12-43-74 *70333*

**TWO-DIMENSIONAL FLUID-HAMMER ANALYSIS
BY THE METHOD OF CHARACTERISTICS
IN A CLOSED AXISYMMETRIC CYLINDRICAL DOMAIN**

by

Yong W. Shin and Richard A. Valentin

BASE TECHNOLOGY

MASTER

U of C-AUA-USAEC

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

Prepared for the U.S. ATOMIC ENERGY COMMISSION

Division of Reactor Research and Development

for **under Contract W-31-109-Eng-38**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona	Kansas State University	The Ohio State University
Carnegie-Mellon University	The University of Kansas	Ohio University
Case Western Reserve University	Loyola University	The Pennsylvania State University
The University of Chicago	Marquette University	Purdue University
University of Cincinnati	Michigan State University	Saint Louis University
Illinois Institute of Technology	The University of Michigan	Southern Illinois University
University of Illinois	University of Minnesota	The University of Texas at Austin
Indiana University	University of Missouri	Washington University
Iowa State University	Northwestern University	Wayne State University
The University of Iowa	University of Notre Dame	The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy \$5.45; Microfiche \$2.25

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

TWO-DIMENSIONAL FLUID-HAMMER ANALYSIS
BY THE METHOD OF CHARACTERISTICS
IN A CLOSED AXISYMMETRIC CYLINDRICAL DOMAIN

by

Yong W. Shin and Richard A. Valentin

Components Technology Division

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

August 1974

MASTER

TABLE OF CONTENTS

	<u>Page</u>
NOMENCLATURE	6
ABSTRACT	7
I. INTRODUCTION.	7
II. GOVERNING DIFFERENTIAL EQUATIONS.	10
III. THE METHOD OF CHARACTERISTICS	12
A. Characteristic Surfaces	12
B. Bicharacteristics	13
C. Compatibility Equations	15
D. Boundary- and Axial-point Conditions.	16
IV. NUMERICAL PROCEDURE	17
A. System of Integral Equations and Their Finite-difference Approximation	17
B. Rectangular Spatial Grids	19
C. Geometry of Physical Domain and Boundary Conditions	19
D. Stepwise Integration Procedure	22
V. COMPUTER PROGRAM.	23
A. Description of Input Requirements	23
B. Limitations of NATRAN-II	30
VI. NUMERICAL CALCULATION.	31
A. Description of Sample Problem	31
B. Results and Discussion.	31
C. Further Remarks	34
APPENDIXES	
A. Approximate Solution of Bicharacteristic Curves	35
B. FORTRAN Program Listing	37

TABLE OF CONTENTS

	<u>Page</u>
C. Input Data for Sample Problem	71
D. Output of Sample-problem Calculation.	72
E. Source-pressure Calculation by Sodium-Water Reaction.	77
REFERENCES	79

LIST OF

<u>No.</u>	<u>Tit</u>
1.	Geometric Description of Streamline Vector $\vec{\lambda}$, and Generalized Cone w Time-Space
2.	Streamline and Bicharacteristics t t = t + Δt Plane and Their Interce
3.	Geometry of Two-dimensional Dor
4.	Geometric Configuration of Sample
5.	Pressure vs Time at Selected Poin dimensional Code NATRAN-II . . .
6.	Pressure vs Time at Selected Poin dimensional Code NATRANSIENT;
7.	Comparison of Results of NATRAN NATRANSIENT (1-D code) for Sam Ratio of 2
8.	Pressure vs Time at Selected Poin dimensional Code NATRANSIENT;
9.	Pressure vs Time at Selected Poin dimensional Code NATRAN-II with
10.	Pressure vs Time at Selected Poin dimensional Code NATRANSIENT

TWO-DIMENSIONAL FLUID-HAMMER ANALYSIS
BY THE METHOD OF CHARACTERISTICS
IN A CLOSED AXISYMMETRIC CYLINDRICAL DOMAIN

by

Yong W. Shin and Richard A. Valentin

ABSTRACT

A numerical-integration technique, based on the method of characteristics, is formulated for the system of quasi-linear hyperbolic equations describing fluid-hammer phenomena in a closed two-dimensional axisymmetric domain. The set of non-linear partial differential equations is first converted to a system of integral equations written along bicharacteristics. The solution is constructed in a step-by-step procedure by approximating the integral equations by the trapezoidal formula and using the resulting difference relations along four bicharacteristics and one streamline through each point. Spatial derivatives of the dependent variables are eliminated from the difference relations. The solution at any point then is obtained explicitly by a linear combination of the remaining difference relations.

The method is programmed for an unsteady motion of liquids in a closed system. The one-dimensional fluid-hammer computer program NATRANSIENT is incorporated, and the combined program is capable of performing a two-dimensional analysis in any one section of a complex system while treating the remaining sections as one-dimensional. As an example of the use of the program, sample problems of a simple geometry are computed and discussed. One-dimensional calculations for these same problems are compared.

I. INTRODUCTION

Basically, there are two different methods of numerical integration of hyperbolic systems of partial differential equations in three independent variables (e.g., two space and one time coordinates): (1) the direct finite-difference method, and (2) the method of characteristics. The direct finite-difference method uses either the artificial viscosity¹ or a special differencing scheme.² Both schemes use a rectangular grid and are readily amenable to direct machine coding. Nevertheless, the solution is inherently diffused, and any discontinuities or rapid variations that may exist in the solution can be unrealistically

spread out; hence, the method requires a small time step. Difficulties of this type do not exist in the method of characteristics since the equations are rendered in characteristic form before applying the finite-difference relations; hence, the differential equations include only those derivatives in the direction that the solution is extendable. Consequently, the method is potentially more accurate and appropriate for hyperbolic systems; however, it is logically more complex and hence requires an extensive programming effort. Nevertheless, this disadvantage in the method of characteristics can be well justified in view of its accuracy and the comparatively short machine time required to achieve a desired accuracy.

A direct extension of the two-variable method to the three-variable (or more) method of characteristics is not possible. In the two-variable method, a great simplification occurs, since the partial differential equations, when written in a characteristic form, reduce to ordinary differential equations along the characteristic lines. In the three-variable method, the reduced equations remain partial differential equations (but involving one less independent variable) with the property that they do not contain derivatives normal to the characteristic surfaces. The three-variable method is weaker in that infinitely many compatibility relations can be obtained, corresponding to a single infinity of characteristic curves. Nonetheless, the reduction in the number of independent variables, and hence, the number of partial derivatives, is still a simplification. Derivatives in characteristic directions can be integrated directly by numerically evaluating derivatives in other directions. The solution to the characteristic conditions determines the region of dependence at any point. From this, the criterion for numerical stability and convergence is easily determined. This criterion is sufficiently simple that a close relationship exists between the physical phenomenon and the numerical procedure. The envelope of all characteristic surfaces at a point forms a cone called the characteristic cone, inside which is the domain of dependence of the point under consideration.

The two-variable method has been used extensively in all areas of science and engineering. The method of characteristics, as applied to problems involving more than two independent variables, has been considered since late in the 1940's. Some of the early works include Thornhill,³ Holt,⁴ and Butler,⁵ and, more recently, Sauer⁶ and Richardson.⁷ Slight variations exist in the methods used by various investigators. These result from the nature of the weak formulation in the three-variable method. Holt used two bicharacteristics and a curve that is the intersection of the two characteristic surfaces through the two chosen bicharacteristics. Sauer used curves called nearcharacteristics that lie outside the characteristic cone. In both of these methods, the solution procedure refers directly to conditions outside the true domain of dependence of the differential system; as a result, it is not clear what effects this will have on the numerical solution. Butler and Richardson used four bicharacteristics and a streamline that always stays inside the cone; hence, the solution is determined solely by conditions inside the cone. The four-variable

method (three space coordinates and time) has also been considered.^{8,9} However, an actual numerical calculation has not been made, due to limitations in existing computing machines. A survey of the literature revealed no applications of the method of characteristics to multidimensional fluid-hammer problems in a closed system.

This report discusses in detail the characteristic formulation of the two-dimensional fluid-hammer equations, the geometric interpretation of the characteristic conditions, parametric representation of bicharacteristics and compatibility relations, and a method for obtaining a numerical solution in an explicit manner. The original partial differential equations are cast in a form that does not include derivatives normal to characteristic surfaces through a point, and the equations are written as integrals along bicharacteristics. A set of finite-difference relations is obtained from the set of integral equations by approximating them by a trapezoidal formula. After a judicious choice of sufficient bicharacteristics (four) from the infinity available through a point, it is found that the corresponding reduced equations still contain derivatives along directions outside the characteristic cone. These derivatives at an unknown point are eliminated by linearly combining the difference relations applied along four bicharacteristics and the streamline. The solution is then obtained from the remaining algebraic relations in an explicit form in which the derivatives in the initial surface are evaluated numerically. Since the sonic speed remains constant and the initial condition is given on a constant time plane, a fixed spatial grid with equal spacings and a constant time step are advantageously employed in the numerical procedure. Initial values needed in the solution procedure are interpolated using a four-point bivariate formula. One-time-step numerical integration consists of repeating the above procedure for all unknown grid points, thereby obtaining the solution in the entire new time plane.

A computer program, NATRAN-II, has been written to perform the indicated two-dimensional solution; in addition, it incorporates the one-dimensional system code NATRANSIENT.¹⁰ NATRAN-II is thus capable of computing pressure transients in a two-dimensional axisymmetric domain connected to a system of one-dimensional pipe sections. The source pressure can be either a pressure-time function specified at some point in the two-dimensional domain or a pressure-time function (or a sodium-water reaction) specified at some location in the one-dimensional domain.

Finally, simple two-dimensional problems are calculated by NATRAN-II, and the results are compared with the one-dimensional calculations by NATRANSIENT.

II. GOVERNING DIFFERENTIAL EQUATIONS

In flows involving a large pressure variation or a rapidly changing unsteady motion, the small compressibility of liquids is an important feature and can no longer be neglected. For problems primarily concerned with short-term transients, the passive viscous effects will be small in comparison to the more important unsteady and pressure terms. The basic differential equations considered here are the conservation equations of mass, momentum, and energy (in the absence of viscosity, body force, and heat transfer) and an additional equation, the equation of state. The equations governing the inviscid unsteady compressible flow for a two-dimensional axisymmetric cylindrical domain are:

Continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial r}(\rho u) + \frac{\rho u}{r} + \frac{\partial}{\partial z}(\rho w) = 0; \quad (1)$$

Momentum

$$\rho \frac{Du}{Dt} + \frac{\partial p}{\partial r} = 0; \quad (2)$$

$$\rho \frac{Dw}{Dt} + \frac{\partial p}{\partial z} = 0; \quad (3)$$

Energy

$$\rho \frac{Dh}{Dt} - \frac{Dp}{Dt} = 0; \quad (4)$$

Equation of State

$$h = h(p, \rho); \quad (5)$$

where the substantial derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial r} + w \frac{\partial}{\partial z}. \quad (6)$$

Combining Eqs. 4 and 5 results in

$$-\rho \left(\frac{\partial h}{\partial \rho} \right)_p \left[\rho \left(\frac{\partial h}{\partial p} \right)_\rho - 1 \right]^{-1} \frac{D\rho}{Dt} - \frac{Dp}{Dt} = 0. \quad (7)$$

The isentropic relation

$$\left(\frac{\partial p}{\partial \rho} \right)_s = a^2, \quad (8)$$

when combined with the isentropic energy relation

$$\frac{Ds}{Dt} = 0, \quad (9)$$

yields

$$a^2 \frac{D\rho}{Dt} - \frac{Dp}{Dt} = 0, \quad (10)$$

where s represents the entropy. A comparison of Eqs. 7 and 10 identifies the coefficient

$$a^2 = -\rho \left(\frac{\partial h}{\partial \rho} \right)_p \left[\rho \left(\frac{\partial h}{\partial p} \right)_\rho - 1 \right]^{-1}. \quad (11)$$

Combining Eq. 1 with Eq. 10 yields

$$\frac{Dp}{Dt} + \rho a^2 \left(\frac{\partial u}{\partial r} + \frac{u}{r} + \frac{\partial w}{\partial z} \right) = 0. \quad (12)$$

The set of three equations, Eqs. 2, 3, and 12, are used in place of the original Eqs. 1-5. Equations 2, 3, and 12 are a set of first-order, quasi-linear partial differential equations of hyperbolic type for which real characteristics exist. These equations are used to obtain a new set of equations, written in a characteristic form, that are especially suitable for numerical integration.

III. THE METHOD OF CHARACTERISTICS

The system of first-order quasi-linear partial differential equations as derived in Sec. II is rearranged in a form suitable for characteristic formulation:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + w \frac{\partial u}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial r} = 0, \quad (13)$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial r} + w \frac{\partial w}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial z} = 0, \quad (14)$$

and

$$\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} + \frac{1}{\rho a^2} \left(\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial r} + w \frac{\partial p}{\partial z} \right) + \frac{u}{r} = 0. \quad (15)$$

A. Characteristic Surfaces

Following the classical theory of the method of characteristics,^{11,12} the generalized surface S (often called "manifold") in the three-dimensional time-space (t - r - z) across which the normal derivatives of the solution are indeterminate is called a characteristic surface belonging to the system of partial differential equations. An equation of the form

$$f(t, r, z) = 0$$

defines such a manifold. To calculate the normal derivatives, a new set of independent variables φ , ξ , η is introduced such that ξ and η determine a point on S and φ is the normal coordinate. Using the chain rule, the system of governing equations can be written in terms of the new coordinates and rearranged viewing the normal derivatives $\partial u / \partial \varphi$, $\partial w / \partial \varphi$, and $\partial p / \partial \varphi$ as unknowns. Since these normal derivatives are indeterminate, their coefficient determinant (called the characteristic determinant) must vanish:

$$\begin{vmatrix} \frac{\partial \varphi}{\partial t} + u \frac{\partial \varphi}{\partial r} + w \frac{\partial \varphi}{\partial z} & 0 & \frac{\partial \varphi}{\partial r} \\ 0 & \frac{\partial \varphi}{\partial t} + u \frac{\partial \varphi}{\partial r} + w \frac{\partial \varphi}{\partial z} & \frac{\partial \varphi}{\partial z} \\ a^2 \frac{\partial \varphi}{\partial r} & a^2 \frac{\partial \varphi}{\partial z} & \frac{\partial \varphi}{\partial t} + u \frac{\partial \varphi}{\partial r} + w \frac{\partial \varphi}{\partial z} \end{vmatrix} = 0. \quad (16)$$

This condition yields three first-order partial differential equations for φ . If the vector $\vec{\lambda}$ represents a unit vector normal to the characteristic manifold S , its components in t , r , z coordinates λ_0 , λ_1 , λ_2 are the corresponding gradient components of φ , $\partial \varphi / \partial t$, $\partial \varphi / \partial r$, and $\partial \varphi / \partial z$.

Each vector $\vec{\lambda}$ through a point P on S satisfying Eq. 16 determines a characteristic direction, and, geometrically, it represents a generator through P of a pair of generalized cones with their apex at P. The characteristic determinant is a homogeneous polynomial of degree three in the derivatives (λ 's) and can be written as

$$H = (\lambda_0 + u\lambda_1 + w\lambda_2)(\lambda_0 + u\lambda_1 + w\lambda_2 + a\sqrt{\lambda_1^2 + \lambda_2^2})(\lambda_0 + u\lambda_1 + w\lambda_2 - a\sqrt{\lambda_1^2 + \lambda_2^2}) = 0. \quad (17)$$

B. Bicharacteristics

The three first-order partial differential equations resulting from the characteristic determinant,

$$\left. \begin{aligned} \lambda_0 + u\lambda_1 + w\lambda_2 &= 0, \\ \lambda_0 + u\lambda_1 + w\lambda_2 + a\sqrt{\lambda_1^2 + \lambda_2^2} &= 0, \\ \lambda_0 + u\lambda_1 + w\lambda_2 - a\sqrt{\lambda_1^2 + \lambda_2^2} &= 0, \end{aligned} \right\} \quad (18)$$

and

are of the form

$$\frac{\partial \varphi}{\partial t} + g\left(t, r, z, u, w, p, \frac{\partial \varphi}{\partial r}, \frac{\partial \varphi}{\partial z}\right) = 0. \quad (19)$$

These are equations of the Hamilton-Jacobi type which arise in analytical mechanics, and their solutions are the canonical equations that are also known as bicharacteristics or rays of the original partial differential equations. The three bicharacteristics corresponding to Eqs. 18 are

$$\left. \begin{aligned} \frac{dr}{dt} &= \frac{\partial g_1}{\partial \lambda_1} = u, \\ \frac{dz}{dt} &= \frac{\partial g_1}{\partial \lambda_2} = w; \end{aligned} \right\} \quad (20a)$$

$$\left. \begin{aligned} \frac{dr}{dt} &= \frac{\partial g_2}{\partial \lambda_1} = u + a \frac{\lambda_1}{\sqrt{\lambda_1^2 + \lambda_2^2}}, \\ \frac{dz}{dt} &= \frac{\partial g_2}{\partial \lambda_2} = w + a \frac{\lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2}}, \end{aligned} \right\} \quad (20b)$$

and

$$\left. \begin{aligned} \frac{dr}{dt} &= \frac{\partial g_3}{\partial \lambda_1} = u - a \frac{\lambda_1}{\sqrt{\lambda_1^2 + \lambda_2^2}}, \\ \frac{dz}{dt} &= \frac{\partial g_3}{\partial \lambda_2} = w - a \frac{\lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2}}. \end{aligned} \right\} \quad (20c)$$

Equation 20a represents the locus of the streamline at P, whereas Eqs. 20b and 20c define two rays through P. The rays defined by the latter two equations are generators of the upper and lower characteristic cones that also lie in characteristic surfaces having oppositely directed normal vectors $\vec{\lambda}$. Figure 1 describes the geometry of the cone, normal vector $\vec{\lambda}$, streamline, and bicharacteristics at point P that satisfy Eqs. 20. It suffices to consider only the lower cone. The line $\overline{OP}$ represents the streamline defined by the characteristic Eq. 20a. This equation, rewritten below, will be referred to as the streamline equation:

$$\left. \begin{aligned} dr &= u dt, \\ dz &= w dt. \end{aligned} \right\} \quad (21)$$

Fig. 1

Geometric Description of Streamline, Bicharacteristics, Normal Vector $\vec{\lambda}$, and Generalized Cone with Apex at P in t-r-z Time-Space. ANL Neg. No. 113-5379.

Let θ be the angle measured counterclockwise from positive r axis; then the two bicharacteristics defined by Eqs. 20b and 20c are seen to be 180° apart. The angle θ is used for a single-parameter representation of the bicharacteristics corresponding to all possible $\vec{\lambda}$. Equations 20b and 20c are now parametrically expressed by

$$\left. \begin{aligned} dr &= (u - a \cos \theta) dt, \\ dz &= (w - a \sin \theta) dt. \end{aligned} \right\} \quad (22)$$

C. Compatibility Equations

The compatibility equations written along bicharacteristics are obtained from the three original partial differential equations Eqs. 13-15, by adding and subtracting the quantities

$$\frac{\partial g}{\partial \lambda_1} \frac{\partial u}{\partial r} + \frac{\partial g}{\partial \lambda_2} \frac{\partial u}{\partial z}, \frac{\partial g}{\partial \lambda_1} \frac{\partial w}{\partial r} + \frac{\partial g}{\partial \lambda_2} \frac{\partial w}{\partial z}, \text{ and } \frac{\partial g}{\partial \lambda_1} \frac{\partial p}{\partial r} + \frac{\partial g}{\partial \lambda_2} \frac{\partial p}{\partial z},$$

respectively. The system of original differential equations is already in a characteristic form along a streamline; hence, any one may be taken as the compatibility equation along a streamline. It is convenient to choose Eq. 15 as the compatibility equation, which is written in a characteristic form as

$$\frac{dp}{dt} = -\rho a^2 \left(\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} + \frac{u}{r} \right), \quad (23)$$

where the operator d/dt denotes the differentiation along the streamline. Along bicharacteristics (defined by Eq. 22), the above procedure yields the following set of equations:

$$\frac{du}{dt} = -\frac{1}{\rho} \frac{\partial p}{\partial r} - a \cos \theta \frac{\partial u}{\partial r} - a \sin \theta \frac{\partial u}{\partial z},$$

$$\frac{dw}{dt} = -\frac{1}{\rho} \frac{\partial p}{\partial z} - a \cos \theta \frac{\partial w}{\partial r} - a \sin \theta \frac{\partial w}{\partial z},$$

and

$$\frac{dp}{dt} = -\rho a^2 \left(\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} + \frac{u}{r} \right) - a \cos \theta \frac{\partial p}{\partial r} - a \sin \theta \frac{\partial p}{\partial z},$$

where the operator d/dt in these equations and Eq. 23 denotes a differentiation along bicharacteristics. These three equations are then combined to yield a single-parameter infinite family of compatibility equations:

$$\begin{aligned} \cos \theta \frac{du}{dt} + \sin \theta \frac{dw}{dt} - \frac{1}{\rho a} \frac{dp}{dt} - a \left[\sin^2 \theta \frac{\partial u}{\partial r} - \sin \theta \cos \theta \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \right) \right. \\ \left. + \cos^2 \theta \frac{\partial w}{\partial z} + \frac{u}{r} \right] = 0. \end{aligned} \quad (24)$$

The compatibility equations, Eqs. 23 and 24, contain four partial derivatives $\partial u/\partial r$, $\partial u/\partial z$, $\partial w/\partial r$, $\partial w/\partial z$, in addition to three unknowns u , w , and p ; thus, a simultaneous solution requires seven equations: Eq. 23 and six equations from the parametric Eq. 24 for six different values of θ . A judicious selection of the parameter θ can simplify Eq. 24. Considering $\theta = 0, \pi/2, \pi$, and $3\pi/2$, the partial derivatives $\partial u/\partial z$ and $\partial w/\partial r$ are eliminated, and now four equations

for four different values of θ plus Eq. 23 are sufficient for a simultaneous determination of the five unknowns, u , w , p , $\partial u/\partial r$, and $\partial w/\partial z$. The compatibility Eq. 24 is identical to the one derived in the two-dimensional characteristic formulation¹³ following von Mises.¹⁴ A similar equation for a two-dimensional Cartesian problem also has been derived by Butler.⁵ The angle θ varies along bicharacteristics, depending on local flow conditions. The change in the angle from the vertex P of the cone (refer to Fig. 1) as a function of velocity gradients accurate to the second order of time step Δt is

$$\theta = \alpha + \left(-\sin \alpha \cos \alpha \frac{\partial u}{\partial r} + \cos^2 \alpha \frac{\partial u}{\partial z} - \sin^2 \alpha \frac{\partial w}{\partial r} + \sin \alpha \cos \alpha \frac{\partial w}{\partial z} \right) \Delta t + O(\Delta t)^2, \quad (25)$$

where α is the angle at P , θ is the angle away from P on the same bicharacteristic, and the partial derivatives are to be evaluated in the time plane where the solution is known (initial plane). A detailed derivation of Eq. 25 is given in Appendix A.

D. Boundary- and Axial-point Conditions

Conditions to be satisfied at boundaries and along the axis of symmetry are the ones that are consistent with inviscid flows. At solid boundaries, the normal velocity component must vanish, whereas the tangential component cannot be specified. At axial points, the radial velocity component is zero and the term u/r is replaced by $\partial u/\partial r$. For problems involving regular geometries, no other special conditions exist. Boundary-point calculations can be carried out directly using the appropriate boundary conditions and compatibility equations.

IV. NUMERICAL PROCEDURE

A. System of Integral Equations and Their Finite-difference Approximation

The compatibility Eqs. 23 and 24 are integrated along the streamline (defined by Eq. 21) and bicharacteristics (defined by Eq. 22), respectively:

$$\int dp = -\rho a^2 \int \left(\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} + \frac{u}{r} \right) dt; \quad (26a)$$

$$\int dr = \int u dt; \quad \int dz = \int w dt; \quad (26b)$$

$$\begin{aligned} \int \cos \theta du + \int \sin \theta dw - \frac{1}{\rho a} \int dp - a \int \left[\sin^2 \theta \frac{\partial u}{\partial r} - \sin \theta \cos \theta \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \right) \right. \\ \left. + \cos^2 \theta \frac{\partial w}{\partial z} + \frac{u}{r} \right] dt = 0; \end{aligned} \quad (27a)$$

$$\int dr = \int (u - a \cos \theta) dt; \quad \int dz = \int (w - a \sin \theta) dt, \quad (27b)$$

where θ in Eq. 27 assumes the values $0, \pi/2, \pi$, and $3\pi/2$.

The bicharacteristic angle θ varies linearly with two spatial derivatives $\partial u/\partial z$ and $\partial w/\partial r$ (for $\alpha = 0, \pi/2, \pi$, and $3\pi/2$) and time t . An accurate determination of the derivatives at points falling between grid points in a constant time plane of a finite-difference system is a difficult task and hence the angle θ is assumed constant along bicharacteristics for a small time step Δt . Equations 26 and 27 represent a system of integral equations and are numerically approximated by a trapezoidal formula. With the assumption of constant angle θ , the system of equations, for an ordinary point, includes five unknowns, $u, w, p, \partial u/\partial r$, and $\partial w/\partial z$. The nonlinear nature of the streamline Eq. 26b and the bicharacteristics Eq. 27b necessitates an iteration procedure. The five compatibility equations, Eqs. 26a and 27a, are first written in a finite-difference form and combined linearly to yield an explicit solution for the five unknowns.

The solution applying to an interior grid point is

$$\left. \begin{aligned} u &= \frac{1}{2} \left[u_1 + u_3 - \sigma_1 + \sigma_3 + \frac{1}{2} a \Delta t (s_1 - s_3) \right]; \\ w &= \frac{1}{2} \left[w_2 + w_4 - \sigma_2 + \sigma_4 + \frac{1}{2} a \Delta t (s_2 - s_4) \right]; \\ \sigma &= \frac{1}{2} \left[-u_1 + u_3 - w_2 + w_4 + \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 - 2\sigma_5 \right. \\ &\quad \left. - \frac{1}{2} a \Delta t \left(s_1 + s_2 + s_3 + s_4 - 2s_5 + 2 \frac{u}{r} \right) \right], \end{aligned} \right\} \quad (28)$$

where, as shown in Fig. 2, subscripts 1, 2, 3, and 4 refer to intercepts with plane $t = t$ of the bicharacteristics corresponding to $\theta = 0, \pi/2, \pi$, and $3\pi/2$; the subscript 5 refers to the intercept of the streamline with this plane; and

$$\sigma = \frac{p}{\rho a}, \quad (29)$$

$$\left. \begin{aligned} s_1 &= \left(\frac{\partial w}{\partial z} + \frac{u}{r} \right)_1, \\ s_2 &= \left(\frac{\partial u}{\partial r} + \frac{u}{r} \right)_2, \\ s_3 &= \left(\frac{\partial w}{\partial z} + \frac{u}{r} \right)_3, \\ s_4 &= \left(\frac{\partial u}{\partial r} + \frac{u}{r} \right)_4, \\ \text{and} \\ s_5 &= \left(\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} + \frac{u}{r} \right)_5. \end{aligned} \right\} \quad (30)$$

The streamline Eq. 26b and bicharacteristics Eq. 27b in a finite-difference form are

$$\left. \begin{aligned} r - r_5 &= \frac{1}{2}(u + u_5)\Delta t, \\ z - z_5 &= \frac{1}{2}(w + w_5)\Delta t; \end{aligned} \right\} \quad (31)$$

$$\left. \begin{aligned} r - r_1 &= \left[\frac{1}{2}(u + u_1) - a \right] \Delta t, \\ z - z_1 &= \frac{1}{2}(w + w_1)\Delta t; \end{aligned} \right\} \quad (32a)$$

$$\left. \begin{aligned} r - r_2 &= \frac{1}{2}(u + u_2)\Delta t, \\ z - z_2 &= \left[\frac{1}{2}(w + w_2) - a \right] \Delta t; \end{aligned} \right\} \quad (32b)$$

$$\left. \begin{aligned} r - r_3 &= \left[\frac{1}{2}(u + u_3) + a \right] \Delta t, \\ z - z_3 &= \frac{1}{2}(w + w_3)\Delta t; \end{aligned} \right\} \quad (32c)$$

and

$$\left. \begin{aligned} r - r_4 &= \frac{1}{2}(u + u_4)\Delta t, \\ z - z_4 &= \left[\frac{1}{2}(w + w_4) + a \right] \Delta t. \end{aligned} \right\} \quad (32d)$$

Fig. 2

Streamline and Bicharacteristics through a Grid Point P in $t = t + \Delta t$ Plane and Their Intercepts with $t = t$ Plane. ANL Neg. No. 113-5380.

B. Rectangular Spatial Grids

In the stepwise integration procedure, it is assumed that a flow condition (usually steady state) is given at time t , and it is required to follow the transients of a pressure disturbance. In Fig. 2, a spatial grid system with equal spacings (both in radial and axial coordinates) is described. Point P represents a typical interior grid point in plane at $t = t + \Delta t$, where one streamline and four bicharacteristics are directed toward P from the initial time plane $t = t$.

A typical step in the numerical calculation consists of extending the solution for all grid points to an advanced plane $t = t + \Delta t$ from solutions in the initial plane $t = t$. Repetition of the step calculation yields a complete numerical solution, if the finite-difference scheme is numerically stable. The time step Δt must be chosen such that the initial points 1, 2, 3, 4, and 5 all lie inside the rectangle BCED in Fig. 2. In this manner, the Courant-Friedrichs-Lewy¹⁵ criterion for convergence and stability is satisfied, since the domain of dependence of the finite-difference system contains the domain of dependence of the differential system.

C. Geometry of Physical Domain and Boundary Conditions

The geometry of the physical domain considered here is an axisymmetric cylinder of finite length. Also added as part of the domain are smaller-diameter pipe sections upstream and downstream in order to attain a smooth transition from a two-dimensional to a one-dimensional flow and vice versa. The purpose of this study is to perform a two-dimensional analysis in a section connected to a network of one-dimensional sections. Hence, the small-diameter pipe extensions upstream and downstream are physically a part of the one-dimensional domain and are included in the two-dimensional domain for purposes of analysis. (These extended regions are divided in the same spacings, both axially and radially, as the central 2-D region.)

Figure 3 describes the geometrical configuration of the two-dimensional domain and its interfaces with the connected one-dimensional system. The boundary conditions on the solid surfaces (rigid boundary) are those conditions

that are consistent with the inviscid flows, i.e., the condition of vanishing normal velocity component. This condition allows slip velocities along the surfaces. On the nonboundary axial point, $r = 0$, the condition to satisfy is the zero radial velocity, $u = 0$. At the interfaces upstream and downstream, a simplified condition is used. The radial velocity is assumed to vanish and the axial velocity is taken as uniform across the interface.

Fig. 3

Geometry of Two-dimensional Domain under Consideration.
ANL Neg. No. 113-5381.

A linear combination of compatibility equations along streamline and appropriate bicharacteristics together with boundary conditions yields the following solution at the boundaries:

At the axial point ($r = 0$):

$$\left. \begin{aligned} u &= 0; \\ w &= \frac{1}{2}[w_2 + w_4 - \sigma_2 + \sigma_4 + \frac{1}{2}a\Delta t(s_2 - s_4)]; \\ \sigma &= 2\{-u_1 - \frac{1}{4}(w_2 - w_4) + \sigma_1 + \frac{1}{4}(\sigma_2 + \sigma_4) - \sigma_5 - \frac{1}{2}a\Delta t[s_1 + \frac{1}{4}(s_2 + s_4) - s_5]\}. \end{aligned} \right\} \quad (33)$$

On the periphery ($r = R_1$, $r = R$, $r = R_2$):

$$\left. \begin{aligned} u &= 0; \\ w &= \frac{1}{2}[w_2 + w_4 - \sigma_2 + \sigma_4 + \frac{1}{2}a\Delta t(s_2 - s_4)]; \\ \sigma &= u_3 - \frac{1}{2}(w_2 - w_4) + \sigma_3 + \frac{1}{2}(\sigma_2 + \sigma_4) - \sigma_5 - \frac{1}{2}a\Delta t[s_3 + \frac{1}{2}(s_2 + s_4) - s_5]. \end{aligned} \right\} \quad (34)$$

On the boundary ($z = z_1$; $R_1 < r < R$):

$$\left. \begin{aligned} u &= \frac{1}{2}\left[u_1 + u_3 - \sigma_1 + \sigma_3 + \frac{1}{2}a\Delta t(s_1 - s_3)\right]; \\ w &= 0; \\ \sigma &= -\frac{1}{2}(u_1 - u_3) - w_2 + \sigma_2 + \frac{1}{2}(\sigma_1 + \sigma_3) - \sigma_5 - \frac{1}{2}a\Delta t\left[s_2 + \frac{1}{2}(s_1 + s_3) - s_5 + \frac{u}{r}\right]. \end{aligned} \right\} \quad (35)$$

On the boundary ($z = z_1 + L$; $R_2 < r < R$):

$$\left. \begin{aligned} u &= \frac{1}{2} \left[u_1 + u_3 - \sigma_1 + \sigma_3 + \frac{1}{2} a \Delta t (s_1 - s_3) \right]; \\ w &= 0; \\ \sigma &= -\frac{1}{2} (u_1 - u_3) + w_4 + \frac{1}{2} (\sigma_1 + \sigma_3) + \sigma_4 - \sigma_5 - \frac{1}{2} a \Delta t \left[\frac{1}{2} (s_1 + s_3) + s_4 - s_5 + \frac{u}{r} \right]. \end{aligned} \right\} \quad (36)$$

At the upstream interface ($z = 0$; $0 < r < R_1$):

$$\left. \begin{aligned} u &= 0; \\ w &= \frac{1}{2} \left(w_2 + w_A - \sigma_2 + \sigma_A + \frac{1}{2} a \Delta t s_2 - \frac{f_A w_A |w_A| \Delta t}{4R_A} \right); \\ \sigma &= \frac{1}{2} \left(-w_2 + w_A + \sigma_2 + \sigma_A - \frac{1}{2} a \Delta t s_2 - \frac{f_A w_A |w_A| \Delta t}{4R_A} \right). \end{aligned} \right\} \quad (37)$$

At the downstream interface ($z = z_1 + L + z_2$; $0 < r < R_2$):

$$\left. \begin{aligned} u &= 0; \\ w &= \frac{1}{2} \left(w_4 + w_B + \sigma_4 - \sigma_B - \frac{1}{2} a \Delta t s_4 - \frac{f_B w_B |w_B| \Delta t}{4R_B} \right); \\ \sigma &= \frac{1}{2} \left(w_4 - w_B + \sigma_4 - \sigma_B - \frac{1}{2} a \Delta t s_4 - \frac{f_B w_B |w_B| \Delta t}{4R_B} \right). \end{aligned} \right\} \quad (38)$$

At the corner point (point 1 in Fig. 3):

$$\left. \begin{aligned} u &= 0; \\ w &= 0; \\ \sigma &= -w_2 + u_3 + \sigma_2 + \sigma_3 - \sigma_5 - \frac{1}{2} a \Delta t (s_2 + s_3 - s_5). \end{aligned} \right\} \quad (39)$$

At the corner point (point 2 in Fig. 3):

$$\left. \begin{aligned} u &= 0; \\ w &= 0; \\ \sigma &= u_3 + w_4 + \sigma_3 + \sigma_4 - \sigma_5 - \frac{1}{2} a \Delta t (s_3 + s_4 - s_5). \end{aligned} \right\} \quad (40)$$

Points 3 and 4 in Fig. 3 are treated as interior points using Eqs. 28. The subscripts A and B in Eqs. 37 and 38 refer to the intersections in the one-dimensional domain of the characteristics with the constant time line $t = t$ at the upstream and downstream, respectively, where the characteristics are pointed toward the interfaces.

D. Stepwise Integration Procedure

Initially ($t = 0$), some flow condition (usually a steady-state flow) is given at all grid points, and flow transients following any given disturbance are calculated in a stepwise procedure. A typical one-time-step calculation consists of extending the solution from one constant-time plane ($t = t$) to the next plane ($t = t + \Delta t$), using the approximate finite-difference solutions as given in Eqs. 28 for the interior points and Eqs. 33-40 for the axial and boundary points. The solutions are obtained by simultaneously solving the compatibility equations with the appropriate streamline or bicharacteristic equations. Since the streamline and the bicharacteristic Eqs. 31 and 32 contain the unknowns, an iterative scheme is used. The coordinates of the intersections $r_1, r_2, \dots, z_1, z_2, \dots$, are first guessed in the beginning and are used to obtain initial solutions. The initial solutions are then used to improve the initial guesses of the coordinates, and the procedure is repeated until a satisfactory convergence is attained. For mildly nonlinear problems, such as fluid-hammer phenomena involving moderate pressure levels, the convergence is fast and the iteration is often not needed.

V. COMPUTER PROGRAM

The computer program NATRAN-II is written in FORTRAN IV for the IBM S/370-195 computer. Using the ANL-developed one-dimensional system code NATRANSIENT¹⁰ as a subroutine (PTRAN), NATRAN-II as an integrated code is capable of analyzing a two-dimensional axisymmetric domain connected to a system of one-dimensional domains. At each time-step calculation, the subprogram PTRAN is called twice by the main program. The first time, the main program provides information from 2-D domain analysis for the interface-point calculation (boundary) of the 1-D system calculation in PTRAN. The second time, PTRAN supplies inputs necessary for 2-D side boundary-point calculation of the interface. Information is transferred between the 1-D system and 2-D domain calculations by means of subprogram arguments. Other 1-D system-analysis codes can be used in place of NATRANSIENT, if proper provision is made for data transfer for the boundary-point calculation on both sides of the interface.

In the integral code NATRAN-II, the user can input a prescribed pressure-time history as the source pressure and specify its location in the 2-D domain. At the user's option, either the sodium-water reaction (as described in NATRANSIENT) or a pressure-time history can be prescribed at some location in the 1-D domain. NATRAN-II is capable of treating problems involving nonzero initial velocities by calculating the initial steady-state flow conditions consistent with system pressure and velocity in the 1-D system description. This is accomplished by performing the transient-flow computations from an arbitrarily assumed initial condition until the solution converges to a steady-state condition. The program also can be used for solving a completely enclosed flow problem in a cylindrical container bounded by rigid walls on both ends. When this option is chosen, the one-dimensional analysis (subroutine PTRAN) is bypassed. A third option allows the program to be used as a purely one-dimensional system code, bypassing the two-dimensional calculation, in which case the result is identical to that of the NATRANSIENT code. The FORTRAN program listing of NATRAN-II is given in Appendix B.

A. Description of Input Requirements

The following input data cards are required to run the program; they are described in the order they appear in the input deck. Since a few additional cards are needed for the 1-D domain description than the original NATRANSIENT code document, complete input cards are described here, including those required for the 1-D domain calculation.

<u>Card No.</u>	<u>FORTTRAN Name</u>	<u>Format</u>	<u>Description</u>
1		13I5	
	IP1		If IP1 = 1, source pressure-time function is input in 2-D domain. If IP1 = 2, pressure source is absent in 2-D domain.
	IP2		If IP2 = 0, results of all grid points are printed. If IP2 > 0, then results of IP2 selected points only are printed; e.g., if IP2 = 2, two selected points are printed. The grid to be printed is specified on card 23.
	IP3		If IP3 = 1, maximum grid-point values of radial velocities that have existed up to the present time are printed. If IP3 = 2, maximum axial velocities are printed. If IP3 = 3, maximum pressures are printed. If IP3 = 4, maximum values of radial and axial velocities as well as pressures are printed. If IP3 = 0, none of above is printed.
	IP4		If IP4 = 0, the spatial derivatives in the compatibility equations are evaluated by averaging between two time planes. IP4 = 1, the spatial derivatives are evaluated in new time plane. (This was used for numerical experimentation.) Always set IP4 = 0.
	IP5		If IP5 = 1, the source pressure is applied at four grid points: (I = II, J = JJ), (I = II + 1, J = JJ), (I = II, J = JJ + 1), and (I = II + 1, J = JJ + 1). If IP5 = 2, a one-dimensional source pressure is described at axial location J = JJ. II and JJ are given in card 2.
	IP6		If IP6 = 1, the characteristic calculation is replaced by the symmetric condition on the axis ($r = 0$): $\partial p / \partial r = 0$, $\partial w / \partial r = 0$. (This is an alternative way of specifying conditions on the axis.) IP6 = 0 is preferred.
	IP7		If IP7 = 0, the program solves a completely enclosed flow problem in a cylinder. When this option is chosen, the 1-D system analysis is bypassed. If IP7 = 1, the 2-D + 1-D system analysis is performed.
	IP8		If IP8 = 1, the initial steady-state solution is obtained before the pressure-transient analysis is performed
	IP9		If IP9 = 1, intermediate values of the initial steady-state solution are printed (if option IP8 = 1 is used). The frequency of this print-out is given in card 5.
	IP10		If IP10 = 1, a purely 1-D system calculation is carried out. In this option, the 2-D calculation is bypassed and the calculation is identical to that of the NATRANSIENT code. (However, the entire input cards including those for 2-D calculation have to be given.)

<u>Card No.</u>	<u>FORTTRAN Name</u>	<u>Format</u>	<u>Description</u>
2	IP11	8I5	If IP11 = 1, all results for the extended regions upstream and downstream of the 2-D domain are printed.
	IP12		If IP12 = 1, calculation terminates when a negative pressure is obtained.
	IP13		If IP13 = 1, results for selected points (used with option IP2 > 0) are written on unit-10 disk file for later graphical display.
	NR		The number of grid points in r coordinate in the 2-D region; i.e., $RR = (NR - 1)\Delta r$.
	NZ		The number of grid points in z coordinate in the 2-D region; i.e., $ZLEN = (NZ - 1)\Delta z$.
	II		r: coordinate of source point.
	JJ		z: coordinate of source point. (The numbering of the grid begins with J = 1 at the upstream interface.)
	NF		Frequency of printout for the 2-D transient calculation. If NF = 1, every time-step calculation is printed. If NF = 2, once every two-step calculations is printed, etc.
	NIT		Outer-iteration number for nonlinear characteristics. If NIT = 0, there is no iteration. If NIT = 1, there is one iteration for every time-step calculation, etc. Set NIT = 0 for normal runs.
	MIT		Inner-iteration number for calculating interpolated values of unknowns and their spatial derivatives in the current time plane. If MIT = 0, there is no iteration. If MIT = 1, there is one iteration for every time-step calculation, etc. Set MIT = 0 for normal runs.
3	NNH	6E12.5	Number of sets of data for pressure-time history input. The pressure and time data are given in cards 7 and 8.
	DT		Time step, in seconds.
	TI		Initial time the calculation starts, in seconds.
	TF		Final time the calculation terminates, in seconds.
	PR		Initial pressure in 2-D domain, in psi.
	RHO		Density of liquid in 2-D domain, in lb_m/ft^3 .
	C		Sonic speed in 2-D domain, in ft/sec.
4	RR	2E12.5	Radius of 2-D region, in feet.
	ZLEN		Length of 2-D region, in feet, excluding extended regions upstream and downstream.

<u>Card No.</u>	<u>FORTTRAN Name</u>	<u>Format</u>	<u>Description</u>
5		3I5	If IP7 $\neq$ 1, this card is skipped.
	NFS		Frequency of printout for the initial steady-state solution. If NFS = 1, every intermediate-step calculation is printed. If NFS = 2, once every two-step calculations is printed, etc.
	IA		Pipe-section number belonging to the 1-D system connected to the upstream end of the 2-D domain. IA must be the pipe section whose downstream end is connected to the upstream end of the 2-D domain.
	IB		Pipe-section number belonging to the 1-D system connected to the downstream end of the 2-D domain. IB must be the pipe section whose upstream end is connected to the downstream end of the 2-D domain.
6		5E12.5	If IP7 $\neq$ 1, this card is skipped.
	RR1		Radius of the extended region upstream end of the 2-D domain, in feet.
	ZL1		Length of the extended region upstream end of the 2-D domain, in feet.
	RR2		Radius of the extended region downstream end of the 2-D domain, in feet.
	ZL2		Length of the extended region downstream end of the 2-D domain, in feet.
	TFS		Final time the steady-state calculation terminates, in seconds. This parameter is meaningful only when option IP8 = 1 is chosen.
			If IP1 $\neq$ 1, skip the following two cards. The following two sets of cards contain NNH entries each, each card bearing up to six numbers. Argument I ranges from one to NNH.
7 (Multiple Cards)		6E12.5	
	TIME(I)		Times at which pressures (PTM(I)) are specified, in seconds.
8 (Multiple Cards)		6E12.5	
	PTM(I)		Pressures at the corresponding times TIME(I), in psi.
9		8I5	
	LEGNO		Total number of pipe branches.
	JUNNO		Total number of junctions.
	NOT		Number of surge tanks and/or reservoirs.
	NOPRIN		Frequency of printout. If NOPRIN = 1, every step result is printed. If NOPRIN = 2, the result of every second step is printed, etc.

<u>Card No.</u>	<u>FORTTRAN Name</u>	<u>Format</u>	<u>Description</u>
	KS		Parameter specifying details of printout. Results of every KStH node in every branch are printed.
	JJ*		Branch number in which leak site or the pressure source is located.
	IC		If IC = 1, initial velocities and pressures in all branches are input individually. If IC $\neq$ 1, initial velocities and pressures are set to UR and PR, respectively.
	NP		If NP = 1, the friction factor f is set to 0. The result then corresponds to an inviscid analysis. Set NP = 0 for normal run.
10		6E12.5	
	DT		Time step, in seconds, for the 1-D calculation. (This must be the same as that given in card 3.)
	UR		Initial velocity. If IC $\neq$ 1, initial velocities in all branches are set to UR.
	PR		Initial pressure. If IC $\neq$ 1, initial pressures in all branches are set to PR.
	PCR		Critical pressure at or above which rupture disks break, in psi.
	TR		Temperature of the liquid sodium in the network, in °F.
	XI		Initial time at which calculation starts, in seconds.
11		2E12.5	
	XF		Final time at which calculation terminates, in seconds.
	PARM		Interpolation parameter ensuring numerical stability (must be greater than 1). For a normal run, set PARM = 1.1.
12		7I5	
	INDTR		An indicator signaling results of how many branches are to be printed. If INDTR = 2, results of only two branches are printed; if INDTR = 3, results of only three branches are printed; etc. If INDTR = 0, all results are printed. The branches of which results are to be printed are given in card 13.
	NOTT		Number of rupture disks.
	IP1		If IP1 = 0, source pressure by sodium-water reaction located in branch JJ belonging to 1-D domain. If IP1 = 1, pressure-time history is prescribed. If IP1 = 2, pressure source is absent in 1-D domain.

*The input cards 9-22 describe the requirements for the 1-D calculation. The variable names in these cards occupy different common blocks from those used in 2-D calculations. Hence, although the same FORTRAN names occasionally appear in these cards as in the input cards for 2-D calculation, they should be regarded as different variables.

<u>Card No.</u>	<u>FORTTRAN Name</u>	<u>Format</u>	<u>Description</u>
	NNK		Number of sets of data for pressure-time history input. This data is needed when prescribed pressure-time history is located in 1-D domain (option IP1 = 1).
	JK		Branch number of the 2-D domain that is excluded from the 1-D system calculation. This data is needed when pressure source is located in 1-D domain. If IP2 = 0, source pressure is assigned at all nodes of branch JJ. If IP2 = 1, source pressure assigned at two boundary nodes of first node side of branch JJ. If IP2 = 2, source pressure is assigned at two boundary nodes of last node side of branch JJ. If IP2 = 3, source pressure is given at node 1 only of branch JJ. If IP2 = 4, source pressure is given at last node only of branch JJ. Options, IP2 = 3 and 4, should be used only for special purposes; i.e., disturbances do not transmit through the junctions.
	IP3		If IP3 = 1, the density and the sonic speed are input. They are given in card 18.
13 (Multiple)		16I5	If INDTR = 0, skip this card.
	INT(NQ)		The branch numbers of which results are to be printed. NQ ranges from 1 to INDTR. Provide as many cards as necessary.
14 (Multiple Cards)		2I3, 6E12.5, /, 2E12.5	Total LEGNO sets of cards. Each set consists of two cards containing the following data in the order cited:
	NI(L)		Junction number connected to first node of Lth branch.
	NN(L)		Junction number connected to last node of Lth branch.
	A(L)		Net cross-sectional area of Lth branch, in square inches.
	LENGTH(L)		Length of Lth branch, in feet.
	RF(L)		Roughness of Lth branch, in inches.
	H(L)		Thickness of Lth branch, in inches.
	ES(L)		Young's modulus of Lth pipe material, in psi.
	D(L)		Inside diameter of Lth branch, in inches.
	PO(L)		Initial pressure in Lth branch, in psi. This is used with option IC = 1.
	UO(L)		Initial velocity in Lth branch, in ft/sec. This is used with option IC = 1.
15 (Multiple Cards)		40I2	Argument L in above 10 arrays ranges from 1 to LEGNO. The branches are numbered consecutively from 1 to LEGNO in the order of input.
	NJ(J)		Number indicating junction type for junction J, where J ranges from one to JUNNO. Junction-type specifications are as follows:

<u>Card No.</u>	<u>FORTTRAN Name</u>	<u>Format</u>	<u>Description</u>
	NJ(J) (Contd.)		1: Sudden expansion or contraction. 3: Tee (joining three to six branches). 7: Closed end. 8: Surge tank or reservoir. 9: Far end. 6: Dummy junction. 5: Acoustic-impedance discontinuity. 10: Rupture disk.
16		4(E12.5,I5)	NOT sets of entries. If NOT = 0, skip this card. Each set consists of two numbers and contains the following data in the order cited:
	PT(I)		Gas pressure of surge tank or reservoir I, in psi.
	IT(I)		Junction number to which surge tank I is attached. I ranges from one to NOT.
17		4(E12.5,I5)	NOTT sets of entries. If NOTT = 0, skip this card. Each set consists of two numbers and contains the following data in the order cited:
	PT(I)		Pressure of Ith reservoir to which rupture disks open, in psi.
	IT(I)		Junction number to which Ith rupture disk is attached. I ranges from one to NOTT.
18		2E12.5	If IP3 $\neq$ 1, skip this card.
	RHO		Density of liquid in 1-D domain, in lb _m /ft ³ .
	CS		Sonic speed in 1-D domain, in ft/sec.
19		5E12.5	If IP1 $\neq$ 0, skip this card. (The following two cards pertain to the description of sodium-water reaction.)
	PI		System pressure used in source pressure calculation, in psi.
	WZD		Instantaneous leak rate, in lb _m /sec.
	PCT		Ratio of initial-to-final leak areas used for time-dependent rupture-rate study. PCT = 1.0 represents an instantaneous leak.
	TAU		Time constant appearing in the time-dependent rupture-rate model, in seconds (Eq. 36 in Ref. 6).
	EXP		Exponent appearing in time-dependent rupture-rate model (Eq. 36 in Ref. 6).
20		2E12.5	If IP1 $\neq$ 0, skip this card.
	PCRT		Critical pressure, in psi.
	PRES		Reservoir pressure, in psi.
			If IP1 $\neq$ 1, skip the following two cards. The following two cards describe the input pressure-time history in the 1-D domain. Argument I ranges from 1 to NNK. Provide as many cards as necessary.

<u>Card No.</u>	<u>FORTTRAN Name</u>	<u>Format</u>	<u>Description</u>
21 (Multiple Cards)	TIME(I)	6E12.5	Times at which pressures (PTM(I)) are specified, in seconds.
22 (Multiple Cards)	PTM(I)	6E12.5	Pressures at the corresponding times TIME(I), in psi.
23 (Multiple Cards)	IK(M)	7(2I5)	If IP2 = 0, skip this card. When IP2 ≥ 1, this card contains IP2 set of integers, each set containing entries as cited in the following. Argument M ranges from 1 to IP2. Provide as many cards as necessary.
	JK(M)		Grid numbers in z coordinate for which the result is to be printed.

B. Limitations of NATRAN-II

The limitations of NATRAN-II resulting from the dimension statements are:

<u>The Number of</u>	<u>Cannot Exceed</u>
1. Grid points in r-direction	30
2. Grid points in z-direction	60
3. One-dimensional branches	50
4. Nodes in any one-dimensional branch	120
5. Junctions	50
6. Surge tanks	10
7. Branches joined at any one tee junction	6

Further limitations resulting from the simplifying assumptions in the analysis are:

8. In the two-dimensional analysis, the viscosity effect is neglected; hence, only an inviscid treatment is possible.

9. A sufficiently large length-to-diameter ratio has to be allowed for both extended regions upstream and downstream of the 2-D domain, since the matching of the boundary conditions at the interfaces neglects any two-dimensionality close to the interface.

10. The program does not provide shock-wave analysis, hence is limited to shock-free problems.

VI. NUMERICAL CALCULATION

A. Description of Sample Problem

A sample problem as shown in Fig. 4 is considered for numerical calculation by both NATRAN-II and NATRANSIENT. Geometrically, it consists of a two-dimensional domain (0.625-ft radius, 1.25-ft length) with extended upstream (0.3125-ft radius, 6-radii length) and downstream regions (0.3125-ft radius, 4-radii length) connected to a one-dimensional system consisting of two pipe sections of 0.3125-ft radius and 2.00-ft length, one at the upstream interface and the other at the downstream interface. Both the upstream and downstream ends of the entire problem are considered to be the far-end* junction condition at which no reflected waves are allowed. A one-dimensional pressure-time history (as given in Fig. 5) is prescribed at the entire plane (represented by a dotted vertical line in Fig. 4 located 0.3125 ft downstream from the upstream end of the two-dimensional domain). The initial conditions considered are uniformly zero velocity and zero absolute pressure.

Fig. 4

Geometric Configuration of Sample Problem.
ANL Neg. No. 113-5382.

Fig. 5

Pressure vs Time at Selected Points Obtained
by Two-dimensional Code NATRAN-II. ANL
Neg. No. 113-5383.

B. Results and Discussion

The calculational results obtained by NATRAN-II are plotted in Fig. 5 for selected points (points 1, 2, 3, and 4 in Fig. 4) for the two-peak source-pressure pulse as indicated. This problem (diameter ratio of 2) is also

*See Ref. 6 for detailed description of the far-end junction condition.

calculated by the one-dimensional code NATRANSIENT, and the results are plotted in Fig. 6. Figure 7 compares these two results for points 1 and 2. The comparison of the two results indicates a fair agreement for all points considered, except that the one-dimensional analysis predicts slightly higher local peak pressures than the two-dimensional result.

Fig. 6

Pressure vs Time at Selected Points Obtained by One-dimensional Code NATRANSIENT; $\Delta t = 0.015$ msec. ANL Neg. No. 113-5384.

Fig. 7

Comparison of Results of NATRAN-II (2-D Code) and NATRANSIENT (1-D Code) for Sample Problem with Diameter Ratio of 2. ANL Neg. No. 113-5385.

Sodium is used as the working fluid with a constant sonic speed of 5000 ft/sec. In the 2-D calculation, the domain is a 9×17 grid network with equal spacings in each of the r and z coordinates, excluding the extended upstream and downstream regions. A time step $\Delta t = 0.007$ msec (for the grid spacing of 0.0782 ft) is determined by a numerical experimentation, and is found to be the largest Δt permissible to satisfy the criterion for numerical stability and convergence. In the one-dimensional calculation, a larger time step ($\Delta t = 0.015$ msec) is taken for the same axial spacing ($\Delta z = 0.0782$ ft). This indicates that a more stringent criterion for numerical stability is imposed on the two-dimensional calculation and hence requires a smaller time step for the same grid spacing used. As a result, for a fixed stable Δt , the two-dimensional calculation can be subject to a greater interpolation error than that occurring in the one-dimensional calculation.

To examine this interpolation error, an additional one-dimensional calculation is made with a smaller time step ($\Delta t = 0.007$ msec) for the same $\Delta z = 0.0782$ ft. The results (see Fig. 8) are essentially identical to those depicted in Fig. 6. The interpolation error depends on the pulse shape involved; however, the error can be minimized by taking a small grip spacing for problems involving a rapid variation in the solution.

Fig. 8

Pressure vs Time at Selected Points Obtained by One-dimensional Code NATRANSIENT; $\Delta t = 0.007$ msec. ANL Neg. No. 113-5386.

Additional calculations are made for the sample problem with an increased diameter (2.1875 ft) of the two-dimensional domain (diameter ratio = 7). The results plotted in Figs. 9 and 10 are those obtained from NATRAN-II and NATRANSIENT, respectively. These plots indicate that higher pressures were predicted at points close to the region of expansion (points 1 and 3) by the two-dimensional analysis than the one-dimensional analysis. A pronounced two-dimensionality is evident near the expansion region during the early time, in that the pressure on the axis (point 1) is much higher than the pressure obtained on the periphery (point 4). There is no significant difference between the respective pressures calculated at point 2 (a point located downstream of the contraction region).

Fig. 9

Pressure vs Time at Selected Points Obtained by Two-dimensional Code NATRAN-II with Diameter Ratio of 7. ANL Neg. No. 113-5387.

Fig. 10

Pressure vs Time at Selected Points Obtained by One-dimensional Code NATRANSIENT with Diameter Ratio of 7. ANL Neg. No. 113-5388.

The foregoing numerical results indicate that a one-dimensional approximation can result in an overestimate of pressures in problems involving small changes in cross-sectional area, whereas in problems with large area changes, it can yield a significant underestimate of pressures. Consequently, a multi-dimensional treatment is necessary for problems requiring a close assessment of local pressures. Nevertheless, the one-dimensional result represents a good conservative approximation for problems with moderate changes in the cross-sectional area.

The input data used for the sample problem in the two-dimensional calculation with the diameter ratio of 2 are listed in Appendix C. A partial listing of the output of the same problem is given in Appendix D. Furthermore, a summary description of the sodium-water reaction that determines the source pressure in the one-dimensional region is presented in Appendix E.

C. Further Remarks

It has been found, through the sample-problem calculation, that the maximum time step is somewhat smaller than the one predicted by the CFL criterion. The numerical experimentation indicated that the maximum time step allowed can be expressed as

$$\Delta t_{\max} < 0.45 \left[\frac{\text{Min}(\Delta r, \Delta z)}{c + \text{Max}(|u|_{\max}, |w|_{\max})} \right].$$

This equation can be used in making a preliminary estimate for the maximum time step for given grid spacings.

The inner and outer iterations that arise from the nonlinearity of the problem may be used if extreme accuracies are required or highly nonlinear problems are encountered. However, for the sample problem considered, the iterations made practically no change in the solution.

APPENDIX A

Approximate Solution of Bicharacteristic Curves

Let the equations describing the surface of a characteristic cone be

$$r = r(\alpha, \tau), \quad z = z(\alpha, \tau), \quad \text{and} \quad t = t(\alpha, \tau),$$

where τ is directed in the negative time coordinate; i.e., $\tau = -t$. The bicharacteristic equations then are

$$\left. \begin{aligned} \frac{\partial r}{\partial \tau}(\alpha, \tau) &= -u + a \cos \theta \\ \frac{\partial z}{\partial \tau}(\alpha, \tau) &= -w + a \sin \theta \end{aligned} \right\} \quad \text{and} \quad (A.1)$$

Assuming that u , w , and θ are locally constants, the following approximate relationship can be obtained:

$$\cos \theta \frac{\partial r}{\partial \alpha}(\alpha, \tau) = -\sin \theta \frac{\partial z}{\partial \alpha}(\alpha, \tau), \quad (A.2)$$

which becomes exact as $\theta \rightarrow \alpha$ or $\tau \rightarrow 0$. Now, consider a bicharacteristic, $\theta = \alpha$, at a point, e.g., point P in Fig. 1:

$$t = t_0, \quad r = r_0, \quad z = z_0, \quad \tau = 0,$$

and let the parameter τ increase. An approximate expression is sought here of the angle θ at time $t = t_0 - dt$ or $\tau = d\tau$ of a bicharacteristic defined by $\theta = \alpha$ at $t = t_0$ or $\tau = 0$.

Assume that the unknowns along a bicharacteristic are linear functions of τ , where the coefficients of τ are functions of α only:

$$\left. \begin{aligned} u &= u_0 + [u(\alpha) - u_0]\tau, \\ w &= w_0 + [w(\alpha) - w_0]\tau, \\ a &= a_0 + [a(\alpha) - a_0]\tau, \\ \theta &= \alpha + [\theta(\alpha) - \alpha]\tau, \end{aligned} \right\} \quad (A.3)$$

where u_0 , w_0 , and a_0 are values of u , w , and a at the apex of the cone. The analysis is carried out in a general manner so that it applies to a broader class of problems, including fully compressible flows where the sonic speed a also changes.

Substituting Eqs. A.3 into Eq. A.1 and integrating results in

$$\left. \begin{aligned} r(\alpha, \tau) &= r_0 + (-u_0 + a_0 \cos \alpha)\tau + \frac{1}{2}\{-[u(\alpha) - u_0] + \cos \alpha[a(\alpha) - a_0] \\ &\quad - a_0 \sin \alpha[\theta(\alpha) - \alpha]\}\tau^2 + O(\tau^3); \\ z(\alpha, \tau) &= z_0 + (-w_0 + a_0 \sin \alpha)\tau + \frac{1}{2}\{-[w(\alpha) - w_0] + \sin \alpha[a(\alpha) - a_0] \\ &\quad + a_0 \cos \alpha[\theta(\alpha) - \alpha]\}\tau^2 + O(\tau^3), \end{aligned} \right\} \quad (\text{A.4})$$

where r_0 and z_0 are r and z coordinates of the apex of the cone. Hence,

$$\left. \begin{aligned} \frac{\partial r}{\partial \alpha}(\alpha, \tau) &= -a_0(\sin \alpha)\tau + \frac{1}{2}[-u'(\alpha) + a'(\alpha)\cos \alpha - a(\alpha)\sin \alpha + 2a_0 \sin \alpha \\ &\quad - a_0\theta(\alpha)\cos \alpha + a_0\alpha \cos \alpha - a_0\theta'(\alpha)\sin \alpha]\tau^2 + O(\tau^3); \\ \frac{\partial z}{\partial \alpha}(\alpha, \tau) &= a_0(\cos \alpha)\tau + \frac{1}{2}[-w'(\alpha) + a(\alpha)\cos \alpha - 2a_0 \cos \alpha + a'(\alpha)\sin \alpha \\ &\quad + a_0\theta'(\alpha)\cos \alpha - a_0\theta(\alpha)\sin \alpha + a_0\alpha \sin \alpha]\tau^2 + O(\tau^3). \end{aligned} \right\} \quad (\text{A.5})$$

Substituting Eqs. A.5 into Eq. A.2 and equating coefficients of leading terms on both sides (i.e., the coefficients of τ^2), and using the chain rule, we obtain

$$\begin{aligned} \theta - \alpha &= \left[\cos^2 \alpha \frac{\partial u}{\partial z} - \sin^2 \alpha \frac{\partial w}{\partial r} + \sin \alpha \cos \alpha \left(-\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} \right) + \sin \alpha \frac{\partial a}{\partial r} \right. \\ &\quad \left. - \cos \alpha \frac{\partial a}{\partial z} \right] \tau + O(\tau^2), \end{aligned} \quad (\text{A.6})$$

where the partial derivatives are to be evaluated at $\tau = \tau$. Equation A.6 is written for $\tau = \Delta t$ in which the angle θ at $t = t_0 - \Delta t$ is expressed in terms of derivatives in the time plane at $t = t_0 - \Delta t$ and the angle $\theta = \alpha$ at the cone apex:

$$\begin{aligned} \theta - \alpha &= \left[\cos^2 \alpha \frac{\partial u}{\partial z} - \sin^2 \alpha \frac{\partial w}{\partial r} + \sin \alpha \cos \alpha \left(-\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} \right) \right. \\ &\quad \left. + \sin \alpha \frac{\partial a}{\partial r} - \cos \alpha \frac{\partial a}{\partial z} \right] \Delta t + O(\Delta t)^2. \end{aligned} \quad (\text{A.7})$$

APPENDIX B

FORTRAN Program Listing

```

C *****
C *          PROGRAM TWO-DIMENSIONAL HYDRAULIC TRANSIENTS          *
C *  THE METHOD OF CHARACTERISTICS IS USED TO TRACE PRESSURE TRANSIENTS  *
C *  IN CYLINDRICAL DOMAIN WITH AXIAL SYMMETRY. THE PRESSURE SOURCE  *
C *  IS PRESCRIBED AS AN INPUT PRESSURE-TIME HISTORY AND LOCATED IN  *
C *  THE TWO-DIMENSIONAL DOMAIN. AS A USER'S OPTION, PRESSURE SCURC  *
C *  -E CAN ALSO BE LOCATED IN CONNECTED ONE-DIMENSIONAL NETWORK.  *
C *  IN THIS CASE, EITHER AN INPUT PRESSURE-TIME HISTORY OR A SCDIU  *
C *  -M/WATER REACTION CAN BE SPECIFIED. THE ONE-DIMENSIONAL NETWORK  *
C *  -K ANALYSIS IS PERFORMED BY THE SYSTEM CODE NATRANSIENT USED  *
C *  HERE AS A SUBROUTINE.  *
C *  YONG WOO SHIN : NOVEMBER 9, 1973  *
C *****
C
C  MAIN PROGRAM
C  IMPLICIT REAL*8 (A-H,O-Z)
C  COMMON /R1/ IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,
XIP12,IP13
C  1/R2/ II,JJ,NR,NZ,NM,MM,NR1,NR2,NZ1,NZ2,NZP,NZQ,NZC,JJM,JJP,IIP
C  2/R3/ RR,ZLEN,PR,C,RHO,AIMP,RR1,ZL1,RR2,ZL2
C  3/R4/ DT,DR,DZ,DRDZ,T,TF,DMN
C  4/R5/ U(30,60,2),W(30,60,2),P(30,60,2),UMAX(30,60),WMAX(30,60),PMAX
C  5(30,60),R(30),Z(60),SN(4),CSN(4) /R10/ WA,PA,FA,WB,PB,FB,
C  6S2,W2,P2,S4,W4,P4
C
C  DATA NIN,NOUT/5,6/,GC/32.173/
C
C ***** PROGRAM LOGIC INDICATORS *****
C *  IP1 - IF 1, PRESSURE-TIME FUNCTION IS INPUT  *
C *  IF 2, PRESSURE SCURCE IS ABSENT  *
C *  IP2 - IF 0, RESULT PRINTED FOR ALL PCINTS ; IF .GT. 0, RESULT  *
C *  FOR SELECTED POINTS ( IP2 POINTS ) ONLY ARE PRINTED  *
C *  IP3 - IF 1, MAXIMUM NODAL VALUES OF RADIAL VELOCITY ARE PRINTED  *
C *  IF 2, MAXIMUM NODAL VALUES OF AXIAL VELOCITY ARE PRINTED  *
C *  IF 3, MAXIMUM NODAL VALUES OF PRESSURE ARE PRINTED  *
C *  IF 4, MAXIMUM NODAL VALUES OF U, W, AND P ARE PRINTED  *
C *  IP4 - IF 0, TERMS ON RIGHT-HAND SIDE ARE AVERAGED BETWEEN INITIAL  *
C *  AND NEW PLANE  *
C *  IF 1, TERMS ON RIGHT-HAND SIDE ARE ASSUMED TO BE IN NEW PLANE  *
C *  IP5 - IF 1, FOUR SOURCE PCINTS ((II,JJ),(II,JJ+1),(II+1,JJ),  *
C *  (II+1,JJ+1))  *
C *  IF 2, SOURCE PRESSURE ASSIGNED FOR ALL NODES AT JJ (1-D PROB)  *
C *  IP6 - IF 1, SYMMETRIC CONDITION USED ON AXIS  *
C *  IP7 - IF 0, ENCLOSED PROBLEM BYPASSING SYSTEMS CODE  *
C *  IF 1, SYSTEM ANALYSIS CALLING SYSTEMS CODE AS SUBROUTINE  *
C *  IP8 - IF 1, STEADY-STATE SOLUTION PRECEDES PRESSURE TRANSIENTS  *
C *  (SYSTEM ANALYSIS)  *
C *  IP9 - IF 1, INTERMEDIATE VALUES OF STEADY-STATE SOLUTION PRINTED  *
C *  IP10 -IF 1, PURELY 1-D SYSTEM CALCULATION  *
C *  IP11 -IF 1, RESULTS FOR EXTENDED REGION IN SYSTEM ANALYSIS PRINTED  *
C *  IP12 -IF 1, TERMINATE CALCULATION WHEN A NEGATIVE PRESSURE IS  *
C *  OBTAINED  *
C *  IP13 -IF 1, RESULTS FOR SELECTED PCINTS WRITTEN ON UNIT 10  *

```

```

C      *          FOR TEKTRONIX GRAPHICAL DISPLAY          *
C      ****
C
C *** READ INPUT VARIABLES ****
READ (NIN,1000) IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,
XIP12,IP13
READ (NIN,1000) NR,NZ,II,JJ,NF,NIT,MIT,NNH
READ (NIN,1010) DT,TI,TF,PR,RHO,C,RR,ZLEN
IF (IP7.EQ.1) READ (NIN,1000) NFS,IA,IB
IF (IP7.EQ.1) READ (NIN,1020) RR1,ZL1,RR2,ZL2,TFS
C *** PRINT INPUT DATA ****
WRITE (NOUT,200)
WRITE (NOUT,210) IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,
XIP12,IP13
WRITE (NOUT,220) NR,NZ,II,JJ,NF,NIT,MIT,NNH
WRITE (NOUT,230) DT,TI,TF,PR,RHO,C,RR,ZLEN
IF (IP7.EQ.1) WRITE (NOUT,235) NFS,IA,IB
IF (IP7.EQ.1) WRITE (NOUT,240) RR1,ZL1,RR2,ZL2,TFS

C
C ***** DESCRIPTION OF INPUT VARIABLES *****
C *      NR -      NO OF NODES IN R-COORDINATE      *
C *      NZ -      NO OF NODES IN Z-COORDINATE      *
C *      II -      R-COORDINATE OF SOURCE POINT      *
C *      JJ -      Z-COORDINATE OF SOURCE PCINT      *
C *      NF -      FREQUENCY OF OUTPUT              *
C *      DT -      INITIAL DESIRED TIME STEP, SEC    *
C *      TI -      INITIAL TIME AT WHICH CALCULATION STARTS, SEC *
C *      TF -      FINAL TIME AT WHICH CALCULATION TERMINATES, SEC *
C *      PR -      INITIAL PRESSURE, PSI             *
C *      RHO -     DENSITY OF LIQUID, LB/FT3         *
C *      C -       SONIC SPEED, FT/SEC              *
C *      RR -      RADIUS OF CYLINDRICAL DOMAIN, FT  *
C *      ZLEN -    LENGTH OF CYLINDRICAL DOMAIN, FT  *
C *      NIT -     NO OF OUTER ITERATION FOR EACH TIME-STEP CALCULATION *
C *      MIT -     NO OF INNER ITERATION FOR EACH TIME-STEP CALCULATION *
C *      NNH -     NO OF SETS OF DATA FOR PRESSURE-TIME HISTORY INPUT *
C *      NFS -     FREQUENCY OF PRINTOUT FOR STEADY-STATE SOLUTION *
C *      IA -      CONNECTED 1-D SECTION-NUMBER UPSTREAM OF 2-D DOMAIN FOR *
C *                SYSTEM ANALYSIS *
C *      IB -      CONNECTED 1-D SECTION-NUMBER DOWNSTREAM OF 2-D DOMAIN FOR *
C *                SYSTEM ANALYSIS *
C *      RR1 -     RADIUS OF THE EXTENDED REGION UPSTREAM END OF 2-D *
C *                DOMAIN, FT *
C *      ZL1 -     LENGTH OF THE EXTENDED REGION UPSTREAM END OF 2-D *
C *                DOMAIN, FT *
C *      RR2 -     RADIUS OF THE EXTENDED REGION DOWNSTREAM END OF 2-D *
C *                DOMAIN, FT *
C *      ZL2 -     LENGTH OF THE EXTENDED REGION DOWNSTREAM END OF 2-D *
C *                DOMAIN, FT *
C *      TFS -     FINAL TIME AT WHICH STEADY-STATE CALCULATION TERMINATES *
C *****
C
C      T=TI
C      CALL SETUP
C      NC = 0
C      IF (IP8.EQ.0) NFF = NF
C      IF (IP8.EQ.1) NFF = NFS
C      UWX = 0.
C      AIMP = RHO*C/(GC*144.)
C      IF (IP1.EQ.1) CALL PTME (PSRCE,NNH)

```

```

C *** INITIALIZE 1-D DOMAIN CALCULATION *****
  IF (IP7.EQ.1) CALL PTRAN (0,IA,IB,FA,WA,PA,FB,WB,PB,IP10)
  IF (IP10.EQ.1) GO TO 309
C *** INITIALIZE *****
  DO 1 J=1,60
  DO 1 I=1,30
    UMAX(I,J) = 0.
    WMAX(I,J) = 0.
    PMAX(I,J) = 0.
  DO 1 K=1,2
    U(I,J,K) = 0.
    W(I,J,K) = 0.
  1 P(I,J,K) = 0.
  IF (IP7.EQ.1) GO TO 2
C *** ENCLOSED FLOW PROBLEM *****
  TFS = 0.
  DO 5 J=1,NZ
  DO 5 I=1,NR
    UMAX(I,J) = 0.
    WMAX(I,J) = 0.
    PMAX(I,J) = PR/AIMP
  DO 5 K=1,2
    U(I,J,K) = 0.
    W(I,J,K) = 0.
  5 P(I,J,K) = PR/AIMP
  GO TO 11
  2 NRM1 = NR1 - 1
    NRM2 = NR2 - 1
    NZM2 = NZ2 - 1
    NZP1 = NZP + 1
    NZQ1 = NZQ - 1
    NRP1 = NR1 + 1
    NRP2 = NR2 + 1
  DO 6 J=1,NZ1
  DO 6 I=1,NR1
    UMAX(I,J) = 0.
    WMAX(I,J) = 0.
    PMAX(I,J) = PR/AIMP
  DO 6 K = 1,2
    U(I,J,K) = 0.
    W(I,J,K) = WA
  6 P(I,J,K) = PR/AIMP
  DO 7 J = NZP,NZQ
  DO 7 I = 1,NR
    UMAX(I,J) = 0.
    WMAX(I,J) = 0.
    PMAX(I,J) = PR/AIMP
  DO 7 K = 1,2
    U(I,J,K) = 0.
    W(I,J,K) = WA*(RR1/RR)**2
  7 P(I,J,K) = PR/AIMP
  DO 8 J = NZC,NZ2
  DO 8 I = 1,NR2
    UMAX(I,J) = 0.
    WMAX(I,J) = 0.
    PMAX(I,J) = PR/AIMP
  DO 8 K = 1,2
    U(I,J,K) = 0.
    W(I,J,K) = WB

```

```

      8 P(I,J,K) = PR/AIMP
     11 IF (IP8.EQ.1) GO TO 12
C *** PRINT INITIAL DATA *****
     80 CALL CHART (NC,PSRCE)
       IF (IP7.EQ.1) GO TO 70
       DO 13 JK=1,NZ
       DO 13 IK=1,NR
         U(IK,JK,1) = U(IK,JK,2)
     13 P(IK,JK,1) = P(IK,JK,2)
       GO TO 12
     70 DO 71 J=1,NZ1
       DO 71 I=1,NR1
         U(I,J,1) = U(I,J,2)
     71 P(I,J,1) = P(I,J,2)
       DO 72 J=NZP,NZQ
       DO 72 I=1,NR
         U(I,J,1) = U(I,J,2)
     72 P(I,J,1) = P(I,J,2)
       DO 73 J=NZC,NZ2
       DO 73 I=1,NR2
         U(I,J,1) = U(I,J,2)
     73 P(I,J,1) = P(I,J,2)
     12 IF (IP1.EQ.2.OR.IP8.EQ.1) GO TO 15
C *** SOURCE POINT CALCULATION *****
       IF (IP1.EQ.1.AND.T.GT.TI) CALL PTME (PSRCE,NNH)
       P(II,JJ,1)=PSRCE/AIMP
       IF (IP5.EQ.0) P(II+1,JJ,1) = P(II,JJ,1)
       IF (IP5.NE.1) GO TO 14
       P(II+1,JJ,1) = P(II,JJ,1)
       P(II+1,JJ+1,1) = P(II,JJ,1)
       P(II,JJ+1,1) = P(II,JJ,1)
     14 IF (IP5.NE.2) GO TO 15
       DO 18 I=2,NR1
       P(I,JJM,1) = P(II,JJ,1)
     18 P(I,JJ,1) = P(II,JJ,1)
C *** CHECK DT TO SATISFY STABILITY CRITERION *****
     15 DTT=DT
       DTCK = .95*DMN/(UWX + C)
       IF(DT.LE.DTCK) GO TO 20
       DT = 0.5*DT
       WRITE (NCUT,2000) DTT,DT
       GO TO 99
     20 CONTINUE
       IF (IP7.EQ.1) GO TO 106
C
C *** ENCLOSED FLOW PROBLEM *****
C
C *** ORDINARY POINT CALCULATION *****
       DO 30 J=2,MM
       DO 30 I=2,NM
         30 CALL COMPTE(I,J,NIT,MIT,1)
C *** AXIAL POINT CALCULATION *****
       I = 1
       DO 32 J=2,MM
         32 CALL COMPTE(I,J,NIT,MIT,2)
C *** BOUNDARY POINT CALCULATION: R = RR *****
       I = NR
       DO 34 J = 2,MM
         34 CALL COMPTE (I,J,NIT,MIT,3)
C *** BOUNDARY POINT CALCULATION : Z = 0 *****

```



```

      J = 1
      DO 36 I=2,NM
36  CALL COMPTE (I,J,NIT,MIT,4)
C *** BOUNDARY POINT CALCULATION : Z = LENGTH *****
      J = NZ
      DO 38 I=2,NM
38  CALL COMPTE (I,J,NIT,MIT,5)
C *** CORNER POINT CALCULATION: R = 0, Z = 0 *****
      I = 1
      J = 1
      CALL COMPTE (I,J,NIT,MIT,6)
C *** CORNER POINT CALCULATION : R = 0, Z = LENGTH *****
40  I = 1
      J = NZ
      CALL COMPTE (I,J,NIT,MIT,7)
C *** CORNER POINT CALCULATION: R = RR, Z = 0 *****
42  I = NR
      J = 1
      CALL COMPTE (I,J,NIT,MIT,8)
C *** CORNER POINT CALCULATION : R = RR, Z = LENGTH *****
44  I = NR
      J = NZ
      CALL COMPTE (I,J,NIT,MIT,9)
      GO TO 48
C
C *** SYSTEM ANALYSIS *****
C
C *** PROVIDE NECESSARY INFORMATION AT INTERFACES FOR 2-D CALCULATION *****
106 IF (IP8.EQ.1) GO TO 107
      CALL PTRAN (1,IA,IB,FA,WA,PA,FB,WB,PB,IP10)
C *** ORDINARY POINT (KMT = 1) *****
107 DO 108 J=2,NZP
      DO 108 I=2,NRM1
108  CALL COMPTE (I,J,NIT,MIT,1)
      DO 110 J=NZP1,NZQ1
      DO 110 I=2,NM
110  CALL COMPTE (I,J,NIT,MIT,1)
      DO 112 J=NZQ,NZM2
      DO 112 I=2,NRM2
112  CALL COMPTE (I,J,NIT,MIT,1)
C *** AXIAL NCN-BOUNDARY POINT: R=0 (KMT = 2) *****
      I = 1
      DO 114 J=2,NZM2
114  CALL COMPTE (I,J,NIT,MIT,2)
C *** PERIPHERY POINT (KMT = 3) *****
      I = NR1
      DO 116 J=2,NZ1
116  CALL COMPTE (I,J,NIT,MIT,3)
      I = NR
      DO 118 J=NZP1,NZQ1
118  CALL COMPTE (I,J,NIT,MIT,3)
      I = NR2
      DO 120 J=NZC,NZM2
120  CALL COMPTE (I,J,NIT,MIT,3)
C *** BOUNDARY POINT: Z = ZL1 (KMT = 4) *****
      J = NZP
      DO 122 I=NRP1,NM
122  CALL COMPTE (I,J,NIT,MIT,4)
C *** BOUNDARY POINT: Z = ZL1 + ZLEN (KMT = 5) *****
      J = NZQ
      DO 124 I=NRP2,NM
124  CALL COMPTE (I,J,NIT,MIT,5)

```

```

C *** CORNER POINT: R = RR, Z = ZL1 (KMT = 8) *****
  I = NR
  J = NZP
  CALL COMPTE (I,J,NIT,MIT,8)
C *** CORNER POINT: R = RR, Z = ZL1 + ZLEN (KMT = 9) *****
  I = NR
  J = NZQ
  CALL COMPTE (I,J,NIT,MIT,9)
C *** UPSTREAM INTERFACE: Z = 0 (KMT = 11) *****
  IF (NR1.LE.2) GO TO 130
  J = 1
  DO 126 I=2,NRM1
126 CALL COMPTE (I,J,NIT,MIT,11)
C *** UPSTREAM INTERFACE AXIAL POINT: R = 0, Z = 0 (KMT = 10) *****
130 I = 1
  J = 1
  CALL COMPTE (I,J,NIT,MIT,10)
C *** UPSTREAM INTERFACE BOUNDARY POINT: R = RR1, Z = 0 (KMT = 12) *****
  I = NR1
  J = 1
  CALL COMPTE (I,J,NIT,MIT,12)
C *** DOWNSTREAM INTERFACE: Z = ZL1 + ZLEN + ZL2 (KMT = 14) *****
  IF (NR2.LE.2) GO TO 140
  J = NZ2
  DO 132 I = 2,NRM2
132 CALL COMPTE (I,J,NIT,MIT,14)
C *** DOWNSTREAM INTERFACE AXIAL POINT: R = 0, Z = ZL1 + ZLEN + ZL2 (KMT=13) ***
140 I = 1
  J = NZ2
  CALL COMPTE (I,J,NIT,MIT,13)
C *** DOWNSTREAM INTERFACE BOUNDARY POINT: R = RR2, Z = ZL1 + ZLEN + ZL2 *****
C *** (KMT = 15) *****
  I = NR2
  J = NZ2
  CALL COMPTE (I,J,NIT,MIT,15)
C *** SPECIAL POINT: R = RR1, Z = ZL1 (KMT = 16) *****
  I = NR1
  J = NZP
  CALL COMPTE (I,J,NIT,MIT,16)
C *** SPECIAL POINT: R = RR2, Z = ZL1 + ZLEN (KMT =17) *****
  I = NR2
  J = NZQ
  CALL COMPTE (I,J,NIT,MIT,17)
C
  IF (IP8.EQ.1) GO TO 48
C *** PROVIDE NECESSARY INFORMATION AT INTERFACES FOR 1-D SYSTEM-CALCULATION ***
  CALL SUPPL (MIT)
C *** 1-D NETWORK CALCULATION *****
309 CALL PTRAN (2,IA,IB,S2,W2,P2,S4,W4,P4,IP10)
  IF (IP10.EQ.1) GO TO 315
C *** DETERMINE MAXIMUM VALUES *****
48 UMX = 0.
  WMX = 0.
  IF (IP7.EQ.1) GO TO 50
  DO 49 J = 1, NZ
  DO 49 I = 1, NR
  IF (IP12.EQ.1.AND.P(I,J,2).LT.0.) GO TO 90
  UMAX(I,J)=DMAX1(UMAX(I,J),DAPS(U(I,J,2)))
  WMAX(I,J)=DMAX1(WMAX(I,J),DABS(W(I,J,2)))
  PMAX(I,J)=DMAX1(PMAX(I,J),CAPS(P(I,J,2)))
  UMX = DMAX1(UMX,DABS(U(I,J,2)))
49 WMX = DMAX1(WMX,DABS(W(I,J,2)))

```

```

      GO TO 57
50 DO 51 J=1,NZ1
   DO 51 I=1,NR1
   IF (IP12.EQ.1.AND.P(I,J,2).LT.0.) GO TO 90
   UMAX(I,J)=DMAX1(UMAX(I,J),DABS(U(I,J,2)))
   WMAX(I,J)=DMAX1(WMAX(I,J),DABS(W(I,J,2)))
   PMAX(I,J)=DMAX1(PMAX(I,J),DABS(P(I,J,2)))
   UMX = DMAX1(UMX,DABS(U(I,J,2)))
51 WMX = DMAX1(WMX,DABS(W(I,J,2)))
   DO 52 J=NZP,NZQ
   DO 52 I=1,NR
   IF (IP12.EQ.1.AND.P(I,J,2).LT.0.) GO TO 90
   UMAX(I,J)=DMAX1(UMAX(I,J),DABS(U(I,J,2)))
   WMAX(I,J)=DMAX1(WMAX(I,J),DABS(W(I,J,2)))
   PMAX(I,J)=DMAX1(PMAX(I,J),DABS(P(I,J,2)))
   UMX = DMAX1(UMX,DABS(U(I,J,2)))
52 WMX = DMAX1(WMX,DABS(W(I,J,2)))
   DO 53 J=NZC,NZ2
   DO 53 I=1,NR2
   IF (IP12.EQ.1.AND.P(I,J,2).LT.0.) GO TO 90
   UMAX(I,J)=DMAX1(UMAX(I,J),DABS(U(I,J,2)))
   WMAX(I,J)=DMAX1(WMAX(I,J),DABS(W(I,J,2)))
   PMAX(I,J)=DMAX1(PMAX(I,J),DABS(P(I,J,2)))
   UMX = DMAX1(UMX,DABS(U(I,J,2)))
53 WMX = DMAX1(WMX,DABS(W(I,J,2)))
57 UWX = DMAX1(UMX,WMX)
C *** PRINT RESULTS *****
315 NC = NC + 1
   T = T + DT
   IF (IP10.EQ.1) GO TO 309
   IF (T.GT.TF.AND.IP8.EQ.0) GO TO 999
   IF ((NC/NFF*NFF).GE.NC) CALL CHART (NC,PSRCE)
   IF (IP7.EQ.1) GO TO 60
   DO 59 J=1,NZ
   DO 59 I=1,NR
   U(I,J,1) = U(I,J,2)
   W(I,J,1) = W(I,J,2)
59 P(I,J,1) = P(I,J,2)
   GO TO 12
60 DO 61 J=1,NZ1
   DO 61 I=1,NR1
   U(I,J,1) = U(I,J,2)
   W(I,J,1) = W(I,J,2)
61 P(I,J,1) = P(I,J,2)
   DO 62 J=NZP,NZQ
   DO 62 I=1,NR
   U(I,J,1) = U(I,J,2)
   W(I,J,1) = W(I,J,2)
62 P(I,J,1) = P(I,J,2)
   DO 63 J=NZC,NZ2
   DO 63 I=1,NR2
   U(I,J,1) = U(I,J,2)
   W(I,J,1) = W(I,J,2)
63 P(I,J,1) = P(I,J,2)
   IF (T.EQ.TFS.AND.IP8.EQ.1) GO TO 65
   GO TO 12
C *** STEADY-STATE CALCULATION IS COMPLETED *****
65 IP8 = 0
   NC = 0
   NFF = NF
   T = TI
   GO TO 80

```

```

C *** NEGATIVE PRESSURE - STOP COMPUTATION *****
  90 WRITE (NOUT,250)
    GO TO 999
C *** FORMAT STATEMENTS *****
  200 FORMAT (1H1,////,32X,'----- INPUT DATA FOR TWO - DIMENSIONAL CCMA
        XIN CALCULATION -----',//)
  210 FORMAT (20X,'IP1 =',I5,10X,'IP2 =',I5,10X,'IP3 =',I5,10X,
        1'IP4 =',I5,10X,'IP5 =',I5,/,20X,'IP6 =',I5,10X,'IP7 =',I5,
        210X,'IP8 =',I5,10X,'IP9 =',I5,10X,'IP10 =',I5,/,20X,'IP11 =',I5,
        310X,'IP12 =',I5,10X,'IP13 =',I5)
  220 FORMAT (//,20X,'NR =',I5,10X,'NZ =',I5,10X,'II =',I5,10X,
        1'JJ =',I5,10X,'NF =',I5,/,20X,'NIT =',I5,10X,'MIT =',I5,
        210X,'NNH =',I5)
  230 FORMAT (//,20X,'DT =',1PE11.3,4X,'TI =',1PE11.3,4X,'TF =',
        11PE11.3,4X,'PR =',1PE11.3,4X,'RHO =',1PE11.3,/,20X,'C =',
        21PE11.3,4X,'RR =',1PE11.3,4X,'ZLEN =',1PE11.3)
  235 FORMAT (//,20X,'NFS =',I5,10X,'IA =',I5,10X,'IB =',I5)
  240 FORMAT (//,20X,'RR1 =',1PE11.3,4X,'ZL1 =',1PE11.3,4X,
        1'RR2 =',1PE11.3,4X,'ZL2 =',1PE11.3,4X,'TFS =',1PE11.3)
  250 FORMAT (////,35X,'----- NEGATIVE PRESSURE - STOP COMPUTATION --
        X--')
1000 FORMAT (16I5)
1010 FORMAT (6E12.5,/,6E12.5)
1020 FORMAT (6E12.5)
2000 FORMAT(//,5X,'***** DT IS DECREASED FROM',E12.5,'TO' E12.5,' *****
        1')
  99 WRITE (NOUT,2020)
2020 FORMAT (//,5X,'***** DT IS TOO LARGE FOR NUMERICAL STABILITY ----
        XRERUN WITH SMALLER DT *****')
  999 STOP
      END

      IF (RR2.LT.DR) NR2 = 2
      IF (RR2.GE.DR) NR2 = RR2/DR + 1.5
      NZ1 = ZL1/DZ + 0.5
      NZP = NZ1 + 1
      NZQ = NZ1 + NZ
      NZC = NZQ + 1
      MOC = ZL2/DZ + 0.5
      NZ2 = NZQ + MOC
      DO 45 J=2,NZ2
45 Z(J) = Z(J-1) + DZ
      WRITE (6,100) NR1,NZ1,NZP,NR2,NZQ,NZC,NZ2
100 FORMAT (//,20X,'NR1 =',I5,10X,'NZ1 =',I5,10X,'NZP =',I5,10X,
        X'NR2 =',I5,10X,'NZQ =',I5,/,20X,'NZC =',I5,10X,'NZ2 =',I5)
      RETURN
      END

      SUBROUTINE SETUP
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /R1/ IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,IP12,
        1IP13 /R2/ II,JJ,NR,NZ,NM,MM,NR1,NR2,NZ1,NZ2,NZP,NZQ,NZC,JJM,JJP,II
        2/R3/ RR,ZLEN,PR,C,RHO,AIMP,RR1,ZL1,RR2,ZL2 /R4/ DT,DR,DZ,DRDZ,T,TI,
        3TF,DMN /R5/ U(30,60,2),W(30,60,2),P(30,60,2),UMAX(30,60),
        4WMAX(30,60),PMAX(30,60),R(30),Z(60),SN(4),CSN(4)
C
C *** INITIALIZE *****
      DO 5 K=1,30

```

```

5 R(K) = 0.
DO 6 L=1,60
6 Z(L) = 0.
C
  JJM = JJ - 1
  JJP = JJ + 1
  IIP = II + 1
  NM = NR-1
  MM = NZ-1
  DR = RR/FLOAT(NM)
  DZ = ZLEN/FLOAT(MM)
  DRDZ = DR*DZ
  DMN = DMIN1(DR,DZ)
  R(1) = 0.
  DO 10 I=2,NR
10 R(I)=R(I-1)+DR
  Z(1)=0.
  IF (IP7.EQ.1) GO TO 20
  DO 15 J=2,NZ
15 Z(J) = Z(J-1) + DZ
20 SN(1) = 0.
  SN(2) = 1.
  SN(3) = 0.
  SN(4) = -1.
  CSN(1) = 1.
  CSN(2) = 0.
  CSN(3) = -1.
  CSN(4) = 0.
  IF (IP7.EQ.1) GO TO 40
  RR1 = 0.
  ZL1 = 0.
  RR2 = 0.
  ZL2 = 0.
  NR1 = 0
  NR2 = 0
  NZ1 = 0
  NZP = 0
  NZQ = 0
  NZC = 0
  NZ2 = 0
  RETURN
40 IF (RR1.LT.DR) NR1 = 2
  IF (RR1.GE.DR) NR1 = RR1/DR + 1.5
  SUBROUTINE CCMPTE(I,J,NIT,MIT,KMT)
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /R1/ IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,IP12,
  1IP13 /R2/ II,JJ,NR,NZ,NM,MM,NR1,NR2,NZ1,NZ2,NZP,NZQ,NZC /R3/
  2RR,ZLEN,PR,C,RHO,AIMP,RR1,ZL1,RR2,ZL2 /R4/ DT,DR,DZ,DRDZ,T,TT,TF,
  3DMN /R5/ U(30,60,2),W(30,60,2),P(30,60,2),UMAX(30,60),WMAX(30,60),
  4PMAX(30,60),R(30),Z(60),SN(4),CSN(4)
  DIMENSION DUDR(4),DWDZ(4),RO(4),UQ(4),WQ(4),PQ(4),DRQ(4),DZQ(4)
C
C *** THIS SUBPROGRAM CALCULATES ORDINARY, AXIAL AND BOUNDARY POINTS *****
C * DEPENDING ON THE FLAG KMT AS FOLLOWS: *
C * IF KMT = 1; ORDINARY PCINT *
C * IF KMT = 2; AXIAL POINT *
C * IF KMT = 3; BOUNDARY POINT, R = RR, R = RR1, R = RR2 *
C * IF KMT = 4; BOUNDARY POINT, Z = 0 (ENCLOSED FLOW), Z = ZL1 (SYSTEM ANAL) *
C * IF KMT = 5; BOUNDARY POINT, Z = L (ENCLOSE FLOW), Z = ZL1 + ZLEN *
C * (SYSTEM ANALYSIS) *
C * IF KMT = 6; CORNER POINT, R = 0, Z = 0 (ENCLOSED FLOW) *
C * IF KMT = 7; CCRNER PCINT, R = 0. Z = L (ENCLOSED FLOW) *

```

```

C * IF KMT = 8; CORNER POINT, R = RR, Z = 0 *
C * IF KMT = 9; CORNER POINT, R = RR, Z = L *
C * IF KMT = 10; CORNER POINT, R = 0, Z = 0 (SYSTEM ANALYSIS) *
C * IF KMT = 11; UPSTREAM INTERFACE, Z = 0 (SYSTEM ANALYSIS) *
C * IF KMT = 12; BOUNDARY POINT, R = RR1, Z = 0 (SYSTEM ANALYSIS) *
C * IF KMT = 13; CORNER POINT, R = 0, Z = ZL1 + ZLEN + ZL2 (SYSTEM ANALYSIS) *
C * IF KMT = 14; DOWNSTREAM INTERFACE, Z = ZL1 + ZLEN + ZL2 (SYSTEM ANAL) *
C * IF KMT = 15; BOUNDARY POINT, R = RR2, Z = ZL1 + ZLEN + ZL2 (SYSTEM ANAL) *
C *****
C
C *** INITIALIZE *****
DO 5 M=1,4
  DUDR(M) = 0.
  DWDZ(M) = 0.
  RQ(M) = 0.
  UQ(M) = 0.
  WQ(M) = 0.
  PQ(M) = 0.
  DRQ(M) = 0.
5 DZQ(M) = 0.
  MC = -1
  MT = MIT + 1
C *** VALUES AT R AND DISPLACEMENT OF STREAMLINE *****
299 KT = 1
C *** SKIP THIS CALCULATION FOR INTERFACES BETWEEN 2-D AND 1-D DOMAIN *****
IF ((IP7.EQ.1.AND.J.EQ.1).OR.(IP7.EQ.1.AND.J.EQ.NZ2)) GO TO 60
C *** FIRST ESTIMATES *****
  DRR = U(I,J,1)*DT
  DZR = W(I,J,1)*DT
  IF ((KMT.EQ.16.AND.DRR.LT.0..AND.DZR.GT.0.).OR.(KMT.EQ.17.AND.
XDRR.LT.0..AND.DZR.LT.0.)) DRR = 0.
10 IF (KMT.EQ.3.OR.KMT.EQ.8.OR.KMT.EQ.9) GO TO 12
  IF ((KMT.EQ.16.AND.DRR.EQ.0.).OR.(KMT.EQ.17.AND.DRR.EQ.0.))
XGO TO 12
  IF (DRR.LE.0.) GO TO 20
12 I1=I-1
  FRR = 1.-DRR/DR
  GO TO 22
20 I1 = I
  FRR = -DRR/DR
22 IF (KMT.EQ.5.OR.KMT.EQ.7.OP.KMT.EQ.9) GO TO 24
  IF (KMT.EQ.17.AND.DRR.LE.0..AND.DZR.GE.0.) GO TO 24
  IF (DZR.LE.0.) GO TO 25
24 J1 = J-1
  FRZ = 1.-DZR/DZ
  GO TO 30
25 J1 = J
  FRZ = - DZR/DZ
C *** FOUR POINT BIVARIATE INTERPOLATION *****
30 IF (KMT.EQ.16.AND.DRR.LT.0..AND.DZR.GT.0.) GO TO 350
  IF (KMT.EQ.17.AND.DRR.LT.0..AND.DZR.LT.0.) GO TO 350
  URT = (1.-FRR)*(1.-FRZ)*U(I1,J1,1) + FRR*(1.-FRZ)*U(I1+1,J1,1) +
1FRZ*(1.-FRR)*U(I1,J1+1,1) + FRR*FRZ*U(I1+1,J1+1,1)
  WRT = (1.-FRR)*(1.-FRZ)*W(I1,J1,1) + FRR*(1.-FRZ)*W(I1+1,J1,1) +
1FRZ*(1.-FRR)*W(I1,J1+1,1) + FRR*FRZ*W(I1+1,J1+1,1)
  IF (KT.GE.MT) GO TO 35
C *** IMPROVE ESTIMATES BY ITERATION *****
  DRR = URT*DT
  DZR = WRT*DT
  KT = KT + 1
  GO TO 10
35 PRT = (1.-FRR)*(1.-FRZ)*P(I1,J1,1) + FRR*(1.-FRZ)*P(I1+1,J1,1) +

```

```

1FRZ*(1.-FRR)*P(I1,J1+1,1) + FRR*FRZ*P(I1+1,J1+1,1)
  RRT = R(I) - DRR
  IF (IP4.EQ.1) GO TO 60
C *** PARTIAL DERIVATIVES *****
  IF (I.EQ.1.OR.KMT.EQ.3.OR.KMT.EQ.8.OR.KMT.EQ.9) GO TO 40
  IF (KMT.EQ.16.OR.KMT.EQ.17) GO TO 45
  DUDRR = (1.-FRZ)*(U(I+1,J1,1)-U(I-1,J1,1))/(2.*DR) + FRZ*
1(U(I+1,J1+1,1)-U(I-1,J1+1,1))/(2.*DR)
  GO TO 50
40 DUDRR = (1.-FRZ)*(U(I1+1,J1,1)-U(I1,J1,1))/DR + FRZ*
1(U(I1+1,J1+1,1)-U(I1,J1+1,1))/DR
  GO TO 50
45 DUDRR = (U(I+1,J,1)-U(I-1,J,1))/(2.*DR)
50 IF (KMT.EQ.4.OR.KMT.EQ.5.OR.KMT.EQ.6.OR.KMT.EQ.7.OR.KMT.EQ.8.OR.
1KMT.EQ.9) GO TO 55
  IF (KMT.EQ.16.OR.KMT.EQ.17) GO TO 57
  DWDZR = (1.-FRR)*(W(I1,J+1,1)-W(I1,J-1,1))/(2.*DZ) + FRR*
1(W(I1+1,J+1,1)-W(I1+1,J-1,1))/(2.*DZ)
  GO TO 60
55 DWDZR = (1.-FRR)*(W(I1,J1+1,1)-W(I1,J1,1))/DZ + FRR*
1(W(I1+1,J1+1,1)-W(I1+1,J1,1))/DZ
  GO TO 60
57 DWDZR = (W(I,J+1,1)-W(I,J-1,1))/(2.*DZ)
C *** VALUES AT Q AND DISPLACEMENT OF BICHAERACTERISTICS *****
60 KT = 1
  DO 99 K=1,4
  IF (I.EQ.1.AND.K.EQ.3) GO TO 99
  IF ((KMT.EQ.3.AND.K.EQ.1).OR.(KMT.EQ.8.AND.K.EQ.1).OR.(KMT.EQ.9
1.AND.K.EQ.1).OR.(KMT.EQ.12.AND.K.EQ.1).OR.(KMT.EQ.15.AND.K.EQ.1))
2GO TO 99
  IF ((J.EQ.1.AND.K.EQ.4).OR.(KMT.EQ.4.AND.K.EQ.4).OR.(KMT.EQ.8
1.AND.K.EQ.4).OR.(KMT.EQ.6.AND.K.EQ.4)) GO TO 99
  IF ((KMT.EQ.5.AND.K.EQ.2).OR.(KMT.EQ.9.AND.K.EQ.2).OR.(KMT.EQ.7
1.AND.K.EQ.2).OR.(KMT.EQ.13.AND.K.EQ.2).OR.(KMT.EQ.14.AND.K.EQ.
22).OR.(KMT.EQ.15.AND.K.EQ.2)) GO TO 99
  IF ((KMT.EQ.11.AND.K.NE.2).OR.(KMT.EQ.10.AND.K.NE.2).OR.(KMT.EQ.
112.AND.K.NE.2).OR.(KMT.EQ.13.AND.K.NE.4).OR.(KMT.EQ.14.AND.
2K.NE.4).OR.(KMT.EQ.15.AND.K.NE.4)) GO TO 99
C *** FIRST ESTIMATES *****
67 DRQ(K) = (U(I,J,1)-C*CSN(K))*DT
  DZQ(K) = (W(I,J,1)-C*SN(K))*DT
  IF (KMT.EQ.16.AND.K.EQ.4) DRQ(K) = (U(I,J-1,1)-C*CSN(K))*DT
  IF ((KMT.EQ.16.AND.K.EQ.1).OR.(KMT.EQ.17.AND.K.EQ.1)) DZQ(K) =
1(W(I+1,J,1)-C*SN(K))*DT
  IF (KMT.EQ.17.AND.K.EQ.2) DRQ(K) = (U(I,J+1,1)-C*CSN(K))*DT
69 IF (KMT.EQ.3.OR.KMT.EQ.8.OR.KMT.EQ.9.OR.KMT.EQ.12.OR.KMT.EQ.15.
1OR.(KMT.EQ.16.AND.K.EQ.4).OR.(KMT.EQ.17.AND.K.EQ.2)) GO TO 74
  IF (DRQ(K).LE.0.) GO TO 70
74 I1 = I-1
  FRR = 1.-DRQ(K)/DR
  GO TO 72
70 I1 = I
  FRR = -CRQ(K)/DR
72 IF (KMT.EQ.5.OR.KMT.EQ.7.OR.KMT.EQ.9.OR.(KMT.EQ.17.AND.K.EQ.1))
1GO TO 73
  IF (DZQ(K).LE.0.) GO TO 75
73 J1 = J-1
  FRZ = 1.-DZQ(K)/DZ
  GO TO 79
75 J1 = J
  FRZ = -DZQ(K)/DZ
79 UQ(K) = (1.-FRR)*(1.-FRZ)*U(I1,J1,1) + FRR*(1.-FRZ)*U(I1+1,J1,1) +

```

```

1FRZ*(1.-FRR)*U(I1,J1+1,1) + FRR*FRZ*U(I1+1,J1+1,1)
WQ(K) = (1.-FRR)*(1.-FRZ)*W(I1,J1,1) + FRR*(1.-FRZ)*W(I1+1,J1,1) +
1FRZ*(1.-FRR)*W(I1,J1+1,1) + FRR*FRZ*W(I1+1,J1+1,1)
IF (KT.GE.MT) GO TO 85
C *** IMPROVE ESTIMATES BY ITERATION *****
IF ((KMT.EQ.16.AND.K.EQ.4).OR.(KMT.EQ.17.AND.K.EQ.2)) GO TO 80
DRQ(K) = (UQ(K)-C*CSN(K))*DT
80 IF((KMT.EQ.16.AND.K.EQ.1).OR.(KMT.EQ.17.AND.K.EQ.1)) GO TO 81
CZQ(K) = (WQ(K)-C*SN(K))*DT
81 KT = KT + 1
GO TO 69
85 PQ(K) = (1.-FRR)*(1.-FRZ)*P(I1,J1,1) + FRR*(1.-FRZ)*P(I1+1,J1,1) +
1FRZ*(1.-FRR)*P(I1,J1+1,1) + FRR*FRZ*P(I1+1,J1+1,1)
RQ(K) = R(I) - DRQ(K)
IF (IP4.EQ.1) GO TO 99
C *** PARTIAL DERIVATIVES *****
IF (I.EQ.1.OR.KMT.EQ.3.OR.KMT.EQ.8.OR.KMT.EQ.9.OR.KMT.EQ.12.OR.
XKMT.EQ.15.OR.(KMT.EQ.16.AND.K.EQ.4).OR.(KMT.EQ.17.AND.K.EQ.2))
2GO TO 89
87 DUDR(K) = (1.-FRZ)*(U(I1+1,J1,1)-U(I1-1,J1,1))/(2.*DR) + FRZ*
X(U(I1+1,J1+1,1)-U(I1-1,J1+1,1))/(2.*DR)
GO TO 91
89 DUDR(K) = (1.-FRZ)*(U(I1+1,J1,1)-U(I1,J1,1))/DR + FRZ*
X(U(I1+1,J1+1,1)-U(I1,J1+1,1))/DR
91 IF (J.EQ.1.OR.KMT.EQ.4.OR.KMT.EQ.5.OR.KMT.EQ.6.OR.KMT.EQ.8.OR.
XKMT.EQ.7.OR.KMT.EQ.9.OR.KMT.EQ.13.OR.KMT.EQ.14.OR.KMT.EQ.15.OR.
2(KMT.EQ.16.AND.K.EQ.1).OR.(KMT.EQ.17.AND.K.EQ.1)) GO TO 95
DWDZ(K) = (1.-FRR)*(W(I1,J+1,1)-W(I1,J-1,1))/(2.*DZ) + FRR*
X(W(I1+1,J+1,1)-W(I1+1,J-1,1))/(2.*DZ)
GO TO 99
95 DWDZ(K) = (1.-FRR)*(W(I1,J1+1,1)-W(I1,J1,1))/DZ + FRR*
X(W(I1+1,J1+1,1)-W(I1+1,J1,1))/DZ
99 CONTINUE
CALL SOLVE (I,J,KMT,DUDRR,DWDZR,RRT,URT,PRT,DUDR,DWDZ,RQ,UQ,
1WQ,PQ,DRQ)
MC = MC + 1
IF(MC.GE.NIT) RETURN
C *** REPEAT FOR NONLINEAR ITERATION *****
GO TO 299
350 WRITE (6,1010) KMT,DRR,DZR
1010 FORMAT (//,10X,'---- TROUBLE AT SPECIAL POINT ',I5,4X,
1'DRR =',1PE11.3,4X,'DZR =',1PE11.3,' ---')
T = TF
RETURN
END

SUBROUTINE SUPPL (MIT)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /R2/II,JJ,NR,NZ,NM,MM,NR1,NR2,NZ1,NZ2,NZP,NZQ,NZC
1/R3/ RR,ZLEN,PR,C,RHO,AIMP,RR1,ZL1,RR2,ZL2 /R4/ DT,DR,DZ,
2DRDZ,T,TI,TF,DMN /R5/ U(30,60,2),W(30,60,2),P(30,60,2),
3UMAX(30,60),WMAX(30,60),PMAX(30,60),R(30),Z(60),SN(4),CSN(4)
4/R10/ WA,PA,FA,WB,PB,FB,S2,W2,P2,S4,W4,P4
C
C *** THIS SUBPROGRAM CALCULATES VALUES NEEDED FOR CONNECTED 1-D CALCULATION ***
C
MT = MIT + 1
KT = 1
W2 = 0.
P2 = 0.

```



```

CC 10 I=1,NR1
DZ2 = -(W(I,1,1)-C*SN(2))*DT
2 WW = W(I,1,1)*(DZ-DZ2)/DZ + W(I,2,1)*DZ2/DZ
IF (KT.GE.MT) GO TO 8
DZ2 = -(WW-C*SN(2))*DT
KT = KT + 1
GO TO 2
8 PP = P(I,1,1)*(DZ-DZ2)/DZ + P(I,2,1)*DZ2/DZ
W2 = W2 + WW
10 P2 = P2 + PP
W2 = W2/FLOAT(NR1)
P2 = P2/FLCAT(NR1)
KT = 1
W4 = 0.
P4 = 0.
DO 20 I = 1,NR2
DZ4 = (W(I,NZ2,1)-C*SN(4))*DT
12 WW = W(I,NZ2-1,1)*DZ4/DZ + W(I,NZ2,1)*(DZ-DZ4)/DZ
IF (KT.GE.MT) GO TO 18
DZ4 = (WW-C*SN(4))*DT
KT = KT + 1
GO TO 12
18 PP = P(I,NZ2-1,1)*DZ4/DZ + P(I,NZ2,1)*(DZ-DZ4)/DZ
W4 = W4 + WW
20 P4 = P4 + PP
W4 = W4/FLOAT(NR2)
P4 = P4/FLOAT(NR2)
S2 = 0.
S4 = 0.
RETURN
END

```

```

SUBROUTINE SOLVE (I,J,KMT,DUDRR,DWDZR,RRT,URT,PRT,DUDR,DWDZ,
1RQ,UQ,WQ,PQ,DRQ)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /R1/ IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,IP12,
1IP13 /R2/ II,JJ,NR,NM,MM,NR1,NR2,NZ1,NZ2,NZP,NZQ,NZC,JJM,JJP,IIP
2/R3/ RR,ZLEN,PR,C,RHO,AIMP,RP1,ZL1,RR2,ZL2 /R4/ DT,DR,DZ,DRDZ,
3T,TI,TF,DMN /R5/ U(30,60,2),W(30,60,2),P(30,60,2),UMAX(30,60),
4WMAX(30,60),PMAX(30,60),R(30),Z(60),SN(4),CSN(4) /R10/ WA,PA,
5FA,WB,PE,FB,S2,W2,P2,S4,W4,P4

```

```

C DIMENSION DUDR(4),DWCZ(4),RC(4),UQ(4),WQ(4),PQ(4),DRQ(4),S(5)
C
C QT = C*DT/2.
IF (IP4.EQ.1) GO TO 190
C *** RIGHT-HAND SIDE TERMS AVERAGED BETWEEN INITIAL AND NEW PLANE *****
IF (RQ(1).LE.0.) S(1) = DWCZ(1) + DUDR(1)
IF (RQ(1).GT.0.) S(1) = DWDZ(1) + UQ(1)/RQ(1)
IF (RQ(2).LE.0.) S(2) = 2.*DUDR(2)
IF (RQ(2).GT.0.) S(2) = DUDR(2) + UQ(2)/RQ(2)
IF (RQ(3).LE.0.) S(3) = DUDR(3) + DWDZ(3)
IF (RQ(3).GT.0.) S(3) = DWCZ(3) + UQ(3)/RQ(3)
IF (RQ(4).LE.0.) S(4) = 2.*DUDR(4)
IF (RQ(4).GT.0.) S(4) = DUDR(4) + UQ(4)/RQ(4)
IF (RRT.LE.0.) S(5) = 2.*DUDRR + DWDZR
IF (RRT.GT.0.) S(5) = DUDRR + DWDZR + URT/RRT
GO TO (10,20,30,40,50,60,70,80,90,100,100,100,110,110,110,10,10),
1KMT
10 U(I,J,2) = (UQ(1) + UQ(3) - PQ(1) + PQ(3) + QT*(S(1)-S(3)))/2.

```

```

      W(I,J,2) = (WQ(2)+WQ(4)-PQ(2)+PQ(4)+QT*(S(2)-S(4)))/2.
      P(I,J,2) = (-UQ(1)+UQ(3)-WQ(2)+WQ(4)+PQ(1)+PQ(2)+PQ(3)+PQ(4)-
12.*PRT-QT*(S(1)+S(2)+S(3)+S(4)-2.*S(5)+2.*U(I,J,2)/R(I)))/2.
      GO TO 150
20  U(I,J,2) = 0.
      IF (IP6.EQ.1) GO TO 25
      W(I,J,2) = (WQ(2)+WQ(4)-PQ(2)+PQ(4)+QT*(S(2)-S(4)))/2.
      P(I,J,2) = 2.*(-UQ(1)-(WQ(2)-WQ(4))/4.+PQ(1)+(PQ(2)+PQ(4))/4.
X-PRT-QT*(S(1)+(S(2)+S(4))/4.-S(5)))
      GO TO 27
25  W(I,J,2) = W(I+1,J,2)
      P(I,J,2) = P(I+1,J,2)
27  GO TO 150
30  U(I,J,2) = 0.
      W(I,J,2) = (WQ(2)+WQ(4)-PQ(2)+PQ(4)+QT*(S(2)-S(4)))/2.
      P(I,J,2) = UQ(3)-(WQ(2)-WQ(4))/2.+PQ(3)+(PQ(2)+PQ(4))/2.-PRT-
1QT*(S(3)+(S(2)+S(4))/2.-S(5))
      GO TO 150
40  U(I,J,2) = (UQ(1)+UQ(3)-PQ(1)+PQ(3)+QT*(S(1)-S(3)))/2.
      W(I,J,2) = 0.
      P(I,J,2) = -(UQ(1)-UQ(3))/2.-WQ(2)+PQ(2)+(PQ(1)+PQ(3))/2.-PRT-
1QT*(S(2)+(S(1)+S(3))/2.-S(5)+U(I,J,2)/R(I))
      GO TO 150
50  U(I,J,2) = (UQ(1)+UQ(3)-PQ(1)+PQ(3)+QT*(S(1)-S(3)))/2.
      W(I,J,2) = 0.
      P(I,J,2) = -(UQ(1)-UQ(3))/2.+WQ(4)+PQ(4)+(PQ(1)+PQ(3))/2.-PRT-
1QT*(S(4)+(S(1)+S(3))/2.-S(5)+U(I,J,2)/R(I))
      GO TO 150
60  U(I,J,2) = 0.
      IF (IP6.EQ.1) GO TO 65
      W(I,J,2) = 0.
      P(I,J,2) = 2.*(-UQ(1)-WQ(2)/2.+PQ(1)+PQ(2)/2.-PRT-QT*(S(1)+
1S(2)/2.-S(5)))
      GO TO 67
65  W(I,J,2) = W(I+1,J,2)
      P(I,J,2) = P(I+1,J,2)
67  GO TO 150
70  U(I,J,2) = 0.
      IF (IP6.EQ.1) GO TO 75
      W(I,J,2) = 0.
      P(I,J,2) = 2.*(-UQ(1)+WQ(4)/2.+PQ(1)+PQ(4)/2.-PRT-QT*(S(1)+
1S(4)/2.-S(5)))
      GO TO 77
75  W(I,J,2) = W(I+1,J,2)
      P(I,J,2) = P(I+1,J,2)
77  GO TO 150
80  U(I,J,2) = 0.
      W(I,J,2) = 0.
      P(I,J,2) = -WQ(2)+UQ(3)+PQ(2)+PQ(3)-PRT-QT*(S(2)+S(3)-S(5))
      GO TO 150
90  U(I,J,2) = 0.
      W(I,J,2) = 0.
      P(I,J,2) = UQ(3)+WQ(4)+PQ(3)+PQ(4)-PRT-QT*(S(3)+S(4)-S(5))
      GO TO 150
100 U(I,J,2) = 0.
      W(I,J,2) = (WQ(2)+WA-PQ(2)+PA+QT*S(2)-FA)/2.
      P(I,J,2) = (-WQ(2)+WA+PQ(2)+PA-QT*S(2)-FA)/2.
      GO TO 150
110 U(I,J,2) = 0.
      W(I,J,2) = (WQ(4)+WB+PQ(4)-PR-QT*S(4)-FB)/2.
      P(I,J,2) = (WQ(4)+WB+PQ(4)+PR-QT*S(4)+FB)/2.
C *** RECALCULATE FOR SOURCE POINTS ( IP5=2 ) *****

```

```

150 IF (IP5.NE.2) GO TO 170
    IF (J.NE.JJ.AND.J.NE.JJM) GO TO 170
    U(I,J,2) = 0.
    P(I,J,2) = P(I,J,1)
    IF (J.EQ.JJM) GO TO 125
    W(I,J,2) = WQ(2) + P(I,J,2) - PQ(2)
    GO TO 170
125 W(I,J,2) = WQ(4) - P(I,J,2) + PQ(4)
170 IF (IP5.EQ.1) GO TO 310
    RETURN
C *** OPTION 1 : RIGHT HAND SIDE ASSUMED TO BE IN NEW PLANE *****
190 GO TO (210,220,230,240,250,260,270,280,290), KMT
210 U(I,J,2) = (UQ(1) + UQ(3) - PQ(1) + PQ(3))/2.
    W(I,J,2) = (WQ(2) + WQ(4) - PQ(2) + PQ(4))/2.
    P(I,J,2) = PQ(1) + PQ(2) - PRT + U(I,J,2)-UQ(1) + W(I,J,2) -
    1WQ(2) -2.*QT*U(I,J,2)/R(I)
    GO TO 300
220 U(I,J,2) = 0.
    IF (IP6.EQ.1) GO TO 225
    W(I,J,2) = (WQ(2) + WQ(4) - (PQ(2) - PQ(4)))/2.
    P(I,J,2) = 2.*PQ(1) + PQ(2) -2.*PRT + (-2.*UQ(1)+W(I,J,2)-WQ(2))
    GO TO 227
225 W(I,J,2) = W(I+1,J,2)
    P(I,J,2) = P(I+1,J,2)
227 GO TO 300
230 U(I,J,2) = 0.
    W(I,J,2) = (WQ(2) +WQ(4) - PQ(2) + PQ(4))/2.
    P(I,J,2) = PQ(2) + PQ(3) -PRT + (UQ(3) + W(I,J,2) - WQ(2))
    GO TO 300
240 U(I,J,2) = (UQ(1) + UQ(3) - PQ(1) + PQ(3))/2.
    W(I,J,2) = 0.
    P(I,J,2) = PQ(1) + PQ(2) - PRT + (U(I,J,2)-UQ(1)-WQ(2)-2.*QT*
    1U(I,J,2)/R(I))
    GO TO 300
250 U(I,J,2) = (UQ(1) + UQ(3) - PQ(1) + PQ(3))/2.
    W(I,J,2) = 0.
    P(I,J,2) = PQ(1) + PQ(4) - PRT + (U(I,J,2) - UQ(1) + WQ(4) -
    12.*QT*U(I,J,2)/R(I))
    GO TO 300
260 U(I,J,2) = 0.
    IF (IP6.EQ.1) GO TO 265
    W(I,J,2) = 0.
    P(I,J,2) = PQ(1) + PQ(2) - PRT - (UQ(1) + WQ(2))
    GO TO 267
265 W(I,J,2) = W(I+1,J,2)
    P(I,J,2) = P(I+1,J,2)
267 GO TO 300
270 U(I,J,2) = 0.
    IF (IP6.EQ.1) GO TO 275
    W(I,J,2) = 0.
    P(I,J,2) = PQ(1) + PQ(4) - PRT + (-UQ(1) + WQ(4))
    GO TO 277
275 W(I,J,2) = W(I+1,J,2)
    P(I,J,2) = P(I+1,J,2)
277 GO TO 300
280 U(I,J,2) = 0.
    W(I,J,2) = 0.
    P(I,J,2) = PQ(2) + PQ(3) - PRT + (UQ(3) - WQ(2))
    GO TO 300
290 U(I,J,2) = 0.
    W(I,J,2) = 0.
    P(I,J,2) = PQ(3) + PQ(4) - PRT + (UQ(3) + WQ(4))

```

```

300 IF (IP5.EQ.1) GO TO 310
    RETURN
C *** RECALCULATE FOUR SOURCE POINTS (IP5=1) -- VELOCITY COMPONENTS *****
C *** ARE CALCULATED ALONG NEAR CHARACTERISTICS FOR GIVEN SOURCE PRESSURES *****
310 IF (I.NE.II.OR.I.NE.IIP) RETURN
    IF (J.NE.JJ.OR.J.NE.JJP) RETURN
    IF (I.NE.1) GO TO 320
    U(I,J,2) = 0.
    P(I,J,2) = P(I,J,1)
    DELZ = C*DT
    IF (J.EQ.JJP) GO TO 312
    WC = W(I,J-1,1) + (W(I,J,1)-W(I,J-1,1))*(DZ-DELZ)/DZ
    DELZ = (WC+C)*DT
    WC = W(I,J-1,1) + (W(I,J,1)-W(I,J-1,1))*(DZ-DELZ)/DZ
    PC = P(I,J-1,1) + (P(I,J,1)-P(I,J-1,1))*(DZ-DELZ)/DZ
    DUDRC = (U(I+1,J-1,1)-U(I,J-1,1))/DR*DELZ/DZ + (U(I+1,J,1)
X-U(I,J,1))/DR*(DZ-DELZ)/DZ
    W(I,J,2) = WC - (P(I,J,2)-PC) - DT*2.*C*DUDRC
    RETURN
312 WC = W(I,J,1) + (W(I,J+1,1)-W(I,J,1))*DELZ/DZ
    DELZ = -(WC-C)*DT
    WC = W(I,J,1) + (W(I,J+1,1)-W(I,J,1))*DELZ/DZ
    PC = P(I,J,1) + (P(I,J+1,1)-P(I,J,1))*DELZ/DZ
    DUDRC = (U(I+1,J,1)-U(I,J,1))/DR*(DZ-DELZ)/DZ + (U(I+1,J+1,1)
X-U(I,J+1,1))/DR*DELZ/DZ
    W(I,J,2) = WC + (P(I,J,2)-PC) + DT*2.*C*DUDRC
    RETURN
320 P(I,J,2) = P(I,J,1)
    DELR = C*DT
    IF (I.EQ.IIP) GO TO 325
    UC = U(I-1,J,1) + (U(I,J,1)-U(I-1,J,1))*(DR-DELR)/DR
    DELR = (UC + C)*DT
    UC = U(I-1,J,1) + (U(I,J,1)-U(I-1,J,1))*(DR-DELR)/DR
    WC = W(I-1,J,1) + (W(I,J,1)-W(I-1,J,1))*(DR-DELR)/DR
    PC = P(I-1,J,1) + (P(I,J,1)-P(I-1,J,1))*(DR-DELR)/DR
    RC = R(I) - DELR
    CUDZC = (U(I-1,J+1,1)-U(I-1,J-1,1))/(2.*DZ)*DELR/DR + (U(I,J+1,1)
X-U(I,J-1,1))/(2.*DZ)*(DR-DELR)/DR
    DWDZC = (W(I-1,J+1,1)-W(I-1,J-1,1))/(2.*DZ)*DELR/DR + (W(I,J+1,1)
X-W(I,J-1,1))/(2.*DZ)*(DR-DELR)/DR
    DPDZC = (P(I-1,J+1,1)-P(I-1,J-1,1))/(2.*DZ)*DELR/DR + (P(I,J+1,1)
X-P(I,J-1,1))/(2.*DZ)*(DR-DELR)/DR
    U(I,J,2) = UC-(P(I,J,2)-PC)-DT*(WC*DUDZC+C*(DWDZC+UC/RC)+WC*DPDZC)
    GO TO 327
325 UC = U(I,J,1) + (U(I+1,J,1)-U(I,J,1))*DELR/DR
    DELR = -(UC-C)*DT
    UC = U(I,J,1) + (U(I+1,J,1)-U(I,J,1))*DELR/DR
    WC = W(I,J,1) + (W(I+1,J,1)-W(I,J,1))*DELR/DR
    PC = P(I,J,1) + (P(I+1,J,1)-P(I,J,1))*DELR/DR
    RC = R(I) + DELR
    DUDZC = (U(I,J+1,1)-U(I,J-1,1))/(2.*DZ)*(DR-DELR)/DR + (U(I+1,J+1,
X1)-U(I+1,J-1,1))/(2.*DZ)*DELR/DR
    DWDZC = (W(I,J+1,1)-W(I,J-1,1))/(2.*DZ)*(DR-DELR)/DR + (W(I+1,J+1,
X1)-W(I+1,J-1,1))/(2.*DZ)*DELR/DR
    DPDZC = (P(I,J+1,1)-P(I,J-1,1))/(2.*DZ)*(DR-DELR)/DR + (P(I+1,J+1,
X1)-P(I+1,J-1,1))/(2.*DZ)*DELR/DR
    U(I,J,2) = UC+(P(I,J,2)-PC)+DT*(-WC*DUDZC+C*(DWDZC+UC/RC)+WC*DPDZC
X)
327 DELZ = C*DT
    IF (J.EQ.JJP) GO TO 330
    WC = W(I,J-1,1) + (W(I,J,1)-W(I,J-1,1))*(DZ-DELZ)/DZ
    DELZ = (WC+C)*DT
    UC = U(I,J-1,1) + (U(I,J,1)-U(I,J-1,1))*(DZ-DELZ)/DZ

```

```

      WC = W(I,J-1,1) + (W(I,J,1)-W(I,J-1,1))*(DZ-DELZ)/DZ
      PC = P(I,J-1,1) + (P(I,J,1)-P(I,J-1,1))*(DZ-DELZ)/DZ
      DUDRC = (U(I+1,J-1,1)-U(I-1,J-1,1))/(2.*DR)*DELZ/DZ + (U(I+1,J,1)
X-U(I-1,J,1))/(2.*DR)*(DZ-DELZ)/DZ
      DWDRC = (W(I+1,J-1,1)-W(I-1,J-1,1))/(2.*DR)*DELZ/DZ + (W(I+1,J,1)
X-W(I-1,J,1))/(2.*DR)*(DZ-DELZ)/DZ
      DPDRC = (P(I+1,J-1,1)-P(I-1,J-1,1))/(2.*DR)*DELZ/DZ + (P(I+1,J,1)
X-P(I-1,J,1))/(2.*DR)*(DZ-DELZ)/DZ
      W(I,J,2) = WC-(P(I,J,2)-PC)-DT*(UC*DWDRC+C*(DUDRC+UC/R(I))+UC*CPDR
XC)
      GO TO 335
330 WC = W(I,J,1) + (W(I,J+1,1)-W(I,J,1))*DELZ/DZ
      DELZ = -(WC-C)*DT
      UC = U(I,J,1) + (U(I,J+1,1)-U(I,J,1))*DELZ/DZ
      WC = W(I,J,1)+(W(I,J+1,1)-W(I,J,1))*DELZ/DZ
      PC = P(I,J,1) + (P(I,J+1,1)-P(I,J,1))*DELZ/DZ
      DUDRC = (U(I+1,J,1)-U(I-1,J,1))/(2.*DR)*(DZ-DELZ)/DZ + (U(I+1,J+1,
X1)-U(I-1,J+1,1))/(2.*DR)*DELZ/DZ
      DWDRC = (W(I+1,J,1)-W(I-1,J,1))/(2.*DR)*(DZ-DELZ)/DZ + (W(I+1,J+1,
X1)-W(I-1,J+1,1))/(2.*DR)*DELZ/DZ
      DPDRC = (P(I+1,J,1)-P(I-1,J,1))/(2.*DR)*(DZ-DELZ)/DZ + (P(I+1,J+1,
X1)-P(I-1,J+1,1))/(2.*DR)*DELZ/DZ
      W(I,J,2) = WC + (P(I,J,2)-PC) + DT*(C*(DUDRC+UC/R(I))-UC*DWDRC +
XUC*DPDRC)
335 RETURN
      END

```

```

      SUBROUTINE PTME (PSRCE,NNH)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /R4/ DT,DR,DZ,DRDZ,T,TI,TF,DMN
      DIMENSION TIME(50),PTM(50)
      DATA NIN,NCUT/5,6/
C *** THIS SUBPROGRAM READS IN PRESSURE-TIME HISTORY AT ANY TIME *****
C *** BY LINEAR INTERPOLATION *****
      IF (T.GT.TI) GO TO 5
C *** INITIALIZE *****
      DO 2 N=1,50
        TIME(N) = 0.
      2 PTM(N) = 0.
C *** READ IN TIME IN SEC AND PRESSURE IN PSI *****
      READ(NIN,100)(TIME(I),I=1,NNH)
      READ(NIN,100)(PTM(I),I=1,NNH)
100 FORMAT(6E12.5)
      WRITE(NCUT,200)
200 FORMAT (////,26X,'----- INPUT PRESSURE - TIME HISTORY FOR TWO - D
XIMENSIONAL CALCULATION -----',///,52X,'TIME, SEC',6X,'PRESSURE, P
2SI',/)
      WRITE(NCUT,300)(TIME(I),PTM(I),I=1,NNH)
300 FORMAT(50X,E12.5,5X,E12.5)
      DO 10 L=1,NNH
        I=L-1
        IF(L.GE.NNH) GO TO 34
        IF(T.EQ.TIME(L)) GO TO 34
      10 IF(T.LT.TIME(L)) GO TO 32
        GO TO 34
      32 PSRCE = PTM(I)+(PTM(I+1)-PTM(I))*(T-TIME(I))/(TIME(I+1)-TIME(I))
        GO TO 35
      34 I=I+1
        PSRCE = PTM(I)
      35 RETURN
      END

```

```

SUBROUTINE CHART (NC,PSRCE)
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /R1/ IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,IP12,
  1IP13 /R2/ II,JJ,NR,NZ,NM,MM,NR1,NR2,NZ1,NZ2,NZP,NZQ,NZC /R3/ RR,
  2ZLEN,PR,C,RHC,AIMP,RR1,ZL1,RR2,ZL2 /R4/ DT,DR,DZ,DRDZ,T,TI,TF,
  3DMN /R5/ U(30,60,2),W(30,60,2),P(30,60,2),UMAX(30,60),
  4WMAX(30,60),PMAX(30,60),R(30),Z(60),SN(4),CSN(4)

C
  DIMENSION IK(30),JK(60)
  DATA NIN,NOUT/5,6/

C
C *** THIS SUBROUTINE PROVIDES PRINTED OUTPUT *****
  IF (T.GT.TI.OR.IP2.LE.0) GO TO 15
  DO 5 N=1,30
    IK(N) = 0
    JK(N) = 0
    READ (NIN,20) (IK(M),JK(M),M=1,IP2)
  20 FORMAT (7(2I5))
    WRITE (NCUT,22)
  22 FORMAT (1H1,////,35X,'----- POINTS AT WHICH RESULTS ARE PRINTED
  X-----',//)
    WRITE (NOUT,25) (IK(M),JK(M),M=1,IP2)
  25 FORMAT (50X,'I =',I5,' J =',I5)
  15 IF (T.EQ.TI) WRITE (NOUT,30)
  30 FORMAT (1H1,42X,'----- RESULT OF PRESSURE TRANSIENT CALCULATION
  1-----',//)

C *** CONVERT UNIT OF PRESSURE TO PSI *****
  IF (IP7.EQ.1) GO TO 36
  DO 35 J=1,NZ
    DO 35 I=1,NR
      U(I,J,1) = PMAX(I,J)*AIMP
  35 P(I,J,1) = P(I,J,2)*AIMP
      GO TO 40
  36 DO 37 J=1,NZ1
    DO 37 I=1,NR1
      U(I,J,1) = PMAX(I,J)*AIMP
  37 P(I,J,1) = P(I,J,2)*AIMP
      DO 38 J=NZP,NZQ
        DO 38 I=1,NR
          U(I,J,1) = PMAX(I,J)*AIMP
  38 P(I,J,1) = P(I,J,2)*AIMP
      DO 39 J=NZC,NZ2
        DO 39 I=1,NR2
          U(I,J,1) = PMAX(I,J)*AIMP
  39 P(I,J,1) = P(I,J,2)*AIMP
  40 IF (IP2.NE.0) GO TO 350

C *** RESULT IS PRINTED FOR ALL POINTS *****
  IF (IP7.EQ.1) GO TO 45
  M1 = 1
  M2 = NZ
  GO TO 48
  45 M1 = NZP
  M2 = NZQ
  48 IF (T.GT.TI) WRITE(NOUT,85)
  85 FORMAT (1F1)
      WRITE (NOUT,90) T,NC,PSRCE
  90 FORMAT (1X,'TIME =',1PE12.5,' SEC      ( NC =',I5,' STEPS ) : SCU
  1RCE PRESSURE =',1PE12.5,' PSI',/)
      IK1 = 1
      IK2 = 10
  50 IF (IK2-NR) 60,60,52
  52 IK2 = NR

```

```

60 WRITE (NOUT,100) (R(I),I=IK1,IK2)
100 FORMAT (16X,'R(I)',1X,1P10E11.3)
    WRITE (NOUT,105)
105 FORMAT (3X,'J',4X,'Z(J)',/)
    DO 250 J=M1,M2
        WRITE (NCUT,110) J,Z(J),(U(I,J,2),I=IK1,IK2)
110 FORMAT (2X,I2,1PE11.3,3X,'U',2X,1P10E11.3)
        WRITE (NOUT,120) (W(I,J,2),I=IK1,IK2)
120 FORMAT (18X,'W',2X,1P10E11.3)
        WRITE (NOUT,125) (P(I,J,1),I=IK1,IK2)
125 FORMAT (18X,'P',2X,1P10E11.3)
        IF (IP3.EQ.1.OR.IP3.EQ.4) WRITE (NOUT,130) (UMAX(I,J),I=IK1,IK2)
130 FORMAT (17X,'UMAX',1X,1P10E11.3)
        IF (IP3.EQ.2.OR.IP3.EQ.4) WRITE (NOUT,135) (WMAX(I,J),I=IK1,IK2)
135 FORMAT (17X,'WMAX',1X,1P10E11.3)
        IF (IP3.EQ.3.OR.IP3.EQ.4) WRITE (NOUT,140) (U(I,J,1),I=IK1,IK2)
140 FORMAT (17X,'PMAX',1X,1P10E11.3)
250 WRITE (NOUT,145)
145 FORMAT (1H )
    IF (IK2-NR) 55,57,57
55 IK1 = IK2+1
    IK2 = IK2+10
    WRITE (NOUT,160)
160 FORMAT (1H1)
    GO TO 50
57 IF (IP7.EQ.0.OR.(IP7.EQ.1.AND.IP11.EQ.0)) RETURN
C *** RESULT FOR EXTENDED REGION UPSTREAM AND DOWNSTREAM *****
    N2 = NR1
    M1 = 1
    M2 = NZ1
    INDTR = 0
550 JK1 = M1
    JK2 = M1 + 9
551 IF (JK2-M2) 560,560,552
552 JK2 = M2
560 WRITE (NOUT,600) (Z(J),J=JK1,JK2)
600 FORMAT (///,16X,'Z(L)',1X,1P10E11.3)
    WRITE (NCUT,605)
605 FORMAT (3X,'I',4X,'R(I)',/)
    DO 590 I=1,N2
        WRITE (NOUT,110) I,R(I),(U(I,J,2),J=JK1,JK2)
        WRITE (NOUT,120) (W(I,J,2),J=JK1,JK2)
        WRITE (NOUT,125) (P(I,J,1),J=JK1,JK2)
        IF (IP3.EQ.1.OR.IP3.EQ.4) WRITE (NOUT,130) (UMAX(I,J),J=JK1,JK2)
        IF (IP3.EQ.1.OR.IP3.EQ.4) WRITE (NOUT,135) (WMAX(I,J),J=JK1,JK2)
        IF (IP3.EQ.1.OR.IP3.EQ.4) WRITE (NOUT,140) (U(I,J,1),J=JK1,JK2)
590 WRITE (NOUT,145)
    IF (JK2-M2) 555,557,557
555 JK1 = JK2+1
    JK2 = JK2 + 10
    WRITE (NOUT,160)
    GO TO 551
557 IF (INDTR.GE.1) RETURN
    N2 = NR2
    M1 = NZC
    M2 = NZ2
    INDTR = INDTR + 1
    JK1 = M1
    JK2 = M1 + 9
    GO TO 551
C *** RESULT FOR SELECTED POINTS ONLY ARE PRINTED *****
350 IF (IP2.LT.0) GO TO 450

```

```

      WRITE (NOUT,400) T,NC,PSRCE
400  FORMAT (/ ,5X, 'TIME  =' ,1PE12.5, ' SEC  ( NC =' ,15, ' STEPS )  :  SCU
      XRCE PRESSURE =' ,1PE12.5, '// ,17X, 'LOCATION',12X, 'R, FT',12X, 'Z, FT',
      210X, 'U, FT/SEC',8X, 'W, FT/SEC',6X, 'PRESSURE, PSI' )
      DO 375 M=1,IP2
      I=IK(M)
      J = JK (M)
375  WRITE (NOUT,410) I,J,R(I),Z(J),U(I,J,2),W(I,J,2),P(I,J,1)
410  FORMAT (12X, 'I =' ,I3,4X, 'J =' ,I3,5X,1PE12.5,5X,1PE12.5,5X,1PE12.5,
      15X,1PE12.5,5X,1PE12.5)
      WRITE (NOUT,420)
420  FORMAT (1H )
      IF (IP13.NE.1) RETURN
      WRITE (10,430) T,NC,PSRCE
430  FORMAT (E12.5,I8,E12.5)
      DO 435 M=1,IP2
      I = IK(M)
      J = JK(M)
435  WRITE (10,440) U(I,J,2),W(I,J,2),P(I,J,1)
440  FORMAT (3E12.5)
      RETURN
450  IF (T.GT.TI) GO TO 451
      TM1 = 0.0005
      TM2 = 0.00075
      TMM1 = TM1 - DT/2.
      TMP1 = TM1 + DT/2.
      TMM2 = TM2 - DT/2.
      TMP2 = TM2 + DT/2.
451  IF (T.GE.TMM1.AND.T.LT.TMP1) GO TO 461
      IF (T.GE.TMM2.AND.T.LT.TMP2) GO TO 461
      GO TO 470
460  FORMAT (E12.5)
480  FORMAT (/)
461  WRITE (10,430) T,NC,PSRCE
      WRITE (10,480)
      DO 462 J=1,NZ2
462  WRITE (10,460) P(1,J,1)
      WRITE (10,480)
      DO 463 J=1,NZ2
463  WRITE (10,460) P(NR1,J,1)
      WRITE (10,480)
      DO 464 J=NZP,NZC
464  WRITE (10,460) P(NR,J,1)
      WRITE (10,480)
470  RETURN
      END

```

```

SUBROUTINE PTRAN (NPTR,JIA,JIB,FTRA,WA,PPA,FTRB,WB,PPB,IP10)
IMPLICIT REAL*8 (A-H,C-Z)

```

```

C
C *****
C *                                     PROGRAM SODIUM TRANSIENT                                     *
C *   THE 'METHOD OF CHARACTERISTICS' IS USED FOR PRESSURE TRANSIENT ANALYSIS IN HYDRAULIC NETWORKS. ZAKER AND SALMON ANALYSIS *
C *   CALCULATES SOURCE PRESSURE. THE PROGRAM CONSIDERS COMPLETE EQUATIONS INCLUDING NONLINEAR CONVECTIVE AND VISCOUS TERMS. *
C *   YONG WOO SHIN : SEPTEMBER 1, 1973 *
C *****
C
C MAIN PROGRAM
COMMON /S1/ N1(50),NN(50),N(50) /S5/ RHO,AMU,DT,PR,UR,LEGNC,NP

```



```

X/S2/ A(50),C(50),DZ(50),D(50),P(50,120,2),U(50,120,2),RF(50)
C
  DIMENSION LENGTH(50),NJ(50),PMA(50),PT(10),IT(10),H(50),ES(50),
XINT(50),TMAX(50),PG(50),UO(50)
C
  DATA CL0/59.566/,DL1/-0.79504E-02/,DL2/-0.2872E-06/,DL3/ 0.60350
XE-10/,CL1/8285./,CL2/-2187./,CL3/ 2290./,JMAX/50/,JMIN/1/,LMAX/50/
2,LMIN/1/,PMA,TMAX/100*0./,GC/32.173/,NIN,NCUT,I1,I2/5,6,1,2/,
3PT/10*0./,IT/10*0/
C ***** PARTIAL LISTING OF INPUT DATA *****
C * LEGNC - NO OF BRANCHES *
C * JUNNO - NO OF JUNCTIONS *
C * NOT - NO OF SURGE TANK AND RESERVOIR *
C * NOPRIN - FREQUENCY OF PRINTOUT *
C * KS - PARAMETER SPECIFYING DETAIL CF PRINTOUT *
C * JJ - BRANCH IN WHICH LEAK SITE IS LOCATED *
C * JK - BRANCH THAT IS TREATED AS A 2-D DOMAIN *
C * IC - IF IC = 1, INITIAL CONDITIONS ARE INPUT *
C * INDIVIDUALLY *
C * NP - IF NP = 1, INVISCID ANALYSIS *
C * DT - TIME STEP IN SECOND *
C * UR - INITIAL VELOCITY *
C * PR - INITIAL PRESSURE *
C * PCR - CRITICAL PRESSURE AT OR ABOVE WHICH RUPTURE *
C * DISK BREAKS *
C * TR - SYSTEM TEMPERATURE IN FAHRENHEIT *
C * XI - INITIAL TIME AT WHICH CALCULATION STARTS *
C * XF - FINAL TIME AT WHICH CALCULATION TERMINATES *
C * INDTR - BRANCH NUMBERS OF WHICH RESULTS ARE PRINTED *
C * IF INDTR = 0, ALL RESULTS ARE PRINTED *
C * NOTT - NO OF RUPTURE DISKS *
C * IP1 - IF 0, SOURCE PRESSURE BY SODIUM-WATER REACTION *
C * IF 1, PRESSURE - TIME HISTORY IS INPUT *
C * IF 2, PRESSURE SOURCE IS ABSENT *
C * IP2 - IF 0, SOURCE PRESSURE ASSIGNED AT ALL NODES OF BRANCH JJ *
C * IF 1, SOURCE PRESSURE GIVEN AT INTERFACE ON NODE #1 SIDE *
C * IF 2, SOURCE PRESSURE GIVEN AT INTERFACE ON NODE #N SIDE *
C * IF 3, SOURCE PRESSURE GIVEN AT NODE #1 ONLY *
C * IF 4, SOURCE PRESSURE GIVEN AT NODE #N ONLY *
C * IP3 - IF 1, SONIC SPEED AND DENSITY ARE INPUT *
C * NNK - NUMBER OF ENTRIES FOR PRESSURE - TIME HISTORY INPUT *
C * PARM - PARAMETER ENSURING NUMERICAL STABILITY - MUST BE .GT. 1 *
C *****
EXP(X) = DEXP(X)
ALOG10(X) = DLOG10(X)
ABS(X) = DABS(X)
AMAX1(X,Y) = DMAX1(X,Y)
SQRT(X) = DSQRT(X)
REAL*8 LENGTH
IF (NPTR.EQ.0) GO TO 30
IF (NPTR.EQ.1) GO TO 500
IF (NPTR.EQ.2) GO TO 220
30 NC=0
C**** READ INITIAL AND JUNCTION CONDITIONS OF EACH BRANCH *****
READ (NIN,1) LEGNO,JUNNO,NOT,NOPRIN,KS,JJ,IC,NP,DT,UR,PR,PCR,TR,
XXI,XF,PARM
READ (NIN,1210) INDTR,NOTT,IP1,NNK,JK,IP2,IP3
IF (INDTR.GT.0) READ (NIN,1215) (INT(NQ),NQ=1,INDTR)
IF (LEGNC.GT.LMAX.OR.LEGNC.LT.LMIN) GO TO 401
IF (JUNNO.GT.JMAX.OR.JUNNO.LT.JMIN) GO TO 402
READ (NIN,2) (N1(L),NN(L),A(L),LENGTH(L),RF(L),H(L),ES(L),D(L),
XPO(L),UC(L),L=1,LEGNO)

```

```

      READ (NIN,3) (NJ(J),J=1,JUNNO)
C**** READ CONDITIONS FOR SURGE TANK OR RESERVOIR *****
      IF (NOT.GE.1) READ(NIN,5) (PT(I),IT(I),I=1,ACT)
      NNTT=NOT+NOTT
      NOT1=NOT+1
C**** READ CONDITIONS FOR RUPTURE DISK *****
      IF (NOTT.GE.1) READ (NIN,5) (PT(I),IT(I),I=NOT1,NNTT)
      TT = XI
C**** CALCULATE DENSITY, VISCOSITY AND SONIC SPEED *****
      RHO=DL0+DL1*TR+DL2*TR**2+DL3*TR**3
      CR=CL1+CL2*(TR-210.)/CL3
      AMU=EXP(2.303*(1.0203+397.17/(TR+460.))-0.4925*ALOG10(TR+460.)))/
X3600.
C**** SONIC SPEED ACCOUNTING FOR PIPE EXPANSION *****
      DO 2002 L=1,LEGNC
      ESS=ES(L)*A(L)/(3.14159265*D(L)**2./4.)
2002 C(L)=CR/SQRT(1.+RHO*CR**2*D(L)/ESS/H(L)/(144.*GC))
      IF (IP3.NE.1) GO TO 203
      READ (NIN,1350) RHO,CS
      DO 2004 L=1,LEGNC
2004 C(L) = CS
C**** DETERMINE DZ(L) AND N(L) IN EACH BRANCH *****
203 DO 204 L=1,LEGNO
      ZM = LENGTH(L)/(C(L)*DT)/PARM
      NM=ZM
      IF (NM.EQ.0) NM=1
      N(L)=NM+1
      IF (N(L).GT.120) GO TO 205
204 DZ(L) = PARM*C(L)*DT
      GO TO 206
C**** RECALCULATE DT TO SATISFY N(L) .LT. 120 *****
205 DTT=2.*DT
      WRITE (NOUT,106) DT,DTT
      DT=DTT
      GO TO 203
C**** INITIAL VALUES OF UO, PO, U AND P *****
206 DO 207 L=1,LEGNO
C**** IF IC = 1, UO AND PO ARE GIVEN INDIVIDUALLY *****
      IF (IC.EQ.1) GO TO 202
      PO(L)=PR
      UO(L)=UR
202 DO 207 K=1,2
      DO 207 I=1,120
      P(L,I,K)=PO(L)
207 U(L,I,K)=UO(L)
      DO 50 L=1,LEGNO
      IF (N1(L).EQ.NN(JJ)) LLN=L
      IF(NN(L).EQ.N1(JJ)) LL1=L
50 CONTINUE
      IF (IP2.EQ.1) LLN = JJ
      IF (IP2.EQ.2) LL1 = JJ
      IF (IP2.EQ.1.CR.IP2.EQ.2) GO TO 54
      NV1=N(LL1)
      NV2=1
      GO TO 57
54 NV1 = N(LL1) - 1
      NV2 = 2
      NKL = N(LL1)
57 TD = XI
C**** PRINT INITIAL CONDITIONS AND DESCRIPTION OF NETWORK ARRANGEMENT *****
      WRITE (NOUT,4)
      WRITE (NOUT,6) LEGNC,JUNNG,ACT,NOTT,JJ,IC,NP,INDTR,NOPRIN,KS,JK,

```

```

XNNK,IP1,IP2,IP3
WRITE (NOUT,1311) UR,PR,PCR,TR,RHO,AMU,CT,XI,XF,PARM
IF (NOT.GE.1) WRITE (NOUT,1313)
IF (NOT.GE.1) WRITE (NOUT,9) (I,PT(I),IT(I),I=1,NCT)
IF (NOT.GE.1) WRITE (NOUT,1315)
IF (NOTT.GE.1) WRITE (NOUT,1312)
IF (NOTT.GE.1) WRITE (NOUT,1314) (I,PT(I),IT(I),I=NOT1,NNTT)
WRITE (NOUT,7)
DO 211 L=1,LEGNO
LN1=N1(L)
LNN=NN(L)
211 WRITE (NOUT,8) (L,LN1,NJ(LN1),LNN,NJ(LNN),N(L),DZ(L),LENGTH(L),
XA(L))
WRITE (NOUT,11)
WRITE (NCUT,12) (L,D(L),H(L),ES(L),C(L),PO(L),UO(L),L=1,LEGNO)
C**** CONVERT INCHES TO FEET *****
PCR=PCR*144.
DO 215 L=1,LEGNO
PO(L)=PO(L)*144.
A(L)=A(L)/144.
D(L)=D(L)/12.
RF(L)=RF(L)/12.
DO 215 K=1,2
DO 215 I=1,120
215 P(L,I,K)=P(L,I,K)*144.
DO 216 K=1,10
216 PT(K)=PT(K)*144.
C**** DETERMINE MAXIMUM PRESSURE AND VELOCITY IN EACH BRANCH *****
FR1=1.0
FR2=1.2
NTPP = NCT + NOTT
IF (IP10.EQ.1) GO TO 220
IF (NPTR.EQ.0) GO TO 500
220 IF (TT.LT.0.0) GO TO 250
UMX=0.
CUMX=0.
DO 213 L=1,LEGNO
NK=N(L)
PPX=PMAX(L)
DO 212 I=1,NK
UMX=AMAX1(UMX,ABS(U(L,I,2)))
212 PMAX(L)=AMAX1(PMAX(L),P(L,I,2))
IF (PMAX(L).GT.PPX) TMAX(L)=TT
CUP=1.+UMX/C(L)
213 CUMX=AMAX1(CUP,CUMX)
C**** REDETERMINE DT FOR NUMERICAL STABILITY *****
IF (CUMX.GE.FR1.AND.CUMX.LT.FR2) GO TO 226
FT1=FR2
FT2=FT1+0.2
IF (CUMX.GT.FT2) GO TO 222
CT=DT*FT1/FT2
FR1=FT1
FR2=FT2
WRITE (NCUT,1225) DT
GO TO 226
222 WRITE (NCUT,1230) CUMX
STOP
226 IF (TT.EQ.XI.AND.IP1.EQ.1) CALL PTIME (NNK,TT,PP)
IF (TT.EQ.XI.AND.IP1.EQ.0) PP = PR
C**** PRINT PRESSURE AND VELOCITY AT SPECIFIED TIME STEP *****
IF (TT.LT.TD) GO TO 250
IF (IP10.EQ.1.AND.TT.EQ.0.) WRITE (NOUT,13)

```

```

      TD=TD+NOPRIN*DT
      IF (TT.LE.0.0) TTT=TT
      IF (TT.GT.0.0) TTT=TT+2.0E-9
      IF (IP10.EQ.1) WRITE (NOUT,16) TTT,NC
      WRITE (NCUT,15)
      DO 242 L=1,LEGNO
      IF (IP10.NE.1.AND.L.EQ.JK) GO TO 242
      IF (INDTR.EQ.0) GO TO 1105
      DO 1101 LL=1,INDTR
1101  IF(L.EQ.INT(LL)) GO TO 1105
      GO TO 242
1105  WRITE (NOUT,17) L
      NK=N(L)
      DO 240 I=1,NK,KS
      IF (I.GE.NK) GO TO 241
      Z=(I-1)*DZ(L)
      PPP=P(L,I,2)/144.
230  WRITE (NCUT,19) Z,PPP,U(L,I,2)
240  CONTINUE
241  Z=LENGTH(L)
      I=NK
      PPP=P(L,I,2)/144.
      WRITE (NOUT,19) Z,PPP,U(L,I,2)
      IF (IP1.NE.2.AND.L.EQ.JJ) WRITE (NOUT,265) PP
242  CONTINUE
250  TT=TT+DT
      IF (TT.GT.XF) GO TO 999
      DO 300 L=1,LEGNO
      NK=N(L)
      DO 300 I=1,NK
      U(L,I,1)=U(L,I,2)
300  P(L,I,1)=P(L,I,2)
      IF (IP1.EQ.2) GO TO 270
C *** SOURCE PRESSURE CALCULATION *****
      IF (IP1.EQ.1) GO TO 268
C *** SODIUM - WATER REACTION *****
      V1 = -U(LL1,NV1,1)*12.
      V2 = U(LLN,NV2,1)*12.
      A1 = A(LL1)*144.
      A2 = A(LLN)*144.
      CALL SOURCE (PP,TT,V1,V2,C(LL1),C(LLN),A1,A2)
      PPP = PP*144.
      GO TO 303
C *** INPUT PRESSURE - TIME *****
268  CALL PTIME (NNK,TT,PP)
      PPP = PP*144.
303  NK = N(JJ)
      IF (IP2.EQ.1) GO TO 443
      IF (IP2.EQ.2) GO TO 445
      IF (IP2.EQ.3) GO TO 447
      IF (IP2.EQ.4) GO TO 449
      DO 306 I=1,NK
306  P(JJ,I,1) = PPP
      GO TO 270
443  P(JJ,1,1) = PPP
      P(LL1,NKL,1) = PPP
      GO TO 270
445  P(JJ,NK,1) = PPP
      P(LLN,1,1) = PPP
      GO TO 270
447  P(JJ,1,1) = PPP
      GO TO 270

```

```

449 P(JJ,NK,1) = PPP
C**** INTERIOR NODE CALCULATION AT EACH TIME STEP *****
270 DO 309 L=1,LEGNC
    IF (IP1C.NE.1.AND.L.EQ.JK) GO TO 309
    NK=N(L)
    NKM=NK-1
    IF (NK.LE.2) GO TO 309
    DO 302 I=2,NKM
        CALL INTER (L,I,I1,UA,PA)
        CALL FACTCR (L,UA,FA)
        CALL INTER (L,I,I2,UB,PB)
        CALL FACTOR (L,UB,FB)
        U(L,I,2)=0.5*(UA+UB+GC*(PA-PB)/(RHO*C(L))-DT/(2.*D(L))*
        X(UA*ABS(UA)*FA+UB*ABS(UB)*FB))
302 P(L,I,2)=0.5*(PA+PB+RHO*C(L)*(UA-UB)/GC-RHO*C(L)*DT/(2.*D(L)*GC)*
    X(UA*ABS(UA)*FA-UB*ABS(UB)*FB))
309 CONTINUE
C**** BOUNDARY NODE CALCULATION AT EACH TIME STEP *****
DO 320 J=1,JUNNC
    IF (IP10.EQ.1) GO TO 509
    IF (J.EQ.N1(JK).OR.J.EQ.NN(JK)) GO TO 320
509 JC = NJ(J)
    IF ((IP2.EQ.1.AND.J.EQ.N1(JJ)).OR.(IP2.EQ.3.AND.J.EQ.NN(JJ)))
    XGO TO 3010
    IF ((IP2.EQ.2.AND.J.EQ.NN(JJ)).OR.(IP2.EQ.4.AND.J.EQ.NN(JJ)))
    XGO TO 3030
    IF (JC.EQ.3) GO TO 313
    DO 310 L=1,LEGNO
        IF (N1(L).EQ.J) J1=L
        IF (NN(L).NE.J) GO TO 310
    JN=L
    I=N(L)
310 CONTINUE
    IF (JC.EQ.7.OR.JC.EQ.8.OR.JC.EQ.9.OR.JC.EQ.10) GO TO 315
    IF (JC.EQ.6) GO TO 314
C**** SUDDEN EXPANSION OR CONTRACTION *****
    IF (NN(JN).NE.N1(J1)) GO TO 404
    JNL=N(JN)
    CALL PIPES (J1,JN,I)
    GO TO 320
313 CALL TEE (J,JJ,IC)
    GO TO 320
C**** DUMMY JUNCTION *****
314 CALL DUMMY (J1,JN,I,JJ)
    GO TO 320
315 IF (NOT.EQ.0) GO TO 325
    DO 311 K=1,NOT
311 IF (IT(K).EQ.J) GO TO 325
325 DO 316 L=1,LEGNO
    IF (N1(L).EQ.J) GO TO 317
316 IF (NN(L).EQ.J) GO TO 318
    GO TO 320
317 IF (JC.EQ.7.OR.JC.EQ.9.OR.JC.EQ.10) GO TO 328
C**** SURGE TANK OR RESERVOIR *****
    P(L,1,2)=PT(K)
    CALL INTER (L,I1,I2,UB,PB)
    CALL FACTOR (L,UB,FB)
    U(L,1,2)=UB+GC*(PT(K)-PB)/(RHO*C(L))-FB*UB*ABS(UB)*DT/(2.*D(L))
    GO TO 320
318 IF (JC.EQ.7.OR.JC.EQ.9.OR.JC.EQ.10) GO TO 329
C**** SURGE TANK OR RESERVOIR *****
    P(L,I,2)=PT(K)

```

```

      CALL INTER (L,I,I1,UA,PA)
      CALL FACTOR (L,UA,FA)
      U(L,I,2)=UA-GC*(PT(K)-PA)/(RHO*C(L))-FA*UA*ABS(UA)*DT/(2.*D(L))
      GO TO 320
328 IF(JC.EQ.9) GO TO 319
C**** CLOSED END *****
      CALL INTER (L,I1,I2,UB,PB)
      CALL FACTOR (L,UB,FB)
      U(L,1,2)=0.
      P(L,1,2)=PB+RHO*C(L)/GC*(-UB+FB*UB*ABS(UB)*DT/(2.*D(L)))
      GO TO 320
C**** FAR END *****
319 U(L,1,2)=UO(L)-((P(L,2,1)-PC(L))*GC/(RHO*C(L))-U(L,2,1)+UO(L))/2.
      P(L,1,2)=PO(L)+(P(L,2,1)-PO(L)-RHO*C(L)*(U(L,2,1)-UO(L))/GC)/2.
      GO TO 320
329 IF(JC.EQ.9) GO TO 322
C**** CLOSED END *****
      CALL INTER (L,I,I1,UA,PA)
      CALL FACTOR (L,UA,FA)
      U(L,I,2)=0.
      P(L,I,2)=PA+RHO*C(L)/GC*(UA-FA*UA*ABS(UA)*DT/(2.*D(L)))
      GO TO 320
C**** FAR END *****
322 U(L,I,2)=UO(L)+((P(L,I-1,1)-PO(L))*GC/(RHO*C(L))+U(L,I-1,1)-
      XUO(L))/2.
      P(L,I,2)=PO(L)+(P(L,I-1,1)-PO(L)+RHO*C(L)*(U(L,I-1,1)-UO(L))/GC)
      X/2.
      GO TO 320
3010 CALL INTER (JJ,I1,I2,UB,PB)
      CALL FACTOR (JJ,UB,FB)
      U(JJ,1,2) = UB + GC*(P(JJ,1,1)-PB)/(RHO*C(JJ)) - FB*UB*ABS(UB)*
      XDT/(2.*C(JJ))
      IF (IP2.EQ.3) GO TO 320
      NK1 = N(LL1)
      CALL INTER (LL1,NK1,I1,UA,PA)
      CALL FACTOR (LL1,UA,FA)
      U(LL1,NK1,2) = UA - GC*(P(LL1,NK1,1)-PA)/(RHO*C(LL1)) - FA*UA*
      XABS(UA)*DT/(2.*D(LL1))
      GO TO 320
3030 NK2 = N(JJ)
      CALL INTER (JJ,NK2,I1,UA,PA)
      CALL FACTOR (JJ,UA,FA)
      U(JJ,NK2,2) = UA - GC*(P(JJ,NK2,1)-PA)/(RHO*C(JJ)) - FA*UA*ABS(UA)
      X*DT/(2.*D(JJ))
      IF (IP2.EQ.4) GO TO 320
      CALL INTER (LLN,I1,I2,UB,PB)
      CALL FACTOR (LLN,UB,FB)
      U(LLN,1,2) = UB + GC*(P(LLN,1,1)-PB)/(RHO*C(LLN)) - FB*UB*ABS(UB)*
      XDT/(2.*C(LLN))
320 CONTINUE
      DO 339 IL=1,LENO
      DO 339 JL=1,120
      IF (ABS(U(IL,JL,2)).LE.1.0E-6) U(IL,JL,2) = 0.
339 CONTINUE
      IF (IP10.EQ.1) GO TO 345
C**** CALCULATE THE BOUNDARY VALUES AT THE INTERFACE *****
      NKA = N(JIA)
      CALL INTER (JIA,NKA,I1,UA,PA)
      CALL FACTOR (JIA,UA,FA)
      PA = PA*GC/(RHO*C(JIA))
      U(JIA,NKA,2) = (WA+UA-PPA+PA+FTRA-FA*UA*ABS(UA)*DT/(2.*D(JIA)))/2.
      P(JIA,NKA,2) = (-WA+UA+PPA+PA-FTRA-FA*UA*ABS(UA)*DT/(2.*D(JIA)))

```

```

X/2.*RHO*C(JIA)/GC
CALL INTER (JIB,1,I2,UB,PB)
CALL FACTOR (JIB,UB,FB)
PB = PB*GC/(RHO*C(JIB))
U(JIB,1,2) = (WB+UB+PPB-PB-FTRB-FB*UB*ABS(UB)*DT/(2.*D(JIB)))/2.
P(JIB,1,2) = (WB-UB+PPB+PB-FTRB+FB*UB*ABS(UB)*DT/(2.*D(JIB)))/2.
X*RHO*C(JIB)/GC
345 IF (NOT.EQ.NTPP) GO TO 390
DO 375 J=1,JUNNO
IF (NJ(J).NE.10) GO TO 375
DO 355 L=1,LEGNO
IF (N1(L).EQ.J) GO TO 357
355 IF (NN(L).EQ.J) GO TO 358
357 IF (P(L,1,2).GE.PCR) NJ(J)=8
IF (P(L,1,2).GE.PCR) NOT=NOT+1
GO TO 375
358 NLL=N(L)
IF (P(L,NLL,2).GE.PCR) NJ(J)=8
IF (P(L,NLL,2).GE.PCR) NOT=NCT+1
375 CONTINUE
390 NC=NC+1
RETURN
C**** DIAGNOSTIC MESSAGE PRINTS *****
401 WRITE (NOUT,101) LMIN,LMAX
GO TO 999
402 WRITE (NCUT,102) JMIN,JMAX
GO TO 999
404 WRITE (NOUT,108)
999 WRITE (NOUT,105)
WRITE (NOUT,110)
DO 998 L=1,LEGNO
PMX=PMAX(L)/144.
998 PRINT 109,L,PMX,TMAX(L)
IF (IP10.NE.1) RETURN
STOP
C**** DATA FOR CONNECTED 2-D STEADY-STATE CALCULATION *****
500 NKA = N(JIA)
CALL INTER (JIA,NKA,I1,WA,PA)
CALL FACTOR (JIA,WA,FA)
PPA = PA*GC/(RHO*C(JIA))
FTRA = FA*WA*ABS(WA)*DT/(2.*D(JIA))
CALL INTER (JIB,1,I2,WB,PB)
CALL FACTOR (JIB,WB,FB)
PPB = PB*GC/(RHO*C(JIB))
FTRB = FB*WB*ABS(WB)*DT/(2.*D(JIB))
RETURN
C**** FORMAT STATEMENTS *****
1 FORMAT (8I5,/,6E12.5,/,2E12.5)
2 FORMAT (2I3,6E12.5,/,2E12.5)
3 FORMAT(40I2)
4 FORMAT (1H1,/,5X,'----- INITIAL CONDITION AND DESCRIPTION OF NE
XTWORK ARRANGEMENT FOR ONE - DIMENSIONAL DOMAIN CALCULATION -----
2',///)
5 FORMAT (4(E12.5,I5))
6 FORMAT (5X,'LEGNC' =',I5,7X,'JUNNO' =',I5,7X,'NOT' =',I5,7X,
X'NOTT' =',I5,7X,'JJ' =',I5,7X,'IC' =',I5,/,5X,'NP' =',
2I5,7X,'INDTR' =',I5,7X,'NOPRIN' =',I5,7X,'KS' =',I5,7X,
3'JK' =',I5,7X,'NNK' =',I5,/,5X,'IP1' =',I5,7X,
4'IP2' =',I5,7X,'IP3' =',I5,///)
7 FORMAT (1H1,/,9X,'BRANCH',14X,'JUNCTION TYPE',14X,'NO',12X,
X'INTERNODAL',12X,'LENGTH',15X,'AREA',/,11X,'NO',12X,'N = 1',11X,
2'N = N',7X,'OF NODES',10X,'DISTANCE',14X,'FEET',14X,'SQR INCH',/)

```

```

8 FORMAT (5X,I4,9X,I4,'(' ,I2,')' ,8X,I4,'(' ,I2,')' ,8X,I4,7X,
XE15.4,5X,E15.4,5X,E15.4)
9 FORMAT (46X,I5,5X,E12.5,6X,I5)
11 FORMAT (1H1,/,11X,'BRANCH',12X,'DIAMETER',8X,'THICKNESS',6X,
X'YOUNGS MODULUS',5X,'SONIC SPEED',7X,'PRESSURE',9X,'VELOCITY',
2/,13X,'NO',16X,'INCH',13X,'INCH',14X,'PSI',12X,'FT/SEC',13X,
3'PSI',12X,'FT/SEC',/)
12 FORMAT (11X,I4,8X,E15.4,2X,E15.4,2X,E15.4,2X,E15.4,2X,E15.4,
X2X,E15.4)
13 FORMAT (1H1,/,40X,'----- RESULT OF PRESSURE TRANSIENT CALCULATI
XON -----',/)
15 FORMAT (47X,'DISTANCE, FT',12X,'PRESSURE, PSI',11X,
X'VELOCITY, FT/SEC')
16 FORMAT (/,9X,'TIME =',E15.5,' SEC',5X,'( NC =',I4,' STEPS )',/)
17 FORMAT (20X,'BRANCH NO =', I5)
19 FORMAT (44X,E15.6,10X,E15.6,10X,E15.6)
101 FORMAT (/,10X,'ERROR IN NUMBER OF BRANCHES ----- LEGNO .LT.',
X,I5,' OR LEGNO .GT.',I5)
102 FORMAT (/,10X,'ERROR IN NUMBER OF JUNCTIONS ----- JUNNO .LT.',
X,I5,' OR JUNNO .GT.',I5)
105 FORMAT (/,45X,'----- END OF CALCULATION -----')
106 FORMAT (' CASE OF N(J).GT.120 ..... RESET DT=',E12.5,' TO DT=',
XE12.5)
108 FORMAT (/,2X,' ERROR IN SUBROUTINE PIPES ..... NN.NE.N1' )
109 FORMAT (25X,'MAXIMUM PRESSURE IN BRANCH',I4,' = ',E12.5,
X' PSI OCCURRED AT ',E12.5,' SEC')
110 FORMAT (1H1,/,30X,'----- MAXIMUM PRESSURES IN EACH BRANCH DURIN
XG CALCULATION -----',///)
265 FORMAT (44X,'SOURCE PRESSURE IS',7X,E15.6)
1210 FORMAT (7I5)
1215 FORMAT (16I5)
1225 FORMAT (5X,/, '***** DT IS DECREASED TO ',E12.5,' *****',/)
1230 FORMAT (5X,/, '***** CUMX IS OUTSIDE RANGE .EQ.',E12.5,' *****',/)
1311 FORMAT (5X,'UR = ',E12.5,5X,'PR = ',E12.5,5X,'PCR = ',E12.5,5X,
X'TR = ',E12.5,5X,'RHC = ',E12.5,5X,/,5X,'AMU = ',E12.5,5X,
2'DT = ',E12.5,5X,'XI = ',E12.5,5X,'XF = ',E12.5,5X,'PARM = ',
3E12.5,///)
1312 FORMAT (53X,'--- RUPTURE DISKS ---',/,47X,'DISK NO',2X,
X'GAS PRESSURE',4X,'JUNCTION NO')
1313 FORMAT (48X,'--- SURGE TANKS AND RESERVOIRS ---',/,42X,
X'TANK NC',2X,'GAS PRESSURE',4X,'JUNCTION NO',/)
1314 FORMAT (46X,I5,5X,E12.5,6X,I5)
1315 FORMAT (///)
1350 FORMAT (2E12.5)
END

```

```

SUBROUTINE INTER (L, LN, IN, UC, PC)
IMPLICIT REAL*8 (A-H, C-Z)
COMMON /S5/ RHO, AMU, DT, PR, UR, LEGNO, NP /S2/ A(50), C(50), CZ(50),
XD(50), P(50, 120, 2), U(50, 120, 2), RF(50)
C**** THIS SUBPROGRAM PERFORMS INTERPOLATION FOR PRESSURE AND VELOCITY*****
THTA=DT/DZ(L)
IF(IN.EQ.2) GO TO 10
UC=(U(L, LN, 1)-C(L)*THTA*(U(L, LN, 1)-U(L, LN-1, 1)))/(1.+THTA*
X(U(L, LN, 1)-U(L, LN-1, 1)))
PC=P(L, LN, 1)-THTA*(C(L)+UC)*(P(L, LN, 1)-P(L, LN-1, 1))
RETURN
10 LC=(U(L, LN, 1)+C(L)*THTA*(U(L, LN+1, 1)-U(L, LN, 1)))/(1.+THTA*
X(U(L, LN+1, 1)-U(L, LN, 1)))
PC=P(L, LN, 1)+THTA*(C(L)-UC)*(P(L, LN+1, 1)-P(L, LN, 1))
RETURN
END

```



```

SUBROUTINE DUMMY (J1,JN,I,JJ)
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /S5/ RHO,AMU,DT,PR,UR,LEGNO,NP /S2/ A(50),C(50),DZ(50),
  XC(50),P(50,120,2),U(50,120,2),RF(50)
  DATA GC/32.173/,I1,I2/1,2/
  ABS(X) = DABS(X)

```

```

C**** THIS SUBPROGRAM DESCRIBES CUMMY JUNCTION AND ACOUSTIC IMPEDANCE *****
C**** DISCONTINUITY *****
  CFF=RHO/(1./C(JN)+1./C(J1))
  CALL INTER (JN,I,I1,UA,PA)
  CALL FACTOR (JN,UA,FA)
  CALL INTER (J1,I1,I2,UB,PB)
  CALL FACTOR (J1,UB,FB)
  P(JN,I,2)=((PA/C(JN)+PB/C(J1))/RHO+(UA-UB)/GC-FA*UA*ABS(UA)*DT/
  X(2.*GC*C(JN))+FB*UB*ABS(UB)*DT/(2.*D(J1)*GC))*CFF
  P(J1,I,2)=P(JN,I,2)
  U(JN,I,2)=0.5*(UA+UB+(PA/C(JN)-PB/C(J1))*GC/RHO-DT/2.*(FA*UA*
  XABS(UA)/D(JN)+FB*UB*ABS(UB)/D(J1)))
  U(J1,I,2)=U(JN,I,2)
  RETURN
END

```

```

SUBROUTINE PIPES (J1,JN,I)
  IMPLICIT REAL*8 (A-H,C-Z)
  COMMON /S5/ RHO,AMU,DT,PR,UR,LEGNO,NP /S2/ A(50),C(50),DZ(50),
  XD(50),P(50,120,2),U(50,120,2),RF(50)
  DATA GC/32.173/,NOUT,I1,I2/6,1,2/

```

```

C**** THIS SUBPROGRAM DESCRIBES SUDDEN EXPANSION AND CONTRACTION *****
  ABS(X) = DABS(X)
  REAL*8 K
  R=A(JN)/A(J1)
  NRPT = 0
  1 UU = U(JN,I,2)
C***** DUMMY JUNCTION CASE *****
  IF (R.EQ.1) GO TO 15
C***** EXPANSION CASE *****
  IF (R.LT.1.0.AND.UU.GE.0.) K=(1.-R)**2
  IF (R.GT.1.0.AND.UU.LE.0.) K=-(R-1.)**2
C***** CONTRACTION CASE *****
  IF (R.GT.1.0.AND.UU.GE.0.) K=.45*R*(R-1.)
  IF (R.LT.1.0.AND.UU.LE.0.) K=-.45*(1.-R)
  DD=1.-R**2-K
  IF (ABS(DD).LE.1.0E-5) WRITE(NOUT,25) DD
  CALL INTER (JN,I,I1,UA,PA)
  CALL FACTOR (JN,UA,FA)
  CALL INTER (J1,I1,I2,UB,PB)
  CALL FACTOR (J1,UB,FB)
  BB=(C(JN)+C(J1)*R)/DD
  CC=(2.*(C(JN)*UA+C(J1)*UB)+2.*GC*(PA-PB)/RHO-DT*(FA*C(JN)*UA*
  XABS(UA)/D(JN)+FB*C(J1)*UB*ABS(UB)/D(J1)))/DD
  U(JN,I,2)=SQROOT(BB,CC,DD)
  U(J1,I,2)=U(JN,I,2)*R
  P(JN,I,2)=PA-RHO*C(JN)*(U(JN,I,2)-UA+FA*UA*ABS(UA)*DT/2./D(JN))/GC
  P(J1,I,2)=PB+RHO*C(J1)*(U(J1,I,2)-UB+FB*UB*ABS(UB)*DT/2./D(J1))/GC
  IF (NRPT.EQ.1) RETURN
  NRPT = NRPT + 1
C ***** RECALCULATE K FOR NEW VALUE OF U(JN,I,2) *****
  GO TO 1
15 WRITE(NOUT,20)
20 FORMAT(5X,'***** ERROR IN SPECIFICATION OF DIAMETER DISCONTINUITY
  X*****')
25 FORMAT(5X,'***** ABS(DD) IS .LE. 1.E-5: DD .EQ.',E12.5,' *****')
  RETURN
END

```

```

FUNCTION SQRCCT (BB,CC,DD)
IMPLICIT REAL*8 (A-H,O-Z)
DATA NOUT/6/
ABS(X) = DABS(X)
SQRT(X) = DSQRT(X)
10 FORMAT (5X,'***** SQROOT HAS NEGATIVE ARGUMENT',E12.5,' *****')
IF (CC.EQ.0.0) GO TO 9
CARG=BB*BB-CC
IF (CARG.LT.0.) GO TO 15
IF (DD.GE.0.) SQROOT=BB-SQRT(CARG)
IF (DD.LT.0.) SQRCCT=BB+SQRT(CARG)
RETURN
9 SQROOT=0.
RETURN
15 WRITE (NOUT,10) CARG
CARG=ABS(CARG)
IF (DD.GE.0.) SQRCCT=BB-SQRT(CARG)
IF (DD.LT.0.) SQROOT=BB+SQRT(CARG)
RETURN
END

```

```

SUBROUTINE FACTOR (L,V,F)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /S5/ RHO,AMU,DT,PR,UR,LEGNC,NP /S2/ A(50),C(50),CZ(50),
XC(50),P(50,120,2),U(50,120,2),RF(50)
DATA NCUT/6/
ABS(X) = DABS(X)
ALOG10(X) = DLOG10(X)
SQRT(X) = DSQRT(X)
C**** THIS SUBPROGRAM DETERMINES 'WEISBACH' FRICTION FACTOR USING *****
C**** 'COLEBROOK-WHITE' CORRELATION *****
N=0
F=0.
Q=ABS(V)
C**** IF NP=1, FRICTION IS NEGLECTED *****
IF (NP.EQ.1) RETURN
RR=RF(L)/D(L)
RE=RHO*C*D(L)/AMU
IF (RE.LE.0.1) RETURN
IF (RE.GT.3000.) GO TO 5
C**** LAMINAR FLOW *****
F=64./RE
RETURN
5 IF (RR.GT.1.E-4) GO TO 10
C**** INITIAL ESTIMATE FOR SMOOTH PIPE BY 'BLASIUS' EQUATION *****
F=0.316/RE**.25
GO TO 25
C**** INITIAL ESTIMATE FOR ROUGH PIPE BY 'VON KARMAN' EQUATION *****
10 F=1./(-2.*ALOG10(RR)+1.14)**2
C**** F IS DETERMINED ITERATIVELY BY 'COLEBROOK-WHITE' CORRELATION *****
25 SF=SQRT(F)
GO TO 35
30 SF=FF
35 DD=1.14-2.*ALOG10(RR+9.35/(RE*SF))
FF=1./DD
IF ((ABS(FF-SF)/SF).LE.0.01) GO TO 50
IF (N.GE.50) GO TO 45
N=N+1
GO TO 30
45 WRITE (NOUT,100)
100 FORMAT (5X,'***** NO CONVERGENCE IS ACHIEVED *****')
50 F=FF**2
RETURN
END

```

```

SUBROUTINE TEE (JT,JJ,IC)
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /S1/ N1(50),NN(50),N(50) /S5/ RHO,AMU,DT,PR,UR,LEGNO,NP
X/S2/ A(50),C(50),DZ(50),D(50),P(50,120,2),U(50,120,2),RF(50)
  DATA GC/32.173/,I1,I2/1,2/
  DIMENSION M1(6),MN(6),UA(6),UB(6),PA(6),PB(6)
  ABS(X) = DABS(X)
C**** THIS SUBPROGRAM DESCRIBES TEE JUNCTION JOINING UP TO 6 BRANCHES *****
  PC=PR*144.
  NN1=0
  NNN=0
  CR=0.
  CS=0.
  UBS=0.
  PBS=0.
  FBS=0.
  UAR=0.
  PAR=0.
  FAR=0.
  DO 4 L=1,LEGNO
    IF (N1(L).NE.JT) GO TO 1
    NN1=NN1+1
    M1(NN1)=L
  1 IF (NN(L).NE.JT)GO TO 4
    NNN=NNN+1
    MN(NNN)=L
  4 CONTINUE
  NT=NN1+NNN
  DO 10 I=1,NT
    IF (I.GT.NN1) GO TO 5
    L=M1(I)
    IF (L.NE.JJ) GO TO 2
    UB(I)=U(L,1,1)
    PB(I)=P(L,1,1)
    GO TO 3
  2 CALL INTER (L,I1,I2,UB(I),PB(I))
  3 DCS=A(L)/(RHO*C(L))*GC
    UBS=UBS+A(L)*UB(I)
    PBS=PBS+DCS*PB(I)
    CS=CS+DCS
    BS=A(L)/(2.*D(L))
    CALL FACTOR (L,UB(I),FB)
    FBS=FBS+FB*BS*UB(I)*ABS(UB(I))*DT
    GO TO 10
  5 M=I-NN1
    L=MN(M)
    NM=N(L)-1
    IF (L.NE.JJ) GO TO 7
    UA(M)=U(L,NM+1,1)
    PA(M)=P(L,NM+1,1)
    GO TO 8
  7 CALL INTER (L,N(L),I1,UA(M),PA(M))
  8 DCR=A(L)/(RHO*C(L))*GC
    UAR=UAR+A(L)*UA(M)
    PAR=PAR+DCR*PA(M)
    CR=CR+DCR
    BR=A(L)/(2.*D(L))
    CALL FACTOR (L,UA(M),FA)
    FAR=FAR+FA*BR*UA(M)*ABS(UA(M))*DT
  10 CONTINUE
    IF (IC.EQ.1) GO TO 13
C**** IF U=0 AND P=PR IN EVERY BRANCH. SET U=0 AND PJT=PR *****

```

```

DO 12 I=1,NT
  IF (I.GT.NN1) GO TO 11
  IF (UB(I).EQ.0.0.AND.PB(I).EQ.PC) GO TO 12
  GO TO 13
11 M=I-NN1
  IF (UA(M).EQ.0.0.AND.PA(M).EQ.PC) GO TO 12
  GO TO 13
12 CONTINUE
  PJT=PC
  GO TO 14
13 PJT=(UAR-UBS+PAR+PBS-FAR+FBS)/(CR+CS)
14 DO 20 I=1,NT
  IF (I.GT.NN1) GO TO 15
  L=M1(I)
  CALL FACTOR (L,UB(I),FB)
  U(L,1,2)=UB(I)+(PJT-PB(I))*GC/(RHO*C(L))-FB*UB(I)*ABS(UB(I))*
  XDT/(2.*D(L))
  P(L,1,2)=PJT
  GO TO 20
15 M=I-NN1
  L=MN(M)
  NL=N(L)
  CALL FACTOR (L,UA(M),FA)
  U(L,NL,2)=UA(M)-(PJT-PA(M))*GC/(RHO*C(L))-FA*UA(M)*ABS(UA(M))
  X*DT/(2.*D(L))
  P(L,NL,2)=PJT
20 CONTINUE
RETURN
END

```

```

SUBROUTINE SOURCE (P,T,V1,V2,C1,C2,A1,A2)
  IMPLICIT REAL*8 (A-F,O-Z)
  COMMON /S5/ RHO,AML,DT,PR,UR,LEGNC,NP
  DATA GC/32.173/

```

```

C**** THIS SUBPROGRAM CALCULATES SOURCE PRESSURE BY ZAKER-SALMON ANALYSIS *****
REAL*8 MDOT
IF (T-DT) 1,1,9

```

```

C**** READ INPUT DATA *****
1 READ (5,1000) P1,WZD,PCT,TAU,EXP,PCRT,PRES
1000 FORMAT(5E12.5,/,2E12.5)

```

```

C ***** DESCRIPTION OF INPUT DATA *****
C * P1 - SYSTEM PRESSURE, PSI *
C * WZD - LEAK RATE, LB/SEC *
C * PCT - FRACTIONAL INITIAL RUPTURE RADIUS OF MAX RADIUS *
C * TAU - TIME CONSTANT APPEARING IN TIME-DEPENDENT RUPTURE MODEL *
C * EXP - EXPONENT APPEARING IN TIME-DEPENDENT RUPTURE MODEL *
C * PCRT - CRITICAL PRESSURE, PSI *
C * PRES - RESERVOIR PRESSURE, PSI *
C *****

```

```

RMAX=(WZD/(3.14159*18.9))**.5
RI=PCT*RMAX

```

```

WRITE(6,2000) P1,WZD,RI,TAU,EXP,PCRT,PRES

```

```

2000 FORMAT (1H1,////,37X,'----- INPUT DATA FOR SOURCE PRESSURE CALCU
XLATION -----',////,47X,'SYSTEM PRESSURE P1 = ',E12.5,' PSI',/,
251X,'LEAK RATE MDOT = ',E12.5,' LB/SEC',/,45X,'INITIAL BREAK RADIU
3S = ',E12.5,' INCH',/,48X,'TIME CONSTANT TAU = ',E12.5,' SEC',/,
449X,'EXPONENT EPSILON = ',E12.5,/,48X,'CRITICAL PRESSURE = ',
5E12.5,' PSI',/,47X,'RESERVOIR PRESSURE = ',E12.5,' PSI')

```

```

C=12.3
B=23.*1728./9./RHO
E=5700.*12.*778.
PL=P1
PU=PRES
PT=P1+100.
RC1=RHO*C1/GC/1728.
RC2=RHO*C2/GC/1728.
K=0
RR=RI+(RMAX-RI)*(T/TAU)**EXP
WZD=3.14159*18.9*RR**2
7  CALL FCFP(FP,PT,WZD,PCRT,PRES)
MDOT=FP
T1=E-C*B*PT
  IF(T1)4,4,15
15 CONTINUE
T2=(PT-P1)/RC1*A1+(PT-P1)/RC2*A2
FP=T1*FP/(1.+C)/T2
  IF(FT-FP)2,3,4
2  PL=PT
  GOTO 5
4  PU=PT
5  ERR=(PU-PL)/PU
  IF(ERR-1.0E-6)3,3,6
6  PT=(PL+PU)/2.
  K=K+1
  IF(K.FQ.200) GOTO 20
  GCTC 7
3  PB=PT-P1
  P=PT
  RETURN
20 WRITE (6,3000) PU,PL,PT
3000 FORMAT (' PU,PL,PT'/(3E15.5))
  CALL EXIT
9  IF(T.GT.TAU) GO TO 19
RR=RI+(RMAX-RI)*(T/TAU)**EXP
WZD=3.14159*18.9*RR**2
19 IF(T.GT.CT) GO TO 101
VIN1=V1
VIN2=V2
PBDOT=0.0
VEE=0.0
VEEDOT=B*MDOT+VIN1*A1+VIN2*A2
101 IF (T.EQ.DT) GO TO 105
CALL FCFP (FP,P,WZD,PCRT,PRES)
VEEDOT=B*FP+V1*A1+V2*A2
IF((V1.EQ.VIN1).AND.(V2.EQ.VIN2)) GO TC 105
PBDOT=((E-C*B*P)*FP-(1.+C)*P*(V1*A1+V2*A2))/C/VEE
105 PB=PB+PBDOT*DT
P=PB+P1
VEE=VEE+VEEDOT*DT
RETURN
END

```

```

SUBROUTINE FOF(P,WD,PCRT,PRES)
IMPLICIT REAL*8 (A-H,O-Z)
SQRT(X) = DSQRT(X)
C**** THIS SUBPROGRAM DETERMINES INSTANTANECUS LEAK RATE *****
FP=WD
  IF(P-PCRT)1,1,2
1 RETURN
2 FP=0.
  IF(P-PRES) 3,1,1
3 FP=WD*SQRT(1.-((P-PCRT)/(PRES-PCRT))**2)
RETURN
END

```

```

SUBROUTINE PTIME (NNK,T,P)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION TIME(50),PTM(50)
DATA NIN,NOUT/5,6/
C**** THIS SUBROUTINE READS IN PRESSURE - TIME HISTORY AND CALCULATES *****
C**** PRESSURE AT ANY TIME BY LINEAR INTERPOLATION *****
IF (T.GT.0.) GO TO 5
C**** READ PRESSURE - TIME HISTORY : PRESSURE IN PSI AND TIME IN SEC *****
READ (NIN,100) (TIME(I),I=1,NNK)
READ (NIN,100) (PTM(I),I=1,NNK)
100 FORMAT (6E12.5)
WRITE (NOUT,200)
200 FORMAT (1H1,////,40X,'----- INPUT DATA FOR PRESSURE - TIME HISTCR
1Y -----',///,52X,'TIME, SEC',6X,'PRESSURE, PSI',/)
WRITE(NOUT,300) (TIME(I),PTM(I),I=1,NNK)
300 FORMAT (50X,1PE12.5,5X,1PE12.5)
5 DO 10 L=1,NNK
  I = L - 1
  IF (L.GE.NNK) GO TO 34
  IF (T.EQ.TIME(L)) GO TO 34
10 IF (T.LT.TIME (L)) GO TO 32
  GO TO 34
32 P = PTM(I)+(PTM(I+1)-PTM(I))*(T-TIME (I))/(TIME(I+1)-TIME(I))
  GO TO 35
34 I = I + 1
  P = PTM(I)
35 RETURN
END

```

APPENDIX C

Input Data for Sample Problem

```

1      5      3      0      2      0      1      0      0      0      0      0      1
9      17      1      5      2      0      0      6
0.70E-5      0.0E00      3.0E-3      0.0E00      0.52762E02      5.000E03
0.6250E00      1.25E00
1      1      3
.3125E 00      1.875E 00      .3125E 00      1.250E 00      .0000E 00
0.0E 00      3.0E-04      5.0E-04      7.0E-04      1.0E-03      1.0E-01
5.E 02      20.E 02      12.E 02      20.E 02      5.0E 02      5.0E 02
3      4      0      2      20      2      0      1
0.07000E-04      0.00000E 00      0.00000E 00      0.83500E 03
0.00300E 00      1.10000E 00
0      0      2      0      2      0      1
1 2 0.44200E 02 0.20000E 01 0.00020E 00 0.25000E 00 0.25000E 08 0.75000E 01
0.0E 00      0.0E 00
2 3 2.16200E 03 0.20000E 01 0.00020E 00 0.50000E 00 0.25000E 08 0.52500E 02
0.0E 00      0.0E 00
3 4 0.44200E 02 0.20000E 01 0.00020E 00 0.25000E 00 0.29000E 08 0.75000E 01
0.0E 00      0.0E 00
9 1 1 9
0.52762E02      5.000E03
1      26      1      50      1      33      9      25      9      33

```

APPENDIX D
Output of Sample-problem Calculation

----- INPUT DATA FOR TWO - DIMENSIONAL DOMAIN CALCULATION -----

IP1 = 1	IP2 = 5	IP3 = 3	IP4 = 0	IP5 = 2
IP6 = 0	IP7 = 1	IP8 = 0	IP9 = 0	IP10 = 0
IP11 = 0	IP12 = 0	IP13 = 1		
NR = 5	NZ = 17	II = 1	JJ = 5	NF = 2
NIT = 0	MIT = 0	NNH = 6		
DT = 7.000D-06	TI = 0.0	TF = 3.000D-03	PR = 0.0	RHO = 5.276D 01
C = 5.000D 03	RR = 6.250D-01	ZLEN = 1.250D 00		
NFS = 1	IA = 1	IB = 3		
RR1 = 3.125D-01	ZL1 = 1.875D 00	RR2 = 3.125D-01	ZL2 = 1.250D 00	TFS = 0.0
NR1 = 5	NZ1 = 24	NZP = 25	NR2 = 5	NZQ = 41
NZC = 42	NZ2 = 57			

----- INPUT PRESSURE - TIME HISTORY FOR TWO - DIMENSIONAL CALCULATION -----

TIME, SEC	PRESSURE, PSI
0.0	0.50000D 03
0.30000D-03	0.20000D 04
0.50000D-03	0.12000D 04
0.70000D-03	0.20000D 04
0.10000D-02	0.50000D 03
0.10000D 00	0.50000D 03

----- INITIAL CONDITION AND DESCRIPTION OF NETWORK ARRANGEMENT FOR ONE - DIMENSIONAL DOMAIN CALCULATION -----

LEGND = 3 JUNNO = 4 NOT = 0 NOTT = 0 JJ = 2 IC = 0
 NP = 1 INDTR = 0 NOPRIN = 2 KS = 20 JK = 2 NNK = 0
 IP1 = 2 IP2 = 0 IP3 = 1

UR = 0.0 PR = 0.0 PCR = 0.0 TR = 0.83500D 03 RHO = 0.52762D 02
 AMU = 0.17293D-03 DT = 0.70000D-05 XI = 0.0 XF = 0.30000D-02 PARM = 0.11000D 01

BRANCH NO	JUNCTION TYPE		NO CF NODES	INTERNODAL DISTANCE	LENGTH FEET	AREA SQR INCH
	N = 1	N = N				
1	1(9)	2(1)	52	0.3850D-01	0.2000D 01	0.4420D 02
2	2(1)	3(1)	52	0.3850D-01	0.2000D 01	0.2162D 04
3	3(1)	4(9)	52	0.3850D-01	0.2000D 01	0.4420D 02

BRANCH NC	DIAMETER INCH	THICKNESS INCH	YOUNGS MODULUS PSI	SONIC SPEED FT/SEC	PRESSURE PSI	VELOCITY FT/SEC
1	0.7500D 01	0.2500D 00	0.2900D 08	0.5000D 04	0.0	0.0
2	0.5250D 02	0.5000D 00	0.2900D 08	0.5000D 04	0.0	0.0
3	0.7500D 01	0.2500D 00	0.2900D 08	0.5000D 04	0.0	0.0

----- POINTS AT WHICH RESULTS ARE PRINTED -----

I = 1 J = 26
 I = 1 J = 50
 I = 1 J = 33
 I = 9 J = 25
 I = 9 J = 33

----- RESULT OF PRESSURE TRANSIENT CALCULATION -----

TIME = 0.0 SEC (NC = 0 STEPS) : SOURCE PRESSURE = 5.000000 02

LOCATION	R, FT	Z, FT	U, FT/SEC	W, FT/SEC	PRESSURE, PSI
I = 1 J = 26	0.0	1.95313D 00	0.0	0.0	0.0
I = 1 J = 50	0.0	3.82813D 00	0.0	0.0	0.0
I = 1 J = 33	0.0	2.50000D 00	0.0	0.0	0.0
I = 9 J = 25	6.25000D-01	1.87500D 00	0.0	0.0	0.0
I = 9 J = 33	6.25000D-01	2.50000D 00	0.0	0.0	0.0

BRANCH NO =	DISTANCE, FT	PRESSURE, PSI	VELOCITY, FT/SEC
1	0.0	0.0	0.0
	0.770000D 00	0.0	0.0
	0.154000D 01	0.0	0.0
	0.200000D 01	0.0	0.0
3	0.0	0.0	0.0
	0.770000D 00	0.0	0.0
	0.154000D 01	0.0	0.0
	0.200000D 01	0.0	0.0

TIME = 1.40000D-05 SEC (NC = 2 STEPS) : SOURCE PRESSURE = 5.35000D 02

LOCATION		R, FT	Z, FT	U, FT/SEC	W, FT/SEC	PRESSURE, PSI
I = 1	J = 26	0.0	1.95313D 00	0.0	0.0	0.0
I = 1	J = 50	0.0	3.82813D 00	0.0	0.0	0.0
I = 1	J = 33	0.0	2.50000D 00	0.0	0.0	0.0
I = 9	J = 25	6.25000D-01	1.87500D 00	0.0	0.0	0.0
I = 9	J = 33	6.25000D-01	2.50000D 00	0.0	0.0	0.0

BRANCH NO =	DISTANCE, FT	PRESSURE, PSI	VELOCITY, FT/SEC
1	0.0	0.0	0.0
	0.770000D 00	0.0	0.0
	0.154000D 01	0.0	0.0
	0.200000D 01	0.0	0.0
3	0.0	0.0	0.0
	0.770000D 00	0.0	0.0
	0.154000D 01	0.0	0.0
	0.200000D 01	0.0	0.0

TIME = 2.80000D-05 SEC (NC = 4 STEPS) : SOURCE PRESSURE = 6.05000D 02

LOCATION		R, FT	Z, FT	U, FT/SEC	W, FT/SEC	PRESSURE, PSI
I = 1	J = 26	0.0	1.95313D 00	0.0	0.0	0.0
I = 1	J = 50	0.0	3.82813D 00	0.0	0.0	0.0
I = 1	J = 33	0.0	2.50000D 00	0.0	0.0	0.0
I = 9	J = 25	6.25000D-01	1.87500D 00	0.0	0.0	0.0
I = 9	J = 33	6.25000D-01	2.50000D 00	0.0	0.0	0.0

BRANCH NO =	DISTANCE, FT	PRESSURE, PSI	VELOCITY, FT/SEC
1	0.0	0.0	0.0
	0.770000D 00	0.0	0.0
	0.154000D 01	0.0	0.0
	0.200000D 01	0.123587D 03	-0.217038D 01
3			

•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•

TIME = 2.98200D-03 SEC (NC = 426 STEPS) : SOURCE PRESSURE = 5.00000D 02

LOCATION		R, FT	Z, FT	U, FT/SEC	W, FT/SEC	PRESSURE, PSI
I = 1	J = 26	0.0	1.95313D 00	0.0	1.16625D 00	2.39990D 02
I = 1	J = 50	0.0	3.82813D 00	0.0	5.82361D 00	3.29189D 02
I = 1	J = 33	0.0	2.50000D 00	0.0	6.67683D 00	1.94514D 02
I = 9	J = 25	6.25000D-01	1.87500D 00	0.0	0.0	2.25354D 02
I = 9	J = 33	6.25000D-01	2.50000D 00	0.0	-2.77531D 00	1.95154D 02

BRANCH NC =	DISTANCE, FT	PRESSURE, PSI	VELOCITY, FT/SEC
1	0.0	0.405929D 03	-0.712873D 01
	0.770000D 00	0.405929D 03	-0.712873D 01
	0.154000D 01	0.405929D 03	-0.712873D 01
	0.200000D 01	0.405929D 03	-0.712873D 01
3	0.0	0.386672D 03	0.679056D 01
	0.770000D 00	0.430118D 03	0.755353D 01
	0.154000D 01	0.533671D 03	0.937209D 01
	0.200000D 01	0.645537D 03	0.113366D 02

TIME = 2.99600D-03 SEC (NC = 428 STEPS) : SOURCE PRESSURE = 5.00000D 02

LOCATION		R, FT	Z, FT	U, FT/SEC	W, FT/SEC	PRESSURE, PSI
I = 1	J = 26	0.0	1.95313D 00	0.0	1.37085D 00	2.42131D 02
I = 1	J = 50	0.0	3.82813D 00	0.0	5.64654D 00	3.19153D 02
I = 1	J = 33	0.0	2.50000D 00	0.0	6.68435D 00	1.90835D 02
I = 9	J = 25	6.25000D-01	1.87500D 00	0.0	0.0	2.24968D 02
I = 9	J = 33	6.25000D-01	2.50000D 00	0.0	-2.68643D 00	1.91115D 02

----- END OF CALCULATION -----

----- MAXIMUM PRESSURES IN EACH BRANCH DURING CALCULATION -----

MAXIMUM PRESSURE IN BRANCH	1 =	0.16221D 04	PSI OCCURRED AT	0.35000D-03	SEC
MAXIMUM PRESSURE IN BRANCH	2 =	0.0	PSI OCCURRED AT	0.0	SEC
MAXIMUM PRESSURE IN BRANCH	3 =	0.16790D 04	PSI OCCURRED AT	0.16030D-02	SEC

APPENDIX E

Source-pressure Calculation by Sodium-Water Reaction

The one-dimensional model as originally developed by Zaker and Salmon for the sodium-water reaction and for the product-zone growth characterizing the source pressure is reproduced here. An instantaneous reaction was assumed, and the break was assumed to occur at a junction connecting two or more pipe branches. Mass and energy balances taken for a control volume enclosing the product zone at the junction lead to the following ordinary differential equations in pressure p and product-zone volume V :

$$\frac{dV}{dt} = B \frac{dm}{dt} + \sum_{i=1}^n u_i A_i \quad (\text{E.1})$$

and

$$\frac{dp}{dt} = \left[(F - CBp) \frac{dm}{dt} - (1 + C)p \sum u_i A_i \right] / CV, \quad (\text{E.2})$$

where dm/dt is the leak rate, which depends on the steam-reservoir condition, the rupture area, and the back pressure (in the reaction-product zone); B , C , and F are constants; and u_i and A_i are the fluid velocity and flow cross-sectional area, respectively, of the i th branch connected to the leak site. For the steam-reservoir state of pressure p_0 , temperature T_0 , and the critical pressure at the break p_c , the leak rate is written

$$\left. \begin{aligned} \frac{dm}{dt} &= \dot{m}_0 && \text{for } p < p_c; \\ \frac{dm}{dt} &= \dot{m}_0 \left[1 - \left(\frac{p - p_c}{p_0 - p_c} \right)^2 \right]^{1/2} && \text{for } p_c \leq p \leq p_0; \\ \frac{dm}{dt} &= 0 && \text{for } p > p_0. \end{aligned} \right\} \quad (\text{E.3})$$

The initial conditions considered are

$$\left. \begin{aligned} \frac{dp}{dt} &= 0, \\ V &= 0, \\ u_i &= \frac{p - p_1}{\rho a_i}, \end{aligned} \right\} \quad (\text{E.4})$$

and

where p_1 is the system fluid pressure existing before the leak starts and a_i is the sonic velocity in the i th branch. The initial pressure thus is determined by Eq. E.2 to be

$$p = \frac{(F - CBp)dm/dt}{(1 + C)\sum u_i A_i}, \quad (E.5)$$

which is solved by an iteration.

The equations and initial conditions, Eqs. E.1-E.5, determine the source pressure p and volume V for given leak rate dm/dt and velocities u_i .

REFERENCES

1. J. Von Neumann and R. D. Richtmyer, *A Method for the Numerical Calculation of Hydrodynamic Shocks*, J. Appl. Phys. 21, 232-237 (1950).
2. L. Fox, *Numerical Solution of Ordinary and Partial Differential Equations*, Addison-Wesley Pub. Co., Reading, Mass., pp. 218-229 (1962).
3. C. K. Thornhill, *The Numerical Method of Characteristics for Hyperbolic Problems in Three Independent Variables*, ARC-RM-2615 (1948).
4. M. Holt, *The Method of Characteristics for Steady Supersonic Rotational Flow in Three Dimensions*, J. Fluid Mech. 1, 409-423 (1956).
5. D. S. Butler, *The Numerical Solution of Hyperbolic Systems of Partial Differential Equations in Three Independent Variables*, Proc. Roy. Soc. London, Series A, 255, 232-252 (1960).
6. R. Sauer, *Differenzenverfahren für hyperbolische Anfangswert-probleme bei mehr als zwei unabhängigen Veränderlichen mit Hilfe von Nebencharakteristiken*, Num. Math. 5, 55-67 (1963).
7. D. J. Richardson, "Solutions of Two-Dimensional Hydrodynamic Equations by the Method of Characteristics," in *Methods in Computational Physics*, Vol. 3, B. Alder, S. Fernbach, and M. Rotenberg, eds., Academic Press, N.Y. pp. 295-318 (1964).
8. G. Bruhn and W. Haack, *Ein Charakteristikenverfahren für drei-dimensionale instationäre Gasströmung*, Math. Phys. Z. Angew 98, 173-190 (1958).
9. C. R. Strom, *The Method of Characteristics for Three-dimensional Steady and Unsteady Reacting Gas Flow*, Ph.D. thesis, University of Illinois, Urbana (1965).
10. Y. W. Shin, *The Method of Characteristics for Analysis of Pressure Transients Resulting from Sodium-Water Reaction in Hydraulic Networks*, ANL-8049 (Dec 1973).
11. A. Jeffrey and T. Taniuti, *Nonlinear Wave Propagation*, Academic Press, N.Y. (1964).
12. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. 2, Interscience Publishers, N.Y. (1962).
13. Y. W. Shin, *The Method of Characteristics for Pressure Transients in Cylindrical Geometry with Axial Symmetry --The Formulation*, Technical Memorandum ANL-CT-73-23 (Oct 1973).
14. R. von Mises, *Mathematical Theory of Compressible Fluid Flow*, Academic Press, N.Y. (1958).
15. R. Courant, K. O. Friedrichs, and H. Lewy, *Über die Partiellen Differenzialgleichungen der Mathematischen Physik*, Math. Ann. 100, 32-74 (1928).