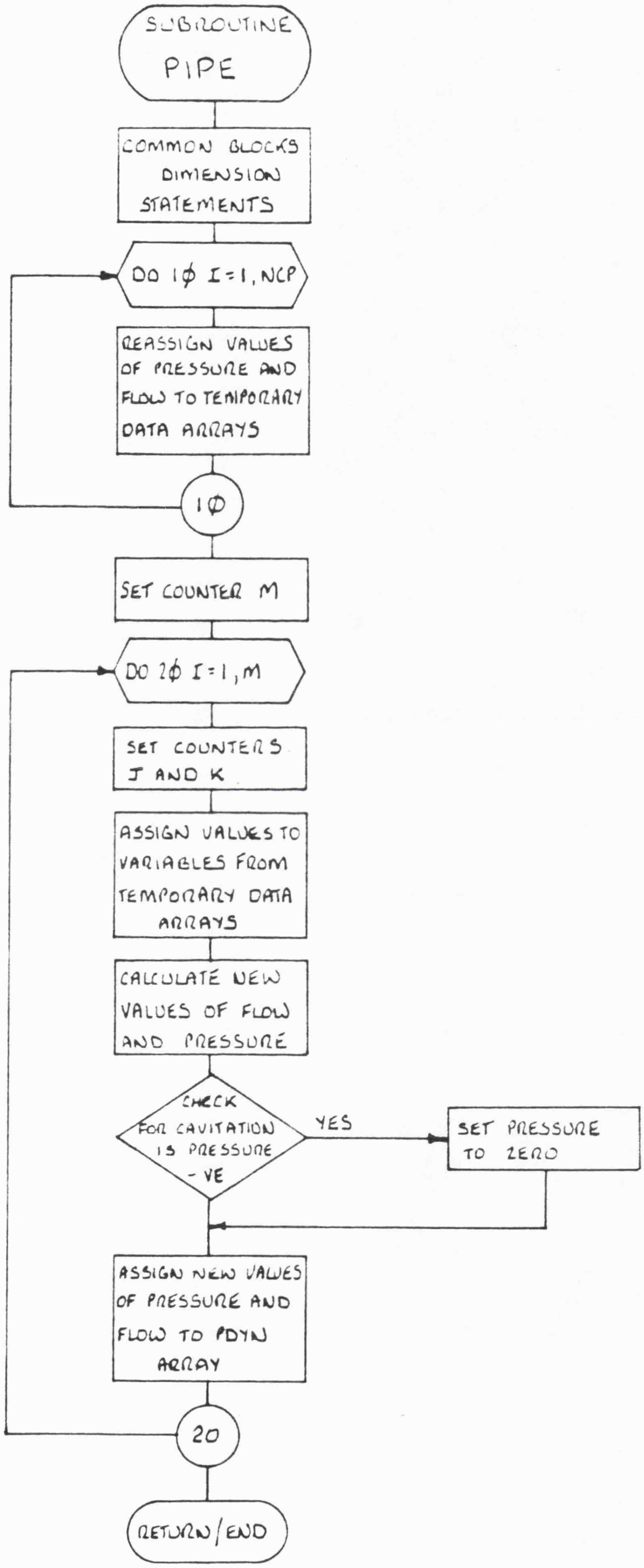

FIGURE 2 FLOWCHART FOR PIPE SUBROUTINE

```
      subroutine pipe(np,it,itm,rho,dt)
c
c     subroutine name   pipe
c
c     library classification
c
c     title    pipe model using the method of characteristics
c
c     author  c.m. skarbek-wazynski
c
c     purpose  pipe is a distributed parameter model of a
c              pipeline carrying a liquid. the method of characteristics
c              is used to calculate the pressure and flow at a number of
c              internal crossections (calculation points), the effects
c              of fluid friction are taken into account.
c
c     associated subroutines
c     wavspd  subr. to eval. wavespeed in each pipe
c     clpnt   subr. to set no. of calc. points in system
c     pqin    subr. to set initial conds. in all pipes
c     fric    calc. of friction factor (trikhas method)
c     fricss  calc. of steady state friction factor
c     zerof   zero friction model
c
c     common blocks
c     common/blk1/ pd       pipe data (constant)
c     common/blk3/pdyn      pipe data (dynamic)
c
c     input information
c     input via argument list
c     dt       timestep  (s)
c     it       time level indicator (time=t)
c     itm      time level indicator (time=t-delta t)
c     np       pipe number
c     rho      fluid density (kg/m3)
c
c     input via common block
c     ap       pipe area     (m2)                               pd(2,-)
c     dia      pipe diameter (m)                                pd(1,-)
c     ncp      number of calculation points                     pd(6,-)
c     c        wavespeed (m/s)                                  pd(7,-)
c
c     output information
c     output via common block
c     pnew     new value of pressure (n/m2)                     pdyn(1,-,-,-)
c     qnew     new value of flow     (m3/s)                     pdyn(2,-,-,-)
c
c     variables (excluding i/o variables)
c     ff       friction factor (forward characteristic)
c     fr       temporary array for friction factor data
c     fs       friction factor (backward characteristic)
c     i        do loop counter
c     j        counter
c     k        counter
c     m        counter
c     pf       pressure (forward characteristic) (n/m2)
c     pr       temporary array for pressure data
```

```
c     ps        pressure(backward characteristic)(n/m2)
c     qr        flow value for temporary array assignment
c     vf        flow velocity (forward characteristic) (m/s)
c     vr        temporary array for flow velocity data (m/s)
c     vs        flow velocity (backward characteric) (m/s)
c     vnew      new value of flow velocity (m/s)
c
      common /blk1/pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
      dimension pr(400),vr(400),fr(400)
c
c     pipe data
      dia=pd(1,np)
      ap =pd(2,np)
      c  =pd(7,np)
      ncp=pd(6,np)
c
c     re-assign pressure and flow values to temporary arrays
      do 10 i=1,ncp
      qr=pdyn(2,i,itm,np)
      pr(i)=pdyn(1,i,itm,np)
      vr(i)=qr/ap
      fr(i)=pdyn(3,i,itm,np)
   10 continue
c
      m=ncp-2
      do 20 i=1,m
c
c     set counters
      j=i+2
      k=i+1
c
c     set values of pressure,flow velocity and friction factor
      vf=vr(i)
      vs=vr(j)
      pf=pr(i)
      ps=pr(j)
      ff=fr(i)
      fs=fr(j)
c
c     calculate new values of pressure and flow
      pnew=(ps+pf)/2.0+rho*c*(vf-vs)/2.0+rho*c*dt*
     1(fs*vs*abs(vs)-ff*vf*abs(vf))/dia
      vnew=-2.0*ff*vf*abs(vf)*dt/dia+vf-(pnew-pf)/(rho*c)
c
c     cavitation check
      if(pnew.lt.0.0)pnew=0.0
c
c     output data
      qnew=vnew*ap
      pdyn(1,k,it,np)=pnew
      pdyn(2,k,it,np)=qnew
   20 continue
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'ZEROF'

Zero friction model

Purpose

This subroutine sets the friction factor at all calculation points to zero.

No associated subroutines

CALL ZEROF(MAXP,IT)

Common block data arrays

COMMON/BLK1/PD	pipe data array (constant)
COMMON/BLK3/PDYN	pipe data array (dynamic)

No user defined information required

Output information

0.0 constant value (zero friction factor) PDYN(3,NCP,IT,MAXP) - R

Program action and algorithm

The subroutine treats each pipe in turn (do loop 10) setting the friction factor to zero at each calculation point by assigning the value 0.0 to the relevant elements in array PDYN.

LIST OF VARIABLES USED

I	do loop counter	-	I
IT	time level indicator time = t	-	I
K	do loop counter	-	I
MAXP	total number of pipes	-	I
NCP	number of calculation points	-	I

```
      subroutine zerof(maxp,it)
c
c     subroutine name zerof
c
c     library classification
c
c     title  zero friction model
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine sets the friction at all pipe
c              calculation points to zero
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/pd      pipe data array (constant)
c     common/blk3/pdyn    pipe data array (dynamic)
c
c     input information
c     input via argument list
c     it    time level indicator          --
c     maxp  total number of pipes         --
c
c     input via common block
c     ncp   number of calculation points   --     pd(6,-)
c
c     output information
c     output via common block
c     0.0    zero friction factor          --     pdyn(3,-,-,-)
c
c     variables (excluding i/o variables)
c     i      counter                       --
c     k      counter                       --
c
      common /blk1/ pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
c
      do 10 i=1,maxp
      ncp=pd(6,i)
      do 20 k=1,ncp
      pdyn(3,k,it,i)=0.0
   20 continue
   10 continue
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'FRICSS'

Calculation of steady state friction factors

Purpose

This subroutine calculates the steady state friction factors which apply at all pipe calculation points, for all the pipes in a given system.

No associated subroutines

CALL FRICSS(MAXP,IT,RHO,AVISC)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)

No user defined information required

Output information (via common block)

F	friction factor (BRITISH)	PDYN(3,NCP,IT,MAXP)	R

Program action and algorithm

The calculation of friction factor is based on two standard steady state equations.

Laminar flow $F = 16/Re$ 1

Turbulent flow $F = 0.079\ (Re)^{-0.25}$ (Blasius formula) 2

Input data is assigned to program variables from the common block data arrays. To prevent floating point over flows, a check is performed to detect very small flows and consequently set the friction factor to zero. The Reynolds number is calculated; for values less than 2000 laminar flow is assumed and equation 1 is used, otherwise equation 2 is used. The calculated value is assigned to data array PDYN. All system pipelines are handled by one call to this subroutine.

LIST OF VARIABLES USED

AP	pipe area	-	M^2	R
AVISC	fluid viscosity (absolute)	-	Ns/M^2	R
DIA	pipe diameter	-	M	R
F	friction factor (British)	f	-	R
I	do loop counter	-	-	I
IT	time level indicator (time = t)	-	-	I
K	do loop counter	-	-	I
MAXP	total number of pipes	-	-	I
NCP	number of calculation points	-	-	I
QN	flow	-	M^3/S	R
RE	Reynolds number	Re	-	R
RHO	fluid density	-	Kg/M^3	R
VN	flow velocity	-	M/S	R

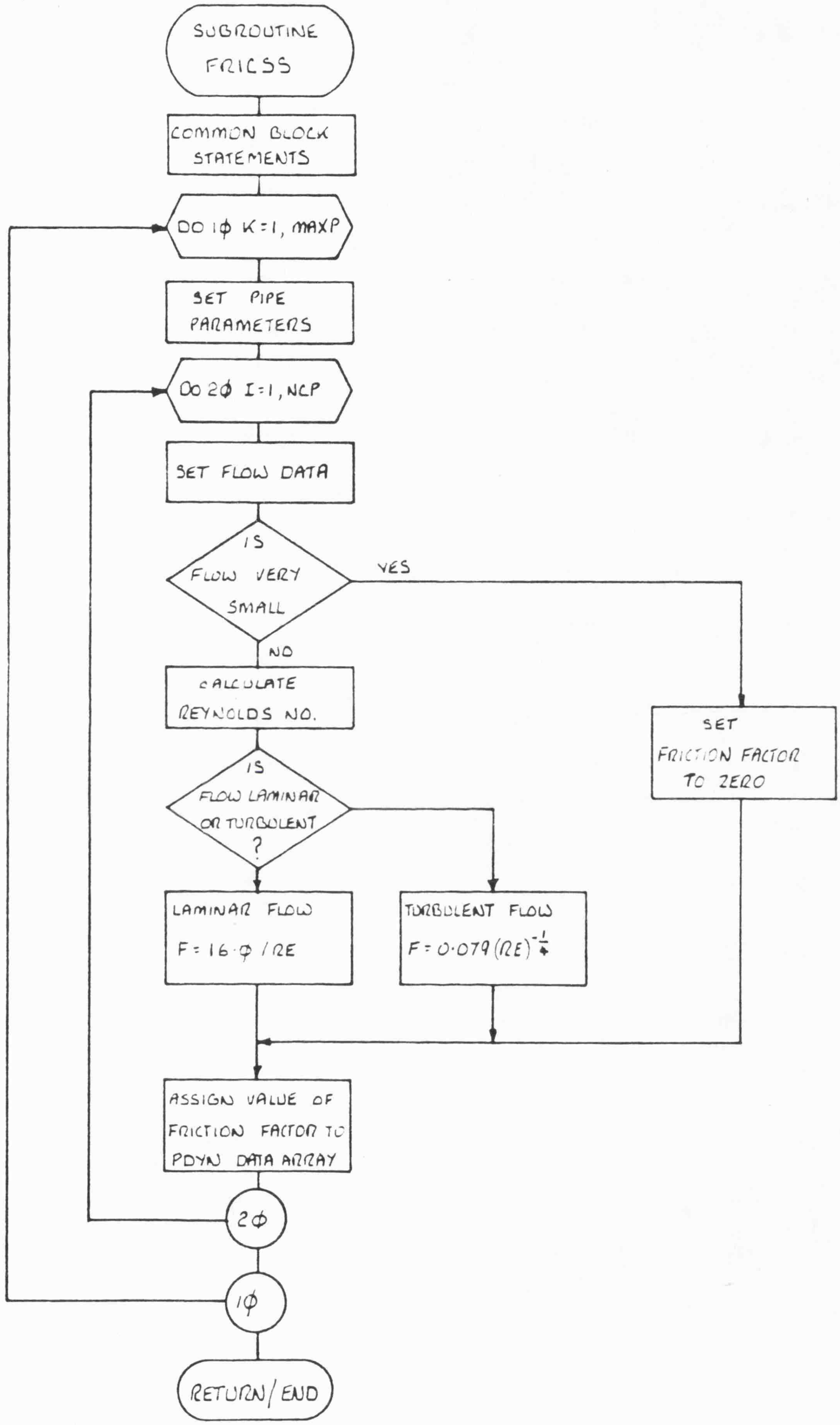

FIGURE 1 FLOWCHART FOR SUBROUTINE 'FRICSS'

```
      subroutine fricss(maxp,it,rho,avisc)
c
c     subroutine name   fricss
c
c     library classification
c
c     title   calculation of steady state friction factors
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine calculates the steady state friction
c              factors which apply at all pipe calculation points for
c              all the pipes in a given system
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/pd       pipe data (constant)
c     common/blk3/pdyn     pipe data (dynamic)
c
c     input information
c     input via argument list
c     avisc  fluid viscosity (absolute)    ns/m2
c     it     time level indicator          --
c     maxp   total number of pipes         --
c     rho    fluid density                 kg/m3
c
c     input via common block
c     ap     pipe area                     m2      pd(2,-)
c     dia    pipe diameter                 m       pd(1,-)
c     ncp    number of calculation points  --      pd(6,-)
c     qn     flow                          m3/s    pdyn(2,-,-,-)
c
c     output information
c     output via common block
c     f      friction factor (british)     --      pdyn(3,-,-,-)
c
c     variables (excluding i/o variables)
c     i      do loop counter               --
c     k      do loop counter               --
c     re     reynolds number               --
c     vn     flow velocity                 m/s
c
      common /blk1/ pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
c
c     loop round for each pipe in the system
      do 10 k=1,maxp
      ncp=pd(6,k)
      dia=pd(1,k)
      ap =pd(2,k)
c
c     loop round for each calculation point
      do 20 i=1,ncp
      qn=pdyn(2,i,it,k)
      vn=qn/ap
```

```
c     check for small flows
      if(abs(qn).le.1.0e-30)go to 100
c
c     calculate reynolds number
      re=abs(vn)*dia*rho/avisc
c     check for laminar flow
      if(re.ge.2000.0)go to 110
c
c     flow is laminar
      f=16.0/re
      go to 120
c
c     flow is turbulent use blasius formula
  110 f=0.079*re**(-0.25)
      go to 120
c
c     zero flow or very small flow
  100 f=0.0
c
c     assign value of friction factor to data array
  120 pdyn(3,i,it,k)=f
c
   20 continue
   10 continue
c
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'FRIC'

Calculation of friction factors using Trikha's method

Purpose

This subroutine calculates the frequency dependent friction factors for unsteady laminar flow using Trikha's method. The friction factor is set at all pipe calculation points at each timestep of the system simulation. When the flow is not laminar a steady state friction factor is calculated using the Blasius formula.

Associated subroutines

BLOCK DATA SUBPROGRAM - Initialise array TWF to zero
PIPE Pipe model using the method of characteristics

CALL FRIC(MAXP,DT,IT,ITM,RHO,AVISC)

Common block data arrays

COMMON/BLK1/PD	pipe data array (constant)
COMMON/BLK3/PDYN	pipe data array (dynamic)
COMMON/BLK27/TWF	Trikha's weighting factors

No user defined information required

Output information via common block

F	friction factor (British)	PDYN(3,I,IT,K)	-	R

Program action and algorithm

Mathematical model

In 1968 W. Zielke (Ref. 1) developed a method for simulating frequency dependent friction in unsteady laminar flow. The technique was simplified and presented in a more practical form by A.K. TRIKHA in 1975 (Ref. 2).

The equations used in this subroutine are those developed by Trikha. The pressure drop per unit length is given by:-

$$f(t) = \underbrace{\frac{8\rho\nu}{r^2} v(t)}_{\text{steady state component}} + \underbrace{\frac{4\rho\nu}{r^2} (y_1 + y_2 + y_3)}_{\text{transient component}} \qquad 1$$

$\nu = \frac{\mu}{\rho}$ (Kinematic viscosity)

where y_1, y_2 and y_3 are weighting factors which are determined by the previous flow history and may be calculated by the following equation.

$$y_i(t) = y_i(t - \Delta t)e^{-ni(\nu r^2)\Delta t} + Mi(v(t) - v(t - \Delta t)) \qquad 2$$

where $i = 1, 2, 3$

$M_1 = 40.0 \quad n_1 = 8000$

$M_2 = 8.1 \quad n_2 = 200$

$M_3 = 1.0 \quad n_3 = 26.4$

$y_i(t)$ represents the current values of the weighting factors. $y_i(t - \Delta t)$ is the value at the previous timestep. Similarly $v(t)$ is the current flow velocity and $v(t - \Delta t)$ is the previous value. Thus stored values of weighting factors and flow velocity at all pipe calculation points can be used in equation 2 to calculate the current weighting factors which in turn are used to calculate the pressure drop/unit length $f(t)$ in equation 1 which applies at a particular calculation point.

The friction factor used in the method of characteristics is based on the D'Arcy equation:-

pressure drop/unit length $f(t) = \frac{4f}{d} \cdot \frac{1}{2} \cdot \rho v^2$

the friction factor is given by

$$f = \frac{2f(t)d}{4v^2\rho} \qquad 3$$

Flow is assumed to remain laminar up to a Reynold's number of 2000. Once this is exceeded Trikha's equations no longer apply and the friction factor is calculated from the Blasius formula for steady state turbulent flow.

Computing procedure

One call to subroutine FRIC at each timestep of the solution evaluates the friction factor at all pipe calculation points in a given system. The calculation is performed inside two do loops (Figure 1). Each pipe is considered in turn (do loop 30) flow conditions are examined at each calculation point (do loop 10) and the friction factor is evaluated accordingly.

Inside do loop 10 a check is made on the flow value, if it is fcund to be zero the program branches and the friction factor is set to zero. For all finite flow values the Reynold's number is calculated and if it is greater than 2000 a steady state turbulent flow friction factor is calculated using the Blasius formula. For laminar flows the current values of the weighting factors are calculated from equation 2 using previous values stored in array TWF and a previous value of flow from data array PDYN. The pressure drop per unit length is then calculated (equation 1), but before evaluating the friction factor (equation 3) a check is made on the expression 2.0*VN*VN*RHO if this is less than 1.0×10^{-50} the friction factor is set to zero. The check is directly analagous to a check for a very small flow and avoids problems with floating point overflow where dividing by a very small number. The resultant value of friction factor is assigned to data array PDYN.

References

1 W. ZIELKE
Frequency-dependent friction in transient pipe flow
TRANS ASME J. BASIC ENG MARCH 1968

2 A.K. TRIKHA
An efficient method for simulating frequency dependent friction in transient liquid flow
TRANS ASME J. FLUIDS ENG MARCH 1975

LIST OF VARIABLES USED

AP	pipe area	-	M^2	R
AVISC	fluid viscosity (absolute)	μ	Ns/M^2	R
DIA	pipe diameter	d	M	R
DT	Timestep	Δt	S	R

F	Friction factor (British)	f	-	R
FL	Frictional pressure drop per unit length	f(t)	N/M^3	R
I	Do loop counter	-	-	I
IT	time level indicator (time = t)	-	-	I
ITM	time level indicator (time = t - Δt)	-	-	I
J	Do loop counter	-	-	I
K	Do loop counter	-	-	I
MAXP	total number of pipes in system	-	-	I
NCP	number of calculation points	-	-	I
Q	flow at time level ITM	-	M^3/S	R
QN	flow at time level IT	-	M^3/S	R
R	radius of pipe	r	M	R
RE	Reynold's number	Re	-	R
RHO	fluid density	ρ	Kg/M^3	R
RM	array containing m coefficients of equation 2	m_i	-	R
RN	array containing n coefficients of equation 2	n_i	-	R
RR	assigned variable for m coefficients in do loop	m_i	-	R
RS	assigned variable for n coefficients in do loop	n_i	-	R
SIGNV	sign of flow velocity term	-	-	R
V	flow velocity at time level ITM	$v(t - \Delta t)$	M/S	R
VN	flow velocity at time level IT	v(t)	M/S	R
YN	updated value of Trikha's weighting factor	y_i	-	R
YO	old value of Trikha's weighting factor	y_i	-	R
Y1	Trikha's weighting factor	y_i	-	R
Y2	Trikha's weighting factor	y_i	-	R
Y3	Trikha's weighting factor	y_i	-	R

```
      subroutine fric(maxp,dt,it,itm,rho,avisc)
c
c     subroutine name   fric
c
c     library classification
c
c     title   calculation of friction factor using trikhas method
c
c     author  c.m. skarbek-wazynski
c
c     purpose   this subroutine calculates the frequency dependent
c               friction factor for unsteady laminar flow using
c               trikhas method. the friction factor is set at all
c               pipe calculation points at each timestep of the
c               system simulation. when the flow is not laminar a
c               steady state friction factor is calculated using
c               the blasius formula
c
c     associated subroutines
c     block data subprogram  -initialise array twf to zero
c     pipe  -pipe model using the method of characteristics
c
c     common blocks
c     common/blk1/ pd     pipe data (constant)
c     common/blk3/ pdyn   pipe data (dynamic)
c     common/blk27/twf    trikhas weighting factors
c
c     input information
c     input via argument list
c     avisc    fluid viscosity (absolute ns/m2)
c     dt       timestep (s)
c     it       timelevel indicator
c     itm      timelevel indicator
c     maxp     total number of pipes
c     rho      fluid density  (kg/m3)
c
c     input via common block
c     ap  pipe area                   m2        pd(2,-)
c     dia pipe diameter               m         pd(1,-)
c     ncp number of calc points       -         pd(6,-)
c     q   flow at time level itm      m3/s      pdyn(2,-,-,-)
c     qn  flow at time level it       m3/s      pdyn(2,-,-,-)
c     yo  old value of weighting factor --      twf(j,i,k)
c     y1  current weighting factor    --        twf(1,-,-)
c     y2  current weighting factor    --        twf(2,-,-)
c     y3  current weighting factor    --        twf(3,-,-)
c
c     output information
c     output via common block
c     f   friction factor (british)   --        pdyn(3,-,-,-)
c
```

```
c     variables (excluding i/o variables)
c     fl        frictional pressure drop per unit length (n/m3)
c     i         do loop counter
c     j         do loop counter
c     k         do loop counter
c     r         pipe radius (m)
c     re        reynolds number
c     rm        array containing m coeffs of trikhas equation
c     rn        array containing n coeffs  of trikhas equation
c     rr        assigned variable for m coeffs inside do loop
c     signv     sign of flow velocity term
c     rs        assigned variable for n coeffs inside do loop
c     v         flow velocity at time level itm
c     vn        flow velocity at time level it
c     yn        updated value of trikhas weighting factor
c
      common /blk1/ pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
      common /blk27/ twf(3,400,2)
      dimension rm(3),rn(3)
c
c     initialise rm and rn arrays
      data rm/40.0,8.1,1.0/
      data rn/8000.0,200.0,26.4/
c
      do 30 k=1,maxp
c
c     pipe parameters
      ncp=pd(6,k)
      dia=pd(1,k)
      ap =pd(2,k)
      r  =dia/2.0
c
      do 10 i=1,ncp
c
c     evaluate friction factor at each calculation point
c     current value of flow
      qn=pdyn(2,i,it,k)
c
c     check for zero flow
      if(qn.eq.0.0)go to 500
c
c     calculate reynolds number
      vn=qn/ap
      re=abs(vn)*dia*rho/avisc
c
```

```
c     check whether flow is laminar or turbulent
      if(re.ge.2000.0)go to 100
c
c     flow is laminar use trikha's equations
c     old value of flow
      q =pdyn(2,i,itm,k)
      v=q/ap
c     calculate new weighting factors
      do 20 j=1,3
      rr=rm(j)
      rs=rn(j)
      yo=twf(j,i,k)
      yn=yo*(exp(-rs*avisc*dt/(rho*r*r)))+rr*(vn-v)
      twf(j,i,k)=yn
   20 continue
c
c     calculate frictional pressure drop
      y1=twf(1,i,k)
      y2=twf(2,i,k)
      y3=twf(3,i,k)
      signv=abs(vn)/vn
      fl=(4.0*avisc/(r*r))*(2.0*abs(vn)+(y1+y2+y3)*signv)
c
c     check for very small flows to prevent floating
c     point overflow
      if((2.0*vn*vn*rho).lt.1.0e-30)go to 500
c
c     calculate friction factor
      f=fl*dia/(2.0*vn*vn*rho)
      go to 700
c
c     for turbulent flow use blasius formula
  100 f=0.079*re**(-0.25)
      go to 700
c
c     zero flow (or flow is very small)
  500 f=0.0
c
c     assign value of friction factor to the pdyn data array
  700 continue
      pdyn(3,i,it,k)=f
   10 continue
   30 continue
      return
      end
```

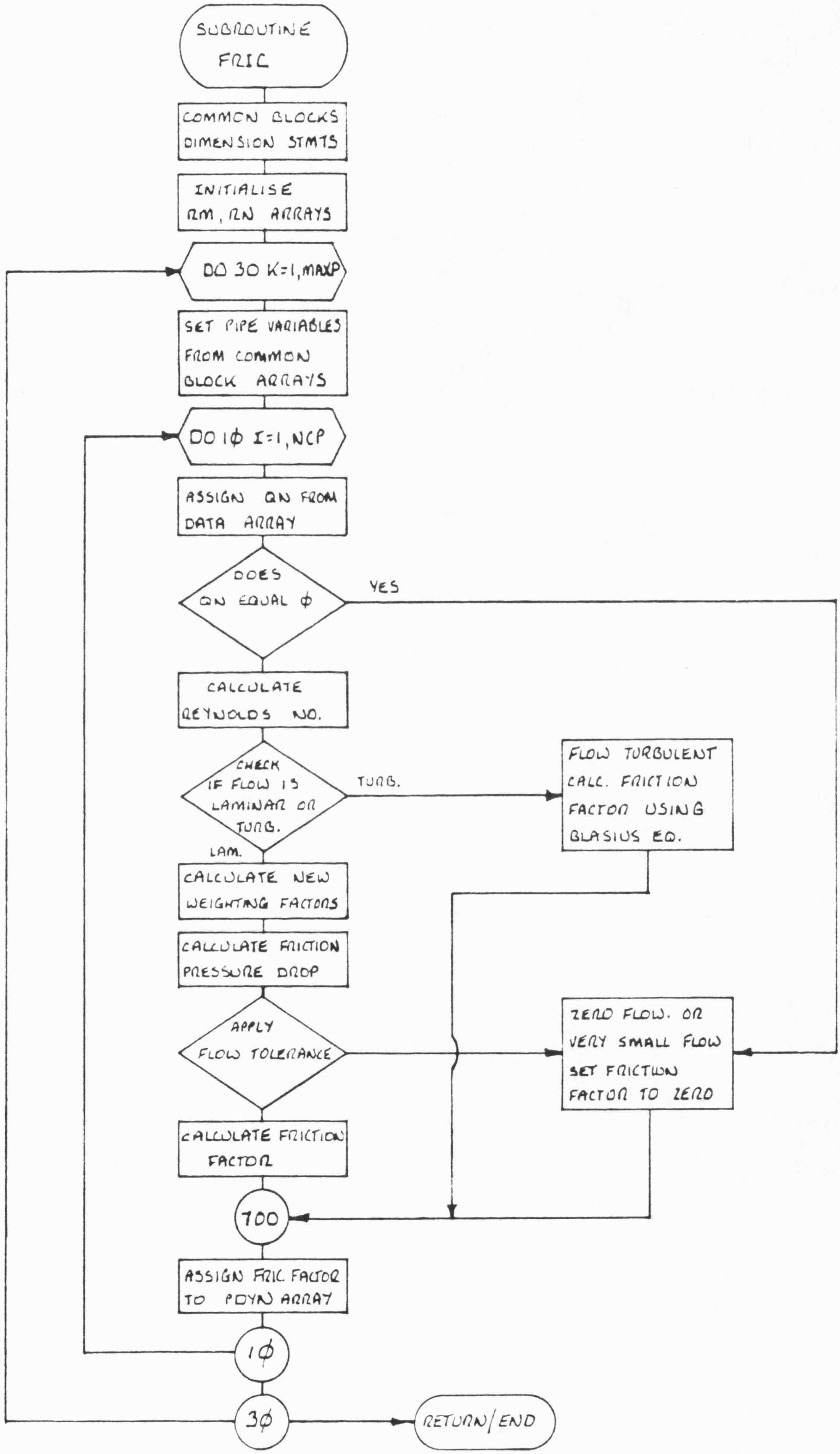

FIGURE FLOWCHART FOR SUBROUTINE FRIC

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'PUMP'

Method of characteristics model of a hydrostatic pump

Purpose

Subroutine PUMP is a steady state model of a hydrostatic pump. All internal dynamics are neglected and the pump operating speed is determined by the associated prime mover model. Pump pressures and flows depend on the interaction with the rest of the system via the inlet and outlet pipelines which are modelled using the method of characteristics.

The model is capable of a ramp change in swash setting from zero to full swash. The start time and duration of the ramp are specified by the user.

Associated subroutines

CONST	Constant speed prime mover model
ENGINE	Governed diesel engine model

CALL PUMP(IT,ITM,DT,RHO,TIME,NC)

NO = pump component number

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK6/PMP	pump data (constant)
COMMON/BLK7/PMPD	pump data (dynamic)
COMMON/BLK26/SWSH	swash setting parameters

User defined information

ARRAY PMP(10,NO)

PMP(1,NO) = DP	pump displacement	M^3/rad	R
PMP(2,NO) = KLP	slip loss coefficient	$(M^3/S)/(N/M^2)$	R
PMP(3,NO) = KTP	slip loss coefficient	$(M^3/S)/(N/M^2)$	R
PMP(4,NO) = KVP	viscous friction coefficient	Nm/(rad/sec)	R

PMP(5,NO)	= IP	pump inertia	KgM^2	R
PMP(6,NO)	= IPN	inlet pipe number	-	I
PMP(7,NO)	= OPN	outlet pipe number	-	I
PMP(8,NO)	= PCD	case drain pressure	N/M^2	R
PMP(9,NO)	= NPRM	prime mover number	-	I
PMP(10,NO)	= NTYP	prime mover selection switch	-	I

ARRAY PMPD(3,IT,NO)

PMPD(1,1,NO)	= WP2	pump speed time = t	rad/sec	R
PMPD(2,1,NO)	= TP2	pump shaft torque time = t	Nm	R
PMPD(3,1,NO)	= XP	pump swash	-	R

Note: the user is only required to specify initial values, subsequent values are calculated by the program.

ARRAY SWSH(2)

SWSH(1)	= T1	starting time for swash change	S	R
SWSH(2)	= T2	finishing time for swash change	S	R

The prime mover selection switch NTYP must be set to 1 if a constant speed prime mover model is required (see subroutine CONST). Alternatively set NTYP = 2 if a governed diesel engine model is required (see subroutine ENGINE).

OUTPUT INFORMATION via the common block

PI	pressure at pump inlet, time = t	PDYN(1,NCPI,IT,NO)	N/M^2	R
PD	pressure at pump outlet, time = t	PDYN(1,1,IT,NO)	N/M^2	R
VI*API	volumetric flow, pump inlet, time = t	PDYN(2,NCPI,IT,NO)	M^3/S	R
VO*APO	volumetric flow, pump outlet, time = t	PDYN(2,1,IT,NO)	M^3/S	R
TP2	pump shaft torque, time = t	PMPD(2,IT,NO)	Nm	R

INBUILT ERROR MESSAGES

'PRIME MOVER NUMBER INCORRECT' - prime mover number specified incorrectly call to prime mover model cannot be made. Fatal error condition which causes the program to stop.

Program action and algorithm

Mathematical model

The pump is treated as a steady state flow source with various internal leakage paths, compressibility effects within the pump are ignored. (Figure 1a.) The inlet and outlet pipelines are modelled using the method of characteristics, case drain pipe dynamics are assumed to be negligible. With the flow directions as specified in Figure 1a the pipe characteristic lines are shown on the time distance plane in Figure 1b .

The pump equations are derived by considering continuity, and a simple torque balance on the pump shaft:-

Flow into pump

$$Q_{IN} = A_{pI}v_I = W_pD_pX_p - K_{LP}(P_o - P_I) + K_{TP}(P_I - P_{CD}) \quad 1$$

Flow out of pump

$$Q_{OUT} = A_{po}v_o = W_pD_pX_p - K_{LP}(P_o - P_I) - K_{TP}(P_o - P_{CD}) \quad 2$$

Shaft torque

$$T_p = (P_o - P_I)D_pX_p + K_{VP}W_p \quad 3$$

The pipe equations are:-

Inlet pipeline

$$\frac{1}{\rho C_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_Rv_R|v_R|\cdot\Delta t}{d_I} = 0 \quad 4$$

Outlet pipeline

$$-\frac{1}{\rho C_o}(P_o - P_s) + (v_o - v_s) + \frac{2f_sv_s|v_s|\Delta t}{d_o} = 0 \quad 5$$

It is assumed that the pump speed (W_p) and pump swash (X_p) have been evaluated for the time level at which the solution of pressures and flows is required. The pump speed is calculated by a call to a prime mover subroutine, the pump inertia is referred through to the prime mover and the prime mover dynamics determine the instantaneous pump speed. In the current model the swash is calculated as a rapid ramp from zero to full swash in a number of seconds specified by the user. However provisions have been made for future modifications to have the swash determined by an external subroutine where the swash follows a particular duty cycle or obeys the dynamics of a servo mechanism.

There are four unknowns P_I, P_O, v_I, v_O for which equations 1 , 2 , 4 and 5 may be solved simultaneously. Once the pressures are known the current shaft torque may be calculated from equation 3 .

Rearranging equations 1 , 2 , 4 , and 5 in the form:-

1 → $$A_1 v_I + A_2 P_I + A_3 P_O = A_4 \qquad (6)$$

2 → $$B_1 v_O + B_2 P_I + B_3 P_O = B_4 \qquad (7)$$

4 → $$C_1 v_I + C_2 P_I = C_3 \qquad (8)$$

5 → $$D_1 v_O + D_2 P_D = D_3 \qquad (9)$$

where

$$A_1 = A_{PI} \qquad (10)$$

$$A_2 = -K_{TP} - K_{LP} \qquad (11)$$

$$A_3 = +K_{LP} \qquad (12)$$

$$A_4 = W_p D_p X_p - K_{TP} P_{CD} \qquad (13)$$

$$B_1 = A_{PO} \qquad (14)$$

$$B_2 = -K_{LP} \qquad (15)$$

$B_3 = K_{TP} + K_{LP}$ 16

$B_4 = W_p D_p X_p + K_{TP} P_{CD}$ 17

$C_1 = 1.0$ 18

$C_2 = 1/\rho C_I$ 19

$C_3 = -2f_R v_R |v_R| \Delta t / d_I + v_R + P_R / \rho C_I$ 20

$D_1 = 1.0$ 21

$D_2 = -1/\rho C_o$ 22

$D_3 = -2f_s v_s |v_s| \Delta t / d_o + v_s - P_s / \rho C_o$ 23

Equations 6 - 9 may be solved simultaneously and the following expressions obtained.

$$P_o = \left[A_4 - A_1 \left(\frac{C_3}{C_1} - \frac{C_2}{C_1} R \right) - A_2 R \right] \Big/ \left[A_2 Q + A_3 - \frac{A_1 C_2 Q}{C_1} \right] \qquad 24$$

$$v_I = \frac{C_3}{C_1} - \frac{C_2 R}{C_1} - \frac{C_2 Q P_o}{C_1} \qquad 25$$

$$P_I = R + QP_o \qquad 26$$

$$v_o = (D_3 - D_2 P_o) / D_1 \qquad 27$$

where

$$R = \frac{B_4}{B_2} - \frac{B_1 D_3}{B_2 D_1} \qquad 28$$

$$Q = \frac{B_1 D_2}{B_2 D_1} - \frac{B_3}{B_2} \qquad 29$$

Computing procedure

The subroutine assigns all pump data from common block array PMP to program variable names. Depending on the value of NTYP a prime mover model is chosen and the appropriate subroutine is called. The prime mover model calculates the current pump shaft speed which is transferred via common block array PMPD to program variable WP2. The current value of pump swash is calculated and finally all pipe data is assigned to program variable names.

The equation coefficients A_1, A_2 .... R, Q (equations 10 - 23 , 28 and 29) are calculated. Solving equations 24 - 27 evaluates the current values of pressure and flow at the pump ports. A cavitation check is performed to ensure negative pressures are set to zero. The computed data is output to the common block arrays.

Operational status

The program coding was written with a view to easy debugging and testing as a result it is not particularly efficient. Future versions of this program will be modified to increase the speed of execution and minimise storage requirements.

LIST OF VARIABLES USED

API	inlet pipe area	A_{PI}	M^2	R
APO	outlet pipe area	A_{po}	M^2	R
Al - A4	flow into pump equation coefficient	A_1, A_4	-	R
Bl - B4	flow out of pump equation coefficient	B_1, B_4	-	-
CI	inlet pipe wavespeed	C_I	M/S	R
CO	outlet pipe wavespeed	C_o	M/S	R
Cl - C3	inlet pipe equation coefficient	C_1, C_3	-	R
DI	inlet pipe diameter	d_I	M	R
DO	outlet pipe diameter	d_o	M	R
DP	pump displacement	D_p	M^3/rad	R
DT	timestep	Δt	s	R
Dl - D3	outlet pipe equation coefficient	D_1, D_3	-	R
FR	friction factor, forward characteristic	f_R	-	R
FS	friction factor, backward characteristic	f_s	-	R
IE	not used	-	-	-

IP	pump inertia	-	K_gM^2	R
IPN	inlet pipe number	-	-	I
IT	time level indicator time = t	-	-	I
ITM	time level indicator time = t - Δt	-	-	I
KLP	slip loss coefficient	K_{LP}	$M^3/S/(N/M^2)$	R
KTP	slip loss coefficient	K_{TP}	$M^3/S/(N/M^2)$	R
KVP	viscous friction coefficient	K_{VP}	NM/(rad/s)	R
NCM	number of calculation points minus one	-	-	I
NCPI	number of calculation points on inlet pipe	-	-	I
NO	pump number	-	-	I
NPRM	prime mover number	-	-	I
NTYP	prime mover selection switch	-	-	I
OPN	outlet pipe number	-	-	I
PCD	pump case drain pressure	P_{CD}	N/M^2	R
PI	pressure at pump inlet time = t	P_I	N/M^2	R
PO	pressure at pump outlet time = t	P_o	N/M^2	R
PR	pressure, forward characteristic	P_R	N/M^2	R
PS	pressure, backward characteristic	P_s	N/M^2	R
Q	equation coefficient	-	-	R
QR	flow, forward characteristic	-	M^3/S	R
QS	flow, backward characteristic	-	M^3/S	R
R	equation coefficient	-	-	R
RHO	fluid density	ρ	Kg/M^3	R
TIME	time	-	S	R
TP2	pump shaft torque (time = t)	T_p	Nm	R
T1	starting time for swash change	-	S	R
T2	finishing time for swash change	-	S	R
VI	flow velocity at pump inlet time = t	v_I	M/S	R
VO	flow velocity at pump outlet time = t	v_o	M/S	R
VR	flow velocity, forward characteristic	v_R	M/S	R
VS	flow velocity, backward characteristic	v_s	M/S	R
WP2	pump shaft speed time = t	W_p	rad/s	R
XP	pump swash time = t	X_p	-	R

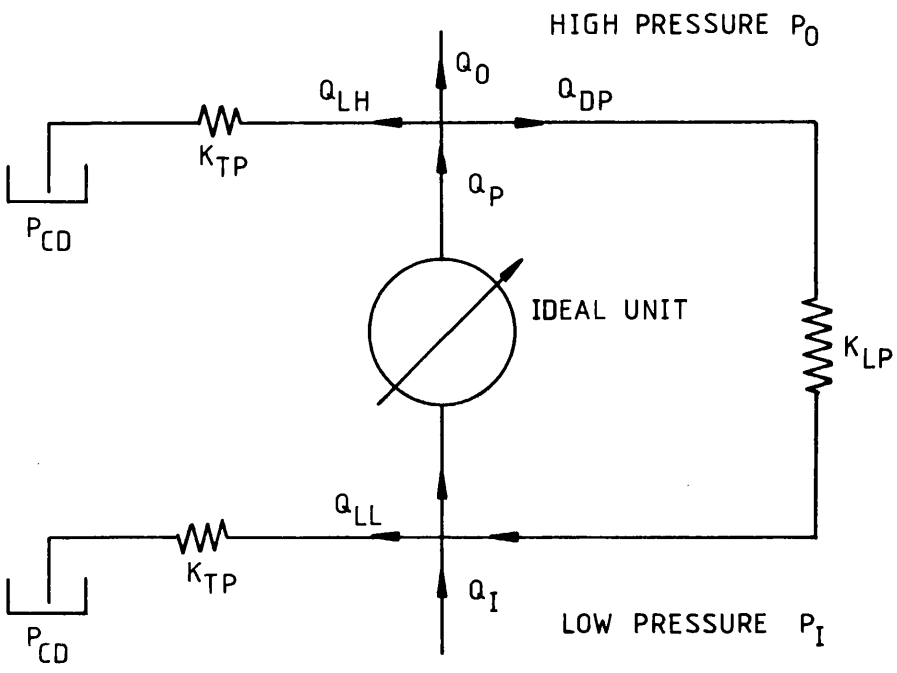

FIGURE 1a SCHEMATIC DIAGRAM OF A PUMP

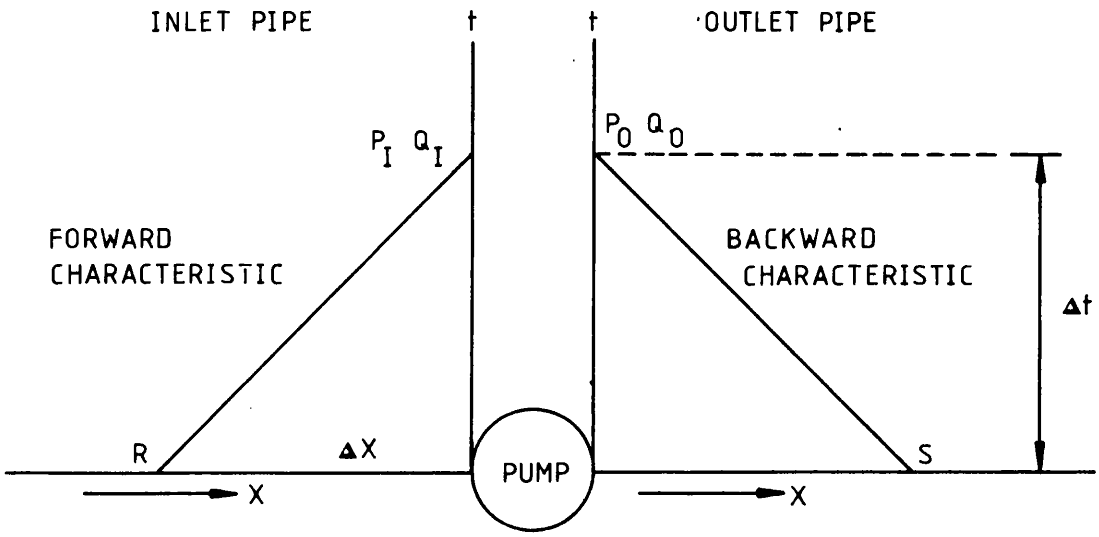

FIGURE 1b PUMP MODEL ON THE TIME DISTANCE PLANE

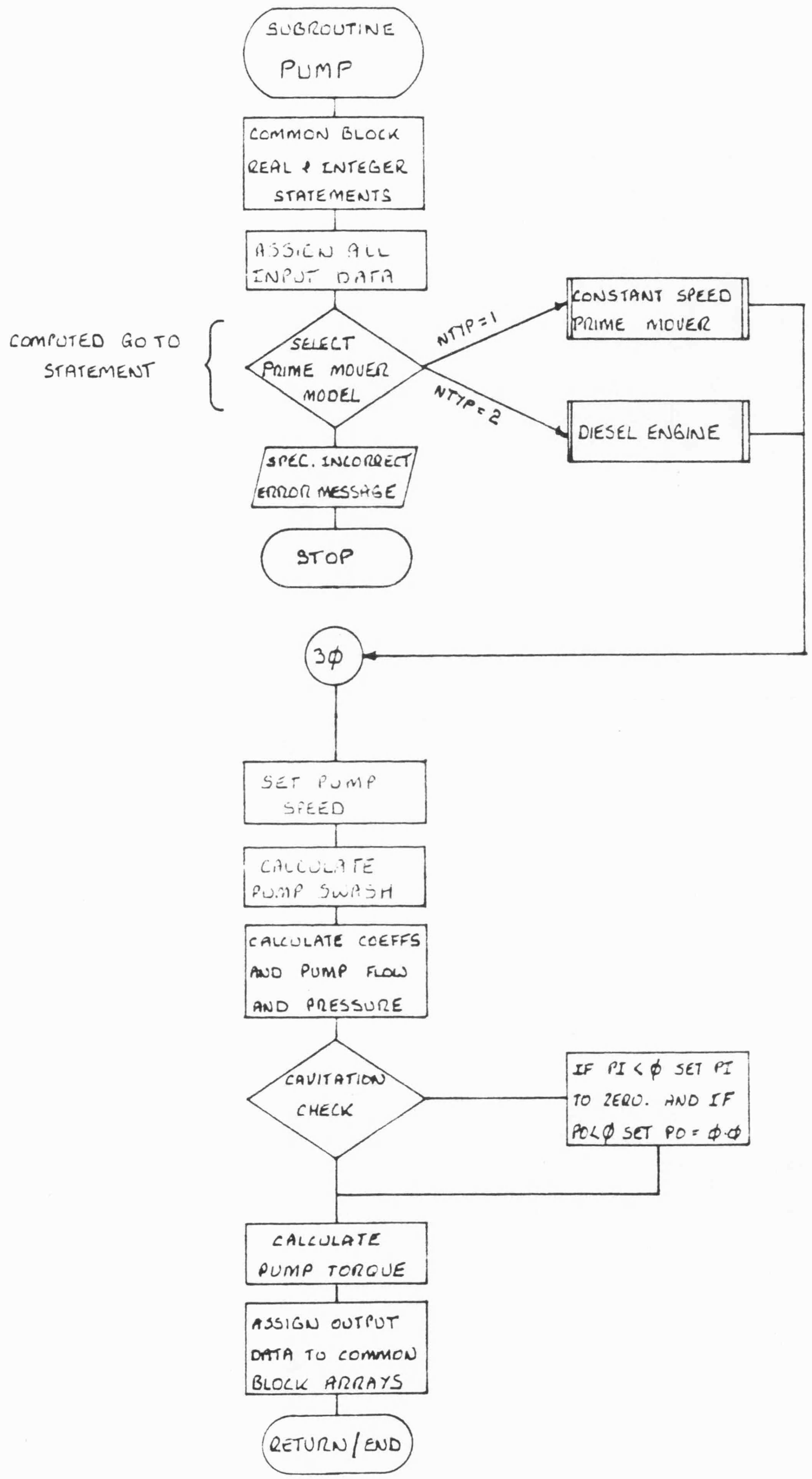

FIGURE 2 FLOWCHART FOR SUBROUTINE PUMP

```
      subroutine pump(it,itm,dt,rho,time,no)
c
c     subroutine name  pump
c
c     library classification
c
c     title  method of characteristics model of a hydrostatic pump
c
c     author  c.m. skarbek-wazynski
c
c     purpose   this subroutine is a steady state model of a hydrostatic
c               pump, all internal dynamics are neglected, the pump speed
c               is determined by the prime mover dynamics. pump
c               pressures and flows depend on the interaction with the
c               rest of the system via the inlet and outlet pipes which
c               are modelled using the method of characteristics
c               (model is capable of ramp changes in swash setting)
c
c     associated subroutines
c     const   constant speed prime mover model
c     engine  governed diesel engine model
c
c     common blocks
c     common/blk1/ pd    pipe data (constant)
c     common/blk3/ pdyn  pipe data (dynamic)
c     common/blk6/ pmp   pump data (constant)
c     common/blk7/ pmpd  pump data (dynamic)
c     common/blk26/swsh  swash setting parameters
c
c     input information
c     input via argument list
c     dt   timestep                        s
c     it   time level indicator            --
c     itm  time level indicator            --
c     no   pump component number           --
c     rho  fluid density                   kg/m3
c     time simulation time                 s
c
c     input via common block
c     api   inlet pipe area                  m2        pd(2,-)
c     apo   outlet pipe area                 m2        pd(2,-)
c     ci    wavespeed,inlet pipe             m/s       pd(7,)
c     co    wavespeed,outlet pipe            m/s       pd(7,-)
c     di    diameter, inlet pipe             m         pd(1,-)
c     do    diameter, outlet pipe            m         pd(1,-)
c     dp    pump displacement                m3/rad    pmp(1,-)
c     fr    friction factor fwrd/charac      --        pdyn(3,-,-,-)
c     fs    friction factor bkwrd/charac     --        pdyn(3,-,-,-)
c     ip    pump inertia                     kgm2      pmp(5,-)
c     ipn   inlet pipe number                --        pmp(6,-)
c     klp   slip loss coefficient            m5/ns     pmp(2,-)
c     ktp   slip loss coefficient            m5/ns     pmp(3,-)
c     kvp   viscous friction coefficient     nms/rad   pmp(4,-)
c     ncpi  no of calc points inlet pipe     --        pd(6,-)
c     nprm  prime mover number               --        pmp(9,-)
c     ntyp  prime mover selection switch     --        pmp(10,-)
c     opn   outlet pipe number               --        pmp(7,-)
c     pcd   pump case drain pressure         n/m2      pmp(8,-)
c     pr    pressure fwrd/characteristic     n/m2      pdyn(1,-,-,-)
```

```
c     ps    pressure bwrd/characteristic     n/m2      pdyn(1,-,-,-)
c     qr    flow fwrd/characteristic         m3/s      pdyn(2,-,-,-)
c     qs    flow bwrd/characteristic         m3/s      pdyn(2,-,-,-)
c     t1    start time of swash change       s         swsh(1)
c     t2    finish time of swash change      s         swsh(2)
c     wp2   pump shaft speed                 rad/s     pmpd(1,-,-)
c     xp    pump swash setting               --        pmpd(3,-,-)
c
c     output information
c     output via common block
c     pi      pressure at inlet              n/m2      pdyn(1,-,-,-)
c     po      pressure at outlet             n/m2      pdyn(1,-,-,-)
c     tp2     pump shaft torque              nm        pmpd(2,-,-)
c     vi*api  flow at inlet                  m3/s      pdyn(2,-,-,-)
c     vo*apo  flow at outlet                 m3/s      pdyn(2,-,-,-)
c
c     variable names (excluding i/o variables)
c     a1...a4   equation coefficients                   --
c     b1...b4   equation coefficients                   --
c     c1...c3   equation coefficients                   --
c     d1...d3   equation coefficients                   --
c     ncm       no of calc points minus 1               --
c     q         equation coefficient                    --
c     r         equation coefficient                    --
c     vi        flow velocity inlet pipe                m/s
c     vo        flow velocity outlet pipe               m/s
c     vr        flow velocity forward characteristic    m/s
c     vs        flow velocity backward characteristic   m/s
c
      common /blk1/ pd(8,10)
      common /blk3/ pdyn(3,20,10,10)
      common /blk6/ pmp(10,2)
      common /blk7/ pmpd(3,10,2)
      common /blk26/swsh(2)
      real ip,ie,ktp,kvp,klp
      integer opn
c
c     input data
c     pump data
      dp =pmp(1,no)
      klp=pmp(2,no)
      ktp=pmp(3,no)
      kvp=pmp(4,no)
      ip =pmp(5,no)
      ipn=pmp(6,no)
      opn=pmp(7,no)
      pcd=pmp(8,no)
      nprm=pmp(9,no)
      ntyp=pmp(10,no)
c
c     pipe data
      ncpi=pd(6,ipn)
      ncm=ncpi-1
      api=pd(2,ipn)
      apo=pd(2,opn)
      di =pd(1,ipn)
      do =pd(1,opn)
      ci =pd(7,ipn)
      co =pd(7,opn)
      pr=pdyn(1,ncm,itm,ipn)
      fr=pdyn(3,ncm,itm,ipn)
      qr=pdyn(2,ncm,itm,ipn)
      vr=qr/api
```

```
      ps=pdyn(1,2,itm,opn)
      fs=pdyn(3,2,itm,opn)
      qs=pdyn(2,2,itm,opn)
      vs=qs/apo
c
c     select prime mover model
      go to (10,20),ntyp
c     print error message
      write(6,801)
  801 format(1h0,'prime mover number incorrect')
      stop
   10 call const(nprm,it,no)
      go to 30
   20 call engine(it,itm,dt,no,nprm)
   30 continue
c
c     set pump speed
      wp2=pmpd(1,it,no)
c
c     calculate pump swash
      t1=swsh(1)
      t2=swsh(2)
      xp=(time-t1)/t2
      if(xp.gt.1.0)xp=1.0
c
c     calculate equation coefficients
      a1=api
      a2=-ktp-klp
      a3=+klp
      a4=wp2*dp*xp-ktp*pcd
      b1=apo
      b2=-klp
      b3=ktp+klp
      b4=wp2*dp*xp+ktp*pcd
      c1=1.0
      c2=1.0/(rho*ci)
      c3=-2.0*fr*vr*abs(vr)*dt/di+vr+pr/(rho*ci)
      d1=1.0
      d2=-1.0/(rho*co)
      d3=-2.0*fs*vs*abs(vs)*dt/do+vs-ps/(rho*co)
      r=b4/b2-b1*d3/(b2*d1)
      q=b1*d2/(b2*d1)-b3/b2
c
c     calculate pressures and velocities
      po=(a4-a1*(c3-c2*r)/c1-a2*r)/(a2*q+a3-a1*c2*q/c1)
      vi=c3/c1-c2*r/c1-c2*q*po/c1
      pi=r+q*po
      vo=(d3-d2*po)/d1
c
c     cavitation check
      if(pi.lt.0.0)pi=0.0
      if(po.lt.0.0)po=0.0
c
c     calculate pump load torque
      tp2=(po-pi)*dp*xp+kvp*wp2
c
c     output data to arrays
      pdyn(1,ncpi,it,ipn)=pi
      pdyn(2,ncpi,it,ipn)=vi*api
      pdyn(1,1,it,opn)=po
      pdyn(2,1,it,opn)=vo*apo
      pmpd(2,it,no)=tp2
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'MOTOR'

Method of characteristics model of a hydrostatic motor

Purpose

This subroutine is a steady state model of a hydrostatic motor, all internal dynamics are neglected, the motor speed is determined by the load dynamics, pressures and flows at the motor speed depend on the interaction with the rest of the system via the inlet and outlet pipes which are modelled using the method of characteristics.

Associated subroutines

LOADCH	simple inertia load model
BOAT	load model of a propeller driven boat

CALL MOTOR(IT,ITM,DT,RHO,MNO)

MNO = motor component number (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK9/HMD	motor data (constant)
COMMON/BLK10/HMDD	motor data (dynamic)

User defined information

ARRAY HMD(11,MNO)

HMD(1,MNO)	= DM	motor displacement	M^3/rad	R
HMD(2,MNO)	= KLM	slip loss coefficient	$M^3/S/(N/M^2)$	R
HMD(3,MNO)	= KTM	slip loss coefficient	$M^3/S/(N/M^2)$	R
HMD(4,MNO)	= KVM	viscous friction coefficient	NM/(rad/s)	R
HMD(5,MNO)	= IPN	inlet pipe number	-	I
HMD(6,MNO)	= OPN	outlet pipe number	-	I
HMD(7,MNO)	= IM	motor inertia	KgM^2	R
HMD(8,MNO)	= KPM	pressure dependent loss coefficient	$NM/(N/M^2)$	R

HMD(9,MNO) = PCD	tank or case drain pressure	N/M^2	R
HMD(10,MNO) = LNO	load model component number	-	I
HMD(11,MNO) = NTYP	load model selection switch	-	I

ARRAY HMDD(3,IT,MNO)

HMDD(1,1,MNO) = XM	motor swash setting	-	R
HMDD(2,1,MNO) = TM1	motor shaft torque	NM	R
HMDD(3,1,MNO) = WM	motor speed	rad/s	R

Note: The user is only required to specify initial values subsequent values are calculated by the program.

The load model selection switch NTYP must be set to 1 if a simple load model of an inertia plus viscous friction is required (see subroutine LOADCH). Alternatively if NTYP is set to 2 the load model defined by subroutine BOAT is called.

Output information via common block

PI	pressure at motor inlet, time = t	PDYN(1,NCPI,IT,IPN)	N/M^2	R
PO	pressure at motor outlet, time = t	PDYN(1,1,IT,OPN)	N/M^2	R
VI*API	volumetric flow at motor inlet	PDYN(2,NCPI,IT,IPN)	M^3/S	R
VO*APO	volumetric flow at motor outlet	PDYN(2,1,IT,OPN)	M^3/S	R
TM	torque on motor shaft	HMDD(2,IT,MNO)	NM	R

INBUILT ERROR MESSAGES

'LOAD TYPE NUMBER INCORRECT' - load type number specified incorrectly, call to load model cannot be made. Fatal error therefore the program is stopped.

Program action and algorithm

Mathematical model

The motor is treated as an ideal unit with various leakage paths around it, internal compressibility losses are ignored (Figure 1a). The inlet and outlet pipelines are modelled using the method of characteristics, case drain pipe dynamics are assumed to be negligible. With the flow directions as specified in Figure 1a the pipe characteristic lines are shown on the time distance plane in Figure 1b. Motor inertia is referred through to the load model and load model dynamics determined the instantaneous motor speed. The motor swash is assumed to be constant.

The motor equations are derived by considering continuity and a simple torque balance on the motor shaft.

<u>Flow into motor</u>

$$Q_I = A_{PI}v_I = W_m D_m X_m + K_{LM} (P_I - P_O) \; K_{TM}(P_I - P_{CD}) \qquad 1$$

<u>Flow out of motor</u>

$$Q_O = A_{PO}v_O = W_m D_m X_m + K_{LM} (P_I - P_O) - K_{TM}(P_O - P_{CD}) \qquad 2$$

<u>Shaft torque developed by motor</u>

$$T_m = (P_I - P_O)D_m X_m - K_{vm}W_m - K_{pm}|P_I - P_O| \operatorname{sign}W_m \qquad 3$$

The pipe equations are:-

<u>Inlet pipeline</u>

$$\frac{1}{\rho C_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad 4$$

<u>Cutlet pipeline</u>

$$-\frac{1}{\rho C_O}(P_O - P_S) + (v_O - v_S) + \frac{2f_S v_S |v_S| \Delta t}{d_O} = 0 \qquad 5$$

Equations 1 , 2 , 4 and 5 may be solved simultaneously for the four unknowns P_I, P_o, v_I, v_o. Once these values are known equation 3 may be used to calculate motor torque.

Rearranging equations 1 , 2 , 4 and 5

$$1 \rightarrow \quad A_1 v_I + A_3 P_I + A_4 P_o = A_5 \qquad 6$$

$$2 \rightarrow \quad B_2 v_o + B_3 P_I + B_4 P_o = B_5 \qquad 7$$

$$4 \rightarrow \quad C_1 v_I + C_3 P_I = C_5 \qquad 8$$

$$5 \rightarrow \quad D_2 v_o + D_4 P_o = D_5 \qquad 9$$

where

$$A_1 = A_{PI} \qquad 10$$

$$A_3 = -K_{LM} - K_{TM} \qquad 11$$

$$A_4 = K_{LM} \qquad 12$$

$$A_5 = W_m D_m X_m - K_{TM} P_{CD} \qquad 13$$

$$B_2 = A_{po} \qquad 14$$

$$B_3 = -K_{LM} \qquad 15$$

$$B_4 = K_{LM} + K_{TM} \qquad 16$$

$$B_5 = W_m D_m X_m + K_{TM} P_{CD} \qquad 17$$

$$C_1 = 1.0 \qquad 18$$

$$C_3 = 1/\rho C_I \qquad 19$$

$$C_5 = P_R/\rho C_I + v_R - 2f_R v_R |v_R| \Delta t/d_I \qquad 20$$

$$D_2 = 1.0 \qquad 21$$

$$D_4 = -1/\rho C_o \qquad 22$$

$$D_5 = -P_s/\rho C_o + v_s - 2f_s v_s |v_s| \Delta t/d_o \qquad 23$$

Equations 6 - 9 may be solved simultaneously to yield the following expressions:-

$$P_I = (R - S)\big/(P - Q) \qquad 24$$

$v_I = (C_5 - C_3P_I)/C_1$ 25

$P_o = (A_5 - A_3P_I - A_1v_I)/A_4$ 26

$v_o = (D_5 - D_4P_o)/D_2$ 27

where the coefficient P, Q, R, S are defined as follows:-

$P = (C_1A_3 - A_1C_3)/C_1$ 28

$Q = A_4B_3D_2/(D_2B_4 - B_2D_4)$ 29

$R = (C_1A_5 - A_1C_5)/C_1$ 30

$S = A_4(D_2B_5 - B_1D_5)/(D_2B_4 - B_2D_4)$ 31

Computing procedure

All motor and connected pipe data is assigned to program variable names from common block data arrays. A load model is selected, depending on the value of the switch NTYP, the appropriate subroutine is called which returns a value for the current motor speed, which is directly transferred via common block to program variable name Wm.

The equation coefficients A_1, A_2 .... D_4, D_5 (expressions 10 - 23) are evaluated, hence the coefficients P, Q, R, S are calculated from equations 28 - 31 , and finally the current values of pressure and flow velocity at the motor ports are calculated from equations 24 - 27 . A cavitation check is performed to ensure negative pressures are set to zero. The shaft torque developed by the motor is calculated from equation 3 . A problem arises when the motor speed Wm is zero, the sign of the pressure dependent loss term is indeterminate. To avoid this difficulty it is assumed that in cases of zero velocity the pressure dependent torque opposes the motor torque and the sign of the $K_{pm}|P_I - P_o|$ term is determined by the last known value of motor torque TMl. (See program listing.)

Operational status

The original versions of this program were written with the intention of using a matrix method for solving the set of simultaneous equations. The coefficients A1, A2 etc. were set up with this in mind. Subsequent program alterations to a direct solution have made this form of coding inefficient. Future versions of this subroutine will be modified to increase speed of execution and to minimise storage requirements.

The motor swash is set to 1.0 in the program and the value stored in array element HMDD(1,1,MNO) has no effect.

LIST OF VARIABLES USED

API	inlet pipe area	A_{PI}	M^2	R
APO	outlet pipe area	A_{PO}	M^2	R
A1, A5	equation coefficient	A_1, A_5	-	R
B2, B5	equation coefficient	B_2, B_5	-	R
CI	wavespeed, inlet pipe	C_I	M/S	R
CO	wavespeed, outlet pipe	C_o	M/S	R
C1, C5	equation coefficient	C_1, C_5	-	R
DI	pipe diameter, inlet pipe	d_I	M	R
DM	motor displacement	d_m	M^3/rad	R
DO	pipe diameter, outlet pipe	d_o	M	R
DT	timestep	Δt	S	R
D2, D5	equation coefficient	D_2, D_5	-	R
FR	friction factor, forward characteristic	f_R	-	R
FS	friction factor, backward characteristic	f_s	-	R
IM	motor inertia	I_m	KgM^2	R
IPN	inlet pipe number	-	-	I
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
KLM	slip loss coefficient	K_{LM}	$M^3/S/(N/M^2)$	R
KPM	pressure dependent loss coefficient	K_{PM}	$NM/(N/M^2)$	R
KTM	slip loss coefficient	K_{TM}	$M^3/S/(N/M^2)$	R
KVM	viscous friction coefficient	K_{VM}	NM/(rad/s)	R
LNO	load model component number	-	-	I
MNO	motor component number	-	-	I

NCPI	number of calculation points on inlet line	-	-	I
NCR	NCPI minus 1	-	-	I
NTYP	load model selection switch	-	-	I
OPN	outlet pipe number	-	-	I
P	equation coefficient	P	-	R
PCD	case drain or tank pressure	P_{CD}	N/M^2	R
PI	pressure, motor inlet, time = t	P_I	N/M^2	R
PO	pressure, motor outlet, time = t	P_O	N/M^2	R
PR	pressure, forward characteristic	P_R	N/M^2	R
PS	pressure, backward characteristic	P_s	N/M^2	R
Q	equation coefficient	Q	-	R
QR	flow, forward characteristic	-	M^3/S	R
QS	flow, backward characteristic	-	M^3/S	R
R	equation coefficient	R	-	R
RHO	fluid density	ρ	Kg/M^3	R
S	equation coefficient	S	-	R
TM	motor torque, time = t	T_m	NM	R
TM1	motor torque, time = t - Δt	-	NM	R
VI	flow velocity, at motor inlet	v_I	M/S	R
VO	flow velocity, at motor outlet	v_O	M/S	R
VR	flow velocity, forward characteristic	v_R	M/S	R
VS	flow velocity, backward characteristic	v_s	M/S	R
WM	motor speed	W_m	rad/s	R
XM	motor swash setting	X_m	-	R

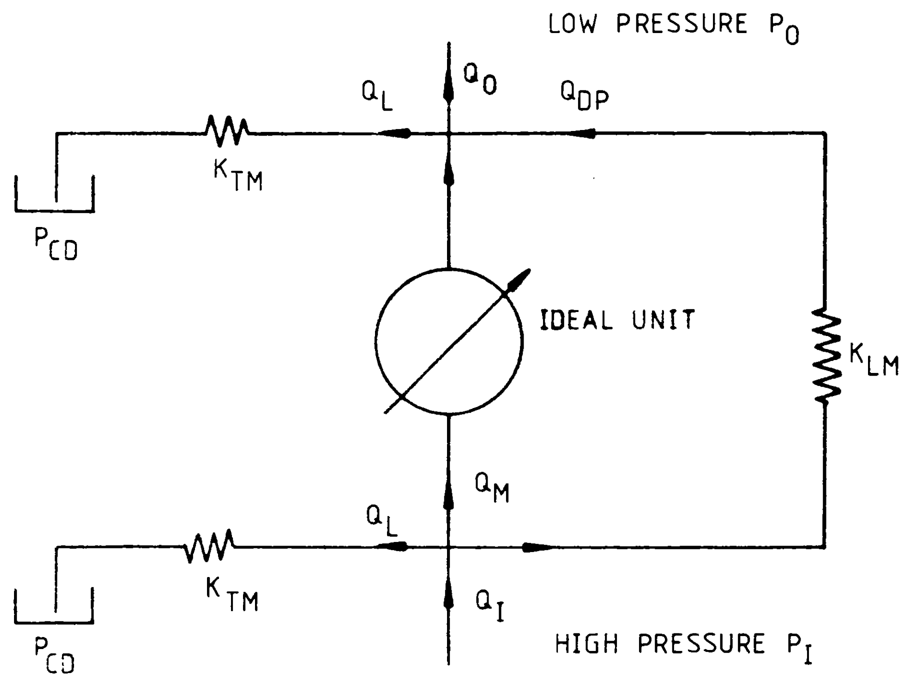

FIGURE 1a SCHEMATIC DIAGRAM OF A MOTOR

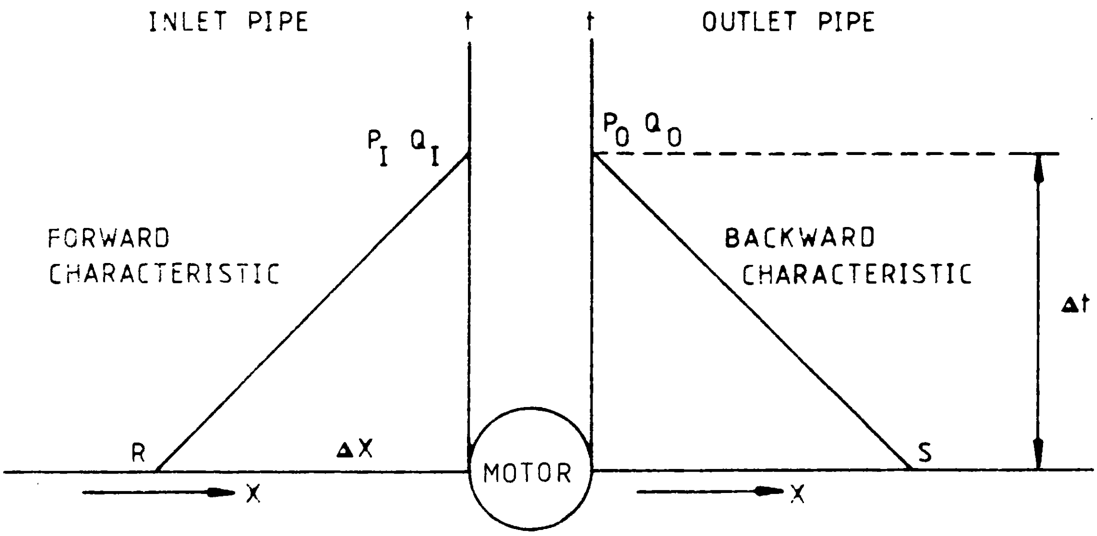

FIGURE 1b MOTOR MODEL ON THE TIME DISTANCE PLANE

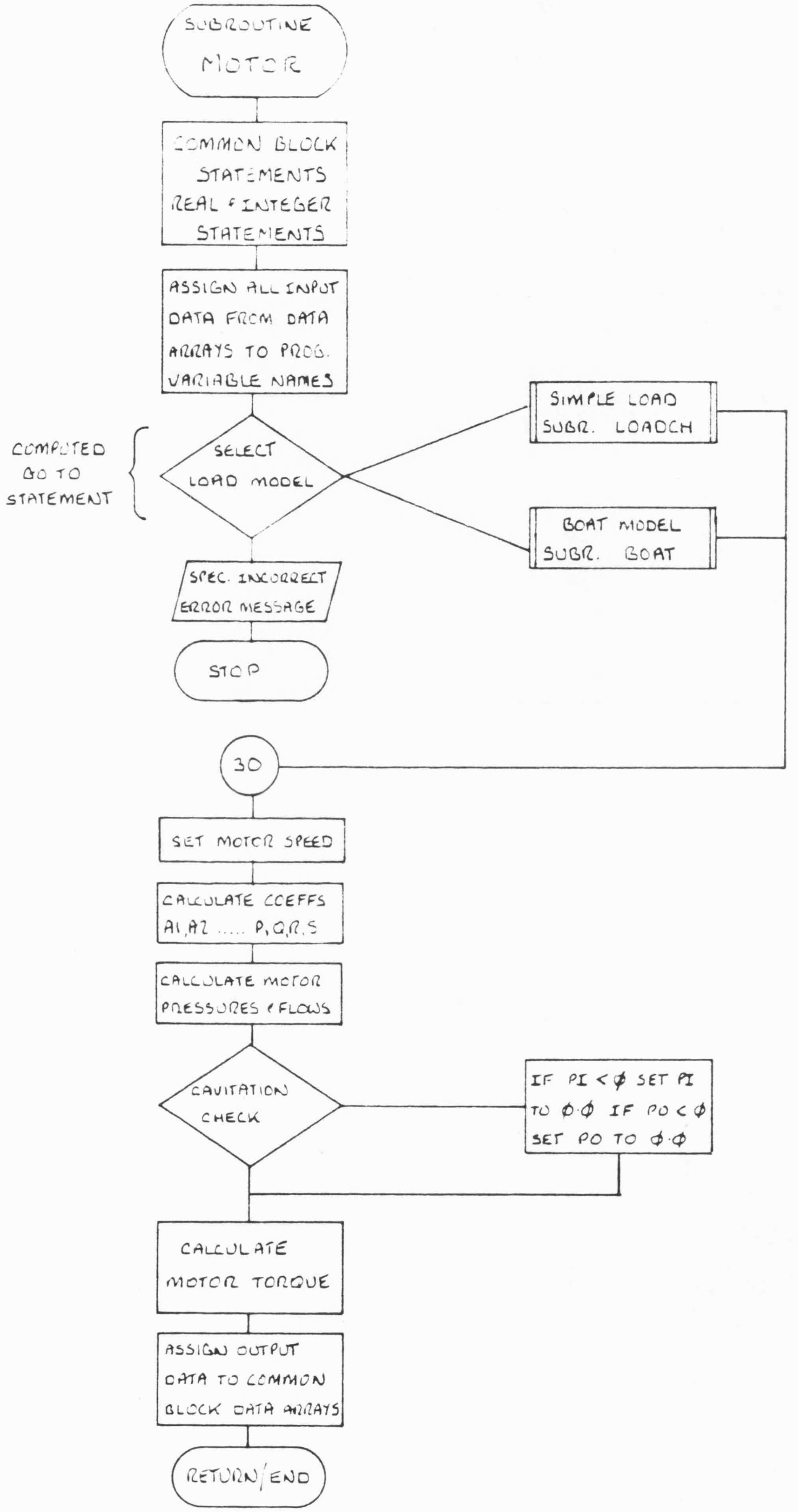

FIGURE 2 FLOWCHART FOR SUBROUTINE MOTOR

```
      subroutine motor(it,itm,dt,rho,mno)
c
c     subroutine name   motor
c
c     library classification
c
c     title  method of characteristics model of a hydrostatic motor
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine is a steady state model of a hydrostatic
c              motor, all internal dynamics are neglected, the motor
c              speed is determined by the load dynamics. pressures and
c              flows at the motor depend on the interaction with the
c              rest of the system via the inlet and outlet pipes which
c              are modelled using the method of characteristics.
c
c     associated subroutines
c     loadch   simple inertia load model
c     boat     load model of a propeller driven boat
c
c     common blocks
c     common/blk1/ pd      pipe data (constant)
c     common/blk3/ pdyn    pipe data (dynamic)
c     common/blk9/ hmd     motor data(constant)
c     common/blk10/hmdd    motor data(dynamic)
c
c     input information
c     input via argument list
c     dt   timestep                       s
c     it   time level indicator           --
c     itm  time level indicator           --
c     mno  motor component number         --
c     rho  fluid density                  kg/m3
c
c     input via common block
c     api   inlet pipe area                 m2        pd(2,-)
c     apo   outlet pipe area                m2        pd(2,-)
c     ci    wavespeed,inlet pipe            m/s       pd(7,-)
c     co    wavespeed,outlet pipe           m/s       pd(7,-)
c     di    diameter, inlet pipe            m         pd(1,-)
c     dm    motor displacement              m3/rad    hmd(1,-)
c     do    diameter, outlet pipe           m         pd(1,-)
c     fr    friction factor fwrd/charac     --        pdyn(3,-,-,-)
c     fs    friction factor bkwrd/ch(rac    --        pdyn(3,-,-,-)
c     ipn   inlet pipe number               --        hmd(5,-)
c     klm   slip loss coefficient           m5/ns     hmd(2,-)
c     kpm   pressure dependent loss coeff   m3        hmd(8,-)
c     ktm   slip loss coefficient           m5/ns     hmd(3,-)
c     kvm   viscous friction coefficient    nms/rad   hmd(4,-)
c     lno   load model component no.        --        hmd(10,-)
c     ncpi  no of calc points inlet pipe    --        pd(6,-)
c     ntyp  load model selection switch     --        hmd(11,-)
c     opn   outlet pipe number              --        hmd(6,-)
c     pcd   case drain or tank pressure     n/m2      hmd(9,-)
c     pr    pressure fwrd/characteristic    n/m2      pdyn(1,-,-,-)
c     ps    pressure bwrd/characteristic    n/m2      pdyn(1,-,-,-)
c     qr    flow fwrd/characteristic        m3/s      pdyn(2,-,-,-)
c     qs    flow bwrd/characteristic        m3/s      pdyn(2,-,-,-)
```

```
c      tm!   motor torque                         nm        hmdd(2,-,-)
c      wm    motor speed                          rad/s     hmdd(3,-,-)
c      xm    motor swash                          --        hmdd(1,-,-)
c
c      output information
c      output via common block
c      pi        pressure at inlet                n/m2      pdyn(1,-,-,-)
c      po        pressure at outlet               n/m2      pdyn(1,-,-,-)
c      vi*api    flow at inlet                    m3/s      pdyn(2,-,-,-)
c      vo*apo    flow at outlet                   m3/s      pdyn(2,-,-,-)
c      tm        motor shaft torque               nm        hmdd(2,-,-)
c
c      variable names (excluding i/o variables)
c      a1...a5    equation coefficients                       --
c      b2...b5    equation coefficients                       --
c      c1...c5    equation coefficients                       --
c      d2...d5    equation coefficients                       --
c      ncr        no of calc points minus 1                   --
c      p          equation coefficient                        --
c      q          equation coefficient                        --
c      r          equation coefficient                        --
c      s          equation coefficient                        --
c      vi         flow velocity inlet pipe                    m/s
c      vo         flow velocity outlet pipe                   m/s
c      vr         flow velocity forward characteristic        m/s
c      vs         flow velocity backward characteristic       m/s
c
       common /blk1/ pd(8,10)
       common /blk3/ pdyn(3,20,10,10)
       common /blk9/ hmd(11,2)
       common/blk10/ hmdd(3,10,2)
       integer opn
       real klm,ktm,kvm,kpm
c
c      input data
       ipn=hmd(5,mno)
       opn=hmd(6,mno)
       klm=hmd(2,mno)
       ktm=hmd(3,mno)
       kvm=hmd(4,mno)
       kpm=hmd(8,mno)
       dm =hmd(1,mno)
       pcd=hmd(9,mno)
       lno=hmd(10,mno)
       ntyp=hmd(11,mno)
       xm=1.0
       api =pd(2,ipn)
       apo =pd(2,opn)
       ncpi=pd(6,ipn)
       di  =pd(1,ipn)
       do  =pd(1,opn)
       ci  =pd(7,ipn)
       co  =pd(7,opn)
       ncr=ncpi-1
c
       pr=pdyn(1,ncr,itm,ipn)
       fr=pdyn(3,ncr,itm,ipn)
       qr=pdyn(2,ncr,itm,ipn)
       vr=qr/api
       ps=pdyn(1,2,itm,opn)
       fs=pdyn(3,2,itm,opn)
```

```
      qs=pdyn(2,2,itm,opn)
      vs=qs/apo
c
c     select load model
      go to (10,20),ntyp
c     print error message
      write(6,800)
  800 format(1h0,'load type number incorrect')
      stop
   10 call loadch(it,itm,dt,lno,mno)
      go to 30
   20 call boat(it,itm,dt,mno)
   30 continue
c
      wm=hmdd(3,it,mno)
      tm1=hmdd(2,itm,mno)
c
c     set coefficients
      a1=api
      a3=-klm-ktm
      a4=klm
      a5=wm*dm*xm-ktm*pcd
      b2=apo
      b3=-klm
      b4=klm+ktm
      b5=wm*dm*xm+ktm*pcd
      c1=1.0
      c3=1.0/(rho*ci)
      c5=pr/(rho*ci)+vr-2.0*fr*vr*abs(vr)*dt/di
      d2=1.0
      d4=-1.0/(rho*co)
      d5=-ps/(rho*co)+vs-2.0*fs*vs*abs(vs)*dt/do
c
      p=(c1*a3-a1*c3)/c1
      q=a4*b3*d2/(d2*b4-b2*d4)
      r=(c1*a5-a1*c5)/c1
      s=a4*(d2*b5-b2*d5)/(d2*b4-b2*d4)
c
      pi=(r-s)/(p-q)
      vi=(c5-c3*pi)/c1
      po=(a5-a3*pi-a1*vi)/a4
      vo=(d5-d4*po)/d2
c
c     cavitation check
      if(pi.lt.0.0)pi=0.0
      if(po.lt.0.0)po=0.0
c
c     calculation of torque
      if(wm)40,50,40
   40 tm=(pi-po)*dm*xm-kvm*wm-kpm*sign((pi-po),wm)
      go to 60
   50 tm=(pi-po)*dm*xm-kvm*wm-kpm*sign((pi-po),tm1)
   60 continue
c
c     output data
      pdyn(1,ncpi,it,ipn)=pi
      pdyn(2,ncpi,it,ipn)=vi*api
      pdyn(1,1,it,opn)=po
      pdyn(2,1,it,opn)=vo*apo
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'BOAT'

Load model of a propeller driven boat

Purpose

Subroutine BOAT is a load model of a propeller driven boat for use in conjunction with subroutine MOTOR. It calculated the load torques developed by the propeller and determines the acceleration of the propeller shaft to which the hydraulic motor is directly connected. Integration gives a new value of shaft speed which is transferred to the motor subroutine where the resulting load pressures and flows are calculated. The model is not valid for negative values of propeller speed.

Associated subroutines

MOTOR	method of characteristics model of a hydrostatic motor
INTRPI	smoothed one dimensional interpolation and linear extrapolation subroutine.

CALL BOAT(IT,ITM,DT,MNO)

MNO = component number of motor driving the boat (integer)

Common block data arrays

COMMON/BLK9/HMD	motor data (constant)
COMMON/BLK10/HMDD	motor data (dynamic)
COMMON/BLK30/DLCD	boat data (constant)
COMMON/BLK21/DLDP	boat data (dynamic)
COMMON/BLK19/EFF	propellor efficiency values
COMMON/BLK20/THR	propellor thrust coefficient values
COMMON/BLK28/ARATIO	corresponding advance ratio values

User defined information

ARRAY DLCD(6)

DLCD(1) = PDIA	propellor diameter	M	R
DLCD(2) = CD	boat drag coefficient	-	R
DLCD(3) = AB	boat frontal area	M^2	R
DLCD(4) = IP	propeller inertia	KgM^2	R

DLCD(5) = BMS	boat mass	Kg	R
DLCD(6) = ROE	density of water in which the boat is floating	Kg/M^3	R

ARRAY DLDP(2,IT)

DLDP(1,ITM) = VB1	boat velocity	M/S	R
DLDP(2,ITM) = WP1	propeller shaft speed	rad/s	R

Note: The user is only required to specify initial values, subsequent values are calculated by the program.

ARRAY EFF(31)

EFF(1→31) up to 31 values of propeller efficiency defining the efficiency curve (Fig. 1)

ARRAY THR(31)

THR(1→31) up to 31 values of propeller thrust coefficient defining the thrust coefficient curve (Fig. 1)

ARRAY ARATIO(31)

ARATIO(1→31) up to 31 values of advance ratio corresponding to the specified values of propeller efficiency and thrust coefficient

Note: Arrays, EFF, THR, ARATIO must all have the same dimensions.

Output information via common block

VB2	updated boat velocity	DLDP(1,IT)	R
WP2	updated propeller shaft speed	DLDP(2,IT) HMDD(3,IT,MNO)	R

Program action and algorithm

Mathematical model

The propeller characteristics are defined as functions of the boat advance ratio, (Fig. 1) which is the ratio of the distance the boat has actually travelled to the distance the propellor would have travelled if it were a perfect screw. The advance ratio is usually expressed in the form:- (Ref. 1)

$$A_R = v_{B1}/\omega_{p1}D_\rho \qquad 1$$

The graphs shown in Figure 1 are stored as arrays of points (EFF, THR) and once a value of A_R is specified the corresponding values of efficiency (η) and thrust coefficient (KT) may be obtained by smoothed interpolation (subroutine INTRPI).

The following equations define the boat performance:-

<u>Propeller thrust</u> $$F_{T1} = K_T\, \rho\, \omega_{p1}^{2} D_p^{4} \qquad 2$$

<u>Propeller torque</u> $$T_{p1} = F_T v_{B1}/\omega_{p1}\eta \qquad 3$$

<u>Boat drag</u> $$F_{D1} = 0.5\rho\, v_{B1}^{2} C_D A_B \qquad 4$$

<u>Acceleration of boat</u> $$v_B = (F_T - F_D)/M_B \qquad 5$$

<u>Accleration of propeller shaft</u> $$\omega_p = (T_{M1} - T_{p1})/(I_p + I_m) \qquad 6$$

<u>Computing procedure</u>

The purpose of the subroutine is to calculate a value of propeller shaft speed at a given time level using previously calculated values of boat velocity (v_{B1}) and propellor speed (ω_{p1}) to set up equations 1 - 6 . The sequence of calculations is shown in flowchart form in Figure 2 .

The values of boat acceleration and propeller shaft acceleration are integrated using Simple Euler to yield new values of boat speed and propeller speed. (Equations 7 and 8).

$$v_{B2} = v_B \, \Delta t + v_{B1} \qquad 7$$

$$\omega_{p2} = \omega_p \, \Delta t + \omega_{p1} \qquad 8$$

Operational status

The equations used assume a positive value of propeller speed. Interpolation will fail for negative values and the program will crash.

References

1 DUNCAN, THOM, YOUNG
Mechanics of fluids (pg. 671)
EDWARD ARNOLD LTD 1970

LIST OF VARIABLES USED

AB	boat frontal area	A_B	M^2	R
ACCP	acceleration of propeller shaft	ω_p	rad/s	R
AR	advance ratio	A_R	-	R
BMS	boat mass	M_B	Kg	R
CD	boat drag coefficient	C_D	-	R
DT	timestep	Δt	S	R
FD1	boat drag (old value)	F_{D1}	N	R
FT1	propellor thrust (old value)	F_{T1}	N	R
IM	hydraulic motor inertia	I_m	KgM^2	R
IP	propellor inertia	I_p	KgM^2	R
IT	time level indicator (time = t)	-	-	I
ITM	time level indicator (time = t - Δt)	-	-	I
MNO	hydraulic motor number	-	-	I
PDIA	propellor diameter	D_p	M	R
PEFF	propellor efficiency	η	-	R
ROE	fluid density	ρ	Kg/M^3	R
TC	thrust coefficient	K_T	-	R

TM1	hydraulic motor torque (old value)	T_{M1}	Nm	R
TP1	propeller thrust (old value)	T_{p1}	N	R
VBDOT	acceleration of boat	v_B	M/S^2	R
VB1	boat velocity (old value)	v_{B1}	M/S	R
VB2	boat velocity (new value)	v_{B2}	M/S	R
WP1	propeller shaft speed (old value)	ω_{p1}	rad/s	R
WP2	propeller shaft speed (new value)	ω_{p2}	rad/s	R

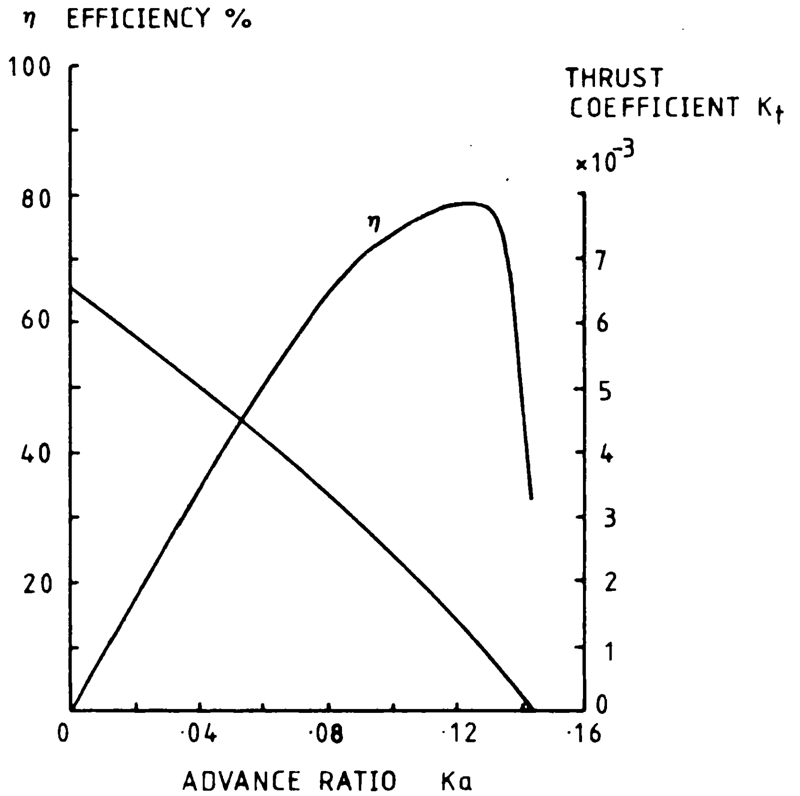

FIGURE 1 PROPELLER CHARACTERISTICS

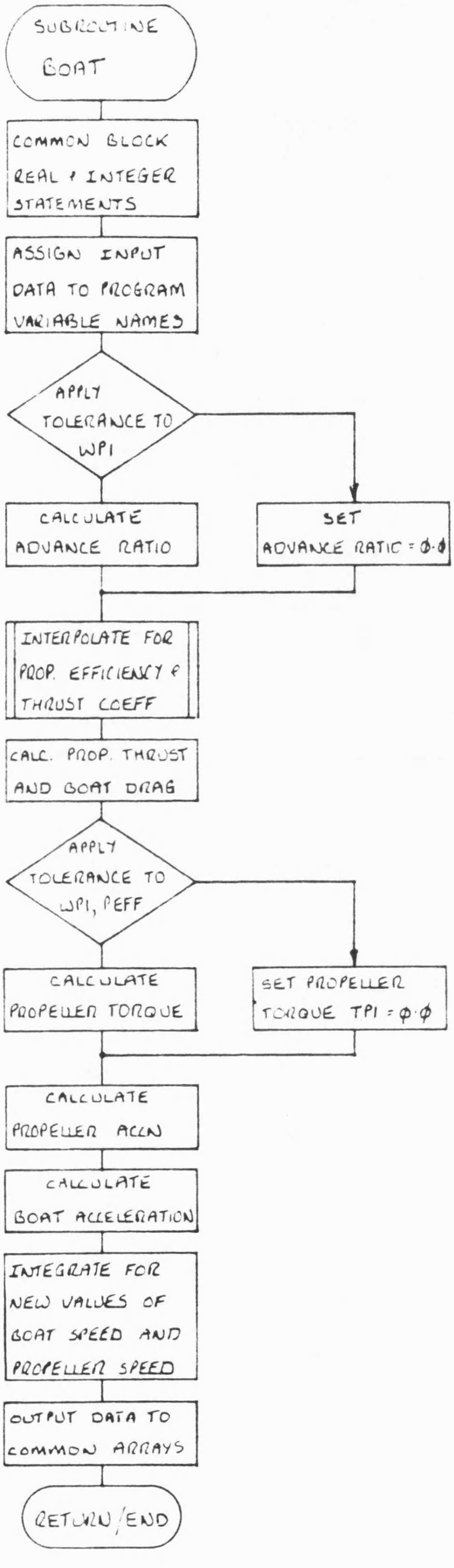

FIGURE 2 FLOWCHART FOR SUBROUTINE BOAT

```
      subroutine boat(it,itm,dt,mno)
c
c     subroutine name  boat
c
c     library classification
c
c     title  load model of a propeller driven boat
c
c     author  c.m. skarbek-wazynski
c
c     purpose   this subroutine is a load model of a propeller driven
c               boat for use in conjunction with subroutine motor. it
c               calculates the load torques developed by the propeller
c               and determines the acceleration of the propeller shaft
c               to which the hydraulic motor is directly connected.
c               integration gives a new value of shaft speed which is
c               transferred to the motor subroutine where the resulting
c               load pressures and flows are calculated.
c               (subroutine only valid for +ve propeller speeds)
c
c     associated subroutines
c     motor    method of characteristics model of a hydrostatic motor
c     intrp1   smoothed one dimensional interpolation and linear
c              extrapolation subroutine
c
c     common blocks
c     common/blk9/ hmd    motor data (constant)
c     common/blk10/hmdd   motor data (dynamic)
c     common/blk30/dlcd   boat data (constant)
c     common/blk21/dldp   boat data (dynamic)
c     common/blk19/eff    propeller efficiency values
c     common/blk20/thr    propeller thrust coefficient values
c     common/blk23/aratio corresponding advance ratio values
c
c     input information
c     input via argument list
c     dt   timestep                   s
c     it   time level indicator       --
c     itm  time level indicator       --
c     mno  component no. of drive motor   --
c
c     input via common block
c     ab    boat frontal area          m2       dlcd(3,-)
c     bms   boat mass                  kg       dlcd(5,-)
c     cd    boat drag coefficient      --       dlcd(2,-)
c     im    hydraulic motor inertia    kgm2     hmd(7,-)
c     ip    propeller inertia          kgm2     dlcd(4,-)
c     pdia  propeller diameter         m        dlcd(1,-)
c     roe   fluid density              kg/m3    dlcd(6,-)
c     tm1   previous motor torque      nm       hmdd(2,-,-)
c     vb1   previous boat velocity     m/s      dldp(1,-)
c     wp1   previous propeller speed   rad/s    dldp(2,-)
c
c     output information
c     output via common block
c     vb2   boat velocity              m/s      dldp(1,-)
c     wp2   propeller speed            rad/s    dldp(2,-)
c                                               hmdd(3,-,-)
```

```
c
c     variable names (excluding i/o variables)
c     accp    acceleration of propeller shaft        rad/s2
c     ar      advance ratio                          --
c     fd1     previous value of boat drag            n
c     ft1     previous value of propeller thrust     n
c     peff    propeller efficiency                   --
c     tc      thrust coefficient                     --
c     tp1     previous value of propeller torque     n
c     vbdot   acceleration of boat                   m/s2
c
      common /blk9/ hmd(11,2)
      common/blk21/ dldp(2,10)
      common/blk10/hmdd(3,10,2)
      common /blk28/ aratio(31)
      common /blk19/ eff(31)
      common /blk20/ thr(31)
      common /blk30/ dlcd(6)
      real ip,kpm,im
      integer opn
c
c     input data
      pdia=dlcd(1)
      cd  =dlcd(2)
      ab  =dlcd(3)
      ip  =dlcd(4)
      bms =dlcd(5)
      roe =dlcd(6)
      im  =hmd(7,mno)
      vb1=dldp(1,itm)
      wp1=dldp(2,itm)
      tm1=hmdd(2,itm,1)
c
c     calculate advance ratio
c     apply tolerance
      if(wp1.lt.1.0e-30)go to 100
      ar=vb1/(wp1*pdia)
      go to 200
  100 ar=0.0
  200 continue
c
c     propellor efficiency interpolation (smoothed)
      call intrp1(ar,peff,aratio,eff,31)
c
c     thrust coefficient interpolation (smoothed)
      call intrp1(ar,tc,aratio,thr,31)
c
c     calculate propeller thrust and boat drag
      ft1=tc*roe*wp1*wp1*pdia**4
      fd1=0.5*roe*vb1*vb1*cd*ab
c
```

```
c     apply tolerance
      if(wp1.lt.1.0e-10)go to 300
      if(peff.lt.1.0e-10)go to 300
c
c     calculate propeller torque
      tp1=ft1*vb1/(wp1*peff)
      go to 400
  300 tp1=0.0
  400 continue
c
c     calculate propeller acceleration
      accp=(tm1-tp1)/(ip+im)
c
c     calculate boat acceleration
      vbdot=(ft1-fd1)/bms
c
c     integrate for boat speed and propeller speed
      vb2=vbdot*dt+vb1
      wp2=accp*dt+wp1
c
c     output data
      dldp(1,it)=vb2
      dldp(2,it)=wp2
      hmdd(3,it,mno)=wp2
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'JUNC'

Method of characteristics model of a cross line relief valve and boost pump circuit

Purpose

Subroutine JUNC is an idealised model which simulates the steady state effects of a crossline relief valve and boost pump circuit in a network of pipes modelled using the method of characteristics. For computing convenience the behaviour of the circuit (Fig. 1) is modelled separately for the supply line and the return line by a component JUNC. This simplification is valid provided both lines are not above relief valve cracking pressure simultaneously. Separate calls to subroutine JUNC are required to calculate effects in each line.

No associated subroutines

CALL JUNC(IT,ITM,DT,RHO,JN)

JN = component number of junction model

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK25/DJNC	relief valve circuit (constant)

User defined information

ARRAY DJNC(7,JN)

DJNC(1,JN) = IPN	inlet pipe number	-	
DJNC(2,JN) = OPN	outlet pipe number	-	I
DJNC(3,JN) = PCNR	check valve cracking pressure	N/M^2	R
DJNC(4,JN) = PB	boost pressure	N/M^2	R
DJNC(5,JN) = GNRV	check valve gradient	$M^3/S/(N/M^2)$	R
DJNC(6,JN) = PCRV	relief valve cracking pressure	N/M^2	R
DJNC(7,JN) = GRV	relief valve gradient	$M^3/S/(N/M^2)$	R

Output information (via common block)

PI	pressure at junction	PDYN(1,NCPI,IT,IPN)	N/M^2	R
		PDYN(1,1,IT,OPN)	N/M^2	R
QI	flow into junction	PDYN(2,NCPI,IT,IPN)	M^3/S	R
QO	flow out of junction	PDYN(2,1,IT,OPN)	M^3/S	R

Program action and algorithm

Mathematical model

The boost pump and relief valve circuit is idealised by neglecting all dynamic effects in the circuit pipes and by assuming instantaneous response of all the values. Further more the boost pump is assumed to supply unlimited flow at a constant pressure specified by the user. In other words the boost pump and its relief valve is taken to be a constant pressure source or sink, allowing the supply and return lines to be treated separately.

Consider the supply line and the circuit in Figure 2 , three operating conditions exist.

(1) Boost flow ($P_I < P_B - P_{CNR}$)

The boost pump supplies flow to the junction in accordance with the check valve characteristic.

$$Q_R = G_{NRV}(P_B - P_I - P_{CNR}) \qquad 1$$

(2) Relief valve flow ($P_I > P_{CRV} + P_{CNR} + P_B$)

Flow passes from junction through the check valve and the relief valve.

$$Q_R = G_{NRV}(P_I - P_{RV} - P_{CNR}) \qquad 2$$

$$Q_R = G_{RV}(P_{RV} - P_B - P_{CRV}) \qquad 3$$

Equations 2 and 3 may be combined to eliminate P_{RV}

$$Q_R = G_{RV}(P_I - P_{CNR} - P_B - P_{CRV})/(1 + G_{RV}/G_{NRV}) \qquad 4$$

(3) Intermediate pressure condition ($P_B - P_{CNR} \leqslant P_I \leqslant P_{CRV} + P_{CNR} + P_B$)

No flow to or from the junction $Q_R = 0$ therefore $Q_I = Q_o$

The following equations also apply at all three conditions:-

Inlet pipeline (forward characteristic)

$$\frac{1}{\rho C_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad 5$$

outlet pipeline (backward characteristic)

$$-\frac{1}{\rho C_o}(P_I - P_s) + (v_o - v_s) + \frac{2f_s v_s |v_s| \Delta t}{d_o} = 0 \qquad 6$$

flow continuity

$$A_{PI}v_I - A_{Po}v_o + Q_R = 0 \qquad 7$$

Q_R is defined as positive when flow is into the junction.

The equations given above apply equally well to the return line, therefore the boost pump and relief valve circuit may be treated as two identical components (JUNC) with different inlet and outlet pipelines. This approach is attractive because it reduces the size of the modelling subroutine, the treatment is only unvalid in circumstances when both the supply and return lines are above relief valve cracking pressure. The relief valve then passes flow from both lines and the solution given by the model would be inaccurate due to underestimation of the relief valve pressure drop. However this condition does not arise in most systems.

Solution of equations for the three operating conditions.

Case (1) boost flow

Combining the continuity equation 7 with flow equation 1 yields equation 8 and rewriting the pipe equations 5 and 6 .

7 and 1 → $A_1v_I + A_2v_o + A_3P_I = A_4$ 8

5 → $B_1v_I + B_2P_I = B_3$ 9

6 → $C_1v_o + C_2P_I = C_3$ 10

where

$A_1 = A_{PI}$ 11

$A_2 = -A_{PO}$ 12

$A_3 = -G_{NRV}$ 13

$A_4 = G_{NRV}(P_{CNR} - P_B)$ 14

$B_1 = 1.0$ 15

$B_2 = 1.0/(\rho C_I)$ 16

$B_3 = -2f_Rv_R|v_R|\Delta t/d_I + v_R + P_R/(\rho C_I)$ 17

$C_1 = 1.0$ 18

$C_2 = -1.0/(\rho C_o)$ 19

$C_3 = -2f_sv_s|v_s|\Delta t/d_o + v_s - P_s/(\rho C_o)$ 20

Equations 8 , 9 and 10 may be solved simultaneously to give expressions

$P_I = (A_4 - A_1B_3/B_1 - A_2C_3/C_1)/(A_3 - A_2C_2/C_1 - A_1B_2/B_1)$ 21

$v_I = (B_3 - B_2P_I)/B_1$ 22

$v_o = (C_3 - C_2P_I)/C_1$ 23

Case (2) Relief valve flow

Combining equations 7 and 4 gives

7 and 4 → $A_1v_I + A_2v_o + A_3P_I = A_4$ 24

where

$A_1 = A_{PI}$ 25

$A_2 = -A_{PO}$ 26

$A_3 = -G_{RV}\,G_{NRV}/(G_{NRV} - G_{RV})$ 27

$A_4 = -A_3\,(-P_{CNR} - P_B - P_{CRV})$ 28

Equations 9 and 10 may be solved simultaneously with equation 24 and expression produced for the three unknowns P_I, v_I, v_o. However rather than using expressions 21 , 22 and 23 which are valid in this case, the order of solution has been changed and an alternative set of expressions developed (See operational status).

$$v_o = (R + P\,Q)/(A_2 + C_1B_2Q/(C_2B_1) \qquad 29$$

$$v_I = -\,(P - C_1B_2v_o/C_2B_1) \qquad 30$$

$$P_I = (B_3 - B_1P_I)/B_2 \qquad 31$$

where

$$P = (C_3B_2 - C_2B_3)/C_2B_1 \qquad 32$$

$$Q = (A_1 - A_3B_1/B_2) \qquad 33$$

$$R = (A_4 - A_3B_3/B_2) \qquad 34$$

(3) Intermediate pressure conditions

Q_R is zero, therefore rearranging the continuity equation

$$v_o = A_{PI}v_I/A_{PO}$$

The pipe equations 9 and 10 may be written

$$B_1v_I + B_2P_I = B_3 \qquad 9$$

$$10 \rightarrow \quad C_1v_I + C_2P_I = C_3 \qquad 35$$

where

$$C_1 = A_{PI}/A_{PO} \qquad 36$$

solving 9 and 10 simultaneously gives the expressions

$$P_I = (C_3 - C_1B_3/B_1)/(C_2 - C_1B_2/B_1) \qquad 37$$

$$v_I = (B_3 - B_2P_I)/B_1 \qquad 38$$

Computing procedure

All input data is assigned to program variable names from common block data arrays. Coefficients B_1, B_2, B_3, C_1, C_2 and C_3 are evaluated from expressions 15 - 20 , these coefficients remain constant during each call of the subroutine. The last known value of pressure at the junction (PI1) is checked to determine which of the three operating conditions applies, the program branches accordingly, equation coefficients are set and the pressures and flows are calculated using the appropriate sets of equations. Various tolerances are applied whilst solving the equations to ensure an accurate solution, in each case a cavitation check is performed. The final solution is output via the common block data arrays.

Operational status

Various tests have shown that the output from this subroutine is acceptably accurate. However the coding is still of an experimental nature. The program coding was written with a view to easy debugging and testing, as a result it is not particularly efficient. Tolerances have been applied to the evaluation of certain variables (the 'in program' documentation explains each tolerance fully), the need for all these tolerances has not been completely proven. In the interests of efficiency some or all of the tolerances may be omitted, however the user should then check the output from the subroutine with particular care.

LIST OF VARIABLES USED

API	area, inlet pipe	A_{PI}	M^2	R
APO	area, outlet pipe	A_{PO}	M^2	R
A1, A4	equation coefficients	A_1, A_4	-	R
B1, B3	equation coefficients	B_1, B_3	-	R
CI	wavespeed, inlet pipe	C_I	M/S	R
CO	wavespeed, outlet pipe	C_O	M/S	R

C1, C3	equation coefficients	C_1, C_3	-	R
DI	diameter, inlet pipe	d_I	M	R
DO	diameter, outlet pipe	d_O	M	R
DT	timestep	Δt	S	R
FR	friction factor, forward characteristic	f_R	-	R
FS	friction factor, backward characteristic	f_s	-	R
GNRV	check valve gradient	G_{NRV}	$M^3/S/(N/M^2)$	R
GRV	relief valve gradient	G_{RV}	$M^3/S/(N/M^2)$	R
IPN	inlet pipe number	-	-	I
IT	time level indicator	-	-	I
ITM	time level indicator	-	-	I
JN	component number of junction model	-	-	I
NCM	NCPI minus 1	-	-	I
NCPI	number of calculation points, inlet pipe	-	-	I
OPN	outlet pipe number	-	-	I
P	equation coefficient	P	-	R
PB	boost pump pressure	P_B	N/M^2	R
PCNR	check valve cracking pressure	P_{CNR}	N/M^2	R
PCRV	relief valve cracking pressure	P_{CRV}	N/M^2	R
PI	pressure at junction	P_I	N/M^2	R
PI1	old value of pressure at junction	-	N/M^2	R
PO1	old value of pressure at junction	-	N/M^2	R
PR	pressure, forward characteristic	P_R	N/M^2	R
PS	pressure, backward characteristic	P_s	N/M^2	R
Q	equation coefficient	Q	-	R
QI	volumetric flow inlet pipe	Q_I	M^3/S	R
QI1	old value of volumetric flow, inlet pipe	-	M^3/S	R
QO	volumetric flow outlet pipe	Q_O	M^3/S	R
QO1	old value of volumetric flow, outlet pipe	-	M^3/S	R
QR	volumetric flow, forward characteristic	Q_R	M^3/S	R
QS	volumetric flow, backward characteristic	Q_S	M^3/S	R
R	equation coefficient	R	-	R

RHO	fluid density	ρ	Kg/M^3	R
VI	flow velocity, inlet pipe	v_I	M/S	R
VI1	old value of flow velocity, inlet pipe	-	M/S	R
VO	flow velocity, outlet pipe	v_O	M/S	R
VO1	old value of flow velocity, outlet pipe	-	M/S	R
VR	flow velocity, forward characteristic	v_R	M/S	R
VS	flow velocity, backward characteristic	v_S	M/S	R

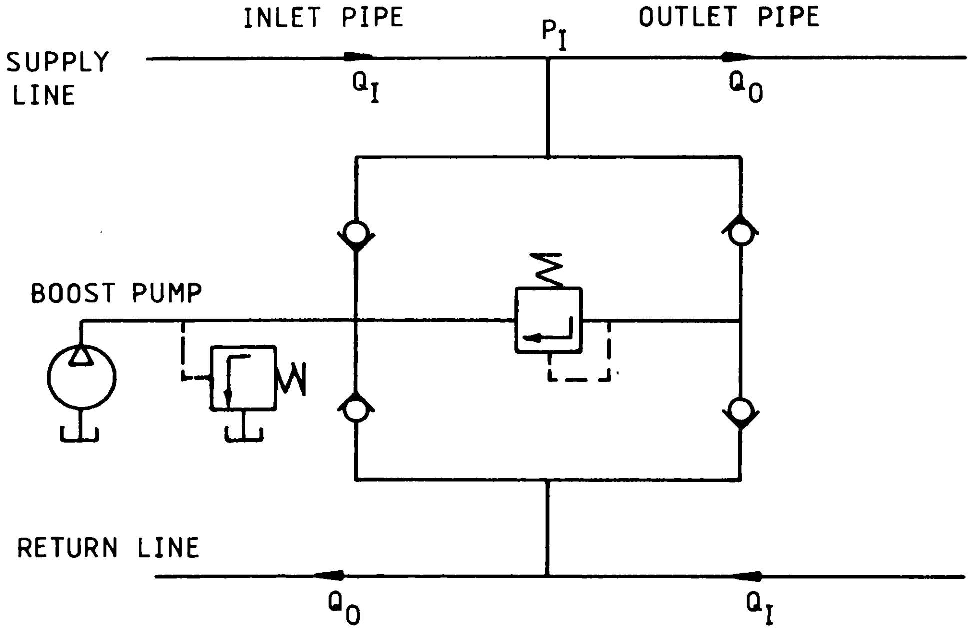

FIGURE 1 BOOST PUMP AND CROSS LINE RELIEF VALVE

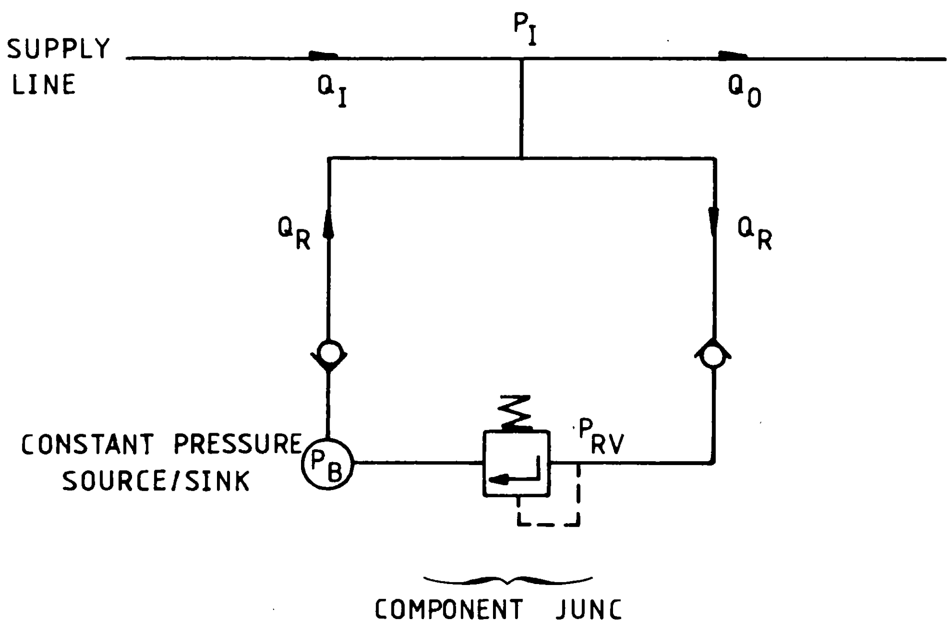

FIGURE 2 IDEALISED FORM OF BOOST PUMP AND CROSS LINE RELIEF VALVE CIRCUIT

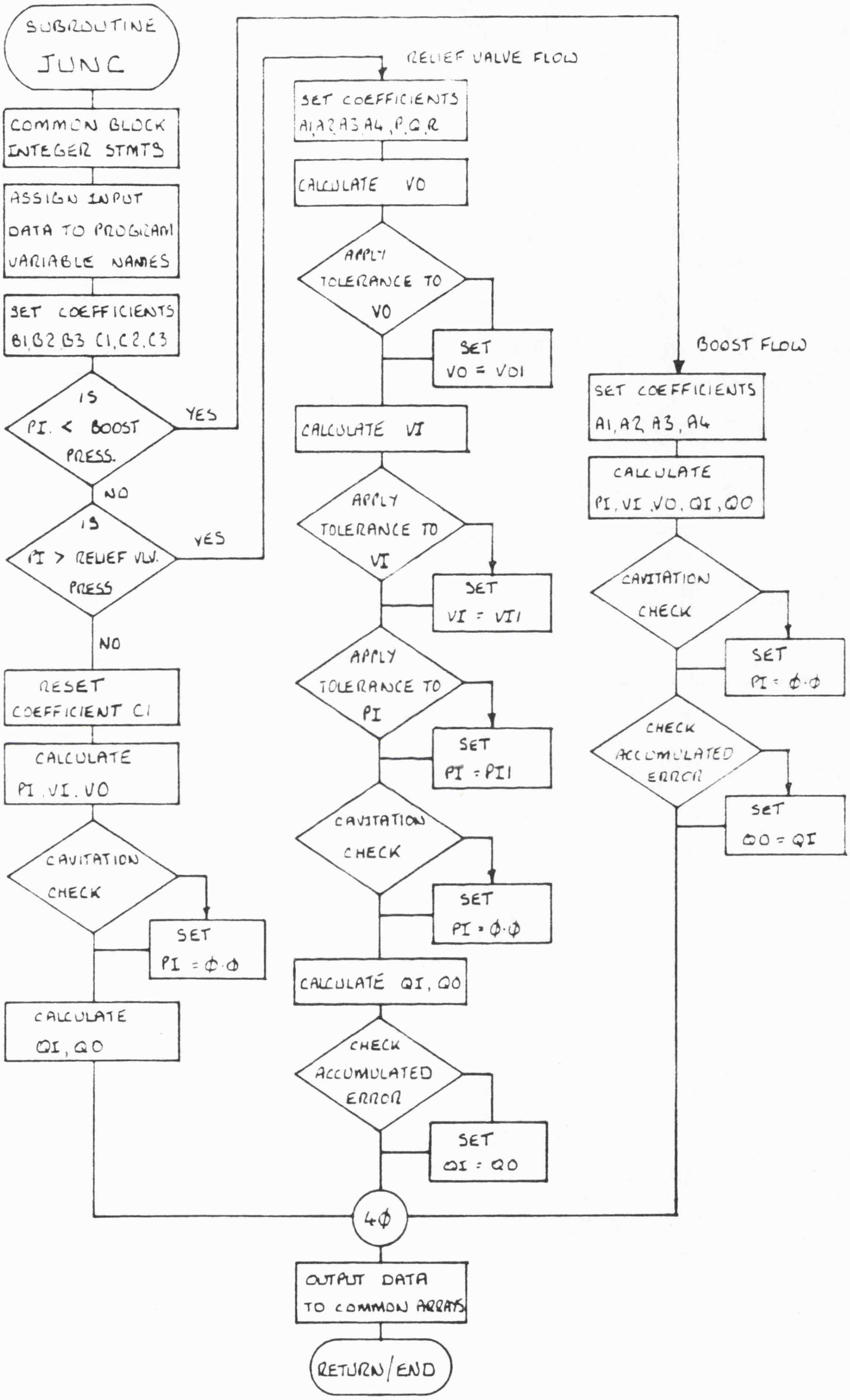

FIGURE 3 FLOWCHART FOR SUBROUTINE JUNC

```
      subroutine junc(it,itm,dt,rho,jn)
c
c     subroutine name   junc
c
c     library classification
c
c     title  method of characteristics model of a crossline relief
c            valve and boost pump circuit
c
c     author  c.m. skarbek-wazynski
c
c     purpose   subroutine junc is an idealised model which simulates
c               the steady state effects of a crossline relief valve
c               and boost pump circuit in a network of pipes modelled
c               using the method of characteristics. for computing
c               convenience the behaviour of the circuit is modelled
c               seperatly for the supply line and return line by a
c               component 'junc'. this simplification is valid provided
c               both lines are not above relief valve cracking pressure
c               simultaneously. separate calls to subroutine junc are
c               required to calculate effects in each line.
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/ pd     pipe data (constant)
c     common/blk3/ pdyn   pipe data (dynamic)
c     common/blk25/djnc   relief valve circuit data (constant)
c
c     input information
c     input via common block
c     dt     timestep                                    s
c     it     time level indicator                        --
c     itm    time level indicator                        --
c     jn     component no.of junction model              --
c     rho    fluid density                               kg/m3
c
c     input via common block
c     api  area inlet pipe                        m2     pd(2,-)
c     apo  area outlet pipe                       m2     pd(2,-)
c     ci   wavespeed inlet pipe                   m/s    pd(7,-)
c     co   wavespeed outlet pipe                  m/s    pd(7,-)
c     di   diameter inlet pipe                    m      pd(1,-)
c     do   diameter outlet pipe                   m      pd(1,-)
c     fr   friction factor  f/characteristic      --     pdyn(3,-,-,-)
c     fs   friction factor  b/characteristic      --     pdyn(3,-,-,-)
c     gnrv check valve gradient                   m5/ns  djnc(5,-)
c     grv  relief valve gradient                  m5/ns  djnc(7,-)
c     ipn  inlet pipe number                      --     djnc(1,-)
c     ncpi no of calculation points,inlet pipe    --     pd(6,-)
c     opn  outlet pipe number                     --     djnc(2,-)
c     pb   boost pump pressure                    n/m2   djnc(4,-)
c     pcnr check valve cracking pressure          n/m2   djnc(3,-)
c     pcrv relief valve cracking pressure         n/m2   djnc(6,-)
c     pi1  old value of pressure at junction      n/m2   pdyn(1,-,-,-)
c     po1  old value of pressure at junction      n/m2   pdyn(1,-,-,-)
```

```
c      pr   pressure  forward characteristic      n/m2   pdyn(1,-,-,-)
c      ps   pressure  backward charteristic       n/m2   pdyn(1,-,-,-)
c      qi1  old value of flow,inlet pipe          m3/s   pdyn(2,-,-,-)
c      qo1  old value of flow,outlet pipe         m3/s   pdyn(2,-,-,-)
c      qr   flow  forward characteristic          m3/s   pdyn(2,-,-,-)
c      qs   flow  backward characteristic         m3/s   pdyn(2,-,-,-)
c
c      output information
c      output via common block
c      pi   pressure at junction                  n/m2   pdyn(1,-,-,ipn)
c                                                        pdyn(1,-,-,opn)
c      qi   inlet pipe flow                       m3/s   pdyn(2,-,-,ipn)
c      qo   outlet pipe flow                      m3/s   pdyn(2,-,-,opn)
c
c      variable names (excluding i/o variables)
c      a1...a4  equation coefficients                          --
c      b1...b3  equation coefficients                          --
c      c1...c3  equation coefficients                          --
c      ncm      no of calc points  minus 1                     --
c      p        equations coefficient                          --
c      q        equation coefficient                           --
c      r        equation coefficient                           --
c      vi       flow velocity, inlet pipe                      m/s
c      vi1      old value of flow velocity, inlet pipe         m/s
c      vo       flow velocity, outlet pipe                     m/s
c      vo1      old value of flow velocity, outlet pipe        m/s
c      vr       flow velocity  forward characteristic          m/s
c      vs       flow velocity  backward characteristic         m/s
       common/blk1/ pd(8,10)
       common /blk3/ pdyn(3,20,10,10)
       common/blk25/ djnc(7,2)
       integer opn
c
c      input data
       ipn =djnc(1,jn)
       opn =djnc(2,jn)
       pcnr=djnc(3,jn)
       pb  =djnc(4,jn)
       gnrv=djnc(5,jn)
       pcrv=djnc(6,jn)
       grv =djnc(7,jn)
c
       di  =pd(1,ipn)
       do  =pd(1,opn)
       ci  =pd(7,ipn)
       co  =pd(7,opn)
       api =pd(2,ipn)
       apo =pd(2,opn)
       ncpi=pd(6,ipn)
       ncm=ncpi-1
c
       pi1=pdyn(1,ncpi,itm,ipn)
       qi1=pdyn(2,ncpi,itm,ipn)
       vi1=qi1/api
       po1=pdyn(1,1,itm,opn)
       qo1=pdyn(2,1,itm,opn)
       vo1=qo1/apo
       pr=pdyn(1,ncm,itm,ipn)
       fr=pdyn(3,ncm,itm,ipn)
       qr=pdyn(2,ncm,itm,ipn)
       vr=qr/api
```

```
      ps=pdyn(1,2,itm,opn)
      fs=pdyn(3,2,itm,opn)
      qs=pdyn(2,2,itm,opn)
      vs=qs/apo
c
c     calculate coefficients
      b1=1.0
      b2=1.0/(rho*ci)
      b3=-2.0*fr*vr*abs(vr)*dt/di+vr+pr/(rho*ci)
      c1=1.0
      c2=-1.0/(rho*co)
      c3=-2.0*fs*vs*abs(vs)*dt/do+vs-ps/(rho*co)
c
c     is junction pressure less than boost pressure
      if(pi1.lt.(pb-pcnr)) go to 10
c
c     is junction pressure greater than relief valve cracking pressure
      if(pi1.gt.(pcrv+pb+pcnr)) go to 20
c
c     there is no flow to or from relief valve circuit
c     reset coefficient c1
      c1=api/apo
c
c     calculate pressures and flows
      pi=(c3-c1*b3/b1)/(c2-b2*c1/b1)
      vi=(b3-b2*pi)/b1
      vo=api*vi/apo
      qi=api*vi
      qo=apo*vo
c
c     cavitation check
      if(pi.lt.0.0)pi=0.0
      go to 40
c
c     boost flow
c     calculate coefficients
   10 a1=api
      a2=-apo
      a3=-gnrv
      a4=gnrv*(pcnr-pb)
c
c     calculate pressures and flows
      pi=(a4-a1*b3/b1-a2*c3/c1)/(a3-a1*b2/b1-a2*c2/c1)
      vi=(b3-b2*pi)/b1
      vo=(c3-c2*pi)/c1
      qi=vi*api
      qo=vo*apo
c
c     cavitation check
      if(pi.lt.0.0)pi=0.0
c
c     accumulated error check
      if(qo.lt.qi)qo=qi
      go to 40
c
c      flow through relief valve
   20 a1=api
      a2=-apo
      a3=-grv*gnrv/(gnrv+grv)
      a4=-a3*(-pcnr-pb-pcrv)
      p=(c3*b2-c2*b3)/(c2*b1)
```

```
      q=a1-a3*b1/b2
      r=a4-a3*b3/b2
c
c     calculate vo
      vo=(r+p*q)/(a2+c1*b2*q/(c2*b1))
c     apply tolerance
      if(abs(vo-vo1).lt.1.0e-3)vo=vo1
c     calculate vi
      vi=-(p-c1*b2*vo/(c2*b1))
c     apply tolerance
      if(abs(vi-vi1).lt.1.0e-3)vi=vi1
c     calculate pi
      pi=(b3-b1*vi)/b2
c     apply tolerance
      if(abs(pi-pi1).lt.5.0)pi=pi1
c
c     cavitation check
      if(pi.lt.0.0)pi=0.0
c
      qi=vi*api
      qo=vo*apo
c
c     accumulated error check
      if(qo.gt.qi)qo=qi
c
   40 continue
c
c     output data to arrays
      pdyn(1,ncpi,it,ipn)=pi
      pdyn(2,ncpi,it,ipn)=qi
      pdyn(1,1,it,opn)=pi
      pdyn(2,1,it,opn)=qo
c
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'ENGINE'

Governed diesel engine model

Purpose

This subroutine is a model of a governed diesel engine connected to a hydrostatic pump. The pump and engine inertias are lumped together and simple Euler integration is used to determine the instantaneous pump-engine speed. The governor characteristics at a given setting are calculated by subroutine GOVINT.

Associated subroutines

GOVINT	interpolation subroutine to obtain governor droop equation
PUMP	method of characteristics model of a hydrostatic pump

CALL ENGINE(IT,ITM,NPMP,NO)

NPMP = component number of pump to which engine is connected (integer)

NO = engine component number (integer)

Common block data arrays

COMMON/BLK22/ENG	engine data (constant)
COMMON/BLK23/ENGD	engine data (dynamic)
COMMON/BLK6/PMP	pump data (constant)
COMMON/BLK7/PMPD	pump data (dynamic)

User defined information

ARRAY ENG(3,NO)

ENG(1,NO) = IE	engine inertia	KgM^2	R
ENG(2,NO) = GS	governor setting	rpm	R

ARRAY ENGD(2,IT,NO)

ENGD(1,1,NO) = TE1	engine torque, initial value	NM	R
ENGD(2,1,NO) = WP1	engine speed, initial value	rad/s	R

Note: the user is only required to specify initial values, subsequent values are calculated by the program.

Output information via the common block

TE2	engine torque, subsequent value	ENGD(1,IT,NO)	Nm	R
WP2	engine torque, subsequent value	PMPD(1,IT,NPMP)	rad/s	R

Program action and algorithm

Mathematical model

A governed diesel engine is assumed to have a linear torque v speed characteristic, with the general form

$$N_E = G_{RAD}\, T_{E2} + G_S \qquad 1$$

In this model the governor characteristics are stored in an array and a separate subroutine GOVINT is used to calculate the governor gradient at a given governor setting. The engine is assumed to be a pure inertia connected directly to a pump. The instantaneous speed of the pump and engine is determined by the following procedure.

A torque balance on the pump-engine shafts yields the following expression.

$$(T_{E1} - T_p) = (I_E + I_p)\frac{d\omega_p}{dt} \qquad 2$$

Assuming that values of torque and shaft speed are known, equation 2 may be integrated to give a value of shaft speed a timestep Δt later. The easiest procedure is simple Euler integration where by equation 2 becomes

$$(T_{E1} - T_{p1}) = (I_E + I_p)\frac{(\omega_{p2} - \omega_{p1})}{\Delta t} \qquad 3$$

rearranging 3

$$\omega_{p2} = \frac{(T_{E1} - T_{p1})\Delta t}{(I_p + I_E)} + \omega_{p1} \qquad 4$$

Thus knowing previous values of torque and speed the current value of shaft speed can be calculated. This is transferred to subroutine PUMP via common block where the current value of pump torque is evaluated. The current engine torque is obtained from the governor characteristic equation.

rearranging 1

$$T_{E2} = \frac{(N_E - G_s)}{G_{RAD}} \qquad 5$$

Computing procedure

The computing procedure is very straightforward and may be followed by examining the program listing. Data is assigned to program variable names from common block arrays. The current shaft speed is calculated from equation 4 and the engine torque from equation 5 . This data is assigned to the appropriate common block arrays.

LIST OF VARIABLES USED

DT	timestep	Δt	s	R
GRAD	gradient of governor drop	G_{RAD}	rpm/Nm	R
GS	governor setting	G_s	rpm	R
IE	engine inertia	I_E	Kgm^2	R
IP	pump inertia	I_p	Kgm^2	R
IT	time level indicator time = t	-	-	I
ITM	time level indicator time = t - Δt	-	-	I
NO	engine component number	-	-	I
NPMP	pump component number	-	-	I
TE1	engine torque, initial value	T_{E1}	NM	R
TE2	engine torque, subsequent value	T_{E2}	NM	R
TP1	pump torque, intial value	T_{p1}	NM	R
WP1	pump and engine speed initial value	ω_{p1}	rad/s	R
WP2	pump and engine speed subsequent value	ω_{p2}	rad/s	R

```
      subroutine engine(it,itm,dt,npmp,no)
c
c     subroutine name   engine
c
c     library classification
c
c     title  governed diesel engine model
c
c     author  c.m. skarbek-wazynski
c
c     purpose   this subroutine is a model of a governed diesel engine
c               connected to a hydrostatic pump. the pump and engine
c               inertias are lumped together and simple eule
c               integration is used to determine the instantaneous
c               pump-engine speed. the governor characteristics at a
c               given setting are calculated by subroutine govint
c
c     associated subroutines
c     govint  interpolation subroutine to obtain governor droop equation
c     pump    method of characteristics model of a hydrostatic pump
c
c     common blocks
c     common/blk22/eng   engine data (constant)
c     common/blk23/engd  engine data (dynamic)
c     common/blk6/ pmp   pump data  (constant)
c     common/blk7/ pmpd  pump data  (dynamic)
c
c     input information
c     input via argument list
c     dt    timestep                  s
c     it    time level indicator      --
c     itm   time level indicator      --
c     no    engine component number   --
c     npmp  pump component number     --
c
c     input via common block
c     grad  gradient of governor droop       rpm/nm      eng(3,-)
c     gs    governor setting                 rpm         eng(2,-)
c     ie    engine inertia                   kgm2        eng(1,-)
c     ip    pump inertia                     kgm2        pmp(5,-)
c     te1   initial value of engine torque   nm          engd(1,-,-)
c     tp1   initial value of pump torque     nm          pmpd(2,-,-)
c     wp1   initial pump/engine speed        rad/s       pmpd(1,-,-)
c
c     output information
c     output via common block
c     te2   engine torque             nm       engd(1,-,-)
c     wp2   pump/engine speed         rad/s    engd(2,-,-)
c                                              pmpd(1,-,-)
c
c
      common /blk22/ eng(3,2)
      common /blk23/ engd(2,10,2)
      common /blk7/  pmpd(3,10,2)
      common /blk6/  pmp(10,2)
      real ip,ie
c
```

```
c     input data
      ie  =eng(1,no)
      gs  =eng(2,no)
      grad=eng(3,no)
      ip =pmp(5,npmp)
      tp1=pmpd(2,itm,npmp)
      wp1=pmpd(1,itm,npmp)
      te1=engd(1,itm,no)
c
c     integrate for prime mover speed
      wp2=((te1-tp1)/(ip+ie))*dt+wp1
c
c     calculate prime mover torque
c     the coeff. 9.54929 converts rad/s to rpm
      te2=(9.54929*wp2-gs)/grad
c
c     output data
      pmpd(1,it,npmp)=wp2
      engd(1,it,no)  =te2
      engd(2,it,no)=wp2
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'GOVINT'

Interpolation subroutine to obtain governor droop equation

Purpose

The user is required to define the gradient of the governor droop at five governor settings. This subroutine calculates the gradient at any intermediate setting by linear interpolation. The subroutine needs to be called only once at the start of the program.

Associated subroutines

ENGINE governed diesel engine model

CALL GOVINT(NENG)

NENG component number of engine model for which the governor gradient is to be calculated

Common block data arrays

COMMON/BLK24/GOV governor gradients

COMMON/BLK22/ENG engine data

User defined information

ARRAY GOV(5)

GOV(1-5) five values of governor droop gradient at governor settings of 800, 1200, 1600, 2000, 2400 rpm respectively rpm/Nm R

Note: the governor setting is specified as a part of the engine data and is held in array ENG. Transfer to subroutine GOVINT is made automatically via common block.

Output information via common block

GRAD governor droop gradient at specified governor setting ENG(3,NENG) rpm/Nm R

Program action and algorithm

A simple linear interpolation procedure is followed. The diesel governor setting is assigned to variable GS from common block array ENG. The variable PDS is the position of GS with respect to the five input governor settings at which gradient values are stored in array GOV. The two values of gradient straddling the required governor setting are determined and the corresponding gradient is given by linear interpolation.

```
c
c     subroutine name  govint
c
c     library classification
c
c     title    interpolation subroutine to obtain governor droop equ.
c
c     author  c.m. skarbek-wazynski
c
c     purpose  the user is required to define the gradient of the
c              governor droop at five governor settings. this subroutine
c              calculates the gradient at any intermediate setting by
c              linear interpolation.
c              (the subroutine needs to be called only once at the start
c              of the program)
c
c     associated subroutines
c     engine   governed diesel engine model
c
c     common blocks
c     common/blk24/gov   governor gradients
c     common/blk22/eng   engine data (constant)
c
c     input information
c     input via argument list
c     neng  component number of associated engine model
c
c     input via common block
c     gs    required governor setting          rpm      eng(2,-)
c
c     output information
c     output via common block
c     grad  governor droop gradient            rpm/nm   eng(3,-)
c
c     variable names (excluding i/o variables)
c     dgrad  difference between gradu and gradl
c     gradl  gradient at nearest setting below that specified
c     gradu  gradient at nearest setting above that specified
c     npos   truncated value of pos
c     npos1  value npos plus one
c     pos    position of required setting
c
      common /blk24/ gov(5)
      common /blk22/ eng(3,2)
c
c      input data
      gs=eng(2,neng)
c
c     calculation
      pos=gs/400.0-1.0
      npos=pos
      npos1=npos+1
      npos1=npos+1
      gradu=gov(npos1)
      gradl=gov(npos)
      dgrad=gradu-gradl
      grad=gradl+dgrad*(pos-npos)
c
c     output data
      eng(3,neng)=grad
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'LOADCH'

Simple inertia load model

Purpose

Subroutine LOADCH models a simple inertia load plus viscous friction for use in conjunction with subroutine MOTOR.

Associated subroutines

MOTOR   method of characteristics model of a hydrostatic motor

CALL LOADCH(IT,ITM,DT,LNO,MNO)

LNO   load model component number (integer)

MNO   component number of motor driving the load (integer)

Common block data arrays

COMMON/BLK32/RLD	load model data (constant)
COMMON/BLK33/RLDD	load model data (dynamic)
COMMON/BLK10/HMDD	motor data (dynamic)

User defined information

ARRAY RLD(2,LNO)

RLD(1,LNO) = RIL	load inertia	Kgm^2	R
RLD(2,LNO) = RKVISC	viscous friction coefficient	Nm/(rad/s)	R

ARRAY RLDD(1,IT,LNO)

RLDD(1,1,LNO) = WL1	initial load speed	rad/s	R

Note: the user is only required to specify the initial value, subsequent values are calculated by the program.

Output information via the common block

WL2	new value of load speed	RLDD(1,IT,LNO) HMDD(3,IT,MNO)	rad/s	R

Program action and algorithm

The load is defined as obeying the equation

$$I\dot{\omega}_L = T_m - K_V\omega_L \qquad 1$$

The acceleration term $\dot{\omega}_L$ is integrated using simple Euler

$$I_L\frac{(\omega_{L2} - \omega_{L1})}{\Delta t} = T_m - K_V\omega_{L1}$$

hence

$$\omega_{L2} = \frac{(T_m - K_V\omega_{L1})\Delta t}{I_L} + \omega_{L1} \qquad 2$$

Equation 2 is solved directly in the subroutine and the new value of load speed ω_{L2} is output to the common block data arrays.

Optional status

Subroutine LOADCH does not take account of the hydraulic motor inertia, and the user should make an allowance when specifying RIL. It is assumed that the value of the timestep (DT) is sufficiently small to give accurate integration. Checks are not carried out to ensure a stable solution. The user should make certain that DT is at least 1/100th of the system period.

LIST OF VARIABLES USED

DT	timestep	Δt	S	R
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
LNO	load model component number	-	-	I
MNO	motor model component number	-	-	I
RIL	load inertia	I	Kgm^2	R
RKVISC	viscous friction coefficient	K_V	Nm/(rad/s)	R
TM1	motor torque	T_m	Nm	R
WL1	initial load speed	ω_{L1}	rad/s	R
WL2	new load speed	ω_{L2}	rad/s	R

```
      subroutine loadch(it,itm,dt,lno,mno)
c
c     subroutine name  loadch
c
c     library classification
c
c     title   simple inertia load model
c
c     author  c.m. skarbek-wazynski
c
c     purpose   subroutine loadch models a simple inertia load plus
c               viscous friction for use in conjunction with subroutine
c               motor
c
c     associated subroutines
c     motor    method of characteristics model of a hydrostatic motor
c
c     common blocks
c     common/blk32/rld    load model data (constant)
c     common/blk33/rldd   load model data (dynamic)
c     common/blk10/hmdd   motor data (dynamic)
c
c     input information
c     input via argument list
c     dt    timestep                       s
c     it    time level indicator           --
c     itm   time level indicator           --
c     lno   load model component number    --
c     mno   motor model component number   --
c
c     input via common block
c     ril      load inertia                kgm2        rld(1,-)
c     rkvisc   viscous friction coeff      nms/rad     rld(2,-)
c     tm1      motor torque                nm          hmdd(2,-,-)
c     wl1      initial load speed          rad/s       rldd(1,-,-)
c
c     output information
c     output via common block
c     wl2      load speed                  rad/s       rldd(1,-,-)
c                                                      hmdd(3,-,-)
c
      common /blk33/ rldd(1,10,2)
      common /blk10/ hmdd(3,10,2)
      common /blk32/ rld(2,2)
c
c     input data
      ril=rld(1,lno)
      rkvisc=rld(2,lno)
      tm1=hmdd(2,itm,mno)
      wl1=rldd(1,itm,lno)
c
c     integrate for new load speed
      wl2=(tm1-rkvisc*wl1)*dt/ril+wl1
c
c     output data
      rldd(1,it,lno)=wl2
      hmdd(3,it,mno)=wl2
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'CONST'

Constant speed prime mover model

Purpose

Subroutine CONST models the effect of a constant speed prime mover by setting the speed of the associated pump to a constant value specified by the user.

Associated subroutine

PUMP   method of characteristics model of a hydrostatic pump

CALL CONST(NO,IT,NPMP)

NO = constant speed prime mover number

NPMP = number of associated pump model

Common block data arrays

COMMON/BLK7/PMPD	pump data (dynamic)
COMMON/BLK34/CONS	prime mover data (constant)

User defined information

ARRAY CONS(1,NO)

CONS(1,NO) = WP2	prime mover speed constant value	rad/sec	R

Output information via common block

WP2	prime mover speed (constant)	PMPD(1,IT,NPMP)	rad/sec	R

Program action and algorithm

The program simply assigns a constant value of pump speed to array PMPD.

LIST OF VARIABLES USED

IT	time level indicator time = t	-	I

NO	prime mover component number	-	I
NPMP	number of associated pump model	-	I
WP2	prime mover speed constant value	rad/sec	R

```
      subroutine const(no,it,npmp)
c
c     subroutine name  const
c
c     library classification
c
c     title  constant speed prime mover model
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine models the effect of a constant speed
c              prime mover by directly setting the speed of the
c              associated pump model to a constant value specified by
c              the user
c
c     associated subroutine
c     pump   method of characteristics model of a hydrostatic pump
c
c     common blocks
c     common/blk7/ pmpd    pump data (dynamic)
c     common/blk34/cons    prime mover data (constant)
c
c     input information
c     input via argument list
c     it     time level indicator
c     no     prime mover component number
c     npmp   associated pump model number
c
c     input via common block
c     wp2     prime mover speed         rad/s     cons(1,-)
c
c     output information
c     output via common block
c     wp2     prime mover speed         rad/s     pmpd(1,-,-)
c
      common /blk7/ pmpd(3,10,2)
      common /blk34/ cons(1,2)
c
      wp2=cons(1,no)
      pmpd(1,it,npmp)=wp2
c
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'SOURCE'

Method of characteristics model of a constant pressure flow source

Purpose

Subroutine 'SOURCE' models a constant pressure flow source at the upstream end of a pipeline, i.e. a tank supplying flow.

No associated subroutines

CALL SOURCE(IT,ITM,DT,RHO,OPN,NSC)

OPN	outlet pipe number (integer)
NSC	component number of source model (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK31/SCE	source model data

User defined information

ARRAY SCE(1,NSC)

SCE(1,NSC) = PT	constant pressure at source	N/M^2	R

Output information via common block

PT	constant pressure at source	PDYN(1,1,IT,OPN)	N/M^2	R
VO*APO	flow at source	PDYN(2,1,IT,OPN)	M^3/S	R

Program action and algorithm

A constant pressure is imposed at the upstream boundary of a pipeline modelled using the method of characteristics. The flow is determined by the pipe equation.

Pipe equation (backward characteristic)

$$-\frac{1}{\rho C_o}(P_T - P_s) + (v_o - v_s) + \frac{2 f_s v_s |v_s| \Delta t}{d_o} = 0 \qquad 1$$

Where P_T is the constant pressure at the source. Equation 1 is rearranged to solve directly for v_o.

$$v_o = v_s + \frac{(P_T - P_s)}{\rho C_o} - \frac{2f_s v_s |v_s| \Delta t}{d_o} \qquad 2$$

LIST OF VARIABLES USED

APO	area of outlet pipe	A_{po}	M^2	R
CO	wavespeed in outlet pipe	C_o	M/S	R
DO	diameter of outlet pipe	d_o	M	R
DT	timestep	Δt	S	R
FS	friction factor, backward characteristic	f_s	-	R
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
NSC	component number of source model	-	-	I
OPN	outlet pipe number	-	-	I
PS	pressure, backward characteristic	P_s	N/M^2	R
PT	constant pressure at source	P_T	N/M^2	R
QS	flow, backward characteristic	Q_s	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
VO	flow velocity at source	v_o	M/S	R
VS	flow velocity, backward characteristic	v_s	M/S	R

```
      subroutine source(it,itm,dt,rho,opn,nsc)
c
c     subroutine name   source
c
c     library classification
c
c     title   method of characteristics  model of a constant pressure
c             flow source
c
c     author  c.m. skarbek-wazynski
c
c     purpose   subroutine source models a constant pressure flow source
c               at the upstream end of a pipeline, ie. a tank supplying
c               flow
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/ pd      pipe data (constant)
c     common/blk3/ pdyn    pipe data (dynamic)
c     common/blk31/sce     source data
c
c     input information
c     input via argument list
c     dt   timestep                         s
c     it   time level indicator             --
c     itm  time level indicator             --
c     nsc  number of source model           --
c     opn  pipe number (outlet pipe)        --
c     rho  fluid density                    kg/m3
c
c     input via common block
c     apo   outlet pipe area                   m2       pd(2,-)
c     co    wavespeed,outlet pipe              m/s      pd(7,-)
c     do    diameter, outlet pipe              m        pd(1,-)
c     fs    friction factor bkwrd/charac       --       pdyn(3,-,-,-)
c     pt    source pressure                    n/m2     sce(1,-)
c     ps    pressure bwrd/characteristic       n/m2     pdyn(1,-,-,-)
c     qs    flow bwrd/characteristic           m3/s     pdyn(2,-,-,-)
c
c     output information
c     output via common block
c     pt        source pressure         n/m2        pdyn(1,-,-,-)
c     vo*apo    source flow             m3/s        pdyn(2,-,-,-)
c
c     variable names (excluding i/o variables)
c     vo        flow velocity outlet pipe                   m/s
c     vs        flow velocity backward characteristic       m/s
c
      common /blk1/ pd(8,10)
      common /blk3/ pdyn(3,20,10,10)
      common /blk31/ sce(1,2)
      integer opn
c
c     input data
      apo=pd(2,opn)
      do =pd(1,opn)
      co =pd(7,opn)
```

```
      pt=sce(1,nsc)
      ps=pdyn(1,2,itm,opn)
      fs=pdyn(3,2,itm,opn)
      qs=pdyn(2,2,itm,opn)
      vs=qs/apo
c
c     calculate flow velocity
      vo=vs+(pt-ps)/(rho*co)-2.0*fs*vs*abs(vs)*dt/do
c
c     output of data
      pdyn(1,1,it,opn)=pt
      pdyn(2,1,it,opn)=vo*apo
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'SINK'

Method of characteristics model of a constant pressure flow sink

Purpose

Subroutine SINK models a constant pressure flow sink at the downstream end of a pipeline, i.e. a tank accepting flow.

No associated subroutines

CALL SINK(IT,ITM,DT,RHO,IPN,NSN)

IPN	number of inlet pipe (integer)
NSN	component number of sink model (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK51/SNK	sink model data

User defined information

ARRAY SNK(1,NSN)

SNK(1,NSN) = PT	constant pressure at sink	N/M^2	R

Output information via common block

PT	constant pressure at sink	PDYN(1,NCPI,IT,IPN)	N/M^2	R
VI*API	flow at sink	PDYN(2,NCPI,IT,IPN)	M^3/S	R

Program action and algorithm

A constant pressure is imposed at the downstream boundary of a pipeline modelled using the method of characteristics. The flow is determined by the pipe equation

Pipe equation (forward characteristic)

$$\frac{1}{\rho C_I}(P_T - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad 1$$

Where P_T is the constant pressure at the sink. Equation 1 is rearranged to solve directly for v_o.

$$v_o = v_R - \frac{(P_T - P_R)}{\rho C_I} - \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad 2$$

LIST OF VARIABLES USED

API	area of inlet pipe	A_{PI}	M^2	R
CI	wavespeed in inlet pipe	C_I	M/S	R
DI	diameter of inlet pipe	d_I	M	R
DT	timestep	Δt	S	R
FR	friction factor, forward characteristic	-	-	R
IPN	inlet pipe number	-	-	I
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
NCM	NCPI minus 1	-	-	I
NCPI	number of calculation points	-	-	I
NSN	component number of sink model	-	-	I
PR	pressure, forward characteristic	P_R	N/M^2	R
PT	constant pressure at sink	P_T	N/M^2	R
QR	flow, forward characteristic	Q_R	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
VI	flow velocity at sink	v_I	M/S	R
VR	flow velocity, forward characteristic	v_R	M/S	R

```
      subroutine sink(it,itm,dt,rho,ipn,nsn)
c
c     subroutine name  sink
c
c     library classification
c
c     title   method of characteristics model of a constant pressure
c             flow sink
c
c     author  c.m. skarbek-wazynski
c
c     purpose   subroutine sink models a constant pressure flow sink
c               at the downstream end of a pipeline, ie. a tank
c               accepting flow
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/ pd     pipe data (constant)
c     common/blk3/ pdyn   pipe data (dynamic)
c     common/blk51/snk    sink data
c
c     input information
c     input via argument list
c     dt   timestep                        s
c     it   time level indicator            --
c     ipn  pipe number (inlet pipe)        --
c     itm  time level indicator            --
c     nsn  number of sink model            --
c     rho  fluid density                   kg/m3
c
c     input via common block
c     api   inlet pipe area                  m2       pd(2,-)
c     ci    wavespeed,inlet pipe             m/s      pd(7,-)
c     di    diameter, inlet pipe             m        pd(1,-)
c     fr    friction factor fwrd/charac      --       pdyn(3,-,-,-)
c     ncpi  no. of calculation points        --       pd(6,-)
c     pr    pressure fwrd/characteristic     n/m2     pdyn(1,-,-,-)
c     pt    sink pressure                    n/m2     snk(1,-)
c     qr    flow fwrd/characteristic         m3/s     pdyn(2,-,-,-)
c
c     output information
c     output via common block
c     pt        sink pressure          n/m2        pdyn(1,-,-,-)
c     api*vi    sink flow              m3/s        pdyn(2,-,-,-)
c
c     variable names (excluding i/o variables)
c     ncm       no of calc points minus 1               --
c     vi        flow velocity inlet pipe                m/s
c     vr        flow velocity forward characteristic    m/s
c
      common /blk1/ pd(8,10)
      common /blk3/ pdyn(3,20,10,10)
      common /blk51/ snk(1,2)
c
c     pipe data
      ncpi=pd(6,ipn)
      ncm=ncpi-1
```

```
      api=pd(2,ipn)
      di =pd(1,ipn)
      ci =pd(7,ipn)
      pt=snk(1,nsn)
c
      pr=pdyn(1,ncm,itm,ipn)
      fr=pdyn(3,ncm,itm,ipn)
      qr=pdyn(2,ncm,itm,ipn)
      vr=qr/api
c
c     calculate flow velocity
      vi=(pr-pt)/(rho*ci)+vr-2.0*fr*vr*abs(vr)*dt/di
c
c     output of data
      pdyn(1,ncpi,it,ipn)=pt
      pdyn(2,ncpi,it,ipn)=vi*api
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'BARMAG'

Method of characteristics model of a 3 port pressure compensated flow control valve

Purpose

This subroutine simulates a 3 port pressure compensated flow control valve. The component equations are based on experimentally derived data for a Barmag valve by BAKER (Ref. 1). The inlet and outlet pipelines are modelled using the method of characteristics. The return line to tank (bypass flow) operates at low pressure, dynamics are neglected and a constant pressure is assumed to act at the exhaust port. The dynamics of the valve spool and damper are taken into account.

Associated subroutines

BARIN data initialisation routine for subroutine BARMAG.

CALL BARMAG(IT,ITM,DT,RHO,BFL,PMIN,NV)
NV = component number of valve model (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK35/FCV	flow control valve data (constant)
COMMON/BLK36/FCVD	flow control valve data (dynamic)

User defined information

ARRAY FCV(15,NV)

FCV(1,NV)	= A	compensator spool area	M^2	R
FCV(2,NV)	= M	spool mass	Kg	R
FCV(3,NV)	= F	spool viscous damping coefficient	N/(M/S)	R
FCV(4,NV)	= KFF	effective spring stiffness due to flow forces	N/M	R
FCV(5,NV)	= KS	spring stiffness	N/M	R

FCV(6,NV)	= L	valve overlap	M	R
FCV(7,NV)	= VSC	spring chamber volume (at zero lap position)	M^3	R
FCV(8,NV)	= AD	area of damping restrictor	M^2	R
FCV(9,NV)	= Y	preset orifice opening	M	R
FCV(10,NV)	= ISC	initial valve spring compression	M	R
FCV(11,NV)	= IPN	inlet pipe number	-	I
FCV(12,NV)	= OPN	outlet pipe number	-	I
FCV(13,NV)	= PRT	return line pressure	N/M^2	R
FCV(14,NV)	= KLIN	flow coefficient, damping orifice	M^5/Ns	R

ARRAY FCVD(5,IT,NV)

FCVD(1,1,NV)	= XO	spool position	M	R
FCVD(2,1,NV)	= XDO	spool velocity	M/S	R
FCVD(3,1,NV)	= PCT	spring chamber pressure	N/M^2	R
FCVD(4,1,NV)	= PCDT	rate of change of spring chamber pressure	$(N/M^2)/S$	R
FCVD(5,1,NV)	= QTT	return line flow (bypass flow)	M^3/S	R

Note: the user is only required to specify initial values, subsequent values are calculated by the program.

Output information via common block

PI	pressure at inlet	PDYN(1,NCP,IT,IPN)	N/M^2	R
QIN	flow at inlet	PDYN(2,NCP,IT,IPN)	M^3/S	R
PO	pressure at outlet	PDYN(1,1,IT,OPN)	N/M^2	R
QV	flow at outlet	PDYN(2,1,IT,OPN)	M^3/S	R
XN	spool position	FCVD(1,IT,NV)	M	R
XD	spool velocity	FCVD(2,IT,NV)	M/S	R
PC	spring chamber pressure	FCVD(3,IT,NV)	N/M^2	R
PCD	rate of change of spring chamber pressure	FCVD(4,IT,NV)	$(N/M^2)/S$	R
QTN	return line flow, bypass flow	FCVD(5,IT,NV)	M^3/S	R

Program action and algorithm

Mathematical model

Seven equations are used to describe the behaviour of the valve.

(1) Force balance on the compensator spool

$$(P_I - P_c)A = M\ddot{x} + f\dot{x} + kx + F_c \qquad 1$$

Spool displacement x is measured from the position of zero lap. The spring stiffness term k includes the effective spring stiffness due to flow forces acting on the spool.

F_c is the constant force due to the initial spring compression with an adjustment to allow for the position from which x is measured.

(2) Bypass flow

$$Q_T = 0.0514\sqrt{\frac{2}{\rho}}\,(x)^{1.3}\sqrt{|P_I - P_T|}\ \text{sign}\ (P_I - P_T) \qquad 2$$

The bypass flow is metered by the compensator spool. The term $0.0514\sqrt{2/\rho}\,(x)^{1.3}$ is due to the shape of the valve ports. P_T is constant.

(3) Flow through the preset orifice

$$Q_v = A_{po}v_o = 0.06251(y)^{1.3}\sqrt{\frac{2}{\rho}\,|P_I - P_o|}\ \text{sign}\ (P_I - P_o) \qquad 3$$

Q_v is the required constant flow which is set by adjusting the preset orifice. The term $0.06251(y)^{1.3}$ is due to the port shape.

(4) Continuity

$$A_{pI}v_I - A_{po}v_o - Q_T = 0 \qquad 4$$

It is assumed that the bleed off flows to the ends of the compensator spool are small compared with the main line flows and may be neglected.

(5) Spring chamber volume

$$K_{LIN}\,(P_o - P_c) + A\dot{x} = \frac{V_{CH}}{B_F}\,\dot{P}_c \qquad 5$$

Equation 5 describes the dynamic performance of the spring chamber volume and the damping restrictor, the derivation is given in Appendix 1.

(6) Pipe equations

inlet pipe (forward characteristic)

$$\frac{1}{\rho C_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad 6$$

outlet pipe (backward characteristic)

$$-\frac{1}{\rho C_o}(P_o - P_s) + (v_o - v_s) + \frac{2f_s v_s |v_s| \Delta t}{d_o} = 0 \qquad 7$$

The equations above have to be solved simultaneously for the seven unknowns:-

P_I, P_o, P_c, v_I, v_o, Q_T and x. The coefficients of these equations were determined experimentally by Baker (Ref. 1). The solution procedure adopted involves the linearisation of equations 2 and 3 by small perturbations and a finite difference expansion of the dynamic equations 1 and 4 . The resulting algebraic equations are manipulated to provide expressions for the seven unknowns. A predictor-corrector method is used in the computer solution to minimise errors caused by the linearisation.

Linearisation

Consider the bypass flow equation (equation 2) for positive x and $P_I \neq P_T$ (for x<o the valve is shut and $Q_T = 0$, likewise when $P_I = P_T$, $Q_T = 0$). Partial differentiation with respect to x and P_I gives:-

$$\frac{\partial Q_T}{\partial P_I} = \frac{1}{2}\, 0.05214 \sqrt{\frac{2}{\rho}}\ (x)^{1.3} \{|P_I - P_T|\}^{-\frac{1}{2}} = DF1 \qquad 8$$

$$\frac{\partial Q_T}{\partial x} = 1.3\ 0.05214 \sqrt{\frac{2}{\rho}}\ (x)^{0.3} \{|P_I - P_T|\}^{\frac{1}{2}}\ \text{sign}\ (P_I - P_T) = DF2 \qquad 9$$

from the theory of partial differentiation:-

$$\partial Q_T = \frac{\partial Q_T}{\partial P_I}\delta P_I + \frac{\partial Q_T}{\partial x}\delta x \qquad 10$$

In a stepwise solution values of P_I, P_T and x are available from the previous time level and the partial differentials (equations 8 and 9) may be calculated using these old values. Hence writing equation 10 with the δ terms expressed as the difference between the current unknown value and the previously calculated value.

$$(Q_T - Q_{Tt}) = DF1(P_I - P_{It}) + DF2(x - x_t)$$

rearranging

$$Q_T - DF1P_I - DF2x = Q_{Tt} - DF1P_{It} - DF2x_t \qquad 11$$

where the suffix t devotes known values from the previous time level.

Equation 11 is a linearised form of the bypass equation 2 and is valid for small excursions about the operating condition defined by variables P_{It}, Q_{Tt}, x_t.

Applying the same procedure to equation 3 , partial differentiation with respect to P_I and P_o gives:-

$$\frac{\partial v_o}{\partial P_I} = \frac{1}{2}\,\frac{0.06(y)^{1.3}}{A_{po}}\sqrt{\frac{2}{\rho}}\,(|P_I - P_o|)^{-\frac{1}{2}} = DF3 \qquad 12$$

$$\frac{\partial v_o}{\partial P_o} = -\frac{1}{2}\,\frac{0.06(y)^{1.3}}{A_{po}}\sqrt{\frac{2}{\rho}}\,(|P_I - P_o|)^{-\frac{1}{2}} = -DF3 \qquad 13$$

which results in the linearised equation, for the condition $P_I \neq P_o$

$$v_o - DF3P_I + DF3P_o = v_{ot} - DF3P_{It} + DF3P_{ot} \qquad 14$$

Finite difference form of the dynamic equations

Expressing the dynamic equations in finite difference form is the same as performing a Simple Euler integration.

Consider the compensator spool force balance, rewriting in finite difference form

$$(P_I - P_c)A = M\frac{(\dot{x} - \dot{x}_t)}{\Delta t} + f\frac{(x - x_t)}{\Delta t} + Kx + F_c$$

$$(P_I - P_c)A\Delta t = M\frac{(x - x_t)}{\Delta t} - M\dot{x}_t + fx - fx_t + Kx\Delta t + F_c\Delta t$$

Rearranging

$$x_n(M + f\Delta t + K\Delta t^2) + P_I(-A\Delta t^2) + P_c(A\Delta t^2) = (Mx_t + M\dot{x}_t\Delta t + fx_t\Delta t - F_c\Delta t^2) \qquad 15$$

where

(spring stiffness plus effective spring due to flow force) $K = (K_s + K_{FF})$ 16

(spring preload) $F_c = (L \dot{+} I_{sc})K_s$ 17

Writing equation 5 in finite difference form

$$K_{LIN}(P_o - P_c) + A\frac{(x - x_t)}{\Delta t} = \frac{V_{CH}}{B_F}\frac{(P_c - P_{ct})}{\Delta t}$$

rearranging

$$P_o(K_{LIN}\Delta t) + P_c(-K_{LIN}\Delta t - \frac{V_{CH}}{B_F}) + x\,(A) = (Ax_t - \frac{V_{CH}}{B_F}P_{ct}) \qquad 18$$

SOLUTION ($x>o$, $P_I \neq P_T$, $P_I \neq P_o$)

For convenience the equations are rewritten in the following form

Spring chamber volume 18 $A_1P_c + A_2x + A_3P_o = A4$ 19

Preset orifice flow 14 $v_o + B_1P_I - B_1P_o = B_2$ 20

Bypass flow 11 $Q_T + C_1P_I + C_2x = C_3$ 21

Outlet pipe 7 $v_o + D_1P_o = D2$ 22

Inlet pipe 6 $v_I + E_1P_I = E_2$ 23

Continuity 4 $F_1v_I + F_2v_o - Q_T = 0$ 24

Spool dynamics 15 $G_1x_n - G_2P_I + G_2P_c = G_3$ 25

Coefficients A_1, A_2, A_3, etc are defined in the summary.

Solving simultaneously yields the floowing expressions

$$P_I = (-R_7/R_5 - R_3)/(R_4 + R_6/R_5) \quad 26$$

$$P_o = R_3 + R_4/P_I \quad 27$$

$$x = R_2/R_1 + A_3P_o/R_1A_1 + P_I/R_1 \quad 28$$

$$P_c = (A_4 - A_2x - A_3P_o)/A_1 \quad 29$$

$$v_o = D_2 - D_1P_o \quad 30$$

$$v_I = E_2 - E_1P_I \quad 31$$

$$Q_T = F_1v_I + F_2v_o \quad 32$$

where

$$R_1 = G_1/G_2 - A_2/A_1 \quad 33$$

$$R_2 = G_3/G_2 - A_4/A_1 \quad 34$$

$$R_3 = (B_2 - D_2)/(-D_1 - B_1) \quad 35$$

$$R_4 = B_1/(D_1 + B_1) \quad 36$$

$$R_5 = - F_2D_1 + C_2A_3/R_1A_1 \quad 37$$

$$R_6 = - F_1E_1 + C_1 + C_2/R_1 \quad 38$$

$$R_7 = F_1E_2 + F_2D_2 - C_3 + C_2R_2/R_1 \quad 39$$

The above set of expressions define the system solution for the case $x>o$, $P_I \neq P_o$, and $P_I \neq P_T$.

SOLUTION ($x \leqslant o$, and/or $P_I = P_T$)

The exhaust flow Q_T is zero and thus the effective spring stiffness due to the flow force is also zero. The set of equations modelling the value is modified.

$$A_1P_c + A_2x + A_3P_o = A_4 \quad 19$$

$$v_o + B_1P_I - B_1P_o = B_2 \quad 20$$

$$v_o + D_1P_o = D_2 \quad 22$$

$$v_I + E_1P_I = E_2 \quad 23$$

$$F_1v_I + F_2v_o = 0 \quad 40$$

$$G_1x_n - G_2P_I + G_1P_o = G_3 \quad 25$$

Solving simultaneously yields the following expressions:

$$P_I = R_1/R_2 \quad 41$$

$$P_o = (B_2 - D_2 - B_1P_I)/(-D_1 - B_1) \quad 42$$

$$v_I = E_2 - E_1P_I \quad 43$$

$$v_o = D_2 - D_1P_o \quad 44$$

$$P_c = R_3/R_4 \quad 45$$

$$x = (G_3 + G_2P_I - G_2P_c)/G_1 \quad 46$$

where

$$R_1 = (B_2 - D_2)/(- D_1 - B_1) - (F_1E_2 + F_2D_2)/F_2D_1 \quad 47$$

$$R_2 = B_1/(- D_1 - B_1) - F_1E_1/F_2D_1 \quad 48$$

$$R_3 = (A_4 - A_3P_o)/A_2 - (G_3 + G_2P_I)/G_1 \quad 49$$

$$R_4 = - G_2/G_1 + A_1/A_2 \quad 50$$

Solution for small pressure drops across the preset orifice

When the pressure drop across the preset orifice is small the linearising gradient DF3 (equations 12 and 13) tends to infinity, which leads to numerical problems. At low pressure drops the flow through the orifice is laminar and therefore obeys a linear P v Q relationship. For a given orifice setting a threshold pressure drop may be defined (P_{MIN}) below which the flow is laminar.

Thus when $P_I - P_o < P_{MIN}$ the following flow relationship is used:

$$Q_v = A_{po}v_o = K_{OR}(P_I - P_o) \quad 51$$

comparing with equations 20

$$V_o + B_1 P_I - B_1 P_o = B_2 \qquad 20$$

Equations 51 and 20 are equivalent when:-

$$B_1 = - K_{OR}/A_{po} \qquad 52$$

$$B_2 = 0 \qquad 53$$

Both solutions given above are valid with the altered expressions for B_1 and B_2. P_{MIN} and the coefficient K_{OR} are calculated for a given valve setting by the initialising subroutine BARIN.

THE PREDICTOR-CORRECTOR

The use of linearised equations in a transient solution leads to a gradual accumulation of error between the computed values and the true solution, this program uses a form of predictor-corrector to minimise this error (Vol. 1). The linearising gradients DF1, DF2, DF3 are calculated using values of pressure at the previous time level, the solution of the system equations gives a new set of pressures which are used to calculate another set of linearising gradients PDF1, PDF2, PDF3. The average of the new and old gradients is calculated and the system equations are solved again. Thus the initial set of gradients is used to predict solution, a correction is applied by recalculating the gradients and solving the system again to provide an improved solution. The pressures calculated by the second system solution are now substituted directly into the non linear flow equations 2 and 3 to calculate the flows Q_T and Q_V. Q_{IN} is obtained directly from the continuity equation. In this way the flow solution is forced to obey the non linear equations and errors are averaged out between the other component variables.

COMPUTING PROCEDURE

The computing procedure is shown in flowchart form in Figure 2 . Two separate component solutions exist:-

CASE 1 $x \leq o$ or $P_I = P_T$

CASE 2 $|P_I - P_O| < P_{MIN}$

The subroutine checks for these conditions at various stages and branches accordingly. The predictor loop is entered after assigning all common block input data to program variable names and setting a number of variables and counters. Gradients DF1, DF2, DF3 are calculated and the equation coefficients A1, A2, A3 etc. are set. The component equations corresponding to the last known operating conditions are solved. A series of IF statements checks if the valve has hit any endstops and if so the solution is modified. All calculated pressures are checked for cavitation. The predictor variables PPIT, PPOT, PXO are overwritten with new values of inlet pressure, outlet pressure and spool position and the counter IPRCR is incremented. The subroutine loops back and calculates new values for the linearising gradients which are averaged with the old values. The component equations are solved again and all cavitation and endstop checks are repeated. Now the routine branches out of the predictor loop and calculates the flows QTN, QV and QIN by direct substitution in the non linear equations. All calculated data is output via the common block.

Ref.1 BAKER, O. MSc Thesis Bath University 1981

<u>EXPRESSIONS USED IN THE SUBROUTINE</u>

$$PP = (0.06521(\dot{y})^{1.3}\sqrt{2/\rho}\ /A_{po} \quad (54)$$

$$PK = 0.0514\sqrt{2/\rho} \quad (55)$$

$$A_1 = -\ V_{CH}/B_{FL} - K_{LIN}\Delta t \quad (56)$$

$$A_2 = A \quad (57)$$

$$A_3 = K_{LIN}\Delta t \quad (58)$$

$$A_4 = Ax_t - V_{CH}P_{ct}/B_{FL} \quad (59)$$

$$B_1 = -\ DF3 \quad (60)$$

$$B_2 = v_{ot} - B_1(P_{It} - P_{ot}) \quad (61)$$

$$C_1 = -\ DF1 \quad (62)$$

$$C_2 = -\ DF2 \quad (63)$$

$$C_3 = Q_{Tt} + C_1P_{It} + C_2x_t \quad (64)$$

$$D_1 = -\ 1/\rho C_o \quad (65)$$

$$D_2 = (-\ 2f_s\ v_s|v_s|\Delta t)/d_o + v_s - P_s/\rho C_o \quad (66)$$

$$E_1 = 1/\rho C_I \quad (67)$$

$$E_2 = -\ 2f_R v_R|v_R|\ \Delta t/d_I + v_R + P_R/\rho C_I \quad (68)$$

$$F_1 = A_{PI} \quad (69)$$

$$F_2 = A_{po} \quad (70)$$

$$G_1 = M + f\Delta t + K\Delta t^2 \qquad 71$$

$$G_2 = A\Delta t^2 \qquad 72$$

$$G_3 = Mx_t + M\dot{x}_t\Delta t + fx_t\Delta t - f_c\Delta t^2 \qquad 73$$

APPENDIX 1 DYNAMIC BEHAVIOUR OF SPRING CHAMBER VOLUME

The volume of the spring chamber at any instant $\quad V_{CH} = (V_{sc} - Ax_t) \qquad 74$

where V_{sc} is the volume of the spring chamber when x = o, i.e. at the zero lap position.

Flow through the damping restictor (laminar flow) $\quad Q_{sc} = K_{LIN}(P_o - P_c) \qquad 75$

Continuity

flow into spring chamber + rate of change of spring chamber vol. = compressibility

$$Q_{sc} + A\dot{x} = \frac{V_{CH}}{B_{FL}}\frac{dP_c}{dt} \qquad 76$$

Combining equations 74 , 75 and 76

$$K_{LIN}(P_o - P_c) + A\dot{x} = \frac{V_{CH}}{B_{FL}}\dot{P}_c \qquad 5$$

The coefficient K_{LIN} is derived by matching the linear flow equation with a square law characteristic of the form:-

$$Q_{sc} = 0.64A_D\sqrt{\frac{2(P_o - P_c)}{\rho}} \qquad 77$$

LIST OF VARIABLES USED

A	compensator spool area	A	M^2	R
AD	damping restrictor area	A_D	M^2	R
API	area of inlet pipe	A_{PI}	M^2	R
APO	area of outlet pipe	A_{po}	M^2	R
A1, A4	equation coefficients	A_1, A_4	-	R
BFL	bulk modulus of hydraulic fluid	B_{FL}	N/m^2	R
B1, B2	equation coefficients	B_1, B_2	-	R
CI	wavespeed in inlet pipe	C_I	M/S	R
CO	wavespeed in outlet pipe	C_o	M/S	R
C1, C3	equation coefficients	C_1, C_3	-	R
DF1	current value of linearising gradient (equation 8)	DF1	-	R
DF2	current value of linearising gradient (equation 9)	DF2	-	R
DF3	current value of linearising gradient (equation 12)	DF3	-	R
DI	diameter inlet pipe	d_I	M	R
DO	diameter outlet pipe	d_o	M	R
DT	timestep	Δt	S	R
D1, D2	equation coefficients	D_1, D_2	-	R
E1, E2	equation coefficients	E_1, E_2	-	R
F	spool viscous damping coefficient	f	N/(M/S)	R
FR	friction factor (forward characteristic)	f_R	-	R
FS	friction factor (backward characteristic)	f_s	-	R
F1, F2	equation coefficients	F_1, F_2	-	R
G1, G3	equation coefficients	G_1, G_3	-	R
IPN	inlet pipe number	-	-	I
IPRCR	predictor-corrector counter	-	-	I
ISC	initial valve spring compression	I_{SC}	M	R
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
KFF	effective spring stiffness due to flow forces	K_{FF}	N/M	R
KLIN	flow coefficient, damping orifice	K_{LIN}	$M^3/S/(N/M^2)$	R
KOR	linear flow coefficient for preset orifice	K_{OR}	$M^3/S/(N/M^2)$	R

KS	spool spring stiffness	K_s	N/M	R
L	valve overlap	L	M	R
M	spool mass	M	Kg	R
NCM	NCP minus 1	-	-	I
NCP	number of calculation points, inlet pipeline	-	-	I
NV	component number of valve model	-	-	I
OPN	outlet pipe number	-	-	I
PC	spring chamber pressure	P_c	N/M^2	R
PCD	rate of change of spring chamber pressure (dP_{CD}/dt)	P_c	$(N/M^2)/S$	R
PCDT	previous value of dP_{CD}/dt	-	$(N/M^2)/S$	R
PCT	previous value of spring chamber pressure	-	N/M^2	R
PDF1	previous value of linearising gradient (equation 8)	-	-	R
PDF2	previous value of linearising gradient (equation 9)	-	-	R
PDF3	previous value of linearising gradient (equation 12)	-	-	R
PI	current value of inlet pressure	P_I	N/M^2	R
PIT	previous value of inlet pressure	P_{It}	N/M^2	R
PK	equation coefficient	PK	-	R
PMIN	laminar flow threshold pressure	P_{MIN}	N/M^2	R
PO	current value of outlet pressure	P_o	N/M^2	R
POT	previous value of outlet pressure	P_{ot}	N/M^2	R
PP	equation coefficient	PP	-	R
PPIT	inlet pressure predictor variable	-	N/M^2	R
PPOT	outlet pressure predictor variable	-	N/M^2	R
PR	pressure, forward characteristic	P_R	N/M^2	R
PRT	return line pressure	P_T	N/M^2	R
PS	pressure, backward characteristic	P_s	N/M^2	R
PXO	spool position predictor variable	-	M	R
QIN	current value of inlet flow	Q_I	M^3/S	R
QIT	previous value of inlet flow	-	M^3/S	R
QOT	previous value of outlet flow	-	M^3/S	R
QR	flow, forward characteristic	Q_R	M^3/S	R

QS	flow, backward characteristic	Q_s	M^3/S	R
QT	current value of bypass flow in predictor loop	Q_T	M^3/S	R
QTN	current value of bypass flow	-	M^3/S	R
QTT	previous value of bypass flow	Q_{Tt}	M^3/S	R
QV	current value of outlet flow	Q_o	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
R1, R7	equation coefficients	-	-	R
VCH	current value of spring chamber volume	V_{CH}	M^3	R
VI	current value of flow velocity at inlet	v_I	M/S	R
VIT	previous value of flow velocity at inlet	-	M/S	R
VO	current value of flow velocity at outlet	v_o	M/S	R
VOT	previous value of flow velocity at outlet	v_{ot}	M/S	R
VR	flow velocity, forward characteristic	v_R	M/S	R
VS	flow velocity, backward characteristic	v_s	M/S	R
VSC	nominal spring chamber volume	V_{sc}	M^3	R
XD	spool velocity	$\dot{x}$	M/S	R
XDO	previous value of spool velocity	$\dot{x}_t$	M/S	R
XN	current value of spool position	x	M	R
XO	previous value of spool position	x_t	M	R
Y	preset orifice opening	y	M	R

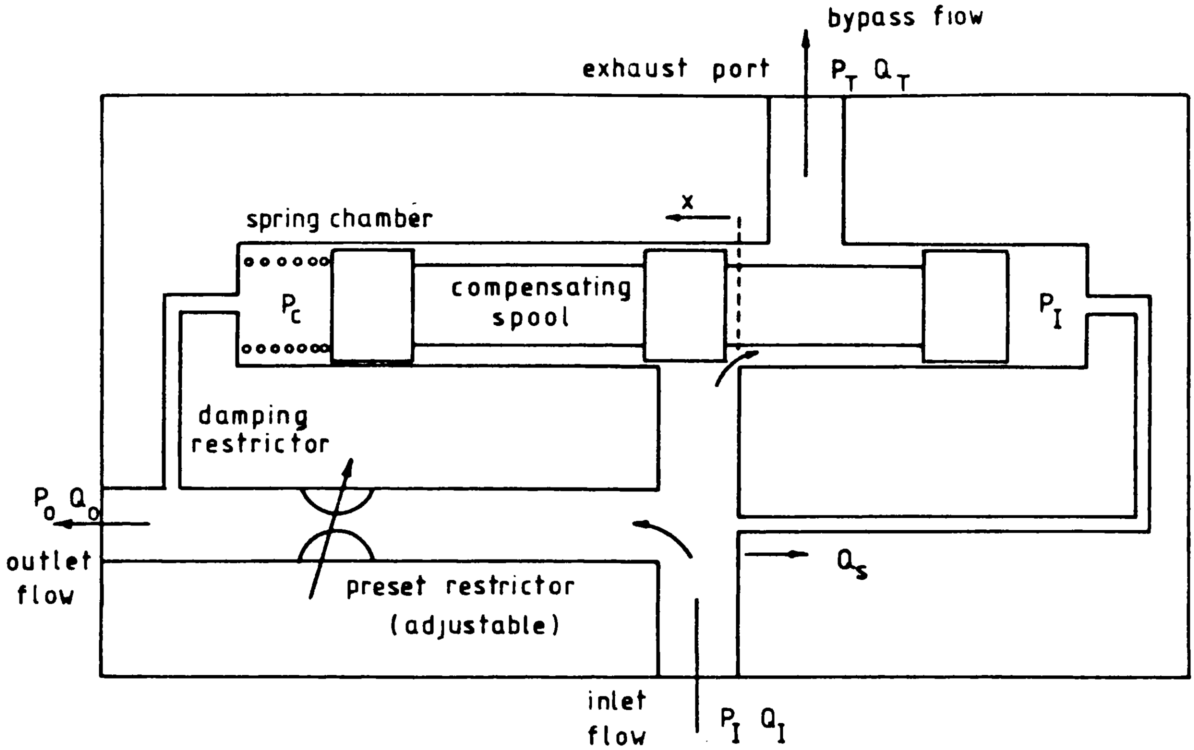

FIGURE 1 SCHEMATIC DIAGRAM OF BARMAG VALVE

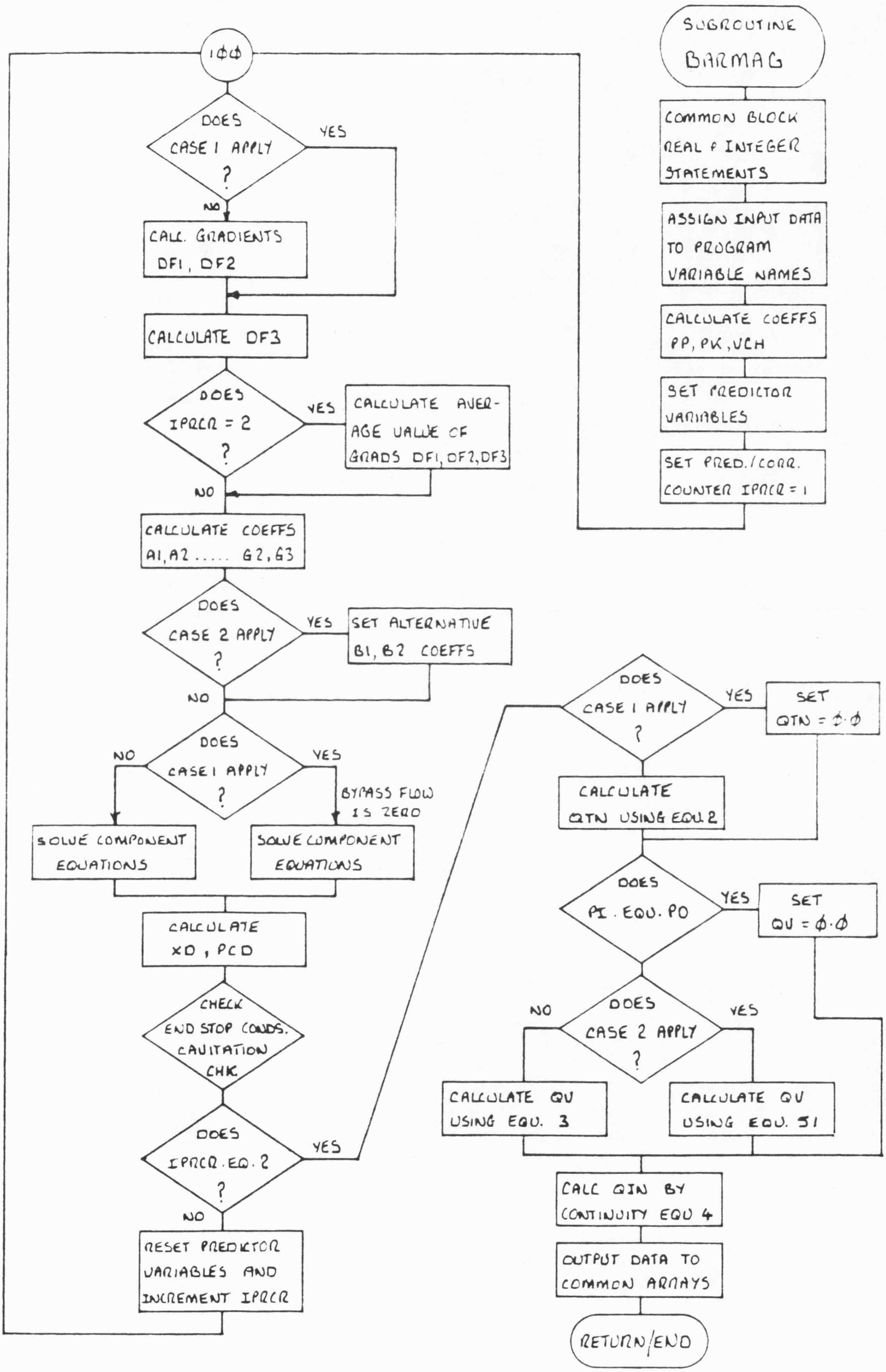

FIGURE 2 FLOWCHART FOR SUBROUTINE BARMAG

```
      subroutine barmag(it,itm,dt,rho,bfl,pmin,nv)
c
c     subroutine name  barmag
c
c     library classification
c
c     title  method of characteristics model of a 3 port pressure
c            compensated flow control valve (barmag type)
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine simulates a 3 port pressure compensated
c              flow control valve. the component equations are based on
c              experimentally derived data for a barmag valve. the
c              inlet and outlet pipelines are modelled using the method
c              of characteristics. the return line (bypass flow)
c              operates at low pressure, dynamics are neglected and a
c              constant pressure assumed to act at the exhaust port.
c              the dynamics of the valve spool and damper are taken into
c              account
c
c     common blocks
c     common/blk1/ pd      pipe data (constant)
c     common/blk3/ pdyn    pipe data (dynamic)
c     common/blk35/fcv     flow control valve data (constant)
c     common/blk36/fcvd    flow control valve data (dynamic)
c
c     input information
c     input via argument list
c     bfl   fluid bulk modulus          n/m2
c     dt    timestep                    s
c     it    time level indicator        --
c     itm   time level indicator        --
c     nv    component number            --
c     pmin  laminar threshold pressure  n/m2
c     rho   fluid density               kg/m3
c
c     input via common block
c     a    compensator spool area         m2          fcv(1,-)
c     ad   damping restrictor area        m2          fcv(8,-)
c     api  area of inlet pipe             m2          pd(2,-)
c     apo  area of outlet pipe            m2          pd(2,-)
c     ci   wavespeed in inlet pipe        m/s         pd(7,-)
c     co   wavespeed in outlet pipe       m/s         pd(7,-)
c     di   inlet pipe diameter            m           pd(1,-)
c     do   outlet pipe diameter           m           pd(1,-)
c     f    spool viscous damping coeff.   n/(m/s)     fcv(3,-)
c     fr   friction factor fwrd charac    --          pdyn(3,-,-,-)
c     fs   friction factor bkwrd charac.  --          pdyn(3,-,-,-)
c     ipn  inlet pipe number              --          fcv(11,-)
c     isc  initial valve spring comp.     --          fcv(10,-)
c     kff  spring stiffness due to flow   n/m         fcv(4,-)
c     klin flow coeff damping orifice     m5/ns       fcv(14,-)
c     kor  flow coeff preset orifice      m5/ns       fcv(15,-)
c     ks   spool spring stiffness         n/m         fcv(5,-)
c     l    valve overlap                  m           fcv(6,-)
c     m    spool mass                     kg          fcv(2,-)
```

```
c     ncp  number of calc points 1/pipe    --          pd(6,-)
c     opn  outlet pipe number              --          fcv(12,-)
c     pcdt previous diff wrt t of pc       (n/m2)/s    fcvd(4,-,-)
c     pct  previous spring chamb. press.   n/m2        fcvd(3,-,-)
c     pit  previous inlet pressure         n/m2        pdyn(1,-,-,-)
c     pot  previous outlet pressure        n/m2        pdyn(1,-,-,-)
c     pr   pressure fwrd caracteristic     n/m2        pdyn(1,-,-,-)
c     prt  return line pressure            n/m2        fcv(13,-)
c     ps   pressure bkwrd characteristic   n/m2        pdyn(1,-,-,-)
c     qit  previous inlet flow             m3/s        pdyn(2,-,-,-)
c     qot  previous outlet flow            m3/s        pdyn(2,-,-,-)
c     qr   flow forward characteristic     m3/s        pdyn(2,-,-,-)
c     qs   flow backward characteristic    m3/s        pdyn(2,-,-,-)
c     qtt  previous bypass flow            m3/s        fcvd(5,-,-)
c     vsc  nom. spring chamber volume      m3          fcv(7,-)
c     xdo  previous value spool velocity   m/s         fcvd(2,-,-)
c     xo   previous value spool position   m/s         fcvd(1,-,-)
c     y    preset orifice opening          m           fcv(9,-)
c
c     output information
c     output via common block
c     pc   new spring chamber pressure     n/m2        fcvd(3,-,-)
c     pcd  new diff wrt t of pc            (n/m2)/s    fcvd(4,-,-)
c     pi   new inlet pressure              n/m2        pdyn(1,-,-,-)
c     po   new outlet pressure             n/m2        pdyn(1,-,-,-)
c     qin  new inlet flow                  m3/s        pdyn(2,-,-,-)
c     qtn  new bypass flow                 m3/s        fcvd(5,-,-)
c     qv   new outlet flow                 m3/s        pdyn(2,-,-,-)
c     xd   new spool velocity              m/s         fcvd(2,-,-)
c     xn   new spool position              m           fcvd(1,-,-)
c
c     variables (excluding i/o variables)
c     a1...a4  equation coefficients                           --
c     b1...b2  equation coefficients                           --
c     c1...c3  equation coefficients                           --
c     df1      current linearising gradient (equ 8 )           --
c     df2      current linearising gradient (equ 9 )           --
c     df3      current linearising gradient (equ 12)           --
c     d1...d2  equation coefficients                           --
c     e1...e2  equation coefficients                           --
c     f1...f2  equation coefficients                           --
c     g1...g2  equation coefficients                           --
c     iprcr    predictor-corrector counter                     --
c     ncm      no of calc points minus 1                       --
c     pdf1     previous linearising gradient (equ 8)           --
c     pdf2     previous linearising gradient (equ 9)           --
c     pdf3     previous linearising gradient (equ 12)          --
c     pk       equation coefficient                            --
c     pp       equation coefficient                            --
c     ppit     inlet pressure predictor variable               n/m2
c     ppot     outlet pressure predictor variable              n/m2
c     pxo      spool position predictor variable               m
c     qt       current value of bypass flow                    m3/s
c     r1...r7  equation coefficients                           --
c     vch      current spring chamber volume                   m3
c     vi       current flow velocity at inlet                  m/s
c     vit      previous flow velocity at inlet                 m/s
c     vo       current flow velocity at outlet                 m/s
c     vot      previous flow velocity at outlet                m/s
c     vr       flow velocity forward characteristic            m/s
c     vs       flow velocity backward characteristic           m/s
```

```
c
      common /blk1/ pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
      common /blk35/fcv(15,2)
      common /blk36/fcvd(5,10,2)
      real m,kff,ks,l,isc,klin,kor
      integer opn
c
c     input data
c     flow control valve data
      a  =fcv(1,nv)
      m  =fcv(2,nv)
      f  =fcv(3,nv)
      kff=fcv(4,nv)
      ks =fcv(5,nv)
      l  =fcv(6,nv)
      vsc=fcv(7,nv)
      ad =fcv(8,nv)
      y  =fcv(9,nv)
      isc=fcv(10,nv)
      ipn=fcv(11,nv)
      opn=fcv(12,nv)
      prt=fcv(13,nv)
      klin=fcv(14,nv)
      kor =fcv(15,nv)
c
      xo  =fcvd(1,itm,nv)
      xdo =fcvd(2,itm,nv)
      pct =fcvd(3,itm,nv)
      pcdt=fcvd(4,itm,nv)
      qtt =fcvd(5,itm,nv)
c
c     pipe data
      ncp=pd(6,ipn)
      ncm=ncp-1
      api=pd(2,ipn)
      apo=pd(2,opn)
      di =pd(1,ipn)
      do =pd(1,opn)
      ci =pd(7,ipn)
      co =pd(7,opn)
c
      pr=pdyn(1,ncm,itm,ipn)
      fr=pdyn(3,ncm,itm,ipn)
      qr=pdyn(2,ncm,itm,ipn)
      vr=qr/api
      ps=pdyn(1,2,itm,opn)
      fs=pdyn(3,2,itm,opn)
      qs=pdyn(2,2,itm,opn)
      vs=qs/apo
c
      pit=pdyn(1,ncp,itm,ipn)
      pot=pdyn(1,1,itm,opn)
      qit=pdyn(2,ncp,itm,ipn)
      qot=pdyn(2,1,itm,opn)
      vit=qit/api
      vot=qot/apo
c
```

```
c     calculate coefficients
      pp=0.06251*(y**1.3)*sqrt(2.0/rho)/apo
      pk=0.05214*sqrt(2.0/rho)
      vch=vsc-a*xo
c
c     set predictor variables
      pxo =xo
      ppit=pit
      ppot=pot
c
c     set predictor-corrector counter
      iprcr=1
c
  100 continue
c
c     calculate gradients
c     is x less than zero or is pressure gradient zero (case 1)
      if(pxo.le.0.0.or.ppit.eq.prt)go to 30
c
      df1=0.5*pk*(pxo**1.3)/sqrt(abs(ppit-prt))
      df2=1.3*pk*(pxo**0.3)*sqrt(abs(ppit-prt))
      df2=sign(df2,(ppit-prt))
   30 continue
c
c     check if ppit-ppot is zero
      if(ppit.eq.ppot)go to 20
      df3=0.5*pp/sqrt(abs(ppit-ppot))
   20 continue
c
      if(iprcr.ge.2)go to 130
      go to 140
c
c     calculate average of gradients
  130 df1=(df1+pdf1)/2.0
      df2=(df2+pdf2)/2.0
      df3=(df3+pdf3)/2.0
  140 continue
c
c     calculate coefficients
      a1=-vch/bfl-klin*dt
      a2=a
      a3=klin*dt
      a4=a*xo-vch*pct/bfl
      b1=-df3
      b2=vot+b1*(pit-pot)
      c1=-df1
      c2=-df2
      c3=qtt+c1*pit+c2*xo
      d1=-1.0/(rho*co)
      d2=-ps/(rho*co)+vs-2.0*fs*vs*abs(vs)*dt/do
      e1=1.0/(rho*ci)
      e2=pr/(rho*ci)+vr-2.0*fr*vr*abs(vr)*dt/di
      f1=api
      f2=-apo
      g1=m+f*dt+(ks+kff)*dt*dt
      g2=a*dt*dt
      g3=m*xo+m*dt*xdo+f*dt*xo-(1+isc)*ks*dt*dt
c
c     is pressure gradient pi-po less than pmin (case 2)
      if(abs(ppit-ppot).lt.pmin)go to 150
      go to 160
```

```
c
c     set alternative coefficients
  150 continue
      b1=-kor/apo
      b2=0.0
  160 continue
c
      if(xo.le.0.0.or.pit.eq.prt)go to 40
c
c     solve component equations
      r1=g1/g2-a2/a1
      r2=g3/g2-a4/a1
      r3=(b2-d2)/(-d1-b1)
      r4=b1/(d1+b1)
      r5=(c2*a3/(r1*a1))-f2*d1
      r6=c1+c2/r1-f1*e1
      r7=f1*e2+f2*d2+c2*r2/r1-c3
c
      pi=(-r7/r5-r3)/(r4+r6/r5)
      po=-(r7+r6*pi)/r5
      xn=(a3*po/a1+pi+r2)/r1
      pc=(a4-a2*xn-a3*po)/a1
      vo=d2-d1*po
      vi=e2-e1*pi
      qt=f1*vi+f2*vo
      go to 50
c
   40 continue
c     case where bypass flow is zero
c
c     reset coefficient g1
      g1=m+f*dt +ks*dt*dt
c
c     solve component equations
c
      r1=((b2-d2)/(-d1-b1))-((f1*e2+f2*d2)/(f2*d1))
      r2=(b1/(-d1-b1))-((f1*e1)/(f2*d1))
      r3=((a4-a3*po)/a2)-((g3+g2*pi)/g1)
      r4=-g2/g1+a1/a2
c
      pi=r1/r2
      po=(b2-d2-b1*pi)/(-d1-b1)
      vi=e2-e1*pi
      vo=d2-d1*po
      pc=r3/r4
      xn=(g3+g2*pi-g2*pc)/g1
c
   50 continue
c
c     calculate rate of change of x and pc
      xd=(xn-xo)/dt
      pcd=(pc-pct)/dt
c
c     endstop checks
      if(xn.gt.3.0e-3)xn=3.0e-3
      if(xn.le.-1.239e-3)xn=-1.239e-3
      if(xn.le.-1.239e-3.and.xd.lt.0.0)xd=0.0
      if(xn.ge.3.0e-3.and.xd.gt.0.0)xd=0.0
```

```
c
c     cavitation check
      if(pi.le.0.0)pi=0.0
      if(po.le.0.0)po=0.0
      if(pc.lt.0.0)pc=0.0
c
      if(iprcr.eq.2)go to 120
c
c     reset predictor variables and increment iprcr
  110 pxo =xn
      ppit=pi
      ppot=po
      pdf1=df1
      pdf2=df2
      pdf3=df3
      iprcr=iprcr+1
      go to 100
c
  120 continue
c
c     correction for inaccuracies in linearisation
c
c     does case 1 apply?
      if(xn.le.0.0.or.pi.eq.prt)go to 90
      qtn=pk*(xn**1.3)*sqrt(abs(pi-prt))
      qtn=sign(qtn,(pi-prt))
      go to 91
   90 qtn=0.0
   91 continue
c
c     calculate flow through valve
c     check for zero pressure gradient
      if(pi.eq.po)go to 92
c     check for laminar flow (case2)
      if(abs(pi-po).lt.pmin)go to 93
      qv=0.06251*(y**1.3)*sqrt(2.0*abs(pi-po)/rho)
      qv=sign(qv,(pi-po))
      go to 94
   92 qv=0.0
      go to 94
   93 qv=kor*(pi-po)
   94 continue
c
c     calculate inlet flow
      qin=qv+qtn
c
c     output data
      fcvd(1,it,nv)=xn
      fcvd(2,it,nv)=xd
      fcvd(3,it,nv)=pc
      fcvd(4,it,nv)=pcd
      fcvd(5,it,nv)=qtn
      pdyn(1,ncp,it,ipn)=pi
      pdyn(2,ncp,it,ipn)=qin
      pdyn(1,1,it,opn)=po
      pdyn(2,1,it,opn)=qv
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'BARIN'

Data initialisation routine for subroutine Barmag

Purpose

For a given opening of the preset orifice in the Barmag valve, subroutine BARIN calculates the pressure gradient at which flow will be laminar and hence calculates a linear pressure flow coefficient to be used when laminar flow conditions exist.

No associated subroutines

CALL BARIN(RHO,AVISC,PMIN,NV)

NV	component number of corresponding valve model	-
PMIN	laminar flow threshold pressure	N/M^2

Common block data arrays

COMMON/BLK35/FCV flow control valve data (constant)

No user defined information required

Output information via argument list

PMIN	laminar flow threshold pressure	N/M^2

Output information via common block

KOR	linear flow coefficient for preset orifice	M^5/Ns	FCV(15,NV)

Program action and algorithm

Flow is assumed to be laminar when the Reynold's number for the orifice is less than 2000.

Reynold's number $$R_e = \frac{v\ y\ \rho}{\mu} \qquad 1$$

when $R_e = 2000$ flow velocity $$v_L = \frac{2000\mu}{y\ \rho} \qquad 2$$

the corresponding flow through the orifice (assuming a circular orifice) $q_L = \frac{\pi y^2 v}{4}$ 3

The non linear flow relationship for the orifice is

$$q = 0.06251\ (y)^{1.3}\sqrt{\frac{2\Delta P}{\rho}} \qquad 4$$

rearranging 4

$$\Delta P = \frac{q^2\rho}{2(0.06251y^{1.3})^2}$$

Substituting q_L in 5 gives P_{MIN} the pressure gradient which will produce a Reynold's number of 2000 at the orifice. For pressure gradients greater than P_{MIN} flow will be turbulent, for gradients less than P_{MIN} flow will be laminar.

A linear pressure flow relationship is assumed under conditions of laminar flow

$$q = K_{OR}\Delta P \qquad 6$$

Expressions 4 and 6 must yield the same value of flow at the trasition point from laminar to turbulent flow.

Therefore $$K_{OR} = \frac{q_L}{P_{MIN}} \qquad 7$$

The subroutine performs the straight forward evaluation of expressions 2 , 3 , 5 and 7 .

LIST OF VARIABLES USED

AVISC	fluid viscosity	μ	Ns/M^2	R
KOR	linear flow coefficient for preset orifice	K_{OR}	M^5/Ns	R
NV	component number of corresponding valve	-	-	I
PMIN	laminar flow threshold pressure	P_{MIN}	N/M^2	R

Q	flow through orifice	q_L	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
V	flow velocity through orifice	v	M/S	R
Y	orifice setting	y	M	R

```
      subroutine barin(rho,avisc,pmin,nv)
c
c     subroutine name barin
c
c     library classification
c
c     title  data initialisation routine for subroutine barmag
c
c     author  c.m. skarbek-wazynski
c
c     purpose   for a given opening of the preset orifice in the barmag
c               valve, subroutine barin calculates the pressure gradient
c               at which flow will be laminar and hence calculates a
c               linear pressure-flow coefficient to be used when laminar
c               for conditions exist
c
c     no associated subroutines
c
c     common blocks
c     common/blk35/ fcv   flow control valve data (constant)
c
c     input information
c     input via argument list
c     avisc  fluid viscosity                           ns/m2
c     nv     component no of corresponding valve       --
c     rho    fluid density                             kg/m3
c
c     input via common block
c     y      valve setting                             m
c
c     output information
c     output via argument list
c     pmin   laminar flow threshold pressure           n/m2
c
c     output via common block
c     kor    linear flow coefficient                   m5/ns
c
c     variables (excluding i/o variables)
c     q      flow through orifice                      m3/s
c     v      flow velocity through orifice             m/s
c
      common /blk35/fcv(15,2)
      real kor
c
c     input data
      y=fcv(9,nv)
c
      v=2000.0*avisc/(y*rho)
      q=3.14159*y*y*v/4.0
      pmin=q*q*rho/(2.0*(0.06*(y**1.3))**2)
      kor=q/pmin
c
c     output data
      fcv(15,nv)=kor
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'CPT'

Method of characteristics model of an idealised actuator load

Purpose

Subroutine CPT is an idealised model of an actuator which acts as a constant pressure load. As it extends the actuator hits an obstruction causing a ramp increase in load to another constant pressure level. (Figure 1.)

No associated subroutines

CALL CPT(IT,ITM,DT,RHO,TIME,NC)

NC = component number of pipe termination model (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK38/CP	pipe termination data

User defined information

ARRAY CP(6,NC)

CP(1,NC) = P1	constant pressure level 1	N/M^2	R
CP(2,NC) = P2	constant pressure level 2	N/M^2	R
CP(3,NC) = T1	starting time of ramp	S	R
CP(4,NC) = T2	duration of the ramp	S	R
CP(5,NC) = T3	not used	-	-
CP(6,NC) = IPN	inlet pipe number	-	I

Output information (via common block)

P	pressure level at pipe termination	PDYN(1,NCP,IT,IPN)	N/M^2	R
VI*API	volumetric flow at pipe termination	PDYN(2,NCP,IT,IPN)	M^3/S	R

Program action and algorithm

This model imposes a pressure level at the last calculation point (downstream boundary) of a pipe modelled using the method of characteristics. The pressure level used follows the duty cycle shown in Figure 1 specified by the user. The flow at the pipe termination is defined by the pipe equation.

pipe equation (forward characteristic)

$$\frac{1}{\rho C_I}(P - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad 1$$

where P is the pressure level specified, rearranging to solve for v_I

$$v_I = -\frac{1}{\rho C_I}(P - P_R) + v_R - \frac{2f_R v_R |v_R| \Delta t}{d_I} \qquad 2$$

Equation 2 is evaluated directly in the subroutine (see program listing). The required pressure level is determined by a simple IF statement.

FIGURE 1

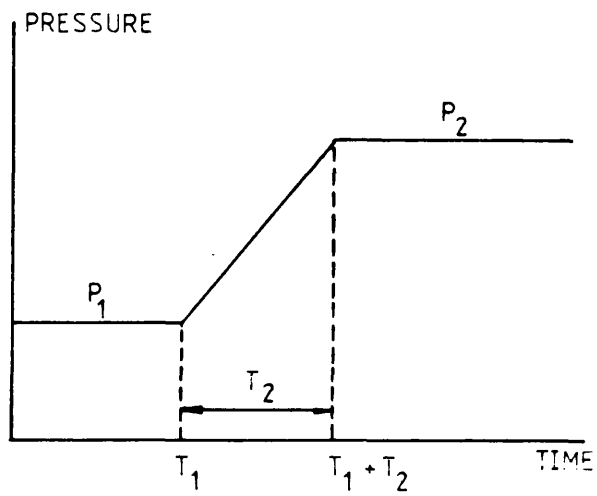

FIG. 1 LOAD MODEL DUTY CYCLE

LIST OF VARIABLES USED

API	area of inlet pipe	A_{PI}	M^2	R
CI	wavespeed in inlet pipe	C_I	M/S	R
DI	diameter of inlet pipe	d_I	M	R
DT	timestep	Δt	S	R
FR	friction factor, forward characteristic	f_R	-	R
IPN	inlet pipe number	-	-	I
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
NC	component number	-	-	I
NCM	number of calculation points minus 1	-	-	I
NCP	number of calculation points	-	-	I
P	current value of pressure	P	N/M^2	R
P1	initial pressure level	-	N/M^2	R
P2	final pressure level	-	N/M^2	R
PR	pressure, forward characteristic	P_R	N/M^2	R
QR	flow, forward characteristic	-	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
T1	starting time of ramp	-	S	R
T2	duration of ramp	-	S	R
TIME	current value of time	-	S	R
VI	flow velocity at component, time = t	v_I	M/S	R
VR	flow velocity, forward characteristic	v_R	M/S	R

```
      subroutine cpt(it,itm,dt,rho,time,nc)
c
c     subroutine name  cpt
c
c     library classification
c
c     title  method of characteristics model of a pipe termination
c            capable of ramp changes from one constant pressure level
c            to another
c
c     author  c.m. skarbek-wazynski
c
c     purpose  subroutine cpt is an idealised model of an actuator
c              which acts as a constant pressure load. as it extends
c              the actuator hits an obstruction causing a ramp increase
c              in load to another constant pressure level
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/ pd        pipe data (constant)
c     common/blk3/ pdvn      pipe data (dynamic)
c     common/blk38/cp        pipe termination data
c
c     input information
c     input via argument list
c     dt      timestep                     s
c     it      time level indicator         --
c     itm     time level indicator         --
c     nc      component number             --
c     rho     fluid density                kg/m3
c     time    time                         s
c
c     input via common block
c     api area of inlet pipe               m2          pd(2,-)
c     ci  wavespeed in inlet pipe          m/s         pd(7,-)
c     di  diameter of inlet pipe           m           pd(1,-)
c     fr  friction factor fwrd charac.     --          pdyn(3,-,-,-)
c     ipn inlet pipe number                --          cp(6,-)
c     ncp number of calculation points     --          pd(6,-)
c     p1  initial pressure level           n/m2        cp(1,-)
c     p2  final pressure level             n/m2        cp(2,-)
c     pr  pressure fwrd characteristic     n/m2        pdyn(1,-,-,-)
c     qr  flow fwrd characteristic         m3/s        pdyn(2,-,-,-)
c     t1  starting time of ramp            s           cp(3,-)
c     t2  duration of ramp                 s           cp(4,-)
c
c     output information
c     output via common block
c     p         current pressure level     n/m2        pdyn(1,-,-,-)
c     vi*api  load flow                    m3/s        pdyn(2,-,-,-)
c
c     variables (excluding i/o variables)
c     ncm  no of calc points minus 1       --
c     vi   flow velocity at load           m/s
c     vr   flow velocity fwrd charac.      m/s
c
```

```
      common /blk1/ pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
      common /blk38/cp(6,2)
c
c     input data
      p1 =cp(1,nc)
      p2 =cp(2,nc)
      t1 =cp(3,nc)
      t2 =cp(4,nc)
      ipn=cp(6,nc)
      di =pd(1,ipn)
      ci =pd(7,ipn)
      api=pd(2,ipn)
      ncp=pd(6,ipn)
      ncm=ncp-1
      pr=pdyn(1,ncm,itm,ipn)
      fr=pdyn(3,ncm,itm,ipn)
      qr=pdyn(2,ncm,itm,ipn)
      vr=qr/api
c
c     calculate current value of pressure
      p=p1
      if(time.ge.t1)go to 10
      go to 20
   10 continue
      p=p1+(p2-p1)*(time-t1)/t2
      if(p.ge.p2)p=p2
   20 continue
c
c     calculate flow velocity at load
      vi=(pr-p)/(rho*ci)+vr-2.0*fr*vr*abs(vr)*dt/di
c
c     output data
      pdyn(1,ncp,it,ipn)=p
      pdyn(2,ncp,it,ipn)=vi*api
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'CFS'

Method of characteristics model of a constant flow source

Purpose

Subroutine CFS is an idealised model of a pump, which is taken to be a source of constant flow at the upstream end of a pipe.

No associated subroutines

CALL CFS(IT,ITM,DT,RHO,NCS)

NCS = component number of constant flow source

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK37/CF	constant flow source data

User defined information

ARRAY CF(2,NCS)

CF(1,NCS) = Q	constant flow at source	M^3/S	R
CF(2,NCS) = OPN	number of pipe connected to source	-	I

Output information (via common block)

PO	pressure at flow source	PDYN(1,1,IT,OPN)	N/M^2	R
Q	constant flow at source	PDYN(2,1,IT,OPN)	M^3/S	R

Program action and algorithm

The subroutine CFS imposes a constant flow at the first calculation point (upstream boundary) of a pipe simulated using the method of characteristics. The pressure is allowed to take any value as determined by the pipe equation pipe equation (backward characteristic)

$$-\frac{1}{\rho C_o}(P_o - P_s) + \left(\frac{Q}{A_{po}} - v_s\right) + \frac{2f_s v_s |v_s| \Delta t}{d_o} = 0 \qquad 1$$

Equation 1 is rearranged to solve for P_o

$$P_o = (-\rho C_o)\left[-\frac{P_s}{\rho C_o} - \frac{Q}{A_{po}} - \frac{2f_s v_s |v_s| \Delta t}{d_o}\right] \qquad 2$$

Equation 2 is evaluated directly in the subroutine (see program listing). A cavitation check is included to ensure that negative pressures are set to zero.

LIST OF VARIABLES USED

APO	area of outlet pipe	A_{po}	M^2	R
CO	wavespeed in outlet pipe	C_o	M/S	R
DO	diameter of outlet pipe	d_o	M	R
DT	timestep	Δt	S	R
FS	friction factor, backward characteristic	f_s	-	R
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
NCS	flow source component number	-	-	I
OPN	outlet pipe number	-	-	I
PO	pressure at flow source, time = t	P_o	N/M^2	R
PS	pressure, backward characteristic	P_s	N/M^2	R
Q	constant flow level	Q	M^3/S	R
QS	flow, backward characteristic	-	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
VS	flow velocity, backward characteristic	v_s	M/S	R

```
      subroutine cfs(it,itm,dt,rho,ncs)
c
c     subroutine name   cfs
c
c     library classification
c
c     title  method of characteristics model of a constant flow source
c
c     author  c.m. skarbek-wazynski
c
c     purpose subroutine cfs is an idealised model of a pump,which is
c             taken to be a source of constant flow at the upstream end
c             of a pipe
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/ pd      pipe data (constant)
c     common/blk3/ pdyn    pipe data (dynamic)
c     common/blk37/cf      constant flow source data
c
c     input information
c     input via argument list
c     dt    timestep                s
c     it   time level indicator    --
c     itm  time level indicator    --
c     ncs  component number        --
c     rho  fluid density           kg/m3
c
c     input via common block
c     apo  outlet pipe area                   m2        pd(2,-)
c     co   wavespeed in outlet pipe           m/s       pd(7,-)
c     do   diameter of outlet pipe            m         pd(1,-)
c     fs   friction factor bkwrds.charac      --        pdyn(3,-,-,-)
c     opn  outlet pipe number                 --        cf(2,-)
c     ps   pressure backward characteristic  n/m2      pdyn(1,-,-,-)
c     q    constant flow level                m3/s      cf(1,-)
c     qs   flow backward characteristic       m3/s      pdyn(2,-,-,-)
c
c     output information
c     output via common block
c     po   pressure at flow source            n/m2      pdyn(1.-,-,-)
c     q    constant flow level                m3/s      pdyn(2,-,-,-)
c
c     variables excluding i/o variables
c     vs  flow velocity, backward charac      m/s
c
```

```
      common/blk1/ pd(8,2)
      common/blk3/ pdyn(3,400,10,2)
      common/blk37/cf(2,2)
      integer opn
c
c     input data
      q=cf(1,ncs)
      opn=cf(2,ncs)
      apo=pd(2,opn)
      co =pd(7,opn)
      do =pd(1,opn)
      ps=pdyn(1,2,itm,opn)
      fs=pdyn(3,2,itm,opn)
      qs=pdyn(2,2,itm,opn)
      vs=qs/apo
c
c     calculate pressure
      po=(-ps/(rho*co)+vs-2.0*fs*vs*abs(vs)*dt/do-q/apo)*(-rho*co)
c
c     output data
      pdyn(1,1,it,opn)=po
      pdyn(2,1,it,opn)=q
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'PRIP'

Method of characteristics model of a hydrostatic pump with flow ripple effects

Purpose

Subroutine PRIP is an idealised model of a hydrostatic piston pump which includes the effects of flow ripple at the pump outlet. The ripple is generated from experimentally determined harmonics at a given set of operating conditions. Full details of model development are given in volume 1.

Associated Subroutines

PIPE	pipe model using the method of characteristics
ZEROFL	method of characteristics model of a zero flow pipe termination

(calls to these routines are made from inside PRIP)

CALL PRIP(IT,ITM,DT,RHO,TIME,NO)

NO = pump component number (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data array (constant)
COMMON/BLK3/PDYN	pipe data array (dynamic)
COMMON/BLK44/RPM	pump data (constant)
COMMON/BLK45/DRPM	pump data (dynamic)
COMMON/BLK46/AMP,FAZE	flow ripple amplitude and phase data arrays
COMMON/BLK47/DTA	flow ripple calculation parameters

User defined information

ARRAY RPM(9,NO)

RPM(1,NO) = QID	ideal pump flow	M^3/S	R
RPM(2,NO) = QPN	outlet pipe number	-	I
RPM(3,NO) = EPN	equivalent pipeline number	-	I
RPM(4,NO) = KR	pump leakage orifice coefficient	Ns/M^5	R
RPM(5,NO) = PT	tank or case drain pressure	N/M^2	R
RPM(6,NO) = NOR	not used	-	-
RPM(7,NO) = NZF	not used	-	-

RPM(8,NO) = NP	number of pistons	-	I
RPM(9,NO) = WP	pump speed	rad/s	R

ARRAY DTA(2)

DTA(1) = T1	starting time for flow ripple	S	R
DTA(2) = THETA	angle turned through by fundamental	rad	R

ARRAY AMP(10), FAZE(10)

AMP(1-10)	amplitude values for 10 harmonics	-	R
FAZE(1-10)	phase values for 10 harmonics	deg	R

Output information via common block

PO	pump outlet pressure	PDYN(1,1,IT,OPN) PDYN(1,1,IT,EPN) DRPM(1,IT,NO)	N/M^2
APO*VO	volumetric flow into outlet pipe	PDYN(2,1,IT,OPN) DRPM(2,IT,NO)	M^3/S
APE*VV	volumetric flow into equivalent pipe	PDYN(2,1,IT,EPN)	M^3/S

Program action and algorithm

Mathematical model

The derivation of the four pump model equations is described in volume 1.

Continuity equation

$$Q_{ID} + Q_s = Q_R + Q_v + Q_o \qquad 1$$

Outlet pipeline (backward characteristic)

$$-\frac{1}{\rho C_o}(P_o - P_s) + (v_o - v_s) + \frac{2 f_s v_s |v_s| \Delta t}{d_o} = 0 \qquad 2$$

equivalent volume pipeline (backward characteristic)

$$-\frac{1}{\rho C_E}(P_o - P_E) + (v_v - v_E) + \frac{2f_E v_E |v_E| \Delta t}{d_E} = 0 \qquad 3$$

leakage orifice

$$Q_R = K_R(P_o - P_T) \qquad 4$$

The pump simulation involves solving the equations above for the four unknowns P_o, v_o, v_v and Q_R. Since only four equations are involved the simplest approach is to solve simultaneously by hand to obtain algebraic expressions for each of the unknowns from which the values can be directly calculated. Numerical simultaneous equation solvers would be inefficient in this case.

Rewriting equations 1 - 4

$$Q_R + A_1 v_v + A_2 v_o = A_3 \qquad 6$$

$$v_o + B_1 P_o = B_2 \qquad 7$$

$$v_v + C_1 P_o = C_2 \qquad 8$$

$$Q_R + D_1 P_o = D_2 \qquad 9$$

Solving simultaneously gives

$$P_o = \frac{(D_2 + A_1 C_2 + A_2 B_2 - A_3)}{(D_1 + A_1 C_1 + A_2 B_1)} \qquad 10$$

$$v_o = B_2 - B_1 P_o \qquad 11$$

$$v_v = C_2 - C_1 P_o \qquad 12$$

$$Q_R = D_2 - D_1 P_o \qquad 13$$

where

$A_1 = A_{PE}$ 14

$A_2 = A_{PO}$ 15

$A_3 = Q_{ID} + Q_s$ 16

$B_1 = -1.0/(\rho C_o)$ 17

$B_2 = -2f_s v_s |v_s| \Delta t/d_o + v_s - P_s/\rho C_o$ 18

$C_1 = -1.0/\rho C_E$ 19

$C_2 = -2f_E v_E |v_E| \Delta t/d_E + v_E - P_E/\rho C_E$ 20

$D_1 = -K_R$ 21

$D_2 = -K_R P_T$ 22

The term Q_s is the contribution to the pump flow due to the flow ripple and is directly analagous to the flow source ripple Q_s used in impedance theory. In this model the flow ripple is stored as ten experimentally determined harmonics in amplitude and phase form. At a given instant in time the value of Q_s is calculated by the summation of the flow contribution made by each of the ten harmonics.

Computing procedure

The computing procedure is shown as a flowchart in Figure 1 . After assigning all input data from common block arrays to subroutine variables the instantaneous value of flow ripple is calculated by summation as decribed above. The coefficients A_1 A_2 A_3 .... etc. are calculated using expressions 14 to 22 , the resultant values are used in equations 10 to 15 to calculate the four unknowns P_o, v_o, v_v, Q_R, which are assigned to common block arrays.

The pump impedance is modelled by the dynamic behaviour of a closed ended pipeline which is considered to be equivalent to the pump clearance volume (called the equivalent pipeline). The performance of the pipe and termination is calculated by calls to the 'PIPE' subroutine and the 'ZEROFL' subroutine.

Operational status

This subroutine was developed to investigate the possibilities of time domain simulation of pump pressure ripple. The model only considers the outlet side of the pump, and no cavitation checks are performed. Modelling the equivalent pipeline by calls to external subroutines is computationally convenient although it is inefficient. The program layout was designed to assist in debugging and testing, as a result the calculations are expressed in a longwinded and inefficient form. Future working versions of this subroutine will remedy these defects.

LIST OF VARIABLES USED

AMPL	do loop variable for ripple amplitude	-	M^3/S	R
ANGLE	angle turned through by K^{th} harmonic	-	rad	R
APE	area of equivalent volume pipeline	A_{PE}	M^2	
APO	area of outlet pipeline	A_{po}	M^2	R
A1, A3	equation coefficients	A1,A3	-	R
B1, B2	equation coefficients	B1, B2	-	R
CE	wavespeed in equivalent pipeline	C_E	M/S	R
CO	wavespeed in outlet pipeline	C_o	M/S	R
C1, C2	equation coefficient	C1, C2	-	R
DE	diameter of equivalent pipeline	d_E	M	R
DO	diameter of outlet pipeline	d_o	M	R
DT	timestep	Δt	S	R
D1, D2	equation coefficients	D1, D2	-	R
EPN	equivalent pipeline number	-	-	I
FE	friction factor, equivalent pipe, backward characteristic	f_E	-	R
FHARM	contribution to ripple made by K^{th} harmonic	-	M^3/S	R
FS	friction factor, outlet pipe, backward characteristic	f_s	-	R
IT	time level indicator (time = t)	-	-	I
ITM	time level indicator (time = t - Δt)	-	-	I
K	do loop counter	-	-	I
KR	pump leakage orifice coefficient	K_R	Ns/M^5	R
NO	pump number	-	-	I

NOR	not used	-	-	I
NP	number of pistons	-	-	I
NZF	not used	-	-	I
OPN	outlet pipe number	-	-	I
PE	pressure, equivalent pipeline, backward characteristic	P_E	N/M^2	R
PHAS	variable used inside do loop for ripple phase values	-	deg	R
PO	pump outlet pressure, time = t	P_o	N/M^2	R
PS	pressure, outlet pipeline, backward characteristic	P_s	N/M^2	R
PT	tank pressure or case drain pressure	P_I	N/M^2	R
QE	flow, equivalent pipeline, backward characteristic	-	M^3/S	R
QID	ideal pump flow	Q_{ID}	M^3/S	R
QR	leakage flow, time = t	Q_R	M^3/S	R
QRIP	contribution to pump flow due to flow ripple	Q_s	M^3/S	R
QS	flow, outlet pipeline, backward characteristic	-	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
THETA	angle turned through by fundamental harmonic	-	rad	R
TIME	time	-	S	R
T1	starting time for flow ripple	-	S	R
VE	flow velocity, equivalent pipeline, backward characteristic	v_E	M/S	R
VO	flow velocity, outlet pipeline, time = t	v_o	M/S	R
VS	flow velocity, outlet pipeline, backward characteristic	v_s	M/S	R
VV	flow velocity, equivalent pipeline, time = t	v_v	M/S	R
WP	pump speed	-	rad/sec	R

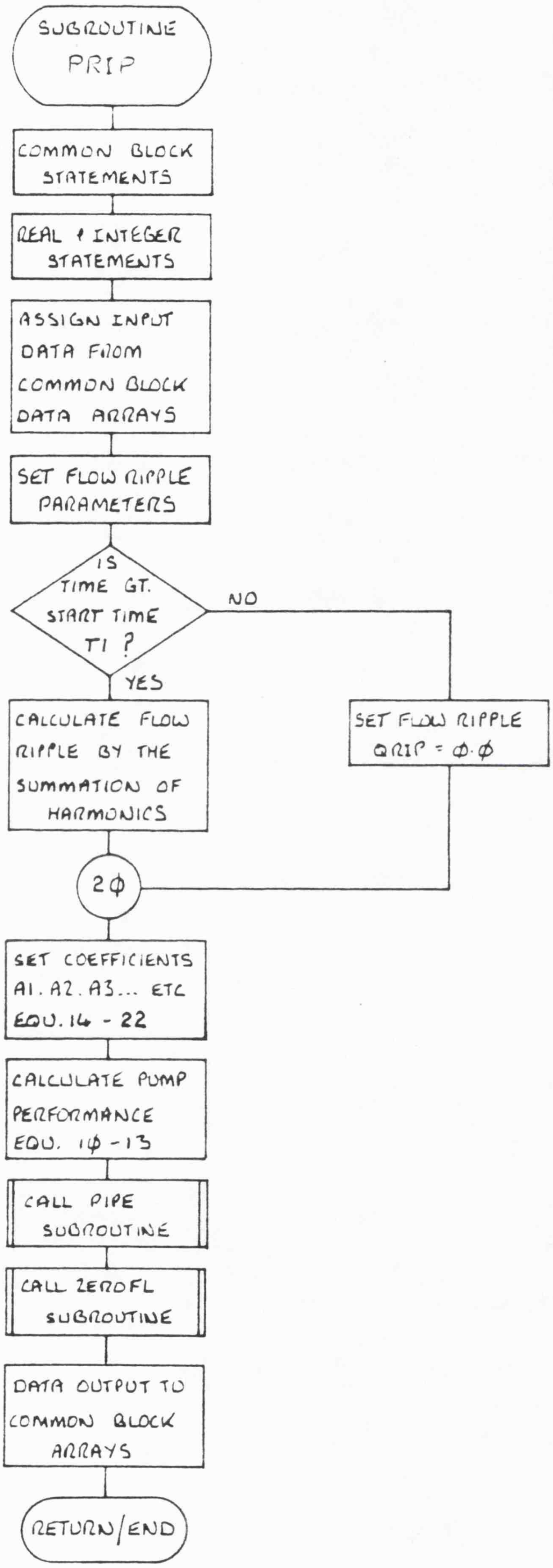

FIGURE 1   FLOWCHART FOR SUBROUTINE 'PRIP'

```
      subroutine prip(it,itm,dt,rho,time,no)
c
c     subroutine name   prip
c
c     library classification
c
c     title   method of characteristics model of a hydrostatic pump
c             with flow ripple effects at the outlet
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine is an idealised model of a hydrostatic
c              piston pump which includes the effects of flow ripple
c              at the pump outlet. the ripple is generated from
c              experimentally determined harmonics at a given set of
c              operating conditions
c
c     associated subroutines
c     pipe    pipe model using the method of characteristics
c     zerofl  method of characteristics model of a zero flow
c             pipe termination
c       (calls to these routines are made from inside prip)
c
c     common blocks
c     common/blk1/ pd          pipe data (constant)
c     common/blk3/ pdyn        pipe data (dynamic)
c     common/blk44/rpm         pump data (constant)
c     common/blk45/drpm        pump data (dynamic)
c     common/blk46/amp,faze    flow ripple amplitude and phase data
c     common/blk47/dta         flow ripple calculation parameters
c
c     input information
c     input via argument list
c     dt    timestep                          s
c     it    time level indicator              --
c     itm   time level indicator              --
c     no    pump component number             --
c     rho   fluid density                     kg/m3
c     time  time                              s
c
c     input via common block
c     ampl  flow ripple amplitude             m3/s       amp(-)
c     ape   area equivalent pipe              m2         pd(2,-)
c     apo   area outlet pipe                  m2         pd(2,-)
c     ce    wavespeed equivalent pipe         m/s        pd(7,-)
c     co    wavespeed outlet pipe             m/s        pd(7,-)
c     de    diameter equivalent pipe          m          pd(1,-)
c     do    diameter outlet pipe              m          pd(1,-)
c     epn   equivalent pipe number            --         rpm(3,-)
c     fe    friction factor e/p b/charac      --         pdyn(3,-,-,-)
c     fs    friction factor o/p b/charac      --         pdyn(3,-,-,-)
c     kr    pump leakage orifice coeff        ns/m5      rpm(4,-)
c     nor   not used                          --         rpm(6,-)
c     np    number of pistons                 --         rpm(8,-)
c     nzf   not used                          --         rpm(7,-)
c     opn   outlet pipe number                --         rpm(2,-)
c     pe    pressure e/p back/charac.         n/m2       pdyn(1,-,-,-)
c     phas  flow ripple phase                 deg        faze(-)
c     ps    pressure o/p back/charac          n/m2       pdyn(1,-,-,-)
c     pt    tank or case drain pressure       n/m2       rpm(5,-)
```

```
c      qe    flow e/p backward/charac        m3/s       pdyn(2,-,-,-)
c      qid   ideal pump flow                 m3/s       rpm(1,-)
c      qs    flow e/p backward/charac        m3/s       pdyn(2,-,-,-)
c      theta angle of 1st harmonic           rad        dta(2)
c      t1    start time of flow ripple       s          dta(1)
c      wp    pump speed                      rad/s      rpm(9,-)
c
c      output information
c      output via common block
c      po       pump outlet pressure   n/m2     pdyn(1,1,it,epn)
c                                               pdyn(1,1,it,opn)
c                                               drpm(1,it,no)
c      apo*vo  flow into outlet pipe   m/s      pdyn(2,1,it,opn)
c                                               drpm(2,it,no)
c      ape*vv  flow into equiv. pipe   m/s      pdyn(2,1,it,epn)
c
c      variable names (excluding i/o variables)
c      angle    angle turned by kth harmonic                rad
c      a1...a3  equation coeffs                             --
c      b1...b2  equation coeffs                             --
c      c1...c2  equation coeffs                             --
c      d1...c2  equation coeffs                             --
c      fharm    contribution to ripple from ith harmonic    m3/s
c      k        counter                                     --
c      qr       leakage flow                                m3/s
c      qrip     contribution to pump flow made by ripple    m3/s
c      ve       flow velocity e/pipe backward charac        m/s
c      vo       flow velocity o/pipe                        m/s
c      vs       flow velocity o/pipe backward charac.       m/s
c      vv       flow velocity e/pipe                        m/s
c
       common /blk1/pd(8,2)
       common /blk3/pdyn(3,400,10,2)
       common /blk44/rpm(9,2)
       common /blk45/drpm(2,10,2)
       common /blk46/amp(10),faze(10)
       common /blk47/dta(2)
       real kr
       integer opn,epn
c
c      data input
c      pump data
       qid=rpm(1,no)
       opn=rpm(2,no)
       epn=rpm(3,no)
       kr =rpm(4,no)
       pt =rpm(5,no)
       nor=rpm(6,no)
       nzf=rpm(7,no)
       np =rpm(8,no)
       wp =rpm(9,no)
c
c      pipe data
c      pipe equivalent to pump volume
       ape=pd(2,epn)
       de =pd(1,epn)
       ce =pd(7,epn)
       pe=pdyn(1,2,itm,epn)
       fe=pdyn(3,2,itm,epn)
       qe=pdyn(2,2,itm,epn)
       ve=qe/ape
```

```
c
c     outlet pipe
      apo=pd(2,opn)
      do =pd(1,opn)
      co =pd(7,opn)
      ps=pdyn(1,2,itm,opn)
      fs=pdyn(3,2,itm,opn)
      qs=pdyn(2,2,itm,opn)
      vs=qs/apo
c
c     set flow ripple parameters
      t1   =dta(1)
      theta=dta(2)
      qrip=0.0
c
c     is time greater than start time
      if(time.lt.t1) go to 20
c
c     calculate flow ripple
      do 50 k=1,10
      ampl=amp(k)
      phas=faze(k)
      angle=(theta*np*k)+phas/57.3
      fharm=ampl*sin(angle)
      qrip=qrip+fharm
   50 continue
      theta=theta+wp*dt
      dta(2)=theta
   20 continue
c
c     set coefficients
      a1=ape
      a2=apo
      a3=qid+qrip
      b1=-1.0/(rho*co)
      b2=-2.0*fs*vs*abs(vs)*dt/do+vs-ps/(rho*co)
      c1=-1.0/(rho*ce)
      c2=-2.0*fe*ve*abs(ve)*dt/de+ve-pe/(rho*ce)
      d1=-kr
      d2=-kr*pt
c
c     pump model
      po=(d2+a1*c2+a2*b2-a3)/(d1+a1*c1+a2*b1)
      vo=b2-b1*po
      vv=c2-c1*po
      qr=d2-d1*po
c
c     call pipe and zerofl subroutines
      call pipe(epn,it,itm,rho,dt)
      call zerofl(it,itm,dt,rho,epn)
c
c     data output
      pdyn(1,1,it,epn)=po
      pdyn(2,1,it,epn)=ape*vv
      pdyn(1,1,it,opn)=po
      pdyn(2,1,it,opn)=vo*apo
      drpm(1,it,no)=po
      drpm(2,it,no)=vo*apo
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'DYNOR'

Dynamic model of a restrictor valve

Purpose

Subroutine DYNOR models a restrictor valve with a constant downstream pressure. Dynamic effects are taken into account by treating the valve impedance as a first order lag.

Associated subroutines

DYNORI initialisation of integration parameters for subroutine DYNOR

CALL DYNOR(IT,ITM,DT,RHO,NO)

NO = restrictor valve component number (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data array (constant)
COMMON/BLK3/PDYN	pipe data array (dynamic)
COMMON/BLK50/DOR	dynamic valve model data array (constant)

User defined information

ARRAY DOR(12,NO)

DOR(1,NO) = CD	coefficient of discharge	-	R
DOR(2,NO) = A	orifice area	M^2	R
DOR(3,NO) = IPN	inlet pipe number	-	I
DOR(4,NO) = PT	constant downstream pressure	N/M^2	R
DOR(5,NO) = RN	valve index	-	R
DOR(6,NO) = T	valve time constant	S	R

Output information via common block

PI	pressure at the valve	PDYN(1,NCP,IT,IPN)	N/M^2	R
API*VI	flow at the valve	PDYN(2,NCP,IT,IPN)	M^3/S	R

Program action and algorithm

Mathematical model

The dynamic equations for the restrictor valve performance are obtained by considering the valve impedance not as a constant function but as a 1st order lag the full derivation is given in volume 1.

Two equations apply at the valve:-

<u>valve equation</u>

$$A_{PI}v_I = K_{NL}P^{\frac{1}{n}} + \frac{K_{NL}}{n} T\, P^{\left(\frac{1}{n} - 1\right)} \frac{dP}{dt} \qquad 1$$

<u>pipe equation</u>
(forward characteristic

$$\frac{1}{\rho C_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_R v_R |v_R| \Delta t}{d_I} = 0 \qquad 2$$

where P is the pressure difference across the orifice i.e. $P = (P_I - P_T)$ and $K_{NL} = Cd\ A(2/\rho)^{1/n}$ Equation 1 may be rearranged as follows.

$$\frac{dP}{dt} = \frac{A_{PI}v_I - K_{NL}P^{\frac{1}{n}}}{\frac{K_{NL}}{n} T\, P^{\left(\frac{1}{n} - 1\right)}} \qquad 3$$

Equation 3 is used to calculate the rate of change of pressure, which is then integrated by the Simple Euler method to give the value of pressure after a timestep.

<u>Simple Euler Integration</u>

$$P_{NEW} = P_{OLD} + \Delta t \frac{dP}{dt} \qquad 4$$

The new value of pressure is substituted in a rearranged form of equation 2 and v_I is calculated (equation 5)

flow velocity

$$v_I = - \frac{2f_R v_R |v_R| \Delta t}{d_I} + v_R - \frac{1}{\rho C_I}(P_I - P_R) \qquad 5$$

Computing procedure

For accurate integration the timestep used should be less than approximately 1/100th of the valve timeconstant. The associated subroutine DYNORI calculates the maximum value of the timestep permissible for accurate integration and sets other integration parameters which are stored in array DOR. Subroutine DYNOR assigns all input data to program variable names, then checks if the method of characteristics timestep Δt is sufficiently small, if so the program branches and the solution of the valve is performed by solving equations 3 , 4 and 5 in one step (Figure 1).

If Δt is too large for accurate integration a multiple step procedure is followed. The internal Δt is divided into an integer number of steps each of which is less than 1/100th of the time constant. At each of these timesteps the integration procedure is carried out by solving equations 3 , 4 and 5 , the values of P_R, v_R and f_R which apply at the intermediate points are obtained by linear interpolation. The integration and interpolation is repeated (do loop 20) until the time internal Δt has been covered.

The final values of pressure and flow are assigned to array PDYN.

LIST OF VARIABLES USED

A	orifice area	-	M^2	R
API	inlet pipe area	A_{PI}	M^2	R
CD	discharge coefficient	-	-	R
CI	inlet pipe wavespeed	C_I	M/S	R
DI	inlet pipe diameter	d_I	M	R
DIFP	pressure gradient dP/dt	dP/dt	$(N/M^2)/S$	R
DT	timestep	Δt	S	R
DTM	maximum timestep for valve integration	-	S	R
DTO	integration timestep for valve model	-	S	R

DX	distance between calculation points	Δx	M	R
DXO	Δx used in interpolation	-	M	R
FI1	old value of friction factor at valve	-	-	R
FR	friction factor, forward characteristic	f_R	-	R
GFR	interpolation gradient for friction factor values	-	-	R
GPR	interpolation gradient for pressure values	-	-	R
GVR	interpolation gradient for flow velocity values	-	-	R
I	do loop counter	-	-	I
IDT	number of integration steps for valve model	-	-	I
IPN	inlet pipe number	-	-	I
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
KNL	orifice coefficient	K_{NL}	-	R
NCM	number of calculation points minus 1	-	-	I
NCP	number of calculation points	-	-	I
NO	valve number	-	-	I
PI	pressure, time = t + Δt	P_I	N/M^2	R
PI1	old value of pressure at valve	-	N/M^2	R
PNEW	updated value of pressure	-	N/M^2	R
PR	pressure, forward characteristic	P_R	N/M^2	R
PT	constant downstream pressure at valve	P_T	N/M^2	R
QI1	old value of flow at valve	-	M^3/S	R
QR	flow, forward characteristic	-	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
RN	index	n	-	R
T	orifice time constant	T	S	R
VI	flow velocity, time = t	v_I	M/S	R
VI1	old value of flow velocity at valve	-	M/S	R
VNEW	updated value of flow velocity	-	M/S	R
VR	flow velocity, forward characteristic	v_R	M/S	R

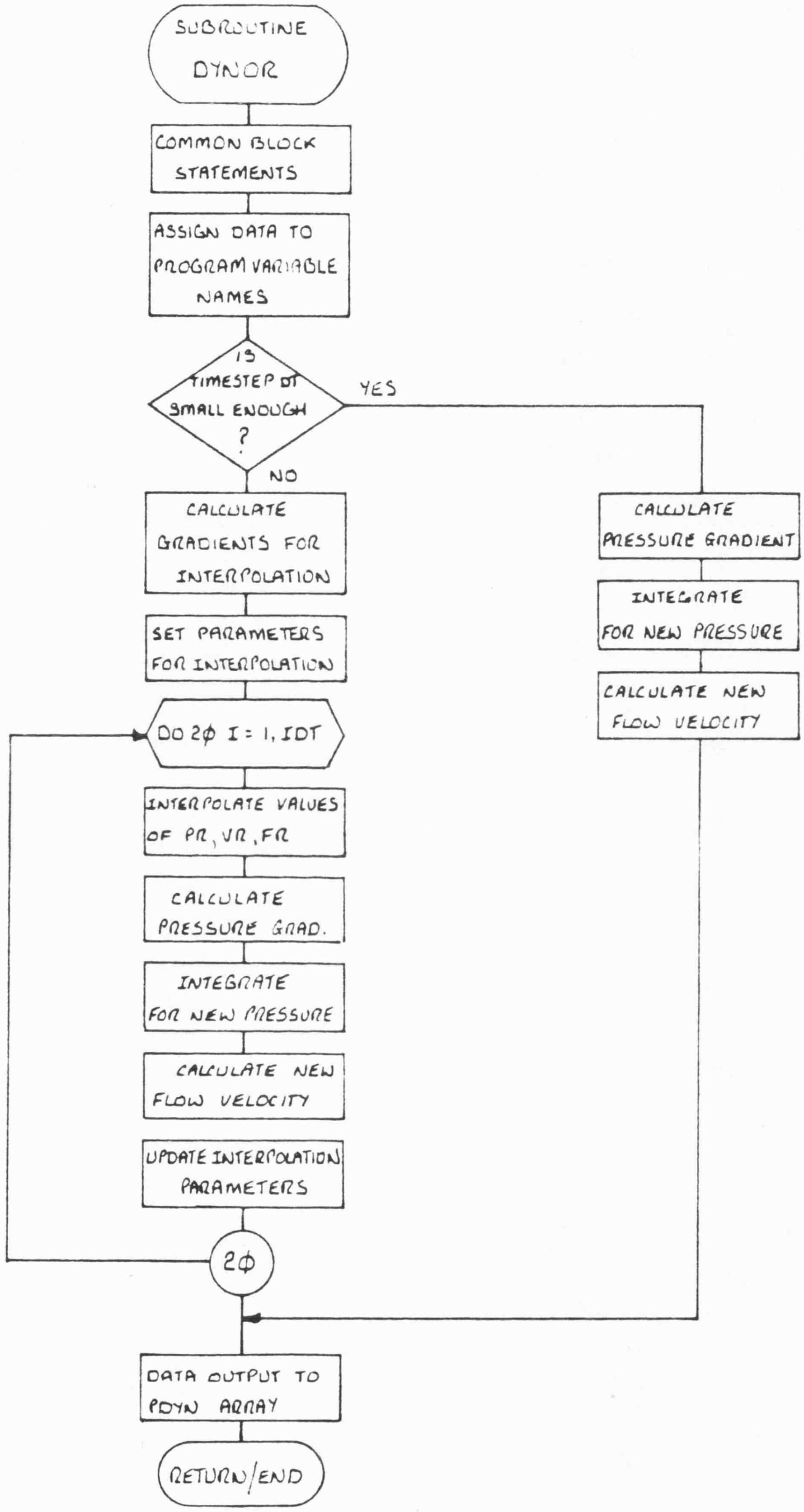

FIGURE 1 FLOWCHART FOR SUBROUTINE 'DYNOR'

```
      subroutine dynor(it,itm,dt,rho,no)
c
c     subroutine name  dynor
c
c     library classification
c
c     title  dynamic model of a restrictor valve
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine models a restrictor valve with a constant
c              downstream pressure. dynamic effects are taken into
c              account by treating the valve impedance as a first order
c              lag.
c
c     associated subroutines
c     dynori   initialisation of integration parameters for subr dynor
c
c     common blocks
c     common/blk1/ pd      pipe data (constant)
c     common/blk3/ pdyn    pipe data (dynamic)
c     common/blk50/dor     dynamic valve model data (constant)
c
c     input information
c     input via argument list
c     dt    timestep                  s
c     it    time level indicator      --
c     itm   time level indicator      --
c     no    component number          --
c     rho   fluid density             kg/m3
c
c     input via common block
c     a   orifice area                          m2    dor(2,-)
c     api inlet pipe area                       m2    pd(2,-)
c     cd  discharge coefficient                 --    dor(1,-)
c     ci  inlet pipe wavespeed                  m/s   pd(7,-)
c     di  inlet pipe diameter                   m     pd(1,-)
c     dtm maximum timestep for valve integ.     s     dor(11,-)
c     dto integration timestep for valve model  s     dor(7,-)
c     dx  distance between calculation points   m     dor(9,-)
c     dxo dx used in interpolation              m     dor(8,-)
c     fi1 previous friction factor at valve     --    pdyn(3,-,-,-)
c     fr  friction factor, forward charac.      --    pdyn(3,-,-,-)
c     idt number of integ. steps for valve      --    dor(10,-)
c     ipn inlet pipe number                     --    dor(3,-)
c     knl orifice coefficient                   --    dor(12,-)
c     ncp number of calculation points          --    pd(6,-)
c     pi1 previous pressure at valve            n/m2  pdyn(1,-,-,-)
c     pr  pressure, forward characteristic      n/m2  pdyn(1,-,-,-)
c     pt  constant downstream pressure          n/m2  dor(4,-)
c     qi1 previous flow at valve                m3/s  pdyn(2,-,-,-)
c     qr  flow,forward characteristic           m3/s  pdyn(2,-,-,-)
c     rn  index                                 --    dor(5,-)
c     t   valve time constant                   s     dor(6,-)
c
c     output information
c     output via common block
c     pi     calculated pressure    n/m2     pdyn(1,ncp,it,ipn)
c     api*vi flow at valve          m3/s     pdyn(2,ncp,it,ipn)
c
```

```
c     variable names (excluding i/o variables)
c     difp   pressure gradient (dp/dt)
c     gfr    interpolation gradient,friction factor       --
c     gpr    interpolation gradient,pressure              --
c     gvr    interpolation gradient,flow velocity         --
c     i      counter                                      --
c     ncm    no of calculation points minus 1             --
c     pnew   updated pressure value                       n/m2
c     vi     calculated flow velocity                     m/s
c     vi1    previous value of flow velocity              m/s
c     vnew   updated flow value                           m/s
c     vr     flow velocity forward characteristic         m/s
c
      common /blk1/ pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
      common /blk50/ dor(12,2)
      real knl
c
c     data input
c     orifice data
      cd =dor(1,no)
      a  =dor(2,no)
      ipn=dor(3,no)
      pt =dor(4,no)
      rn =dor(5,no)
      t  =dor(6,no)
      dto=dor(7,no)
      dxo=dor(8,no)
      dx =dor(9,no)
      idt=dor(10,no)
      dtm=dor(11,no)
      knl=dor(12,no)
c
c     pipe data
      ncp=pd(6,ipn)
      ncm=ncp-1
      api=pd(2,ipn)
      di =pd(1,ipn)
      ci =pd(7,ipn)
c
      pr=pdyn(1,ncm,itm,ipn)
      fr=pdyn(3,ncm,itm,ipn)
      qr=pdyn(2,ncm,itm,ipn)
      vr=qr/api
      pi1=pdyn(1,ncp,itm,ipn)
      fi1=pdyn(3,ncp,itm,ipn)
      qi1=pdyn(2,ncp,itm,ipn)
      vi1=qi1/api
c
c     select integration step length
      if(dt.le.dtm)go to 10
c
```

```
c     calculate gradients for interpolation
      gpr=(pr-pi1)/dx
      gvr=(vr-vi1)/dx
      gfr=(fr-fi1)/dx
c
c     set parameters pi and vi
      pi=pi1
      vi=vi1
c
c     integrate over interval dt using timestep dto
      do 20 i=1,idt
c
c     interpolate
      pr=pi1+gpr*dxo*i
      vr=vi1+gvr*dxo*i
      fr=fi1+gfr*dxo*i
c
c     evaluate dp/dt (difp)
c
      difp=rn*(api*vi-knl*((pi-pt)**(1.0/rn)))/
     1(knl*t*((pi-pt)**(1.0/rn-1.0)))
c
c     integrate for new pressure
      pnew=pi+difp*dto
c
c     calculate new flow velocity
      vnew=-2.0*fr*vr*abs(vr)*dto*i/di+vr-(pnew-pr)/(rho*ci)
c
c     update parameters pi vi
      pi=pnew
      vi=vnew
   20 continue
      go to 30
c
   10 continue
c
c     integration using one timestep dt
c
      difp=rn*(api*vi1-knl*((pi1-pt)**(1.0/rn)))/
     1(knl*t*((pi1-pt)**(1.0/rn-1.0)))
c
      pi=pi1+difp*dt
      vi=-2.0*fr*vr*abs(vr)*dt/di+vr-(pi-pr)/(rho*ci)
c
   30 continue
c
c     data output
      pdyn(1,ncp,it,ipn)=pi
      pdyn(2,ncp,it,ipn)=vi*api
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'DYNORI'

Initialisation of integration parameters for subroutine DYNOR

Purpose

Subroutine DYNORI calculates various parameters required for the successful execution of subroutine DYNOR.

Associated subroutine

DYNOR   dynamic model of a restrictor valve

CALL DYNORI(DT,RHO,NO)

NO = corresponding valve component number (integer)

Common block data arrays

COMMON/BLK1/PD   pipe data array (constant)

COMMON/BLK50/DOR   dynamic valve model data array (constant)

No user defined information required

Output information via common block

DTO	integration timestep for valve model	S	DOR(7,NO)	R
DX	distance between calculation points	M	DOR(9,NO)	R
DXO	Δx used in interpolation	M	DOR(8,NO)	R
DTM	maximum timestep for valve integration	S	DOR(11,NO)	R
KNL	orifice coefficient	-	DOR(12,NO)	R
RRIDT	number of integration steps rounded up	-	DOR(10,NO)	R

Program action and algorithm

This subroutine is a straight forward sequential calculation of a number of parameters required for integration calculations in the subroutine DYNOR.

The following parameters are calculated:-

1 KNL - the valve coefficient calculated from the expression KNL = CD A $(2/RHO)^{1/RN}$

2 RRIDT - the number of integration steps required inside each method of characteristics timestep to give accurate integration. (The timestep should be approximately $1/100^{th}$ of the valve time constant.)

3 DTO - the integration step length corresponding to number RRIDT

4 DXM - the distance travelled by a wave during one timestep DTO, the value of DXO is used when interpolating.

5 DTM - the maximum value of the timestep permissible for accurate integration.

LIST OF VARIABLES USED

A	orifice area	M^2	R
C	wavespeed in inlet pipe	M/S	R
CD	coefficient of discharge	-	R
DT	timestep	S	R
DTM	maximum timestep for valve integration	S	R
DTO	integration timestep for valve model	S	R
DX	distance between calculation points	M	R
DXO	Δx used in interpolation	M	R
IDT	number of integration steps for valve model	-	I
IPN	inlet pipe number	-	I
KNL	orifice coefficient	-	R
NO	valve number	-	I
RHO	fluid density	Kg/M^3	R
RIDT	number of integration steps required (real value)	-	R
RN	index	-	R
RRIDT	number of integration steps rounded up (real value)	-	R
T	valve time constant	S	R

```
      subroutine dynori(dt,rho,no)
c
c     subroutine name  dynori
c
c     library classification
c
c     title  initialisation of integration parameters for subroutine
c            dynor
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine calculated various parameters required
c              for the successful execution of subroutine dynor
c
c     associated subroutine
c     dynor   dynamic model of a restrictor valve
c
c     common blocks
c     common/blk1/ pd      pipe data (constant)
c     common/blk50/dor     dynamic valve model data (constant)
c
c     input information
c     input via argument list
c     dt    timestep                   s
c     no    component number           --
c     rho   fluid density              kg/m3
c
c     input via common block
c     a   orifice area                             m2    dor(2,-)
c     c   wavespeed in inlet pipe                  m/s   pd(7,-)
c     cd  discharge coefficient                    --    dor(1,-)
c     ipn inlet pipe number                        --    dor(3,-)
c     rn  index                                    --    dor(5,-)
c     t   valve time constant                      s     dor(6,-)
c
c     output information
c     output via common block
c     dtm maximum timestep for valve integ.        s     dor(11,-)
c     dto integration timestep for valve model     s     dor(7,-)
c     dx  distance between calculation points      m     dor(9,-)
c     dxo dx used in interpolation                 m     dor(8,-)
c     idt number of integ. steps for valve         --    dor(10,-)
c     knl orifice coefficient                      --    dor(12,-)
c     rridt  number of integration steps           -     dor(10,-)
c
c     variables (excluding i/o variables)
c     idt    no. of integration steps for valve           --
c     ridt   no. of integration steps real value          --
c
```

```
      common /blk50/dor(12,2)
      common /blk1/pd(8,2)
      real knl
c
c     input data
      cd =dor(1,no)
      a  =dor(2,no)
      rn =dor(5,no)
      ipn=dor(3,no)
      t  =dor(6,no)
      c  =pd(7,ipn)
c
      knl=cd*a*((2.0/rho)**(1.0/rn))
      ridt=100.0*dt/t
      idt=ridt+1.0
      dto=dt/idt
      dx=c*dt
      dxo=dx/idt
      rridt=float(idt)
      dtm=0.01*t
c
c     output data
      dor(7,no)=dto
      dor(8,no)=dxo
      dor(9,no)=dx
      dor(10,no)=rridt
      dor(11,no)=dtm
      dor(12,no)=knl
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'ORFCE'

Method of characteristics model of a simple orifice with a constant downstream pressure

Purpose

This subroutine models the effects of a simple orifice obeying a steady state square law pressure-flow relationship. The orifice is located at the downstream end of a pipe and discharges to a constant pressure reservoir.

No associated subroutines

CALL ORFCE(IT,ITM,DT,RHO,NO)
NO = orifice component number (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)
COMMON/BLK41/SVLV	orifice data

User defined information

ARRAY SVLV(4,NO)

SVLV(1,NO) = CD	discharge coefficient	-	R
SVLV(2,NO) = A	orifice area	M^2	R
SVLV(3,NO) = IPN	inlet pipe number	-	I
SVLV(4,NO) = PT	constant downstream pressure	N/M^2	R

Output information via common block

PI	pressure at valve	PDYN(1,NCP,IT,IPN)	N/M^2	R
QI	flow at valve	PDYN(2,NCP,IT,IPN)	M^3/S	R

Inbuilt error messages

'Imaginary roots' - error condition message printed and program stops.

Program action and algorithm

Mathematical model

Two equations apply at the orifice

Orifice equation $$A_{PI}v_I = CdA\sqrt{\frac{2}{\rho}\left|P_I - P_T\right|}\cdot \text{sign}(P_I - P_T) \qquad 1$$

Pipe equation (forward characteristic) $$\frac{1}{\rho C_I}(P_I - P_R) + (v_I - v_R) + \frac{2f_R v_R|v_R|\Delta t}{d_I} = 0 \qquad 2$$

rewriting equation 1 as $$v_I^{\,2} = \left[\frac{CdA}{A_{PI}}\right]^2 \frac{2}{\rho}\left|P_I - P_T\right| \qquad 3$$

Note the $\text{sign}(P_I - P_T)$ term may be omitted if suitable precautions are taken in the program to ensure that flow velocity has the correct sign.

Equations 2 and 3 may be written as

$$2 \rightarrow \quad v_I = A_2 - A_1 P_I \qquad 4$$

$$3 \rightarrow \quad v_I^{\,2} + B_1 P_I = B_2 \qquad 5$$

where

$$A_1 = 1/\rho C_I \qquad 6$$

$$A_2 = -2f_R v_R|v_R|\Delta t/d_I + v_R + P_R/\rho C_I \qquad 7$$

$$B_1 = -(CdA/A_{PI})^2 \cdot 2/\rho \qquad 8$$

$$B_2 = B_1 P_T \qquad 9$$

equations 4 and 5 may be further rearranged to form a quadratic equation in P_I

$$(P_I)^2 + R_1(P_I) + R_2 = 0 \qquad 10$$

with the standard solution $P_I = -\frac{R_1}{2} \pm \frac{1}{2}\sqrt{R_1^2 - 4R_2}$ 11

where

$$R_1 = (B_1 - 2A_2A_1)/A_1^2 \qquad 12$$

$$R_2 = (A_2^2 - B_2)/A_1^2 \qquad 13$$

Equation 11 gives two possible roots for the quadratic equation the two corresponding flows may be calculated from:-

$$Q_I = (A_2 - A_1P_I)A_{PI} \qquad 14$$

Computing procedure

The subroutine assigns all input data from common block arrays to program variable names. The equation coefficients A_1, A_2, B_1, B_2, R_1 and R_2 are calculated from equations 6 - 9 and 12 - 13 . A check is made for the possibility of imaginary roots to the quadratic, if this situation arises a program crash is inevitable, since calculating the roots would involve taking the square root of a negative number. In this context imaginary roots are meaningless and to prevent an uncontrolled crash the program prints an error message and stops execution. If the check shows that the roots will be real the subroutine calculates their values from equation 11 . Two pressures PI1 and PI2 are obtained and the corresponding flows QI1 and QI2 are calculated from equation 14 . Only one pair of pressures and flows can satisfy the original equations 1 and 2 and to acertain which of the two roots is correct the following conditions are tested for:-

1 If (PI1 - PT) is positive and QI1 is negative, the flow is opposing the pressure gradient so PI1 and QI1 do not satisfy the original equations, therefore use values PI2 and QI2.

2 If (PI2 - PT) is positive and QI2 is negative, the flow is opposing the pressure gradient so PI2 and QI2 do not satisfy the original equations, therefore use values PI1 and QI1.

3 If (PI1 - PT) is negative and QI1 is positive, the flow is opposing the pressure gradient so PI1 and QI1 do not satisfy the original equations, therefore use values PI2 and QI2.

4 If (PI2 - PT) is negative and QI2 is positive, the flow is opposing the pressure gradient so PI2 and QI2 do not satisfy the original equations, therefore use values PI1 and QI1.

If the program passes through all these checks both roots must be equal. A final cavitation check is performed setting negative pressure to zero. The final solution is output via the PDYN data array.

LIST OF VARIABLES USED

A	orifice area	A	M^2	R
API	inlet pipe area	A_{PI}	M^2	R
A1, A2	component equation coefficients	A_1, A_2	-	R
B1, B2	component equation coefficients	B_1, B_2	-	R
CD	coefficient of discharge	Cd	-	R
CI	inlet pipe wavespeed	C_I	M/S	R
DI	inlet pipe diameter	d_I	M	R
DT	timestep	Δt	S	R
FR	friction factor, forward characteristic	f_R	-	R
IPN	inlet pipe number	-	-	I
IT	time level indicator, time = t	-	-	I
ITM	time level indicator, time = t - Δt	-	-	I
NCM	number of calculation points minus 1	-	-	I
NCP	number of calculation points	-	-	I
NO	orifice number	-	-	I
PI	pressure at orifice, time = t	P_I	N/M^2	R
PI1	first root of quadratic equation	-	-	R
PI2	second root of quadratic equation	-	-	R
PR	pressure, forward characteristic	P_R	N/M^2	R
PT	constant downstream pressure	P_T	N/M^2	R
QI	flow at orifice, time = t	-	M^3/S	R
QI1	flow corresponding to first root	-	M^3/S	R
QI2	flow corresponding to second root	-	M^3/S	R
QR	flow, forward characteristic	-	M^3/S	R

RHO	fluid density	ρ	Kg/M^3	R
Rl, R2	quadratic equation coefficient	R_1, R_2	-	R
VR	flow velocity, forward characteristic	v_R	M/S	R

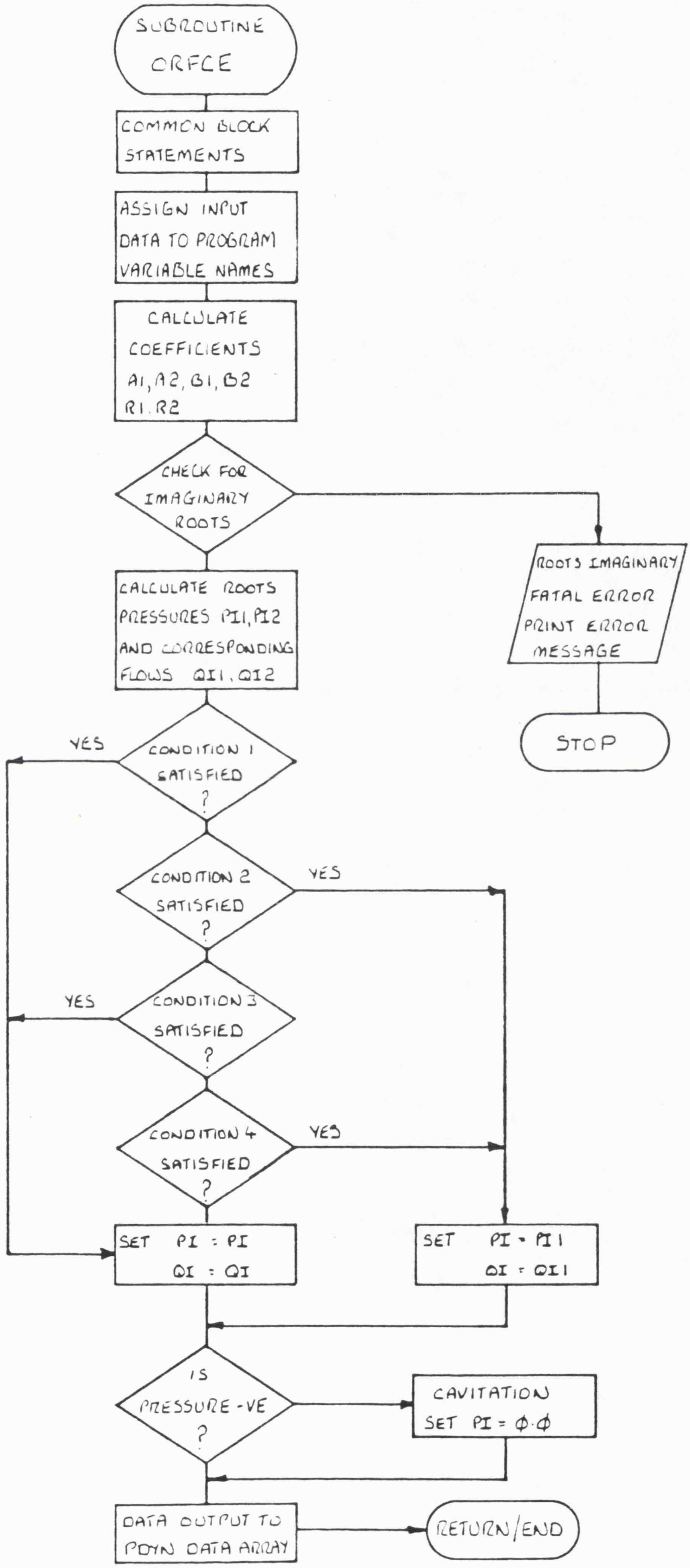

FIGURE 1 FLOWCHART FOR SUBROUTINE ORFCE

```
      subroutine orfce(it,itm,dt,rho,no)
c
c     subroutine name   orfce
c
c     library classification
c
c     title method of characteristics model of a simple orifice with
c           a constant downstream pressure
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine models the effect of a simple orifice
c              obeying a square law pressure-flow relationship.the
c              orifice is located at the downstream end of a pipe and
c              discharges to a constant pressure reservoir
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/ pd     pipe data (constant)
c     common/blk3/ pdyn   pipe data (dynamic)
c     common/blk4/ svlv   orifice data (constant)
c
c     input information
c     input via argument list
c     dt   timestep                        s
c     it   time level indicator            --
c     itm  time level indicator            --
c     no   component number                --
c     rho  fluid density                   kg/m3
c
c     input via common block
c     a    orifice area                     m2      svlv(2,-)
c     api  inlet pipe area                  m2      pd(2,-)
c     cd   discharge coefficient            --      svlv(1,-)
c     ci   inlet pipe wavespeed             m/s     pd(7,-)
c     di   inlet pipe diameter              m       pd(1,-)
c     fr   friction factor fwrd/charac.     --      pdyn(3,-,-,-)
c     ipn  inlet pipe number                --      svlv(3,-)
c     ncp  number of calculation points     --      pd(6,-)
c     pr   pressure fwrd/characteristic     n/m2    pdyn(1,-,-,-)
c     pt   constant downstream pressure     n/m2    svlv(4,-)
c     qr   flow forward characteristic      m3/s    pdyn(2,-,-,-)
c
c     output information
c     output via common block
c     pi   pressure at valve                n/m2    pdyn(1,-,-,-)
c     qi   flow at valve                    m3/s    pdyn(2,-,-,-)
c
c     variable names (excluding i/o variables)
c     a1...a2  component equation coefficients         --
c     b1...b2  component equation coefficients         --
c     ncm      no of calculation points minus 1        --
c     pi1      first root pressure                     n/m2
c     pi2      second root pressure                    n/m2
c     qi1      first root flow                         m3/s
c     qi2      second root flow                        m3/s
c     r1...r2  quadratic equation coefficients         --
c     vr       flow velocity,forward characteristic    m/s
c
      common /blk1/ pd(8,2)
      common /blk3/ pdyn(3,400,10,2)
      common /blk41/ svlv(4,2)
```

```
c
c     data input
c     orifice data
      cd =svlv(1,no)
      a  =svlv(2,no)
      ipn=svlv(3,no)
      pt =svlv(4,no)
c
c     pipe data
      ncp=pd(6,ipn)
      ncm=ncp-1
      api=pd(2,ipn)
      di =pd(1,ipn)
      ci =pd(7,ipn)
      pr=pdyn(1,ncm,itm,ipn)
      fr=pdyn(3,ncm,itm,ipn)
      qr=pdyn(2,ncm,itm,ipn)
      vr=qr/api
c
c     calculate component equation coefficients
      a1=1.0/(rho*ci)
      a2=-2.0*fr*vr*abs(vr)*dt/di+vr+pr/(rho*ci)
      b1=(-cd*a/api)*(cd*a/api)*2.0/rho
      b2=b1*pt
c
c     calculate quadratic coefficients
      r1=(b1-2.0*a2*a1)/(a1*a1)
      r2=(a2*a2-b2)/(a1*a1)
c
c     check for imaginary roots
      if((r1*r1).lt.(4.0*r2))go to 20
c
c     calculate root pressures and flows
      pi1=-r1/2.0+0.5*sqrt(r1*r1-4.0*r2)
      pi2=-r1/2.0-0.5*sqrt(r1*r1-4.0*r2)
      qi1=(a2-a1*pi1)*api
      qi2=(a2-a1*pi2)*api
c
c     check roots and set pressure and flow
      if((pi1-pt).gt.0.0.and.qi1.lt.0.0)go to 33
      if((pi2-pt).gt.0.0.and.qi2.lt.0.0)go to 34
      if((pi1-pt).lt.0.0.and.qi1.gt.0.0)go to 33
      if((pi2-pt).lt.0.0.and.qi2.gt.0.0)go to 34
c
   33 pi=pi2
      qi=qi2
      go to 32
   34 pi=pi1
      qi=qi1
   32 continue
c
c     cavitation check
      if(pi.le.0.0)pi=0.0
c
c     data output
      pdyn(1,ncp,it,ipn)=pi
      pdyn(2,ncp,it,ipn)=qi
      go to 40
c
c     error message
   20 write(6,600)
  600 format(1h0,'***error***imaginary roots****')
      stop
   40 continue
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE 'ZEROFL'

Method of characteristics model of a zero flow pipe termination

Purpose

This subroutine models the effects of a closed end by treating it as a component which imposes a condition of zero flow.

No associated subroutines

CALL ZEROFL(IT,ITM,DT,RHO,IPN)

IPN = inlet pipe number (integer)

Common block data arrays

COMMON/BLK1/PD	pipe data (constant)
COMMON/BLK3/PDYN	pipe data (dynamic)

No user defined information required

Output information via common block

PP	pressure at end of pipe	N/M^2	PDYN(1,NCP,IT,IPN)	R
O.O	zero flow	M^3/S	PDYN(2,NCP,IT,IPN)	R

Program action and algorithm

The equation associated with a forward characteristic applies at a closed end. The flow velocity is set to zero and the equation solved directly for pressure.

$$P_p = \left[- \frac{2 f_R v_R | v_R | \Delta t}{d_I} + v_R + \frac{P_R}{\rho C_I} \right] \rho C_I \qquad 1$$

The subroutine simply evaluates equation 1 and assigns the value of P_p to array PDYN. The flow element of array PDYN is assigned the value 0.0.

N.B. There is no check for cavitation i.e. negative pressures may be obtained.

LIST OF VARIABLES USED

API	inlet pressure area	-	M^2	R
CI	wavespeed in inlet pipe	C_I	M/S	R
DI	diameter of inlet pipe	d_I	M	R
DT	timestep	Δt	S	R
FR	friction factor, forward characteristic	f_R	-	R
IPN	inlet pipe number	-	-	I
IT	time level indicator	-	-	I
ITM	time level indicator	-	-	I
NCM	number of calculation points minus 1	-	-	I
NCP	number of calculation points	-	-	I
PP	pressure, at time = t, at end of inlet pipe	P_p	N/M^2	R
PR	pressure, forward characteristic	P_R	N/M^2	R
QR	flow, inlet pipe, forward characteristic	-	M^3/S	R
RHO	fluid density	ρ	Kg/M^3	R
VR	flow velocity, forward characteristic	v_R	M/S	R

```
c     subroutine name   zerofl
c
c     library classification
c
c     title  method of characteristics model of a zero flow pipe
c            termination (closed end)
c
c     author  c.m. skarbek-wazynski
c
c     purpose  this subroutine models the effects of a closed end by
c              treating it as a component which imposes a condition of
c              zero flow
c
c     no associated subroutines
c
c     common blocks
c     common/blk1/pd     pipe data (constant)
c     common/blk3/pdyn   pipe data (dynamic)
c
c     input information
c     input via argument list
c     dt   timestep                          s
c     it   time level indicator              --
c     itm  time level indcator               --
c     ipn  inlet pipe number                 --
c     rho  fluid density                     kg/m3
c
c     input via common block
c     api  inlet pipe area                      m2      pd(2,-)
c     ci   wavespeed in inlet pipe              m/s     pd(7,-)
c     di   inlet pipe diameter                  m       pd(1,-)
c     fr   friction factor forward charac.      --      pdyn(3,-,-,-)
c     ncp  number of calculation points         --      pd(6,-)
c     pr   pressure forward characteristic      n/m2    pdyn(1,-,-,-)
c     qr   flow forward characteristic          m3/s    pdyn(2,-,-,-)
c
c     output information
c     output via common block
c     pp   calculated pressure                  n/m2    pdyn(1,-,-,-)
c     0.0  zero flow                            m3/s    pdyn(2,-,-,-)
c
c     variable names (excluding i/o variables)
c     ncm  no of calc points minus 1         --
c     vr   flow velocity forward charac.     m/s
c
      common /blk1/pd(8,2)
      common /blk3/pdyn(3,400,10,2)
c     input data
      ncp=pd(6,ipn)
      ncm=ncp-1
      api=pd(2,ipn)
      di =pd(1,ipn)
      ci =pd(7,ipn)
      pr=pdyn(1,ncm,itm,ipn)
      fr=pdyn(3,ncm,itm,ipn)
      qr=pdyn(2,ncm,itm,ipn)
      vr=qr/api
c     calculate pressure
      pp=(-2.0*fr*vr*abs(vr)*dt/di+vr+pr/(rho*ci))*rho*ci
c     data output
      pdyn(1,ncp,it,ipn)=pp
      pdyn(2,ncp,it,tpn)=0.0
      return
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM fan.fortran

LIBRARY CLASSIFICATION

FOURIER ANALYSIS OF A FUNCTION SPECIFIED AS A NUMBER OF DATA POINTS

FORTRAN IV | HONEYWELL MULTICS 1 SEPT. 1980

No special hardware requirements

Author: C.M. SKARBEK-WAŻYNSKI

Purpose

The program takes as data one period of any periodic function y = f(x) defined as a number of points in the x - y space and performs a Fourier analysis. The results are expressed as a number of harmonics in amplitude and phase form and also the corresponding Fourier coefficients are printed out.

Associated subroutines

Subroutine bengf:intrpl.016,01	- smoothed one dimensional interpolation and linear extrapolation routine
Subroutine integs.fortran	- integration of a function using Simpson's rule

Input information (read from a data file)

nf	number of functions	-	I
nch	switch	-	I
nch2	switch	-	I
ndat	number of data points (specified only if all functions have the same number of data points, otherwise set to zero)	-	I

The following items of data must be specified for each function the user wishes to analyse.

nn	number of data points defining a given function	-	I
nh	number of harmonics required as output	-	I
freq	frequency of fundamental harmonic	c/s	R
nchl	switch	-	I
fty	array containing values of the dependent variable y	-	R
ftx	array containing values of the independent variable x	-	R

Note: The array ftx needs to be specified only if the function is defined at irregular spacings of the independent variable x.

Setting switches and producing an input data file

The switches nch2, nch and nchl all describe whether the functions the user wishes to analyse are all supplied to the same values of the independent variable x or whether each function is defined at different values of x, and whether or not the spacing between data points is regular. Naturally if the spacing is regular the user need not specify values of x since they may be inferred directly. The algorithm for setting the switches is presented in Figure 1 . The stacking of the input data file depends on the values of the switches and a guide for producing a data file is given in Figure 2 .

All input data is presented in free format.

Output information

The program prints out all the input data except the values of the switches, plus the following:-

ft	array containing interpolated values of the dependent variable y. This array is only produced if a function is defined by an even number of data points	-	R
amp	amplitude of a given harmonic	-	R
phd	phase of a given harmonic	deg	R

a	array containing Fourier coefficients of cosine terms	-	R
b	array containing Fourier coefficients of sine terms	-	R

A sample results file is shown in Figure 3 .

Inbuilt error messages

No normal failures can occur provided input information is correctly defined.

Timing and storage

The timing and storage depends on the number of functions the user wishes to analyse.

Program action and algorithm

Mathematical model

Any repetitive function y = f(t) may be expressed as a Fourier series.

$$f(t) = \frac{Ac}{2} + A_1\cos\omega ot + A_2\cos 2\omega ot + A_3\cos 3\omega ot + \\ + B_1\sin\omega ot + B_2\sin 2\omega ot + B_3\sin 3\omega ot + \qquad 1$$

General form:-

$$f(t) = M + \sum_{R=1}^{\infty} A_R\cos R\omega_o t + \sum_{R=1}^{\infty} B_R\sin R\omega_o t \qquad 2$$

where M is the mean level of the function.

Manipulation of the sinusoidal terms gives the following series (Ref. 1).

$$f(t) = M + \sum_{R=1}^{\infty} G_R\sin(R\omega_o t + \phi_R) \qquad 3$$

where

$$G_R = \sqrt{A_R^{\,2} + B_R^{\,2}} \qquad 4$$

$$\phi_R = \tan^{-1}\left[\frac{A_R}{B_R}\right] \qquad 5$$

The term $G_R \sin(R\omega_o t + \phi_R)$ represents the R^{th} harmonic of the function f(t). The harmonic is expressed as the amplitude G_R and the phase ϕ_R. The function is defined exactly by the summation of an infinite number of harmonics, however for practical purposes the first 10 or 15 harmonics are usually sufficient. The user is required to define one period of the function as a number of discrete points in the f - t plane.

The program outputs the amplitude and phase for each harmonic as defined by equations 4 and 5 . Also values of the Fourier coefficients A_R and B_R are printed out.

The Fourier coefficients are evaluated using the following expressions:-

$$A_R = \frac{1}{T}\int_{-T}^{T} f(t)\cos R\omega_o t\,dt \qquad 6$$

$$B_R = \frac{1}{T}\int_{-T}^{T} f(t)\sin R\omega_o t\,dt \qquad 7$$

$$Ao = \frac{1}{T}\int_{-T}^{T} f(t)\,dt \qquad 8$$

where T is the half period.

Since f(t) is defined at discrete points in the f - t plane the function $f(t)\cos R\omega_o t$ may also be defined at discrete points by multiplying the value of each data point by the value of $\cos R\omega_o t$ which applies at that point. Once the value of $f(t)\cos R\omega_o t$ is defined the integration may be carried out using a summation method such as the Tapezoidal rule or Simpson's rule.

This work uses Simpson's rule which requires that the values of the dependent variable y are defined at regular intervals of the independent variable t and that an odd number of data points is specified. (Note that in the program the symbol x is used to indicate the independent variable.) The requirement of regular spacing and an odd number of points may not be convenient and the program has been designed so that the function may be expressed by any number of points, up to 200, with arbitrary spacing. The user sets certain switches which describe how the input data is supplied, the program interpolates between irregularly spaced points to produce an odd number of regularly spaced points which are submitted to the integration subroutine.

Computing procedure

The program reads in the number of functions the user wishes to analyse (nf), a couple of switches (nch, nch2) and the number of data points.(Figure 4). Depending on the values of the switches the program branches to read values of the independent variable into array ftx. This is only necessary when all the functions are described at the same values of the independent variable and the spacing is irregular.

The rest of the computing is carried out on each individual function by a do loop (do 10 K = 1,nf). Two separate processes are involved, the input and where necessary the manipulation of data, and the actual Fourier analysis. The program stops once all the functions have been processed. There is no limit on the number of functions which may be handled by one run of the program other than the normal computing time restriction imposed by the system.

The reading and manipulation of input data is a complicated procedure the outcome of which is determined by the input data switches and which is best appreciated by consulting the flowchart (Figure 5). The result of the process is the generation of an array ft which contains an odd number of regularly spaced data points which define the input function.

The Fourier analysis is shown in flowchart form in Figure 6 . For a given function each harmonic is calculated inside a do loop (do 20 i = 1,nh). An array containing values of $f(t)\cos R\omega_o t$ is generated, the integral $\int_{-T}^{T} f(t)\cos R\omega_o t\,dt$ is evaluated by a call to the subroutine integs, and the A_R coefficient is calculated as defined by equation 6 . The same procedure is followed to calculate the B_R coefficients. Do loop 20 continues until the required number of harmonics has been generated. Another call to subroutine integs is used to evaluate $\int_{-T}^{T} f(t)\,dt$ and hence the mean level is calculated (equation 8). Finally the harmonics are expressed in amplitude and phase form according to equations 4 and 5 and the results are printed out.

References

1 RAVEN F.H.
Mathematics of engineering systems (pg 14)
McGRAW HILL 1966

LIST OF VARIABLES USED

a	array containing Fourier coefficients of cosine terms	-	R
acos	result of integral in equation 6 (area under graph)	-	R
aft	result of integral in equation 8 (area under graph)	-	R
amp	amplitude	-	R
ao	coefficient Ao in the Fourier series equation 1	-	R
asin	result of integral in equation 7 (area under graph)	-	R
b	array containing Fourier coefficients of sine terms	-	R
deltax	increment of independent variable when function has even number of points	-	R
fcos	array containing $f(t)\cos R\omega_o t$ terms	-	R
freq	frequency of fundamental harmonic	c/s	R
fsin	array containing $f(t)\sin R\omega_o t$ terms	-	R
ft	array containing interpolated values of the dependent variable	-	R
ftx	array containing values of the independent variable	-	R
fty	array containing values of the dependent variable	-	R

h	increment of independent variable used in Fourier analysis	-	R
i	do loop counter	-	I
j	do loop counter	-	I
k	do loop counter	-	I
n	number of points defining function after interpolation	-	I
nch	input data switch	-	I
nchl	input data switch	-	I
nch2	input data switch	-	I
nf	number of functions to be analysed	-	I
nh	number of harmonics required as output	-	I
nm2	counter used in interpolation	-	I
nn	number of data points defining a given function	-	I
nnchl	variable used to check if number of points is even	-	I
nnch2	variable used to check if number of points is even	-	I
period	period of fundamental harmonic	-	R
phd	phase in degrees	deg	R
phr	phase in radian	rad	R
rmean	mean level of function	-	R
wo	angular frequency of fundamental harmonic	rad/s	R
x	independent variable	-	R
xinc	increment of independent variable used in interpolation	-	R
y	dependent variable	-	R

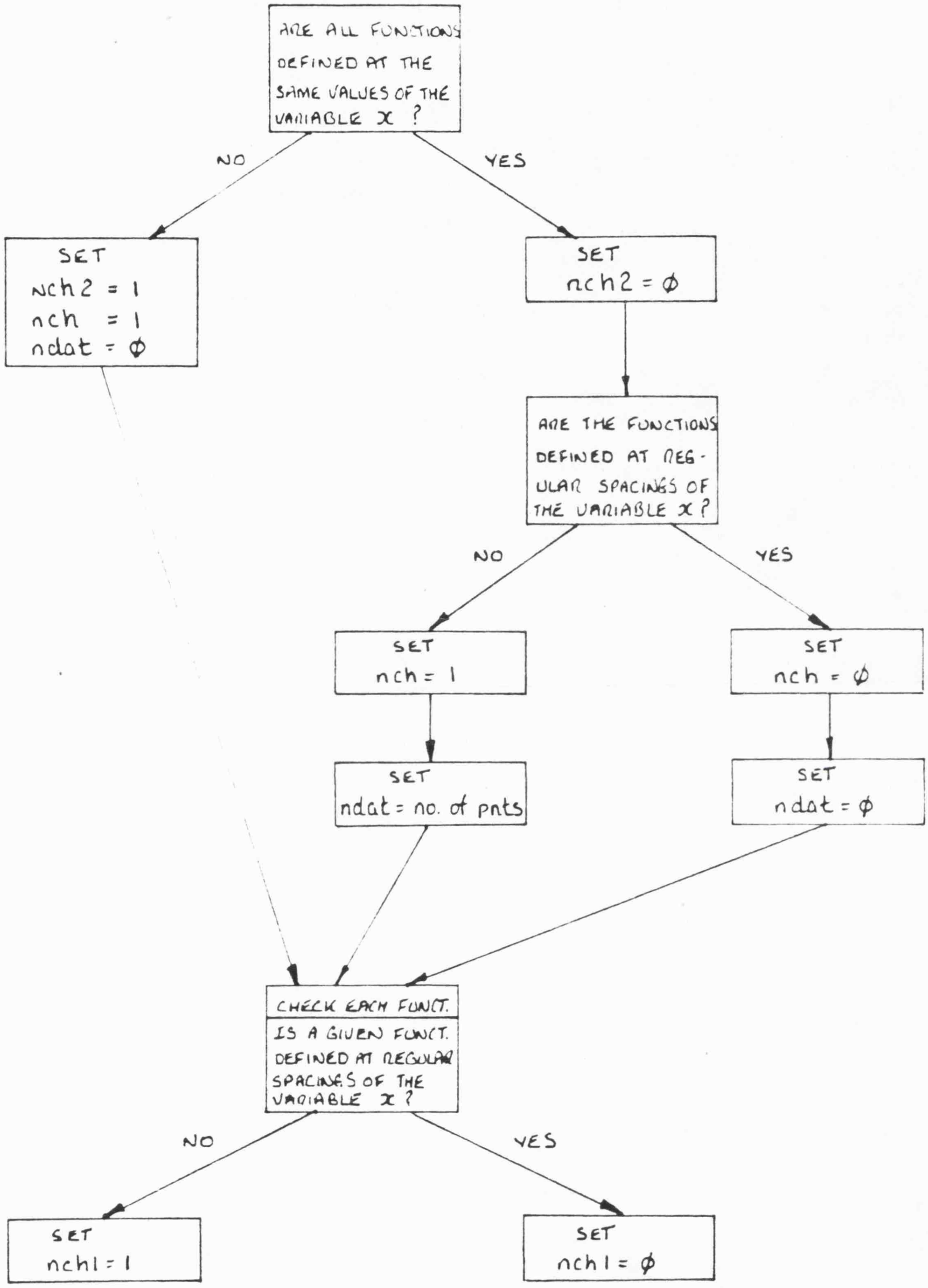

FIGURE 1 ALGORITHM FOR SETTING INPUT DATA SWITCHES

nf nch nch2 ndat

VALUES OF INDEPENDANT VARIABLE x FOR ALL FUNCTIONS FREE FORMAT MAXIMUM 200 POINTS	INCLUDE ONLY IF nch = 1 AND nch2 = 0

nn nh freq nchl

VALUES OF DEPENDENT VARIABLE Y FOR FUNCTION 1. FREE FORMAT		DATA FOR FUNCTION 1
VALUES OF INDEPENDENT VARIABLE x FOR FUNCTION 1. FREE FORMAT	INCLUDE ONLY IF nch2 = 1 AND nch = 1	

nn nh freq nchl

VALUES OF DEPENDENT VARIABLE Y FOR FUNCTION 2. FREE FORMAT		DATA FOR FUNCTION 2
VALUES OF INDEPENDENT VARIABLE x FOR FUNCTION 2. FREE FORMAT	INCLUDE ONLY IF nch2 = 1 AND nch = 1	

ETC

ETC

<u>FIGURE 2 STACKING OF INPUT DATA FILE</u>

```
fourier analysis results
************************

function number  1
==================

number of ordinates   = 135
no of harmonics requ. = 10
fundamental frequency =  171.0 hz

function data supplied at regular values of theindependant variable

data supplied=  dependant variable (v)
```

1.6659e+02	1.6828e+02	1.7002e+02	1.7173e+02	1.7337e+02	1.7490e+02	1.7630e+02	1.7764e+02	1.7893e+02
1.8018e+02	1.8142e+02	1.8265e+02	1.8386e+02	1.8506e+02	1.8625e+02	1.8739e+02	1.8846e+02	1.8947e+02
1.9042e+02	1.9132e+02	1.9221e+02	1.9307e+02	1.9389e+02	1.9472e+02	1.9556e+02	1.9639e+02	1.9720e+02
1.9800e+02	1.9874e+02	1.9944e+02	2.0009e+02	2.0069e+02	2.0125e+02	2.0177e+02	2.0225e+02	2.0271e+02
2.0314e+02	2.0355e+02	2.0393e+02	2.0430e+02	2.0465e+02	2.0497e+02	2.0528e+02	2.0558e+02	2.0539e+02
2.0620e+02	2.0654e+02	2.0691e+02	2.0731e+02	2.0774e+02	2.0821e+02	2.0870e+02	2.0921e+02	2.0971e+02
2.1022e+02	2.1072e+02	2.1117e+02	2.1159e+02	2.1198e+02	2.1235e+02	2.1271e+02	2.1304e+02	2.1335e+02
2.1361e+02	2.1383e+02	2.1404e+02	2.1422e+02	2.1438e+02	2.1451e+02	2.1462e+02	2.1474e+02	2.1485e+02
2.1496e+02	2.1507e+02	2.1517e+02	2.1526e+02	2.1534e+02	2.1539e+02	2.1540e+02	2.1536e+02	2.1529e+02
2.1518e+02	2.1503e+02	2.1486e+02	2.1468e+02	2.1448e+02	2.1427e+02	2.1408e+02	2.1387e+02	2.1364e+02
2.1341e+02	2.1315e+02	2.1298e+02	2.1259e+02	2.1229e+02	2.1198e+02	2.1170e+02	2.1143e+02	2.1118e+02
2.1095e+02	2.1074e+02	2.1053e+02	2.1031e+02	2.1005e+02	2.0974e+02	2.0940e+02	2.0901e+02	2.0850e+02
2.0813e+02	2.0779e+02	2.0738e+02	2.0699e+02	2.0657e+02	2.0606e+02	2.0539e+02	2.0448e+02	2.0322e+02
2.0151e+02	1.9931e+02	1.9658e+02	1.9334e+02	1.8968e+02	1.8569e+02	1.8154e+02	1.7740e+02	1.7342e+02
1.7000e+02	1.6709e+02	1.6489e+02	1.6346e+02	1.6280e+02	1.6284e+02	1.6350e+02	1.6464e+02	1.6610e+02

fourier analysis of function 1 mean level = 2.0045e+02

harmonic no	amplitude	phase (deg)	a	b
1	1.9012e+01	-1.0408e+02	-1.8441e+01	-4.6260e+00
2	7.9010e+00	-9.0069e+01	-7.9010e+00	-9.5267e-03
3	5.2815e+00	-7.9385e+01	-5.1912e+00	9.7290e-01
4	3.1211e+00	-5.2936e+01	-2.4905e+00	1.8811e+00
5	2.2797e+00	-2.6573e+01	-1.0200e+00	2.0388e+00
6	1.3738e+00	-1.0184e+01	-2.4291e-01	1.3522e+00
7	9.5023e-01	1.9420e+01	3.1595e-01	8.9617e-01
8	5.8277e-01	4.0729e+01	3.8025e-01	4.4162e-01
9	3.5301e-01	6.3679e+01	3.1641e-01	1.5652e-01
10	1.6416e-01	8.3378e+01	1.6306e-01	1.8929e-02

FIGURE 3 SAMPLE RESULTS FILE FROM PROGRAM fan.fortran

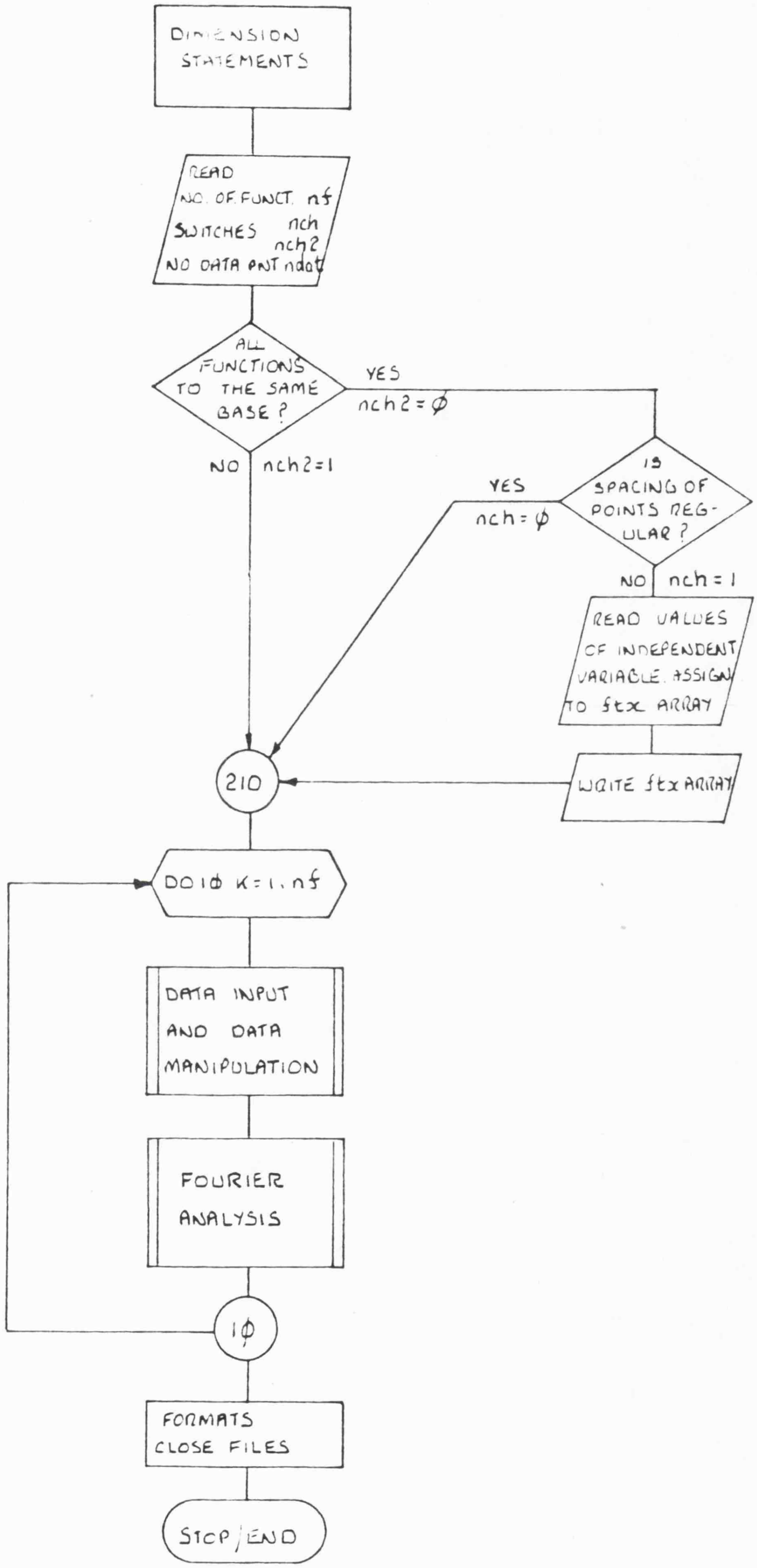

FIGURE 4 FLOWCHART FOR PROGRAM fan. fortran

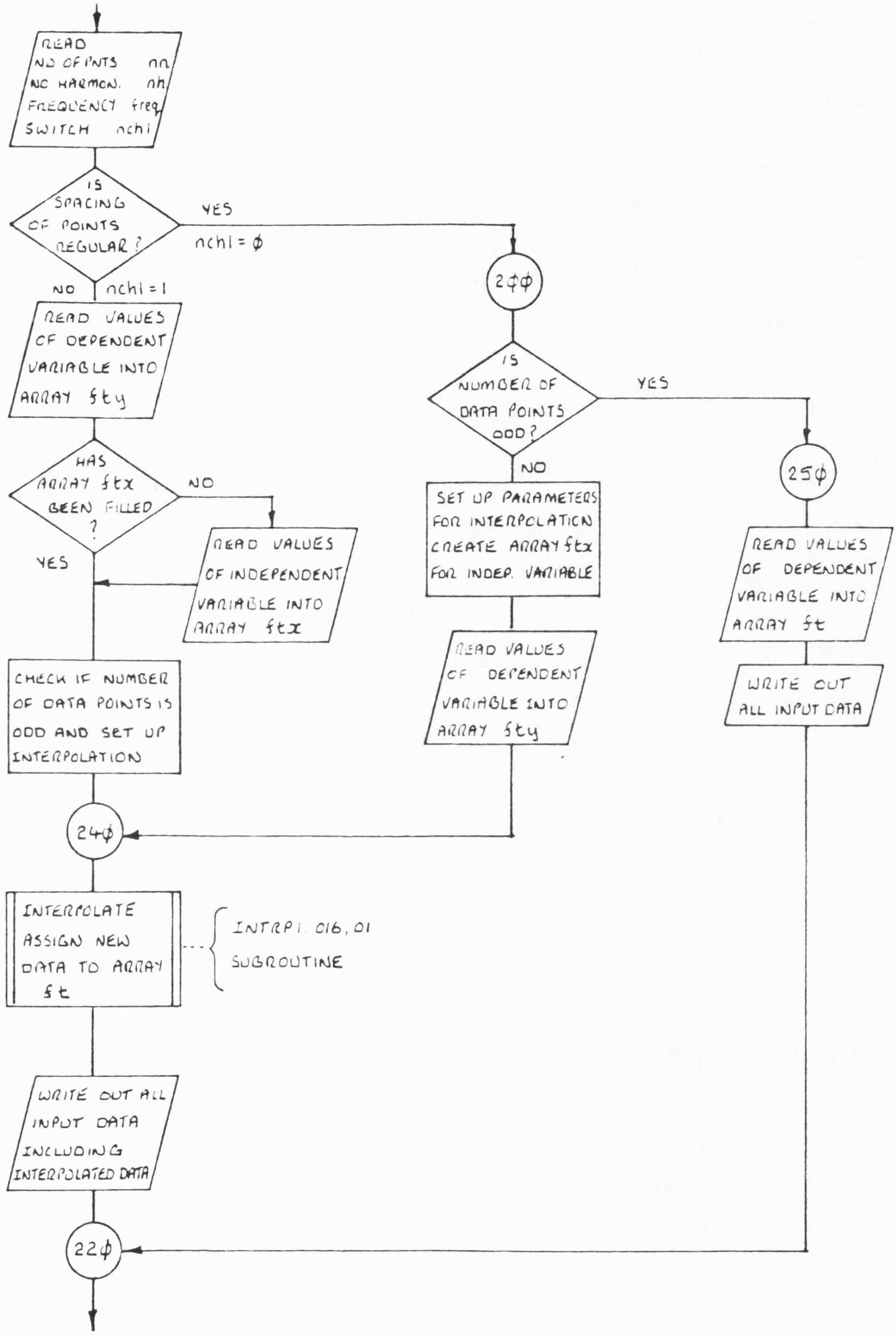

FIGURE 5 FLOWCHART FOR DATA INPUT AND DATA MANIPULATION SECTION

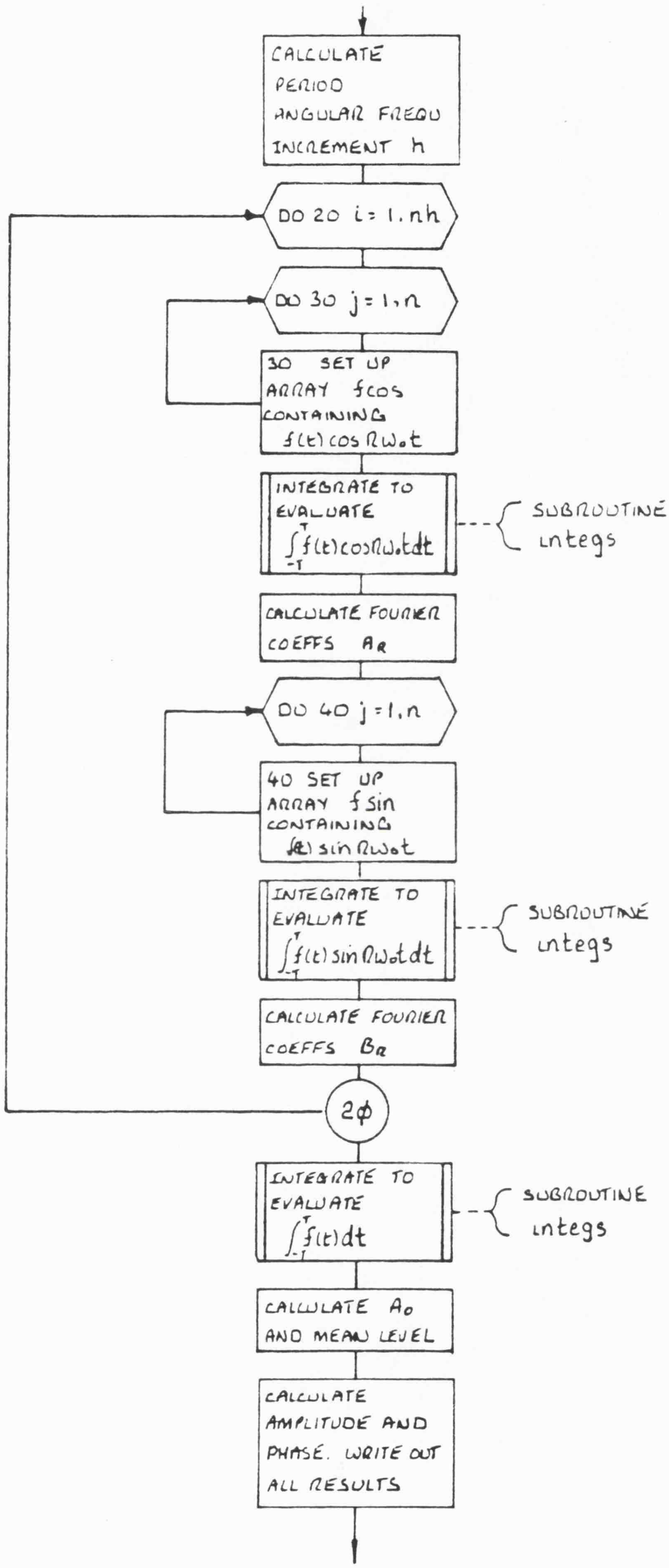

FIGURE 6 FLOWCHART FOR FOURIER ANALYSIS SECTION

```
                    fan.fortran         09/04/80  2234.7 bst Thu

c       PROGRAM NAME fan.fortran
c
c       LIBRARY CLASSIFICATION
c

c       TITLE  Fourier analysis of a function specified as a number of data points
c
c       fortranIV               honeywell multics      3 sept 1980
c
c       no special hardware requirements
c
c       author C.M. Skarbek-Wazynski
c
c       purpose   The program takes as data one period of any periodic
c                 function y=f(x) defined as a number of points in the
c                 x-y space and performs a Fourier analysis. The results
c                 are expressed as a number of harmonics in amplitude and
c                 phase form and also the corresponding Fourier coefficients
c                 are printed out.
c
c       associated subroutines
c
c           subroutine bengf:interp1.016,01
c           subroutine integs.fortran
c
c       input information  (read from a data file)
c
c       nf       number of functions
c       nch      input data switch
c       nch2     input data switch
c       ndat     number of data points
c
c       the following items of data must be specified for each function
c
c       nn       number of data points defining a given function
c       nh       number of harmonics required as output
c       freq     frequency of fundamental harmonic
c       nch1     input data switch
c       fty      array containing values of dependent variable
c       ftx      array containing values of independent variable
c
c       output information
c
c       a        array containing Fourier coeffs. of cosine terms
c       b        array containing Fourier coeffs. of sine terms
c       amp      amplitude
c       phd      phase in degrees
c       ft       array containing values of dependent variable
c                after processing
c
c       variables (excluding i/o variables)
c
c       acos      result of integral equ. 6
c       aft       result of integral equ. 8
c       asin      result of integral equ. 7
c       deltax    increment in x
```

```
c     fcos     array containing f(t)cosRwot terms
c     fsin     array containing f(t)sinRwot terms
c     h        increment in x
c     i        do loop counter
c     j        do loop counter
c     k        do loop counter
c     n        number of points after interpolation
c     nm2      n minus 2
c     nnch1    variable used to check if no. data points even
c     nnch2    variable used to check if no. data points even
c     period   period of fundamental
c     phr      phase in radians
c     rmean    mean level of function
c     wo       angular frequency of fundamental
c     x        independent variable
c     xinc     increment in x
c     y        dependent variable
c
c
c
      dimension ftx(201),fty(201),ft(201),a(20),b(20)
      dimension fsin(201),fcos(201)
c
c
c     read in no. of functions and switches
      read(5,500)nf,nch,nch2,ndat
      write(6,600)
c
c
c     are all functions to the same base ?
c     is spacing of points regular ?
      if(nch2.eq.1)go to 210
      if(nch.eq.0)go to 210
c
c     values of all functions supplied at same values of the
c     independent variable
      read(5,500)(ftx(i),i=1,ndat)
      write(6,610)(ftx(i),i=1,ndat)
c
  210 continue
c
c
c
      do 10 k=1,nf
c
c
c     data input and data manipulation section
c     ========================================
c
      read(5,500)nn,nh,freq,nch1
c
c     is spacing of points regular ?
      if(nch1.eq.0)go to 200
c
c     values of functions not supplied at regular intervals
c     of the independent variable (x)
      read(5,500)(fty(i),i=1,nn)
c
c     has array ftx been filled ?
      if(nch2.eq.1)read(5,500)(ftx(i),i=1,nn)
c
```

```
c     check if no of data points is odd and set up interpolation parameters
      nnch1=nn/2
      nnch2=nnch1*2
      n=nn
      if(nnch2.eq.nn)n=nn+1
      xinc=(ftx(nn)-ftx(1))/(n-1)
c
  240 continue
c
c     interpolation procedure
      nm2=n-2
      do 202 i=1,nm2
      j=i+1
      x=xinc*i
      call intrp1(x,y,ftx,fty,nn)
      ft(j)=y
  202 continue
      ft(1)=fty(1)
      ft(n)=fty(nn)
c
c     write out all input data
      write(6,602)k
      write(6,603)nn,nh,freq
      write(6,611)
      write(6,612)(fty(i),i=1,nn)
      if(nch2.ge.1)write(6,613)(ftx(i),i=1,nn)
      write(6,614)(ft(i),i=1,n)
      go to 220
c
c     data points regularly spaced
  200 continue
c
c     is no. of data points odd ?
      nnch1=nn/2
      nnch2=nnch1*2
      if(nnch2.ne.nn)go to 250
c
c     set up parameters for interpolation
c     and create ftx array
      period=1.0/freq
      deltax=period/(nn-1)
      do 230 i=1,nn
  230 ftx(i)=deltax*(i-1)
      n=nn+1
      xinc=period/(n-1)
c
c     read values of independent variable
      read(5,500)(fty(i),i=1,nn)
      go to 240
c
c     spacing of data is regular and an odd no of points is specified
  250 continue
      n=nn
c
c     read values of independent variable into array ft
      read(5,500)(ft(i),i=1,n)
c
c     write out all input data
      write(6,602)k
      write(6,603)n,nh,freq
      write(6,615)
      write(6,612)(ft(i),i=1,n)
```

```
c
  220 continue
c
c     fourier analysis section
c     ========================
c
c     calculate function parameters
      period=1.0/freq
      wo=2.0*3.14159*freq
      h=period/(n-1)
c
      do 20 i=1,nh
c
      do 30 j=1,n
c     set up array fcos
   30 fcos(j)=ft(j)*cos(i*wo*(j-1)*h)
c
c     integrate
      call integs(n,h,fcos,acos)
c
c     ar coefficients
      a(i)=(2.0/period)*acos
c
      do 40 j=1,n
c     set up array fsin
   40 fsin(j)=ft(j)*sin(i*wo*(j-1)*h)
c
c     integrate
      call integs(n,h,fsin,asin)
c
c     br coefficients
      b(i)=(2.0/period)*asin
c
   20 continue
c
c     integrate
      call integs(n,h,ft,aft)
c
c     ao coefficient (mean level)
      ao=(2.0/period)*aft
      rmean=ao/2.0
c
c     write out results
      write(6,605)k,rmean
      do 60 i=1,nh
      amp=sqrt(a(i)*a(i)+b(i)*b(i))
      phr=atan2(a(i),b(i))
      phd=phr*180.0/3.14159
      write(6,606)i,amp,phd,a(i),b(i)
   60 continue
c
   10 continue
c
c     formats and close files
  500 format(v)
  602 format(1h0,'function number ',i2/1h ,18('='))

  600 format(1h0,'fourier analysis results'/1h ,24('*'))
  603 format(1h0,'number of ordinates   = ',i3/1h ,
     1'no of harmonics requ. = ',i2/1h ,
     1'fundamental frequency = ',f6.1,' hz')
```

```
  605 format(1h0,'fourier analysis of function ',i2,12x,
     1'mean level = ',1pe11.4/1h0,
     16x,'harmonic no',5x,'amplitude',7x,'phase (deg)',
     115x,'a',22x,'b')
  606 format(1h0,10x,i2,9x,1pe11.4,5x,1pe11.4,
     111x,1pe11.4,13x,1pe11.4)
  610 format(1h0,'values of all functions supplied at the ',
     1'same values of the independent variable (x)'/1h0,
     1'independent variable (x)'//1h0,9(3x,1pe11.4))
  611 format(1h0,'function data supplied at irregular intervals ',
     1'of the independent variable (or no of data points is even')
  612 format(1h0,'data supplied-  dependent variable (y)'//
     11h0,9(3x,1pe11.4))
  613 format(1h0,'data supplied- independent variable (x)'//
     11h0,9(3x,1pe11.4))
  614 format(1h0,'interpolated values for the dependent variable (y)'//
     11h0,9(3x,1pe11.4))
  615 format(1h0,'function data supplied at regular values of the',
     1'independent variable')
      close(5)
      close(6)
c
      stop
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR PROGRAM harm.fortran

LIBRARY CLASSIFICATION

SUMMATION OF N HARMONICS TO YIELD RESULTING WAVEFORM

FORTRAN IV MULTICS 25 AUG. 1980

No special hardware requirements

Author: C.M. SKARBEK-WAZYNSKI

Purpose

The program takes as data a number of harmonics, expressed as amplitudes and phases, which are added together to produce the original waveform. Up to a maximum of 20 harmonics can be handled. The resultant waveform is stored as a number of data points and a plot file is generated which may be output using the normal MULTICS graphics commands.

Associated subroutines

Multics subroutines
plot_ $setup (descriptors)
plot_ (descriptors)
For further details consult MULTICS MPM subroutines.

Input information (read from data file)

ng	number of functions to be synthesized	-	I

The following items of data must be supplied for each function the user wishes to synthesize.

n	number of harmonics (maximum 20)	-	I
hz	frequency of fundamental (cycles/sec)	c/s	R
ml	mean level (if known, otherwise set to zero)	-	R

nc	number of cycles of the fundamental to be calculated	-	I
np	number of points per cycle	-	I
amp	array containing amplitude values for n harmonics	-	R
faze	array containing phase values for n harmonics (degrees)	deg	R
title	title to be printed on graph output maximum 50 characters	-	CH
xlabel	label for x axis maximum 20 characters	-	CH
ylabel	label for y axis maximum 20 characters	-	CH

All input data may be submitted in free format. See Figure 1 as an example of an input data file.

Output information

The program prints out all the input data, and the following arrays:-

x	array containing values of the independent variable calculated by program
y	array containing values of the dependent variable calculated by the program

An example of the program printed output is given in Figure 3 . The calls to the plotting subroutine generate a plot file which can be output on the Multics plotter using standard multics graphics commands (Appendix 1). A typical graphical output is shown in Figure 2 .

Inbuilt error messages

No normal failures can occur provided the input data is correct. There are no error messages. Note maximum number of points is 500, i.e. nc*np <500.

Timing and storage requirements

Both timing and storage depend on the number of functions the user wishes to synthesize and the amount of graphical output required.

Program action and algorithm

Mathematical model

Any repetitive function f(t), which has a period P may be expressed as a Fourier Series when $\omega_o = 2\pi/P$.

$$f(t) = \frac{Ao}{2} + A_1\cos\omega_o t + A_2\cos 2\omega_o t + A_3\cos 3\omega_o t \ldots$$
$$+ B_1\sin\omega_o t + B_2\sin 2\omega_o t + B_3\sin 3\omega_o t \ldots \qquad 1$$

general form

$$f(t) = M + \sum_{R=1}^{\infty} A_R\cos R\omega_o t + \sum_{R=1}^{\infty} B_R\sin\omega_o t \qquad 2$$

where M is the mean level of the function.

Manipulation of the sinusoidal terms gives the following series (Ref. 1).

$$f(t) = M + \sum_{R=1}^{\infty} G_R\sin(R\omega_o t + \emptyset_R) \qquad 3$$

where $G_R = \sqrt{A_R^{\;2} + B_R^{\;2}}$

$$\emptyset_R = \tan^{-1}\left[\frac{A_R}{B_R}\right]$$

The term $G_R\sin(R\omega_o t + \emptyset_R)$ represents the R^{th} harmonic of amplitude G and phase $\emptyset$. Thus any repetitive signal may be expressed as a mean level plus a series of harmonics.

The program harm.fortran described here takes as data the mean level of a function and a number of harmonics in amplitude and phase form, these are then added together as in equation 3 thus synthesizing the original function f(t). The function is decribed as a number of points in the f - t space. Note the maximum number of points the program can store is 500, in other words the product of the input variables nc and np must not exceeed 500. Although this restriction may be relaxed by altering the dimensions of the x and y storage arrays.

The program provides a printout of the calculated points and a plot file is generated which may be output using standard Multics terminal commands. (Appendix 1.)

Computing procedure

The summation of harmonics is performed by carrying out the following steps for each set of harmonics supplied as data.

1 The angular frequency of the fundamental harmonic is calculated $\omega f = 2\pi.hz$ and the total number of points at which the function f(t) will be evaluated is given by npt = (nc·np) - (nc - 1). Finally the effective time increment between each of these calculation points is given by:-

period of the fundamental	pf = 1.0/hz
time increment	dt = pf/(np - 1)

2 The evaluation of the function is performed at each of the calculation points by a do loop denoted by counter i (do 10 i = 1,npt). Firstly the value of the function (qs) at the i^{th} point is set to zero, then the angle turned through by the fundamental (thetaf) is calculated.

$$thetaf = \omega f.dt.(i - 1)$$

The actual summation of harmonics is carried out in a nested do loop denoted by counter K (do 20 K = 1,n), K is effectively the harmonic number. The amplitudes and phases of each harmonic in turn are assigned to variables ampl and phas respectively. The angle rotated through by the K^{th} harmonic is obtained from

$$angle = (thetaf.K) + phas/57.3$$

(Note the divisor 57.3 converts the phase in degrees to radians).

The contribution of the K^{th} harmonic to the value of the function at the i^{th} point is calculated as:-

fharm = amp.sin(angle)

The value of fharm is added to the variable qs which contains the summation of all the harmonics so far. The do loop 20 continues until all the harmonics have been summed into variable qs.

The value of time, the independent variable at the i^{th} point is calculated [time = (i - 1).dt] and is assigned to array x(i). The value of the function f(t) at the i^{th} point is qs plus the mean level ml, the sum of these two variables is assigned to array y(i,1).

Do loop 10 continues until values of the function at all the calculation points have been evaluated.

3 The arrays x(i) and y(i,1) are printed out. And finally calls are made to standard plotting routines to set up a plot file.

4 The entire procedure described above is performed within a do loop (do 50 M = 1,ng) which repeats itself for as many functions at the user chooses to input at data. The whole process is shown as a flow diagram in Figure 4 .

References

1 RAVEN, F.H.
'Mathematics of engineering systems' page 14
M^c GRAW HILL 1966

APPENDIX 1

Commands to operate the Multics Graphics facility

sg ␣ - of ␣ filename.graphics	opens a graphic file with name filename

run program using standard Multics commands

rg	closes graphics file

to view graphs on Tektronix terminal

login	login on Tektronix 4012 terminal
stty ␣ - modes ␣ echoplex	sets echo on terminal for convenience
asr ␣ > axl	adds search rules to access plotting commands
sq ␣ - table ␣ tek_4012	sets up graphics facility
show ␣ filename	displays graph on screen if more than 1 graph is contained in graphics file press return

If graphs are alright the graphics file may be used to produce a hard copy on the Bath printer using the dplot command. {NB {t, n, m} are size codes use only one.

dplot ␣ - rqt ␣ bath ␣ sz ␣ {t, n, m} ␣ filename.graphics

LIST OF VARIABLES USED

amp	array containing amplitude values for n harmonics	-	R
ampl	variable used inside do loop for amplitude values	-	R
angle	angle turned through by K^{th} harmonic (radians)	rad	R
base	argument in plot subroutine call statement	-	R
dt	increment between calculation points	-	R
faze	array containing phase values for n harmonics	deg	R
fharm	contribution made to function by K^{th} harmonic at i^{th} point	-	R
hz	fundamental frequency (cycles/sec)	c/s	R
i	do loop counter	-	I
k	do loop counter	-	I
m	do loop counter	-	I
ml	mean level of function	-	R
n	number of harmonics	-	I
nc	number of cycles of the fundamental		

	to be calculated	-	I
npt	total number of points to be calculated	-	I
pf	period of fundamental harmonic (seconds)	-	R
phas	variable used inside do loop for phase values	rad	R
qs	summation of harmonic components	-	R
thetaf	angle turned through by fundamental harmonic (radians)	rad	R
time	time (i.e. values of independent variable for plotting purposes)	-	R
title	title to be printed on output graph (maximum 50 characters)	-	CH
wf	angular frequency of the fundamental harmonic (rad/sec)	rad/s	R
x	array containing values of the independent variable (time)	-	CH
xlabel	label for x axis (maximum 20 characters)	-	CH
y	array containing calculated values of the dependent variable f(t)	-	R
ylabel	label for y axis (maximum 20 characters)		CH

```
2
10 171.0 200.0 1 100
14.02 3.805 6.15 3.67 2.48 1.90 1.835 1.165 1.025 0.5725
-106.0 -39.0 -33.0 -65.0 -43.0 -29.0 -12.0 78.0 57.0 28.0
experimental pressure ripple resonant system
time (seconds)
pressure bar
10 171.0 200.0 1 100
3.733 2.206 5.333 2.80 1.289 1.305 1.705 0.675 0.768 0.522
37.0 -41.0 11.0 173.0 116.0 169.0 -59.0 -94.0 -34.0 48.0
experimental pressure ripple nonresonant system
time (seconds)
pressure bar
```

FIGURE 1 INPUT DATA FILE FOR PROGRAM harm.fortran

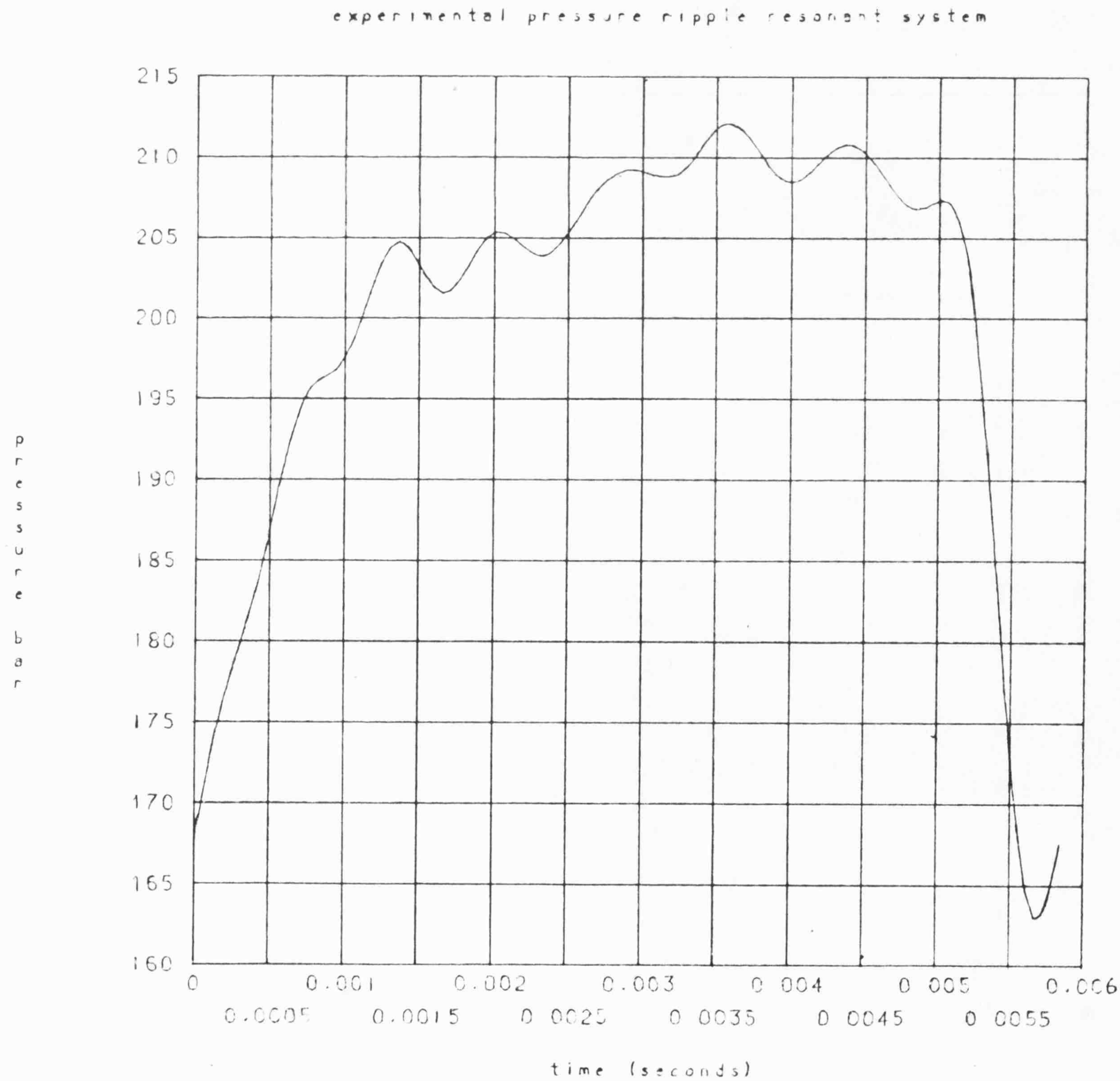

FIGURE 2 GRAPHICAL OUTPUT FROM PROGRAM harm.fortran

```
summation of harmonics results
******************************

function number  1
==================

number of harmonics     = 10
fundamental frequency = 1.[illegible]e+02
mean level              = 2.0000e+02
no of cycles output     =  1
no of points/cycle      = 100

amplitude data

  1.4000e+01  8.2050e+00  6.1500e+00  3.6700e+00  2.4800e+00  1.9000e+00  1.8350e+00  1.1650e+00  1.0250e+00  5.7250e-01

phase data

 -1.0500e+02 -3.9000e+01 -5.3700e+01 -6.5000e+01 -4.3000e+01 -2.9000e+01 -1.2000e+01  7.8000e+01  5.7000e+01  2.8000e+01

independant variable x

  0.0000e+00  5.9070e-05  1.1814e-04  1.7721e-04  2.3628e-04  2.9535e-04  3.5442e-04  4.1349e-04  4.7256e-04  5.3163e-04
  5.9070e-04  6.4977e-04  7.0884e-04  7.6791e-04  8.2698e-04  8.8605e-04  9.4512e-04  1.0042e-03  1.0633e-03  1.1223e-03
  1.1814e-03  1.2405e-03  1.2995e-03  1.3586e-03  1.4177e-03  1.4768e-03  1.5358e-03  1.5949e-03  1.6540e-03  1.7130e-03
  1.7721e-03  1.8312e-03  1.8902e-03  1.9493e-03  2.0084e-03  2.0675e-03  2.1265e-03  2.1856e-03  2.2447e-03  2.3037e-03
  2.3628e-03  2.4219e-03  2.4809e-03  2.5400e-03  2.5991e-03  2.6582e-03  2.7172e-03  2.7763e-03  2.8354e-03  2.8944e-03
  2.9535e-03  3.0126e-03  3.0717e-03  3.1307e-03  3.1898e-03  3.2489e-03  3.3079e-03  3.3670e-03  3.4261e-03  3.4851e-03
  3.5442e-03  3.6033e-03  3.6624e-03  3.7214e-03  3.7805e-03  3.8396e-03  3.8986e-03  3.9577e-03  4.0168e-03  4.0758e-03
  4.1349e-03  4.1940e-03  4.2531e-03  4.3121e-03  4.3712e-03  4.4303e-03  4.4893e-03  4.5484e-03  4.6075e-03  4.6665e-03
  4.7256e-03  4.7847e-03  4.8438e-03  4.9028e-03  4.9619e-03  5.0210e-03  5.0800e-03  5.1391e-03  5.1982e-03  5.2573e-03
  5.3163e-03  5.3754e-03  5.4345e-03  5.4935e-03  5.5526e-03  5.6117e-03  5.6707e-03  5.7298e-03  5.7889e-03  5.8480e-03

dependant variable y

  1.6752e+02  1.7057e+02  1.7347e+02  1.7593e+02  1.7795e+02  1.7971e+02  1.8142e+02  1.3350e+02  1.8582e+02  1.8534e+02
  1.9[illegible]e+02  1.9[illegible]e+02  1.946[illegible]e+02  1.9565e+02  1.9515e+02  1.9447e+02  1.9685e+02  1.9757e+02  1.9874e+02  2.0025e+02
  2.0125e+02  2.0343e+02  2.044[illegible]e+02  2.0470e+02  2.0435e+02  2.0354e+02  2.0253e+02  2.0181e+02  2.0145e+02  2.0171e+02
  2.02[illegible]e+02  2.0323e+02  2.0422e+02  2.0570e+02  2.0534e+02  2.0528e+02  2.0492e+02  2.0444e+02  2.0403e+02  2.0385e+02
  2.0[illegible]7e+02  2.0435e+02  2.0554e+02  2.0598e+02  2.0666e+02  2.0744e+02  2.0812e+02  2.0865e+02  2.0905e+02  2.0917e+02
  2.[illegible]e+02  2.0905e+02  2.0840e+02  2.0825e+02  2.0876e+02  2.0900e+02  2.0948e+02  2.1015e+02  2.1091e+02  2.1154e+02
  2.1[illegible]1e+02  2.12[illegible]e+02  2.1170e+02  2.1116e+02  2.1135e+02  2.0952e+02  2.0885e+02  2.0847e+02  2.0842e+02  2.0867e+02
  2.[illegible]4e+02  2.0969e+02  2.1022e+02  2.1150e+02  2.1077e+02  2.1069e+02  2.1034e+02  2.0977e+02  2.0903e+02  2.0823e+02
  2.[illegible]e+02  2.0[illegible]e+02  2.0674e+02  2.0603e+02  2.0713e+02  2.0737e+02  2.0713e+02  2.0599e+02  2.0354e+02  1.9957e+02
  1.[illegible]e+02  1.9[illegible]e+02  1.8062e+02  1.7397e+02  1.6645e+02  1.6466e+02  1.6289e+02  1.6307e+02  1.6481e+02  1.6752e+02
```

FIGURE 3 SAMPLE RESULTS FILE FROM PROGRAM harm.fortran

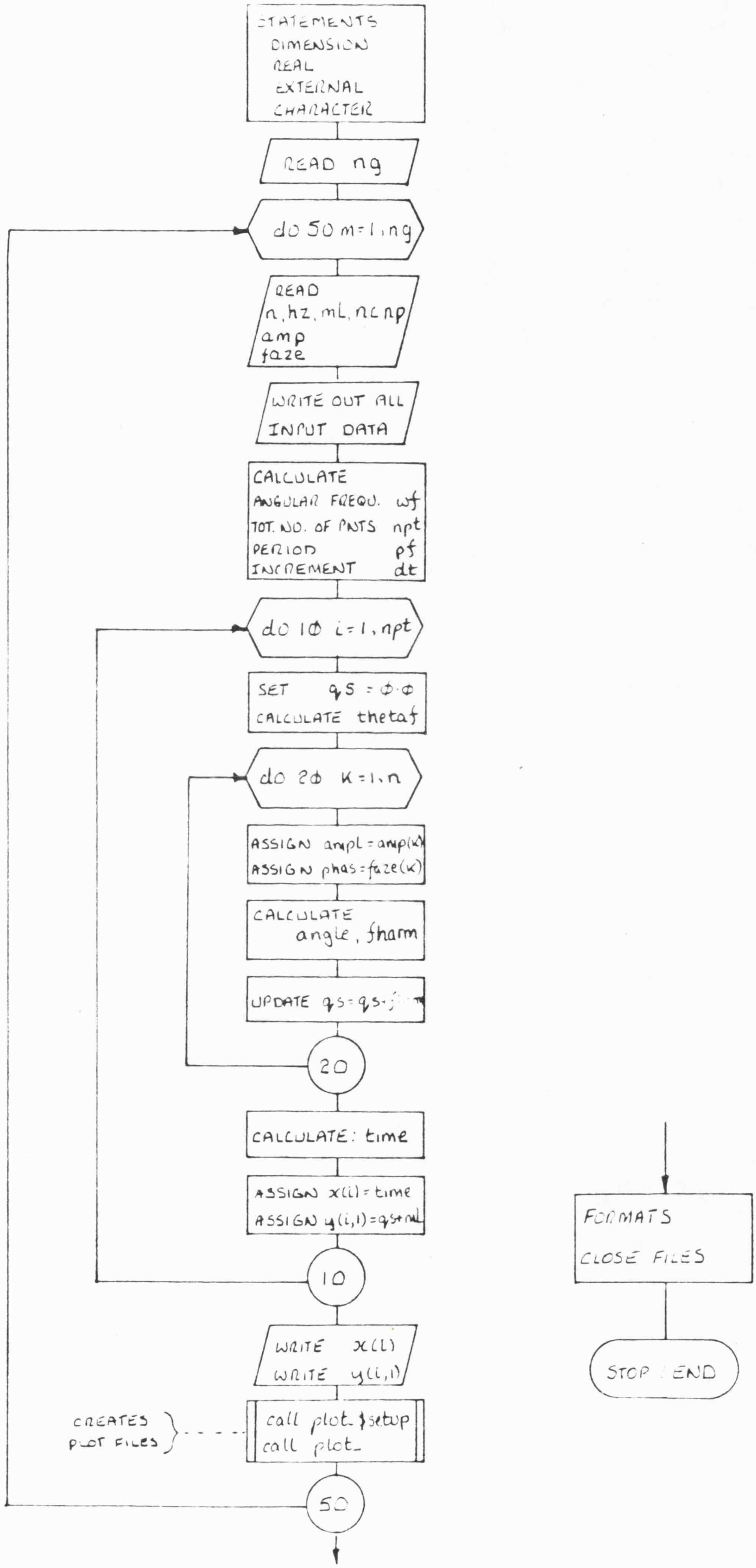

FIGURE 4 FLOWCHART FOR PROGRAM harm.fortran

```
c       PROGRAM NAME    harm.fortran
c
c       LIBRARY CLASSIFICATION
c
c       TITLE   Summation of harmonics to yield original function
c
c       fortranIV           Honeywell Multics      5 sept 1980
c
c       no special hardware
c
c       purpose   The program takes as data a number of harmonics
c                 expressed as amplitude and phase and adds them
c                 together to produce the original function
c
c       associated subroutines
c
c         multics subroutines   plot_$setup
c                               plot_
c
c       input information (read from a data file)
c
c       ng       number of functions
c
c          the following items of data must be specified
c          for each function
c
c       n        number of harmonics
c       hz       frequency of fundamental (hz)
c       ml       mean level  (real variable)
c       nc       number of cycles to be calculated
c       np       number of points per cycle
c       amp      array containing amplitude values
c       faze     array containing phase values (degrees)
c       title    title on graph   (max. 50 charac.)
c       xlabel   label for x axis (max. 20 charac.)
c       ylabel   label for y axis (max. 20 charac.)
c
c       output information
c
c       x       array containing values of independent variable
c       y       array containing values of dependent variable
c
c       variables   (excluding i/o variables)
c
c
c       ampl    amplitude values inside do loop
c       angle   angle turned through by kth harmonic
c       base    argument in plot subroutine
c       dt      time increment
c       fharm   harmonic contribution
c       i       do loop counter
c       k       do loop counter
c       m       do loop counter
c       npt     total number of points to be calculated
c       pf      period of fundamental
c       phas    phase values inside do loop
c       qs      summation of harmonic components
c       thetaf  angle turned through by fundamental
c       time    independent variable for plotting
c       wf      angular frequency of fundamental
```

```
c
      dimension amp(20),faze(20),y(502,1),x(502)
      real ml
      external plot_$setup (descriptors)
      external plot_ (descriptors)
      character*50 title
      character*20 xlabel
      character*20 ylabel
c
c     data input
c     ==========
      read(5,500)ng
      write(6,610)
c
      do 50 m=1,ng
c
      read(5,500)n,hz,ml,nc,np
      read(5,500)(amp(i),i=1,n)
      read(5,500)(faze(i),i=1,n)
      read(5,501)title
      read(5,502)xlabel
      read(5,503)ylabel
c
      write(6,600)m
      write(6,601)n,hz,ml,nc,np
      write(6,602)(amp(i),i=1,n)
      write(6,603)(faze(i),i=1,n)
c
      wf=hz*2.0*3.14159
      npt=(nc*np)-(nc-1)
      pf=1.0/hz
      dt=pf/(np-1)
c
      do 10 i=1,npt
c
      qs=0.0
      thetaf=wf*dt*(i-1)
c
      do 20 k=1,n
      ampl=amp(k)
      phas=faze(k)
c
      angle=(thetaf*k)+phas/57.3
      fharm=ampl*sin(angle)
c
      qs=qs+fharm
c
   20 continue
c
      time=(i-1)*dt
c
      x(i)=time
      y(i,1)=qs+ml
c
   10 continue
c
c      output of results
c     *****************
      write(6,604)(x(i),i=1,npt)
      write(6,605)(y(i,1),i=1,npt)
```

```
c
c     plot graphs
c     ***********
      call plot_$setup(title,xlabel,ylabel,1,base,2,0)
      call plot_ (x,y,npt,1,'.')
c
   50 continue
c
c     formats
c     =======
  500 format(v)
  610 format(1h0,'summation of harmonics results'/1h ,30('*'))
  600 format(1h0,'function number ',i2/1h ,18('='))
  601 format(1h0,'number of harmonics    = ',i2/1h ,
     1'fundamental frequency =',1pe11.4/1h ,
     1'mean level',12x,'=',1pe11.4/1h ,
     1'no of cycles output    = ',i2/1h ,
     1'no of points/cycle     = ',i3)
  602 format(1h0,'amplitude data'//1h ,10(1x,1pe11.4))
  603 format(1h0,'phase data'//1h ,10(1x,1pe11.4))
  501 format(a50)
  502 format(a20)
  503 format(a20)
  505 format(20h                    )
  604 format(1h0,'independant variable x'//1h ,10(1x,1pe11.4))
  605 format(1h0,'dependant variable y'//1h ,10(1x,1pe11.4))
      close(5)
      close(6)
c
      stop
      end
```

COMPUTER PROGRAM DOCUMENTATION FOR SUBROUTINE integs.fortran

LIBRARY CLASSIFICATION

INTEGRATION OF A FUNCTION USING SIMPSON'S RULE

FORTRAN IV HONEYWELL MULTICS 5 SEPT 1980

No special hardware requirements

Author: C.M. SKARBEK-WAZYNSKI

Purpose

Given a function y = f(t) specified as an odd number of discrete points at regular intervals of the independent variable, this routine evaluates the integral of the function, i.e. the area under the graph, using Simpson's rule.

No associated subroutines

All variables transferred via the argument list
call integs(n,h,ft,area)

Input information (via the argument list)

n	number of points defining function	-	I
h	increment in the independent variable	-	R
ft	array containing points defining given function	-	R

Output information (via the argument list)

area	area under the graph	-	R

Limitations and accuracy of program

No data is available on the accuracy which largely depends on the number of points used to specify a given function and the increment. The subroutine is limited to a maximum of 201 points to define the function and the number of points specified must be odd.

Inbuilt error messages

The program checks if the number of data points is odd, if not an error message is printed and the program terminates.

Timing and Storage

Not known.

Program action and algorithm

Simpson's rule is an summation method for finding the area under graphs in other words for evaluating definite integrals numerically. (Ref. 1) (See Figure 1)

$$\text{area under graph} = \frac{h}{3}\left(f_1 + 4f_2 + 2f_3 + 4f_4 + \ldots\ldots 4f_{2m} + f_{2m+1}\right) \qquad 1$$

which may be summarised as

$$\text{area} = \frac{h}{3}\left[f_1 + f_{2m+1} + 4\,\Sigma(\text{even ordinates}) + 2\,\Sigma(\text{odd ordinates})\right] \qquad 2$$

where

h = increment of independent variable between ordinates
f_m = value of m^{th} ordinate. (NB $2m + 1 = n$)

Note the use of 2m implies the area is divided into an even number of strips and hence there must be an odd number of ordinates. (n)

Computing procedure

The computing procedure is very straight forward. After initialising various variables a check is made to see if the function is specified by an odd number of points. If n is even an error message is printed out and the program stops. Otherwise the evaluation of equation 2 is performed in three steps:- (Figure 2)

1 even ordinates are summed, the total being assigned to variable even

2 odd ordinates are summed, the total being assigned to variable odd

3 the area under the graph is calculated and assigned to variable area which is output via the argument list.

Reference

BAJPAI, MUSTOE, WALKER
Engineering Mathematics (page 440)
John Wiley and Sons Limited 1978

LIST OF VARIABLES USED

area	area under the graph	-	R
even	summation of even ordinates	-	R
ft	array containing values of the function	-	R
h	increment of the independent variable	-	R
i	do loop counter	-	I
n	number of points defining the function	-	I
nch1	variable used to check if number of data points is odd	-	I
nch2	variable used to check if number of data points is odd	-	I
nm1	n minus 1, used in do loops	-	I
nm2	n minus 2, used in do loops	-	I
odd	summation of odd ordinates	-	R

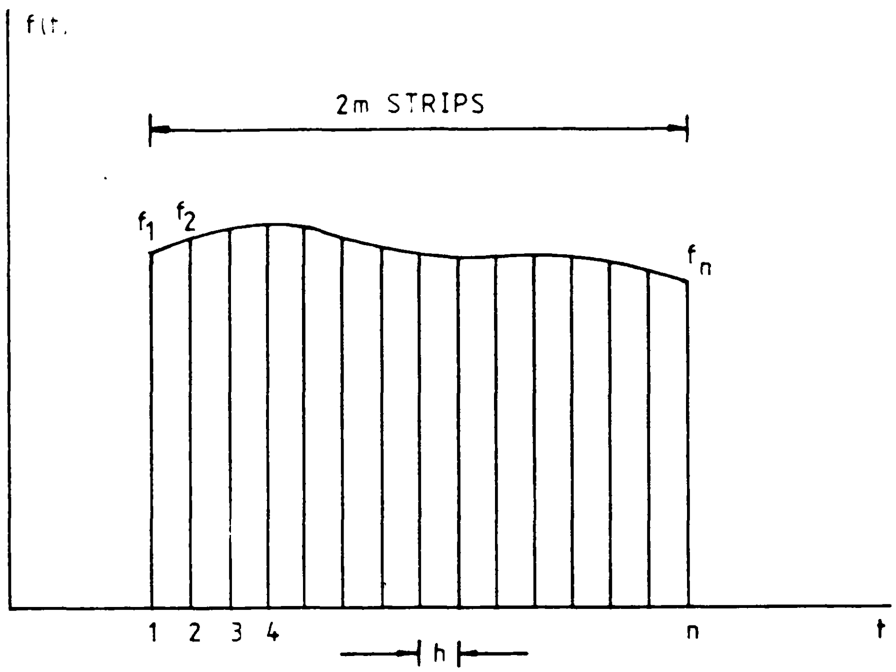

FIGURE 1 INTEGRATION USING SIMPSON'S RULE

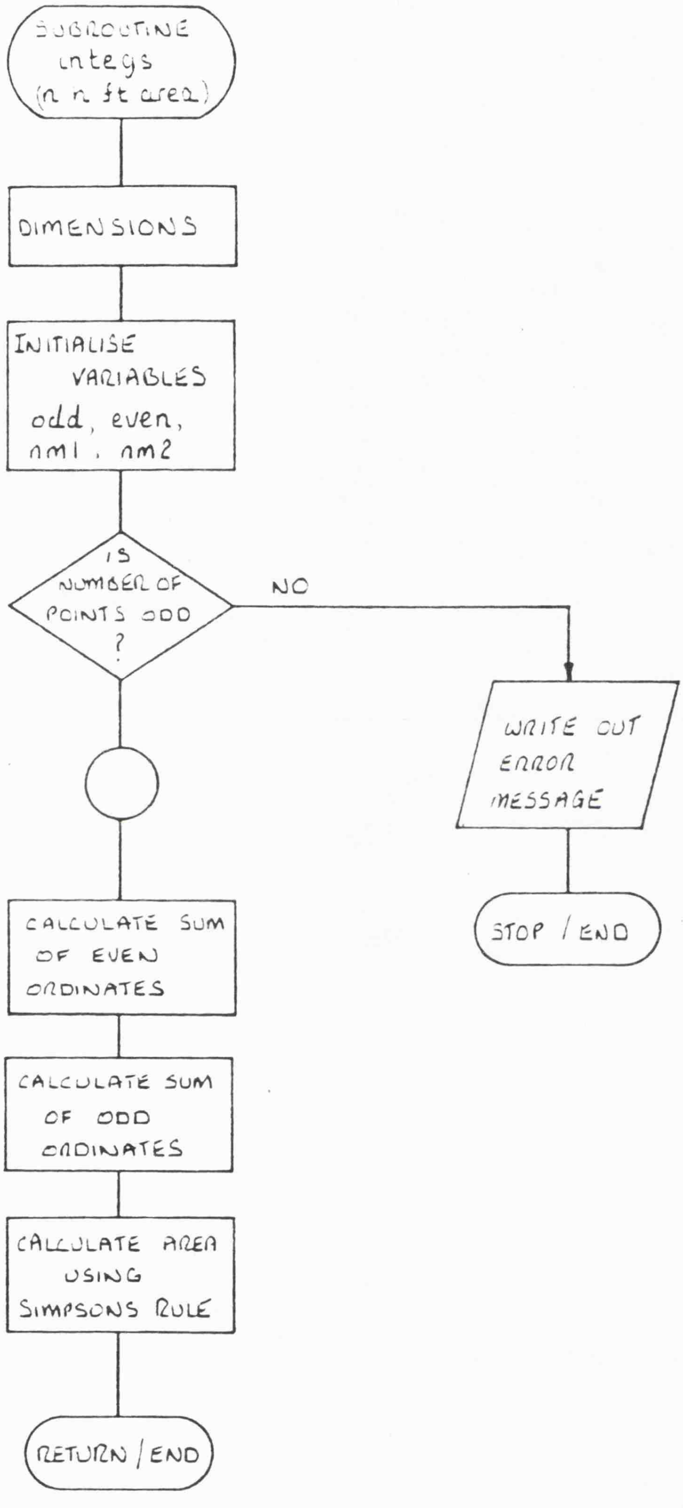

FIGURE 2 FLOWCHART FOR SUBROUTINE integs fortran

```
c
      subroutine integs(n,h,ft,area)
c
c     SUBROUTINE NAME integs
c
c     LIBRARY CLASSIFICATION
c
c     TITLE   Integration of a function using Simpsons rule
c
c     fortranIV            Honeywell Multics       5 sept 1980
c
c
c     author C.M. Skarbek-Wazynski
c
c     no special hardware requirements
c
c     purpose   Given a function y=f(t) specified as an odd number of
c               discrete points at regular intervals of the independent
c               variable, this routine evaluates the integral of the
c               function,ie. the area under the graph, using Simpsons
c               rule
c
c     no associated subroutines
c
c     input information  (via the argument list)
c
c     n        number of points defining the function
c     h        increment in the independent variable
c     ft       array containing values defining function
c
c     output information  (via argument list)
c
c     area     area under graph
c
c    variables   (excluding i/o variables)
c
c     even     summation of even ordinates
c     i        do loop counter
c     nch1     variable used to check if n is odd
c     nch2     variable used to check if n is odd
c     nm1      n minus 1, used in do loops
c     nm2      n minus 2, used in do loops
c     odd      summation of odd ordinates
c
c
      dimension ft(201)
c
c     initialise variables
      even=0.0
      odd =0.0
      nm1=n-1
      nm2=n-2
c
c     check that n is an odd number
      nch1=n/2
      nch2=nch1*2
      if(nch2.eq.n)go to 10
      go to 20
c
```

```
c     write error message
   10 write(6,600)
  600 format(1h0,'**subroutine integs error -n is an even number**')
      stop
c
   20 continue
c
c     sum even ordinates
      do 30 i=2,nm1,2
   30 even=even+ft(i)
c
c     sum odd ordinates
      do 40 i=3,nm2,2
   40 odd=odd+ft(i)
c
c     simpsons rule integration
      area=(h/3.0)*(ft(1)+ft(n)+4.0*even+2.0*odd)
c
      return
      end
```