77-22,454

PEDRIDO, Renée Ramona, 1951-
DYNAMIC ANALYSIS OF STRUCTURES WITH
SOLID-FLUID INTERACTION.

New Jersey Institute of Technology,
D.Eng.Sc., 1977
Engineering, mechanical

Xerox University Microfilms, Ann Arbor, Michigan 48106

DYNAMIC ANALYSIS OF STRUCTURES WITH SOLID-FLUID INTERACTION

BY

RENÉE RAMONA PEDRIDO

A DISSERTATION

PRESENTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE

OF

DOCTOR OF ENGINEERING SCIENCE

AT

NEW JERSEY INSTITUTE OF TECHNOLOGY

Newark, New Jersey
1977

ABSTRACT

Interaction between solid and fluid has been recognized to be an important factor in the areas of aeroelasticity, hydroelasticity, and the study of flow-induced vibration in nuclear reactor components. This study develops a finite element model for interaction between an elastic solid and a fluid medium. Plane triangular finite elements have been used separately for fluid, solid, and solid-fluid continua and the equivalent mass, damping, and stiffness matrices and interaction load arrays for all elements are derived and assembled into global matrices. The global matrix differential equation of motion developed is solved in time to obtain the pressure and velocity distributions in the fluid, as well as the displacements in the solid. Two independent computer programs, each employing different algorithms and numerical solution techniques are used to obtain the dynamic solution. The first program is FLINTS (Fluid Interacting with Solid), a special purpose finite element program developed herein for solid-fluid interaction studies. This program uses the modal superposition technique in which the eigenvalues and eigenvectors for the system are found and used to uncouple the equations. This approach allows an analytic solution in each integration time step. The second program is WECAN (Westinghouse Electric Computer Analysis), a general purpose finite element program in which new element library subroutines for solid-fluid interaction were incorporated. This program can employ a NASTRAN direct integration scheme based on a central difference formula for the acceleration and velocity terms and an implicit representation of the displacement term. This reduces the problem to a matrix equation whose right hand side is updated in every time step and is solved by a variation of the Gaussian elimination method known as the wave front technique. Results have been obtained for the case of water, between two flat elastic parallel plates, initially at rest and accelerated suddenly by applying a step pressure. The results obtained from the above-mentioned two independent finite element programs are in full agreement. This verification provides the confidence needed to initiate parametric studies. Both rigid wall (no solid-fluid interaction) and flexible wall (including solid-fluid interaction) cases were examined. The pressure time histories for the flexible wall configuration show the following features: 1) The observed period of oscillation of the fluid increased with respect to the rigid wall fluid period $2 L/c$ as expected. This is due to a reduction in the effective speed of sound in the fluid resulting from the solid-fluid interaction; 2) The observed pressure in the fluid is generally lower than the pressure in the rigid wall case except when transversal water hammer occurs. This is also a solid-fluid interaction effect, caused by the motion of the wall as the step pressure wave advances along the channel; 3) Transversal flow due to the motion of the wall is also observed. When the motion of the wall reaches its maximum, the transversal flow is decreased resulting in a water hammer phenomenon. This effect exhibits itself in the form of a pressure surge on the response curve.

APPROVAL OF DISSERTATION

DYNAMIC ANALYSIS OF STRUCTURES
WITH SOLID-FLUID INTERACTION

BY

RENÉE RAMONA PEDRIDO

FOR

DEPARTMENT OF MECHANICAL ENGINEERING

NEW JERSEY INSTITUTE OF TECHNOLOGY

BY

FACULTY COMMITTEE

APPROVED: ______________________________, Chairman

____

____

____

____

____

April, 1977

ACKNOWLEDGEMENTS

The author wishes to express her deep and sincere appreciation to Dr. Amir N. Nahavandi, whose continued guidance, insight, and desire to seek the truth has made this study possible. These traits, and his willingness to assist the student in any way to see the research come to a successful conclusion, makes him truly the best advisor anyone could ever ask for. The author only hopes that she can live up to the standards he has set.

The author also wishes to express her special appreciation to her supervisor, Mr. Donn Baker of Bell Laboratories, whose patience, understanding, and generosity with Bell Laboratories facilities and computer time expedited the successful completion of this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PART ONE

DEVELOPMENT OF FINITE ELEMENT MODEL

1 INTRODUCTION

In recent years, there has been a wide interest in the subject of solid-fluid interaction. This area encompasses the interaction of a structure with an incompressible or compressible fluid which is of major concern in many fields of applied science and engineering such as aeroelasticity, hydroelasticity, flow induced vibration, etc. Solid-fluid interaction is also of particular interest in the design and analysis of nuclear reactor components involving geometric complexities and nonlinearities. Experimental studies have been performed that confirm the importance of solid-fluid interaction [10,23][1].

A survey of solid-fluid interaction literature reveals several major categories: 1) Hydrodynamic mass approach; 2) Vibrational behavior of a fluid-conveying pipe or conduit; 3) Interaction between a body and fluid flowing past the body; 4) Interaction between a vibrating panel and a fluid medium; and 5) Generalized solid-fluid interaction.

In the first category, the hydrodynamic mass concept, frequently used in dynamic analyses, has the following limitations. Most practical problems have complicated geometries and the hydrodynamic mass analyses are generally performed for simple geometries with a limited range of applicability [21]. Moreover, for problems involving impact in nuclear power systems, the use of hydrodynamic mass makes the results unrealistically conservative.

[1]Numbers in brackets designate references given in Section 9.

In the second category, flow through vibrating pipes is a historically well known problem and has been examined extensively. Some authors have attempted to approach the problem in a fashion similar to Streeter's water hammer analysis [22]. Chen and Rosenberg [5] studied general fluid-shell interaction characteristics in a small flow velocity range for cylindrical shells conveying fluid. They developed a modified water hammer theory and an approximate bending frequency equation including the effect of the flowing fluid. In the area of plastic response, Fox and Stepnewski [9] developed a one dimensional dynamic analysis which indicated that at pressures exceeding the pipe yield pressure, wave velocities are substantially reduced from the elastic wave velocity, and the waves are dispersed.

The third category, interaction between a body and fluid flowing past the body involving aerodynamic and nuclear problems, has received considerable treatment. In the area of reactor core and heat exchanger analysis, Hine [11] studied the dynamics of slender rods in a cylindrical duct subjected to parallel flow together with an acoustic field. He determined that acoustically induced rod vibrations are critically dependent upon a close structural-acoustic wavelength match. Paidoussis [19] reviewed the existing methods of prediction of the small amplitude vibrations of flexible cylinders subjected to flow that was not purely axial, steady, and uniform. He concluded that experimental verification of this subcritical vibration amplitude is probably best due to the extreme smallness of the vibration and because the force field in the boundary layer cannot be accurately characterized or predicted.

In the fourth category, there has been some examination of solid-fluid interaction with regard to plate-acoustic systems. Craggs [7] studied the behavior of a coupled plate-acoustic cavity system in which the acoustic pressure was spatially uniform. The solution was obtained using finite plate elements by adding an acoustic stiffness matrix, representing the effect of the cavity, to the mechanical stiffness of the plate. Thus, the effect of the acoustic medium was included without having to model the medium itself with finite elements. None of the preceeding examples studied solid-fluid interaction as a phenomenon in itself. These studies concerned themselves with the effect that the fluid has on the system, similar to the hydrodynamic mass concept. Solid-fluid interaction has been studied by Nahavandi, Sun, and Ball [16,17] who examined the problem of a three-dimensional rectangular structure with one elastic wall, subjected to an external harmonic noise source, with and without damping. A concise closed-form solution for the sound pressure level inside the enclosure (for the first wall mode) was determined from the acoustic wave and vibrating plate partial differential equations using the Galerkin method of weighted residuals.

The final category is generalized solid-fluid interaction. In this case, most investigators have developed a finite element model with fluid displacement as the main variable with the ultimate objective of analyzing solid-fluid interaction. Feng and Kiefling [8] derived such a fluid finite element which was tested for the cases of sloshing and fluid compressibility. Tong [24] developed a fluid finite element for viscous flow based on small fluid displacements. The presence of the fluid was accounted for by the addition of equivalent fluid mass and stiffness matrices to the solid matrices. This model was applied to the problem of a liquid and a gas sloshing in an elastic container. In these studies, small fluid displacement was assumed, thereby limiting their scopes to fluid slosh and fluid compressibility analyses. The advantage to using fluid displacement as the dependent variable is that no solid-fluid coupling terms would appear in a solid-fluid interaction formulation. The solid structure and fluid would be treated as continuum in which displacement is the dependent variable. However, limiting the analysis to small fluid displacements severely reduces the applicability of the solution. The applicability may be improved by the choice of a different dependent variable for the fluid. Baker [1,2,3] developed a fluid finite element based on stream-function which is more suited to problems involving fluid flow, but may also be rendered compatible with existing structural elements. However, no generalized solid-fluid interaction package has yet been presented which develops a fluid finite element compatible with existing structural elements and at the same time establishes the interaction between the solid and fluid elements as a phenomenon in itself. This is the purpose of this study.

Specifically, the objectives of this study are:

1) To develop a finite element model for the generalized problem of solid-fluid interaction in a two-dimensional continuum.

2) To develop a computer program based on this finite element model for the treatment of solid-fluid interaction problems.

3) To employ this program for the study of a wave propagation problem consisting of water, between two flat flexible plates, initially at rest and accelerated suddenly by applying a step pressure at one end.

4) To examine the effect of both fluid and solid damping on the above solution.

2 MATHEMATICAL FORMULATION

2.1 Fluid Finite Element

The simplifying assumptions employed in the development of the fluid finite element are

1) The fluid flow is assumed to be compressible and two-dimensional.
2) The fluid pressure is used as the main dependent variable.
3) The density oscillations are assumed to be of small amplitude.
4) The fluid shear stress is assumed to be proportional to its velocity.

The differential equation governing the pressure distribution p in a two-dimensional flow subjected to small amplitude acoustical oscillations is given by the wave equation:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = \frac{1}{c^2}\left[\frac{\partial^2 p}{\partial t^2} + k_f \frac{\partial p}{\partial t}\right] \qquad (2.1\text{-}1)$$

in which c is the velocity of the acoustic waves and k_f is the viscous damping coefficient. Eq. (2.1-1) is discretized on finite element subdivisions of the fluid region by the Galerkin method of weighted residuals. A plane triangular fluid finite element, as shown in Fig. 2-1, is used as the basis for the fluid finite element formulation. It is assumed that the pressure at any point in the triangular element may be expressed as a polynomial in x and y . In this case,

$$p = \alpha_1 + \alpha_2 x + \alpha_3 y \qquad (2.1\text{-}2)$$

in which the coefficients α are functions of time. Employing the above equation at the nodes, it can be shown [26] that

$$p = [N_f(x,y)]\, \{p_e(t)\} \qquad (2.1\text{-}3)$$

in which $\{p_e\}$ is the elemental array of time-dependent nodal pressures

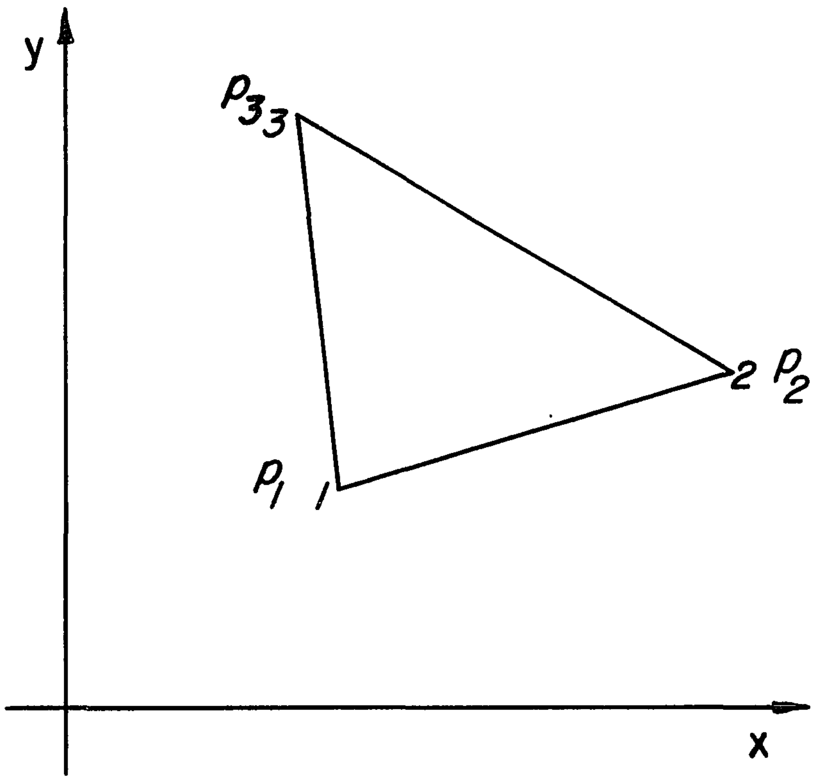


FIG 2-1 FLUID FINITE ELEMENT

and $[N_f]$ is the shape function of the fluid element, given by

$$[N_f] = \frac{1}{2\,A_f}\left[(a_1 + b_1\,x + c_1\,y)\quad(a_2 + b_2\,x + c_2\,y)\quad(a_3 + b_3\,x + c_3\,y)\right] \tag{2.1-4}$$

where

$$\begin{aligned} a_1 &= x_2\,y_3 - x_3\,y_2 \qquad & b_1 &= y_2 - y_3 \qquad & c_1 &= x_3 - x_2 \\ a_2 &= x_3\,y_1 - x_1\,y_3 \qquad & b_2 &= y_3 - y_1 \qquad & c_2 &= x_1 - x_3 \\ a_3 &= x_1\,y_2 - x_2\,y_1 \qquad & b_3 &= y_1 - y_2 \qquad & c_3 &= x_2 - x_1 \end{aligned} \tag{2.1-5}$$

$$A_f = (a_1 + a_2 + a_3)/2.$$

are the local coordinates of the triangle and A_f is the area of the triangular fluid element. Employing the weighted residual method, eq. (2.1-1) becomes

$$\iint_A [N_f]^T \left[\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \frac{k_f}{c^2}\frac{\partial p}{\partial t}\right] dx\,dy = 0 \tag{2.1-6}$$

Direct substitution of eq. (2.1-3) into the above equation and integrating would result in a trivial solution. It is therefore necessary to integrate the spatial derivatives in eq. (2.1-6) by parts in order to reduce the order of the derivative. In terms of continuity requirements, continuity of the variable and its derivatives up to one order less than the highest derivative in the differential equation must be maintained in the solution. In the present form of eq. (2.1-6), it would be necessary to find a solution which would satisfy continuity of p , $\partial p/\partial x$, and $\partial p/\partial y$. This is impossible using piecewise linear polynomials, which are being employed here. Integration by parts would reduce the order of the derivative so that continuity would have to be satisfied on p only, which is tractable. For these two reasons, it is necessary to integrate eq. (2.1-6) by parts. Doing so and substituting in eq. (2.1-3) yields

$$\oint_S [N_f]^T \frac{\partial p}{\partial n}\, dS \; - \; \iint_A \left[\frac{\partial [N_f]^T}{\partial x} \frac{\partial [N_f]}{\partial x} + \frac{\partial [N_f]^T}{\partial y} \frac{\partial [N_f]}{\partial y} \right] dx\, dy\, \{p_e\}$$
$$- \iint_A [N_f]^T \frac{1}{c^2} [N_f]\, dx\, dy\, \{\ddot{p}_e\} \; - \; \iint_A [N_f]^T \frac{k_f}{c^2} [N_f]\, dx\, dy\, \{\dot{p}_e\} = 0 \qquad (2.1\text{-}7)$$

Defining the following matrices and arrays which are evaluated in Appendix 1 :

The inertia matrix $\quad [G_e] = \iint_A [N_f]^T \frac{1}{c^2} [N_f]\, dx\, dy$

The viscous damping matrix $\quad [L_e] = \iint_A [N_f]^T \frac{k_f}{c^2} [N_f]\, dx\, dy$

(2.1-8)

The fluidity matrix $\quad [H_e] = \iint_A \left[\frac{\partial [N_f]^T}{\partial x} \frac{\partial [N_f]}{\partial x} + \frac{\partial [N_f]^T}{\partial y} \frac{\partial [N_f]}{\partial y} \right] dx\, dy$

The boundary integral array $\quad \{F_e\} = \oint_S [N_f]^T \frac{\partial p}{\partial n}\, dS$

eq. (2.1-7) reduces to

$$[G_e]\{\ddot{p}_e\} + [L_e]\{\dot{p}_e\} + [H_e]\{p_e\} = \{F_e\} \qquad (2.1\text{-}9)$$

$\{F_e\}$ is the contribution due to boundary integrals corresponding to the prescribed motion. For fluid elements, this contribution is cancelled out because it is an internal force. For fluid elements attached to a moving boundary, this term couples the fluid with the boundary, which will be discussed in Section 2.3 . Thus, for fluid finite elements the matrix differential equation becomes

$$[G_e]\{\ddot{p}_e\} + [L_e]\{\dot{p}_e\} + [H_e]\{p_e\} = 0 \qquad (2.1\text{-}10)$$

The boundary conditions which may be applied to eq. (2.1-10) are either specified nodal pressures or specified velocity components that can readily be expressed in terms of nodal pressures. For a totally fluid system, this matrix equation is established for each individual finite element, and is then assembled into a global matrix differential equation of the same form for solution.

Once the solution is obtained, the velocity components may be computed on an element basis by the application of the momentum equations

$$\frac{\partial v_x}{\partial t} = -\frac{1}{\rho_f}\frac{\partial p}{\partial x} - k_f\, v_x$$
$$\frac{\partial v_y}{\partial t} = -\frac{1}{\rho_f}\frac{\partial p}{\partial y} - k_f\, v_y \qquad (2.1\text{-}11)$$

Applying a simple finite difference scheme in time and discretizing the pressure yields for the horizontal velocity component v_x

$$\frac{v_x^{n+1} - v_x^{n}}{\Delta t} = -\frac{1}{\rho_f}\frac{\partial [N_f]}{\partial x}\{p_e\} - k_f\, v_x^{n} \qquad (2.1\text{-}12)$$

which when solved for the updated velocity v_x^{n+1}

$$v_x^{n+1} = v_x^{n} + \Delta t\left[-\frac{1}{\rho_f}\frac{\partial [N_f]}{\partial x}\{p_e\} - k_f\, v_x^{n}\right] \qquad (2.1\text{-}13)$$

Similarly, for v_y^{n+1}

$$v_y^{n+1} = v_y^{n} + \Delta t\left[-\frac{1}{\rho_f}\frac{\partial [N_f]}{\partial y}\{p_e\} - k_f\, v_x^{n}\right] . \qquad (2.1\text{-}14)$$

Thus, the velocities are updated in each time step, using the velocities from the previous iteration n .

2.2 Solid Finite Element

The solid finite element used in this study is developed based on the method of virtual work. This method equates the work and change in internal energy in a system generated during a virtual displacement $d\delta$. This procedure is well known and has been documented [20,26].

A plane triangular solid finite element as shown in Fig. 2-2, is used as the basis of the solid finite element formulation. It is assumed that the displacement at any point in the triangular element, u_x and u_y ,

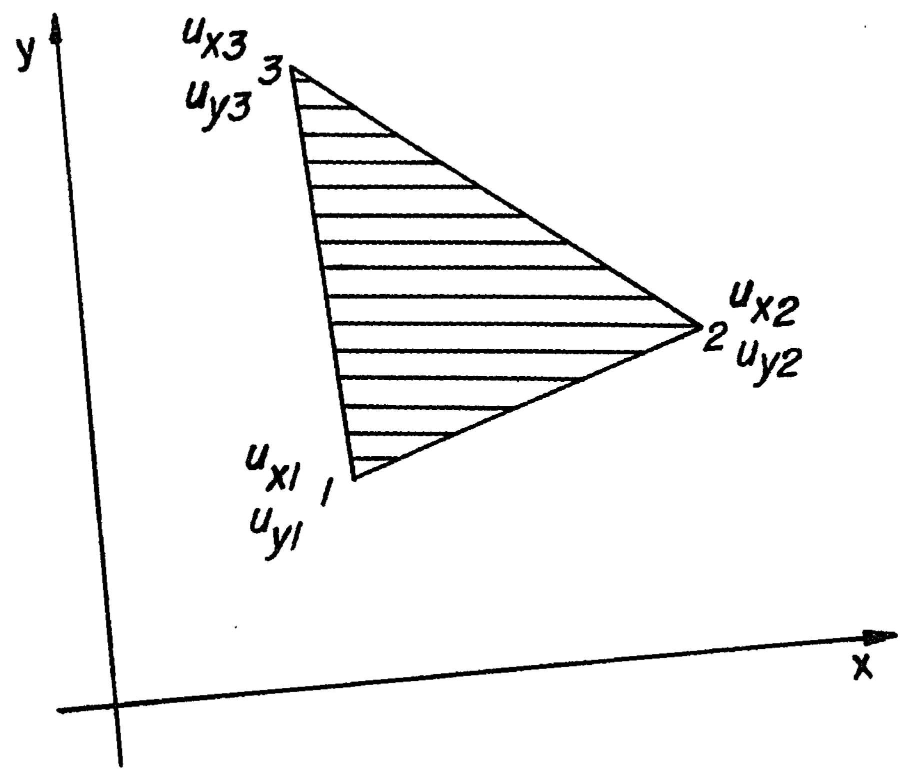


FIG. 2-2 SOLID FINITE ELEMENT

may be expressed as a polynomial in x and y . In this case

$$u_x = \alpha_4 + \alpha_5 x + \alpha_6 y$$
$$u_y = \alpha_7 + \alpha_8 x + \alpha_9 y \qquad (2.2\text{-}1)$$

Eq. (2.2-1) may be employed at the nodes in the same fashion as the pressure in the fluid element, yielding

$$\begin{Bmatrix} u_x \\ u_y \end{Bmatrix} = [N_s(x,y)]\,\{u_e(t)\} = \begin{bmatrix} N_{s1} & 0 & N_{s2} & 0 & N_{s3} & 0 \\ 0 & N_{s1} & 0 & N_{s2} & 0 & N_{s3} \end{bmatrix} \{u_e\} \qquad (2.2\text{-}2)$$

where $\{u_e\}$ is the array of time-dependent nodal displacements and $[N_s]$ is the shape function for the solid triangular element composed of nodal shape functions N_{si} given by

$$N_{si} = \frac{1}{2\,A_s} (a_i + b_i\,x + c_i\,y) \qquad (2.2\text{-}3)$$

in which all terms are defined as for the fluid element. The relationship in eq. (2.2-2) is used in conjunction with the principle of virtual work and the theory of elasticity to yield the matrix differential equation for the nodal displacements of the solid finite element

$$[M_e]\,\{\ddot{u}_e\} + [C_e]\,\{\dot{u}_e\} + [K_e]\,\{u_e\} = \{R_e\} + \{R_e'\} \qquad (2.2\text{-}4)$$

where

The mass matrix $\quad [M_e] = \iint_A [N_s]^T \rho_s [N_s]\, dx\, dy$

The damping matrix $\quad [C_e] = \alpha\,[M_e] + \beta\,[K_e] \qquad (2.2\text{-}5)$

The stiffness matrix $\quad [K_e] = \iint_A [B]^T [D]\,[B]\, dx\, dy$

and $\{R_e\}$ is the applied load array. ρ_s is the density of the solid, $[C_e]$ is the proportional damping matrix related to the mass and stiffness matrices by the damping factors α and β respectively, [B] is the strain-displacement matrix, [D] is the elasticity matrix. $\{R_e'\}$ is the force on the solid due to pressure forces on the boundary, and represents the coupling force which will be evaluated for the solid-fluid superelement. The matrices of eq. (2.2-5) are evaluated in Appendix 1. The boundary conditions which may be applied to eq. (2.2-4) are specified nodal displacements. For a totally solid system, this matrix equation is established for each individual finite element, and is then assembled into a global matrix differentail equation of the same form for solution.

Once the solution is obtained, the stresses may be computed on an element basis by backsubstituting the nodal displacements into the stress-displacement relationship

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D]\,[B]\,\{u_e\} \qquad (2.2\text{-}6)$$

2.3 Solid-Fluid Finite Element

The solid-fluid finite element, as shown in Fig. 2-3, is a super-element consisting of a solid and a fluid finite element. As will be shown later, the equations for the nodal pressures and displacements of the combined element are obtained from the separate fluid and solid parts, but in each case an added term exists which represents the interaction. For the solid part, the interactive term is the pressure force acting normal to the moving boundary. For the fluid part, the interactive term is the inertial force of the moving solid acting on the fluid.

The contribution due to the pressure load on the solid is determined by calculating the work done by the pressure force during the virtual displacement as follows:

$$dW_p = \int_S (du_N)^T\, p\, dS \qquad (2.3\text{-}1)$$

The virtual displacement $d\delta$ is given by

$$d\delta = d \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} = [N_s]\, d\{u_e\} \qquad (2.3\text{-}2)$$

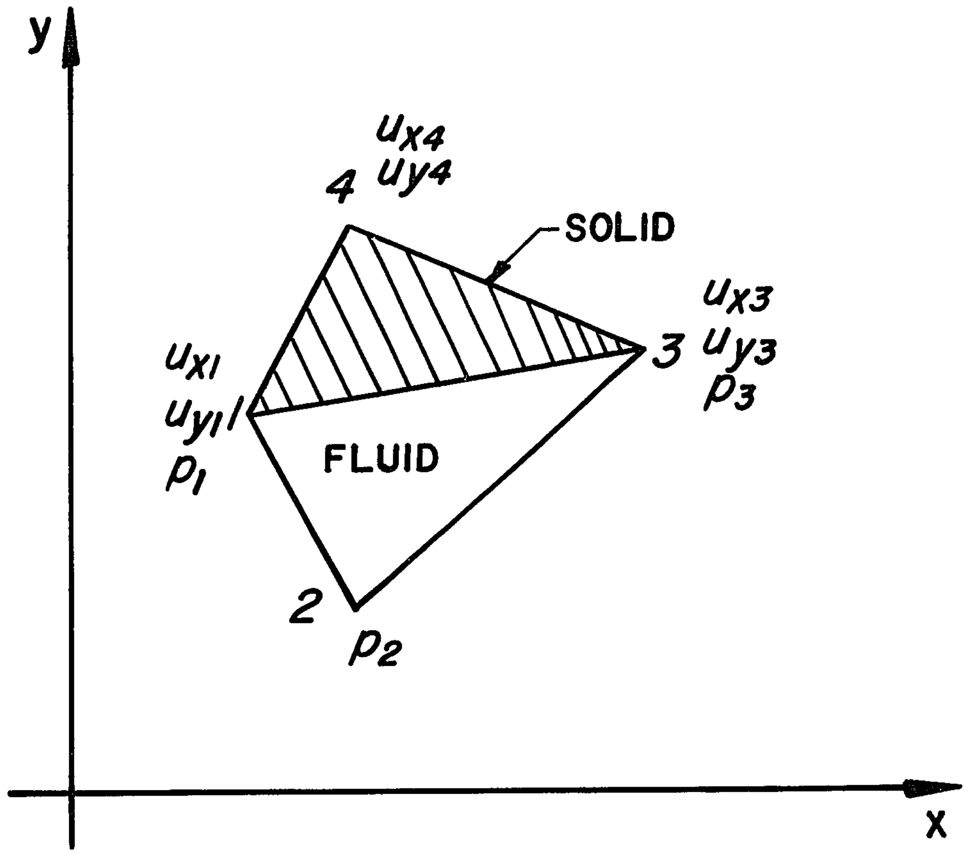


FIG. 2-3 SOLID-FLUID FINITE ELEMENT

The displacement may be resolved into its normal and tangential components du_N and du_T by a coordinate transformation

$$d \begin{Bmatrix} u_T \\ u_N \end{Bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} d \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \tag{2.3-3}$$

in which α is the angle the solid-fluid boundary makes with the horizontal axis, measured counterclockwise. The virtual normal displacement may then be written

$$du_N = [-\sin\alpha \quad \cos\alpha]\,[N_s]\,d\{u_e\} = [\,\bar{N}_s]\,d\{u_e\} \tag{2.3-4}$$

where

$$[\,\bar{N}_s\,] = [-\sin\alpha \quad \cos\alpha]\,[N_s] \tag{2.3-5}$$

$[\,\bar{N}_s\,]$ is the solid boundary shape function. Substituting eq. (2.3-4) into eq. (2.3-1) yields

$$dW_p = d\{u_e\}^T \int_S [\,\bar{N}_s\,]^T\, p\, dS \tag{2.3-6}$$

Employing the shape function of the fluid and the array of nodal pressures, eq. (2.3-6) becomes

$$dW_p = d\{u_e\}^T \int_S [\,\bar{N}_s\,]^T [N_f]\, dS \quad \{p_e\} \tag{2.3-7}$$

The matrix differential equation which results from the inclusion of the coupling term in eq. (2.2-4) is

$$[M_e]\,\{\ddot{u}_e\} + [C_e]\,\{\dot{u}_e\} + [K_e]\,\{u_e\} = \{R_e\} + \frac{1}{\rho_f}\,[S_e]^T\,\{p_e\} \tag{2.3-8}$$

where

$$[S_e] = \int_S [N_f]^T \;\rho_f\, [\,\bar{N}_s\,]\, dS \tag{2.3-9}$$

is the solid-fluid coupling matrix. Eq. (2.3-8) is the matrix differential equation for the solid portion of the solid-fluid superelement.

The contribution due to the inertial load of the solid is determined by evaluating the momentum equation at the boundary. Recalling the matrix differential equation for the general fluid element

$$[G_e]\{\ddot{p}_e\} + [L_e]\{\dot{p}_e\} + [H_e]\{p_e\} = \oint_S [N_f]^T \frac{\partial p}{\partial n}\, dS \qquad (2.1\text{-}9) \text{ rep.}$$

The pressure gradient is expressed in terms of the fluid velocity at the wall using the momentum equation

$$\frac{\partial p}{\partial n} = -\rho_f \frac{\partial v_N}{\partial t} \qquad (2.3\text{-}10)$$

Solid-fluid coupling is established by setting the velocity of the fluid at the wall in eq. (2.3-10) equal to the velocity of the wall itself, resulting in

$$\frac{\partial p}{\partial n} = -\rho_f \ddot{u}_N \qquad (2.3\text{-}11)$$

Employing eq. (2.3-4) for the actual displacement, the pressure gradient becomes

$$\frac{\partial p}{\partial n} = -\rho_f [\,\overline{N}_s\,]\{\ddot{u}_e\} \qquad (2.3\text{-}12)$$

Substituting this into the matrix differential equation (eq. (2.1-9) rep.) yields

$$[G_e]\{\ddot{p}_e\} + [L_e]\{\dot{p}_e\} + [H_e]\{p_e\} = -\int_S [N_f]^T \rho_f [\,\overline{N}_s\,]\, dS\, \{\ddot{u}_e\} \qquad (2.3\text{-}13)$$

or finally,

$$[G_e]\{\ddot{p}_e\} + [L_e]\{\dot{p}_e\} + [H_e]\{p_e\} = -[S_e]\{\ddot{u}_e\} \qquad (2.3\text{-}14)$$

Eq. (2.3-14) is the matrix equation for the fluid portion of the solid-fluid superelement. Equations (2.3-8) and (2.3-14) are rearranged into the form

$$\begin{bmatrix} M_e & 0 \\ 0 & G_e \end{bmatrix} \begin{Bmatrix} \ddot{u}_e \\ \ddot{p}_e \end{Bmatrix} + \begin{bmatrix} C_e & 0 \\ 0 & L_e \end{bmatrix} \begin{Bmatrix} \dot{u}_e \\ \dot{p}_e \end{Bmatrix} + \begin{bmatrix} K_e & 0 \\ 0 & H_e \end{bmatrix} \begin{Bmatrix} u_e \\ p_e \end{Bmatrix} = \begin{Bmatrix} \frac{1}{\rho_f}[S_e]^T\{p_e\} \\ -[S_e]\{\ddot{u}_e\} \end{Bmatrix} + \begin{Bmatrix} R_e \\ 0 \end{Bmatrix} \qquad (2.3\text{-}15)$$

Eq. (2.3-15) is the matrix differential equation for the solid-fluid finite element. The boundary conditions for eq. (2.3-15) are the same as those indicated for the solid and fluid parts.

The applicability of this analysis is not necessarily limited to plane triangular elements. More complex structural elements, with their corresponding mass and stiffness matrices, can also be used to model the solid structure in this solid-fluid interaction formulation.

3 METHOD OF SOLUTION

3.1 Assembly and Reduction of System Matrices

The global matrices are obtained from the elemental matrices by establishing equilibrium throughout the system. The elemental matrices and arrays from equations (2.1-9), (2.2-4), and (2.3-15) are assembled into global matrices and arrays of the form

$$\begin{bmatrix} M & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \ddot{u} \\ \ddot{p} \end{Bmatrix} + \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{p} \end{Bmatrix} + \begin{bmatrix} K & 0 \\ 0 & H \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} = \begin{Bmatrix} \frac{1}{\rho_f} [S]^T \{p\} \\ - [S] \{\ddot{u}\} \end{Bmatrix} + \begin{Bmatrix} R \\ 0 \end{Bmatrix} \qquad (3.1\text{-}1)$$

For the totally solid or fluid finite elements, zero loads are assembled for the coupling terms. The global matrices and arrays are then reduced to condensed form for modal analysis. The specified displacements and inactive freedoms are removed, so that only active nodes are considered in the modal analysis.

The unrestrained global stiffness matrix is singular, so that some constraints must be applied to the system or else the system will show rigid body modes. These constraints are in the form of specified freedoms (displacements and pressures). The specified freedoms are entered into the global equation by inserting the freedom at the proper node, evaluating the force or effect caused by the known freedom, and transferring this effect to the right hand side of the equation. The rows and columns in the global matrices associated with the specified freedoms are then removed for modal analysis.

A further reduction of the system equations is performed for nodes which have inactive freedoms. Each node in the system is assumed to have three degrees of freedom--u_x , u_y , and p . However, only solid-fluid nodes have all three freedoms active. Fluid nodes have only pressure as the active freedom, and for solid nodes only the displacements are active. The inactive freedoms from the solid and fluid nodes are also removed for modal analysis. The final reduced form of the system matrix differential equation for the condensed array of nodal freedoms $\{x\}$ is then

$$[\,\bar{M}\,]\,\{\ddot{x}\} + [\,\bar{C}\,]\,\{\dot{x}\} + [\,\bar{K}\,]\,\{x\} = \{F\} \tag{3.1-2}$$

in which $[\,\bar{M}\,]$, $[\,\bar{C}\,]$, and $[\,\bar{K}\,]$ are the real symmetric condensed mass, damping, and stiffness matrices for the system, and $\{F\}$ is the condensed load array. This latter array consists of the loads applied to the solid, the loads due to specified displacements and pressures, and the solid-fluid interaction loads.

3.2 Modal Superposition and Semi-Analytical Solution

The solution to eq. (3.1-2) is found by the modal superposition technique [20] . The eigenvalues and eigenvectors associated with the system are found and used to form the generalized mass and stiffness matrices. This uncouples the equations and allows an analytic solution in terms of the generalized coordinates. Transforming the dependent variable $\{x\}$ to the generalized variable $\{q\}$ by

$$\{x\} = [\phi]\,\{q\} \tag{3.2-1}$$

where $[\phi]$ is the modal matrix, and substituting this relationship into equation (3.1-2) yields

$$[\,\bar{M}\,]\,[\phi]\,\{\ddot{q}\} + [\,\bar{C}\,]\,[\phi]\,\{\dot{q}\} + [\,\bar{K}\,]\,[\phi]\,\{q\} = \{F\} \tag{3.2-2}$$

Premultiplying eq. (3.2-2) by the transpose of the modal matrix results in

$$[\phi]^T[\,\bar{M}\,]\,[\phi]\,\{\ddot{q}\} + [\phi]^T[\,\bar{C}\,]\,[\phi]\,\{\dot{q}\} + [\phi]^T[\,\bar{K}\,]\,[\phi]\,\{q\} = [\phi]^T\{F\} \tag{3.2-3}$$

For normalized eigenvectors, this equation becomes [25]

$$\{\ddot{q}\} + [2\,\zeta\,\omega_n]\,\{\dot{q}\} + [\omega_n^{\,2}]\,\{q\} = [\phi]^T\{F\} \tag{3.2-4}$$

in which ζ represents the modal damping ratio, and ω_n is the natural frequency. All matrices are now diagonalized and the matrix equation is completely uncoupled and may be solved analytically. The initial displacements and velocities needed to solve eq. (3.2-4) must also be transformed to generalized coordinates in order to be used in the solution. This transformation is accomplished in the following manner. The initial displacements are given by

$$\{x_o\} = [\phi]\ \{q_o\} \tag{3.2-5}$$

Premultiplying eq. (3.2-5) by the matrix product $[\phi]^T[\ \bar{M}\ ]$ yields

$$[\phi]^T[\ \bar{M}\ ]\ \{x_o\} = [\phi]^T[\ \bar{M}\ ]\ [\phi]\ \{q_o\} \tag{3.2-6}$$

For normalized eigenvectors, the matrix product $[\phi]^T[\ \bar{M}\ ]\ [\phi]$ is equal to the identity matrix $[I]$. Thus, the initial displacements in terms of generalized coordinates are

$$\{q_o\} = [\phi]^T[\ \bar{M}\ ]\ \{x_o\} \tag{3.2-7}$$

The same transformation is valid for the array of initial velocities as well. Once the array of unknown generalized freedoms are found, eq. (3.2-1) is used to transform them back to the actual freedoms $\{x\}$. The elemental stresses and velocities may then be found using the nodal displacements and pressures in the manner described in Sections 2.1 and 2.2 . The accelerations are transformed in the same way so that they may be used to update the coupling loads.

If the rigid wall case is being considered and the applied loads and specified freedoms are time independent, the solution is completely analytic and is independent of the time step. However, in the flexible wall case, the coupling loads are functions of the nodal pressures and accelerations. They must be recalculated after every integration time step and the updated values are used as the coupling loads in the next time step. Therefore, in any iteration, the load is constant and the solution is analytic within that time step alone.

The solution to eq. (3.2-4) is of three types: 1) the underdamped solution, in which the modal damping ratio, ζ , is less than 1 ; 2) the critically damped solution, in which $\zeta = 1$; and 3) the overdamped solution, in which $\zeta > 1$. The underdamped solution at any row is given by

$$q = e^{-\zeta \omega_n t} [A \cos \omega_n \sqrt{1 - \zeta^2}\, t + B \sin \omega_n \sqrt{1 - \zeta^2}\, t] + f/\omega_n^2 \qquad (3.2\text{-}8)$$

in which

$$A = q_o - f/\omega_n^2$$

$$B = \frac{1}{\omega_n^2} [\dot{q}_o + \zeta \omega_n (q_o - f/\omega_n^2)]$$

and f is the corresponding element of the generalized force vector $[\phi]^T \{F\}$. The critically damped solution at any row is given by

$$q = A e^{-\omega_n t} + C t e^{-\omega_n t} + f/\omega_n^2 \qquad (3.2\text{-}9)$$

in which

$$C = \dot{q}_o + \omega_n (q_o - f/\omega_n^2)$$

and all other terms are defined as for the underdamped case. The overdamped solution at any row is given by

$$q = D e^{r_1 t} + E e^{r_2 t} + f/\omega_n^2 \qquad (3.2\text{-}10)$$

in which

$$r_1 = - \omega_n (\zeta - \sqrt{\zeta^2 - 1})$$

$$r_2 = -\omega_n(\zeta + \sqrt{\zeta^2 - 1})$$

$$D = \frac{1}{(r_2 - r_1)} [(q_o - f/\omega_n^2) r_2 - \dot{q}_o] \qquad (3.2\text{-}11)$$

$$E = \frac{-1}{(r_2 - r_1)} [(q_o - f/\omega_n^2) r_1 - \dot{q}_o]$$

3.3 NASTRAN Direct Integration Method

The WECAN program, which was used to verify the results from FLINTS is a large general purpose finite element program which solved eq. (3.1-2) using the NASTRAN integration method [14]. This integration scheme employs an elementary central difference formula for the acceleration and velocity terms and an implicit representation of the displacement term, as derived by Chan, Cox, and Benfield [4] in their extension of the Newmark β method [18]

$$\frac{1}{\Delta t^2}[\bar{M}]\{x^{n+2} - 2x^{n+1} + x^n\} + \frac{1}{2\Delta t}[\bar{C}]\{x^{n+2} - x^n\} + [\bar{K}]\{\beta x^{n+2} + (1-2\beta)x^{n+1} + \beta x^n\} = \{\beta\{F\}^{n+2} + (1-2\beta)\{F\}^{n+1} + \beta\{F\}^n\} \qquad (3.3\text{-}1)$$

where n, n + 1, and n + 2 are the past, present, and future (updated) values of the variables and β is a parameter to be selected on the basis of numerical stability and accuracy. With $\beta = \frac{1}{3}$, eq. (3.3-1) reduces to

$$\left[\frac{1}{\Delta t^2}[\bar{M}] + \frac{1}{2\Delta t}[\bar{C}] + \frac{1}{3}[\bar{K}]\right]\{x^{n+2}\} = \frac{1}{3}\{\{F\}^{n+2} + \{F\}^{n+1} + \{F\}^n\} + \left[\frac{2}{\Delta t^2}[\bar{M}] - \frac{1}{3}[\bar{K}]\right]\{x^{n+1}\} + \left[-\frac{1}{\Delta t^2}[\bar{M}] + \frac{1}{2\Delta t}[\bar{C}] - \frac{1}{3}[\bar{K}]\right]\{x^n\} \qquad (3.3\text{-}2)$$

which constitutes a set of linear algebraic equations and is solved by a variation of the Gaussian elimination method known as the wave front technique. The solution is therefore obtained directly without any further transformation as needed in the modal superposition solution.

4 VERIFICATION OF RESULTS

4.1 Rigid Wall Case

The fluid finite element model, developed in this study, is verified for a two-dimensional channel, 24.5 feet long and 2 feet wide with rigid walls, shown in Fig. 4-1. Water initially at rest is accelerated suddenly by applying a step pressure p_o at $x = 0$ while maintaining a zero pressure at $x = L$. All geometric dimensions and flow variables used in this analysis are nondimensionalized so that the final solution is valid for similar problems. The numerical results are verified by comparison with the analytical solution presented in Appendix 2.

Three types of finite element grids were considered: (1) 48 triangular element, uniform grid model (Fig. 4-2); (2) 48 triangular element, non-uniform grid model (Fig. 4-3); and (3) 108 quadrilateral element, uniform grid model (Fig. 4-4). These finite element models were analyzed using WECAN, a large general purpose program whose dynamic capability was extended by incorporating into it fluid and solid-fluid finite elements based on this study. The matrix differential equation for these totally fluid problems was solved using the NASTRAN integration method and the wave front technique as discussed in the previous section.

Three values of viscous damping are studied: (1) inviscid case, $k_f = 0$; (2) slightly viscous case, $k_f = 64.0816 \text{ sec}^{-1}$; (3) highly viscous case, $k_f = 1046.666 \text{ sec}^{-1}$.

Response time histories were obtained both numerically and analytically for all nine cases indicated above. Typical nodal pressure and elemental velocity for one position upstream and one position downstream were plotted

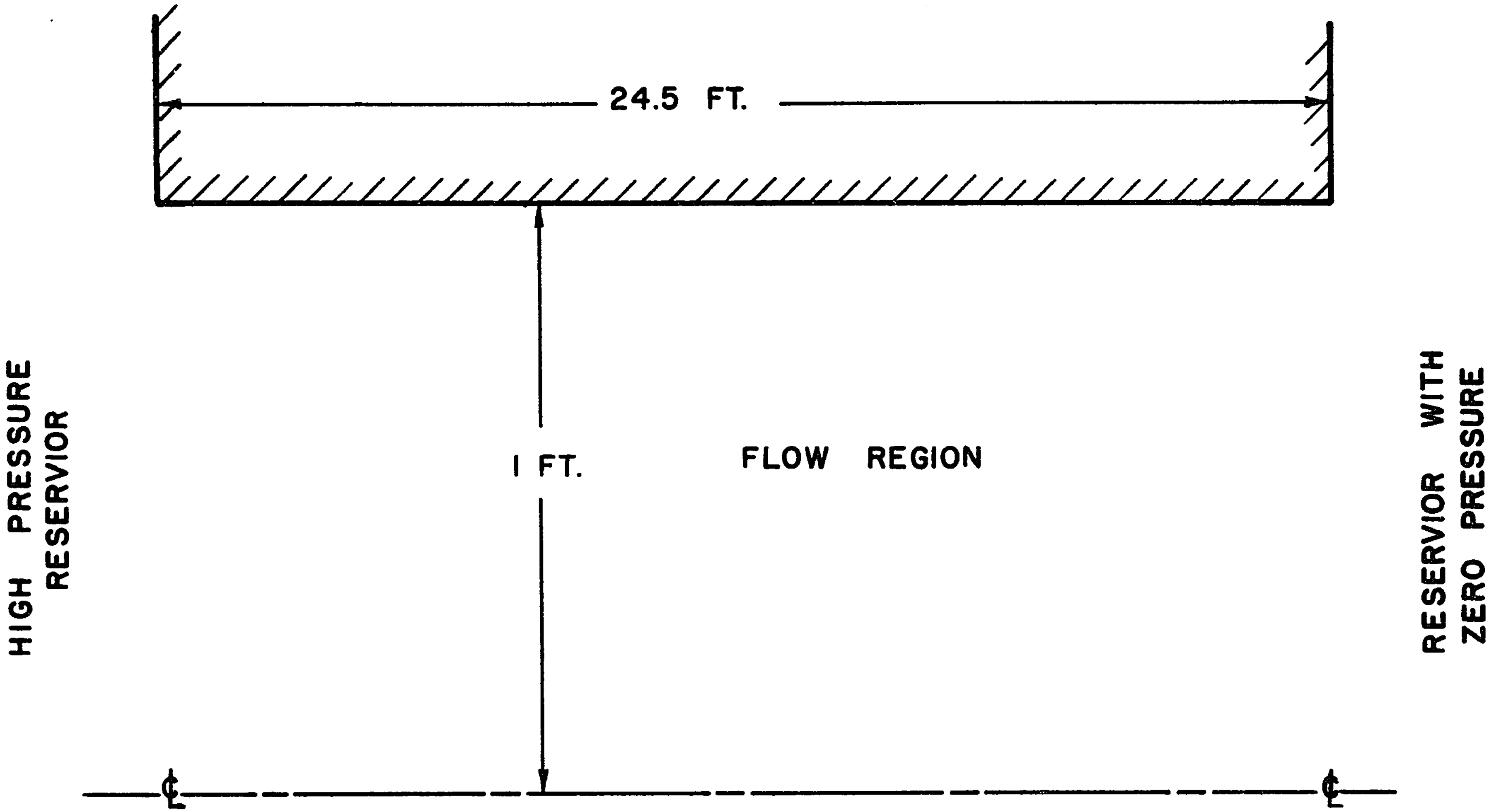


FIG. 4-1 RIGID WALL FLOW CONFIGURATION

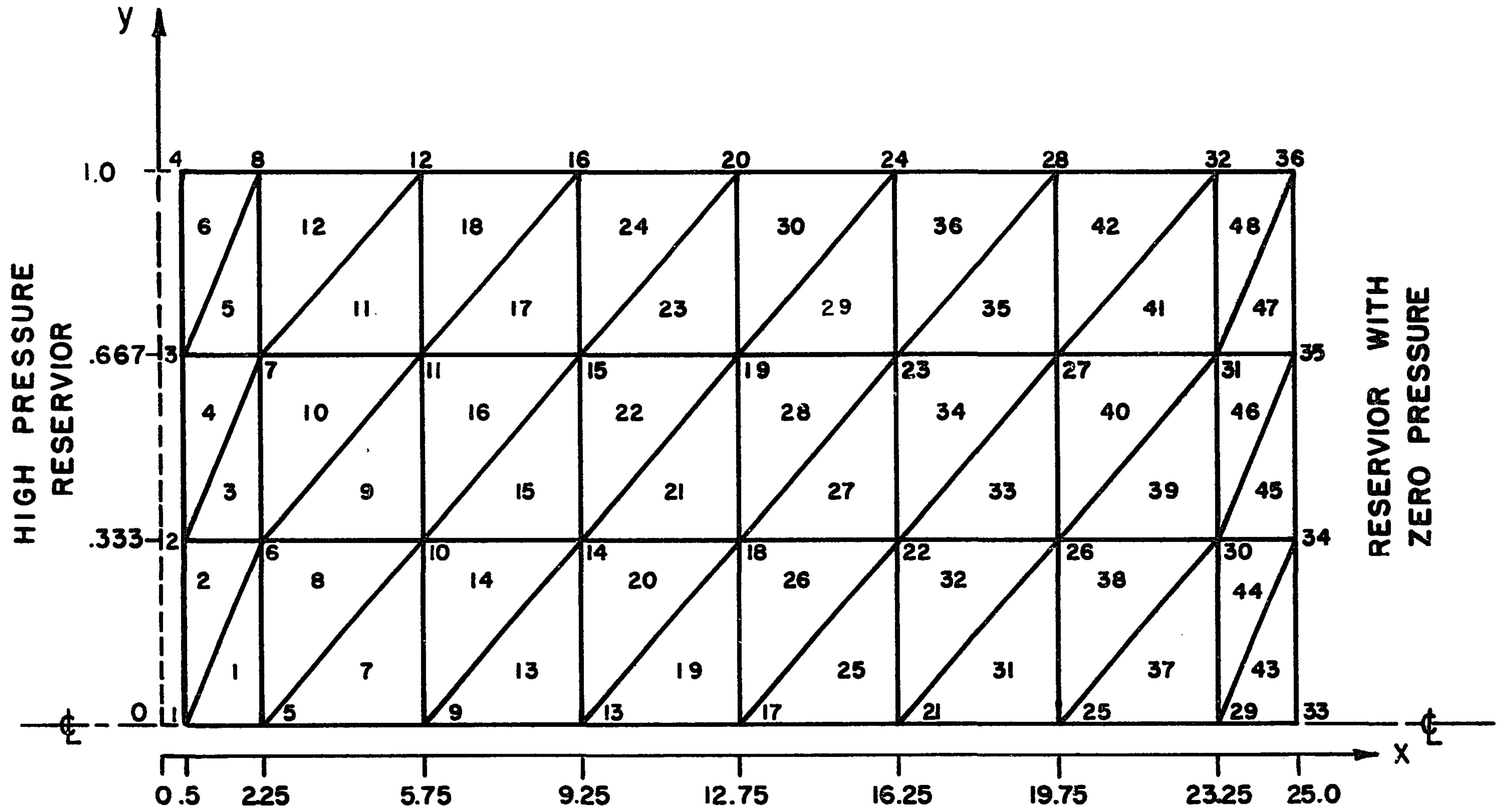


FIG. 4-2 48 TRIANGULAR ELEMENT, UNIFORM GRID MODEL

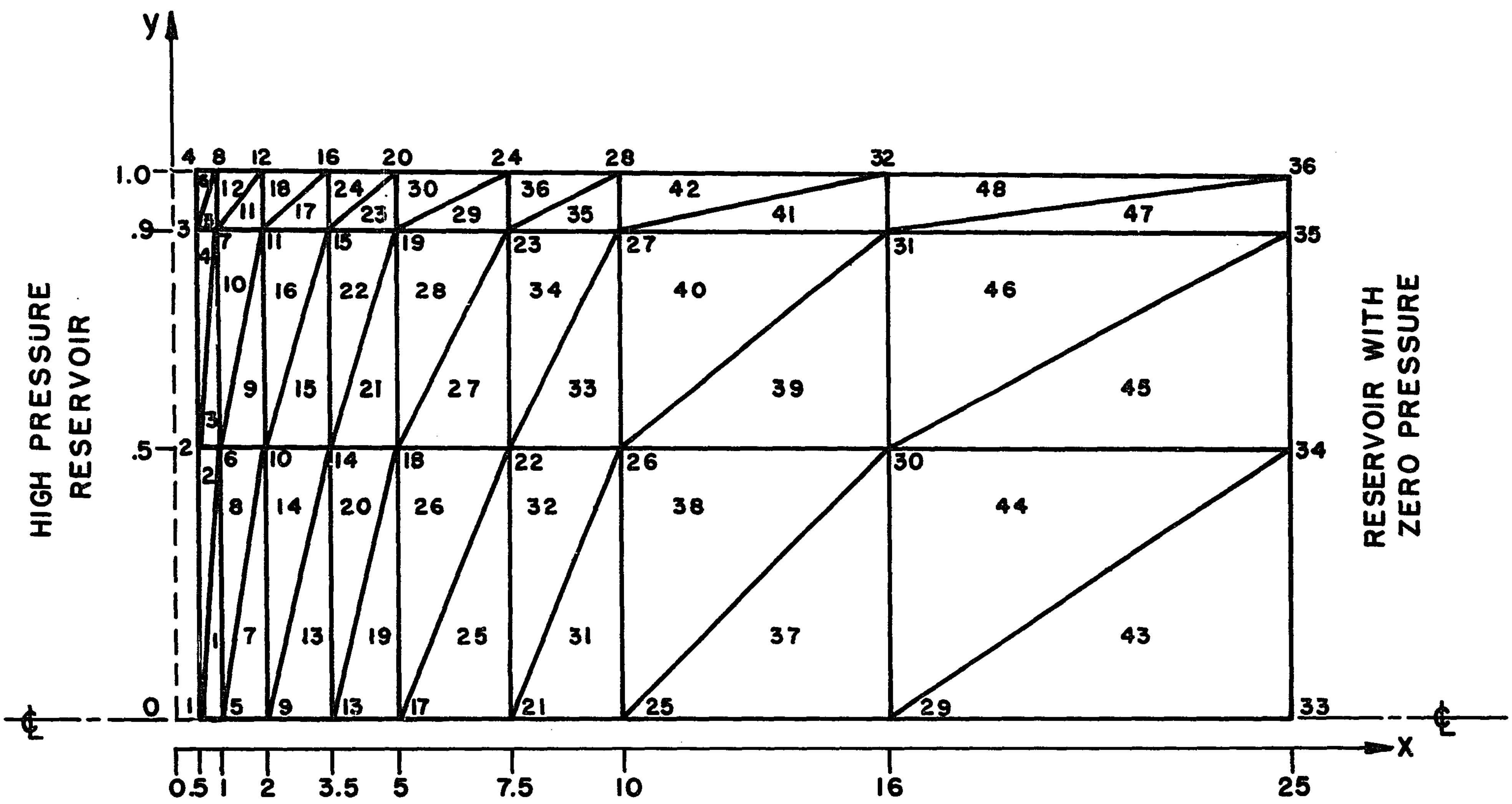


FIG. 4-3 48 TRIANGULAR ELEMENT, NON-UNIFORM GRID MODEL

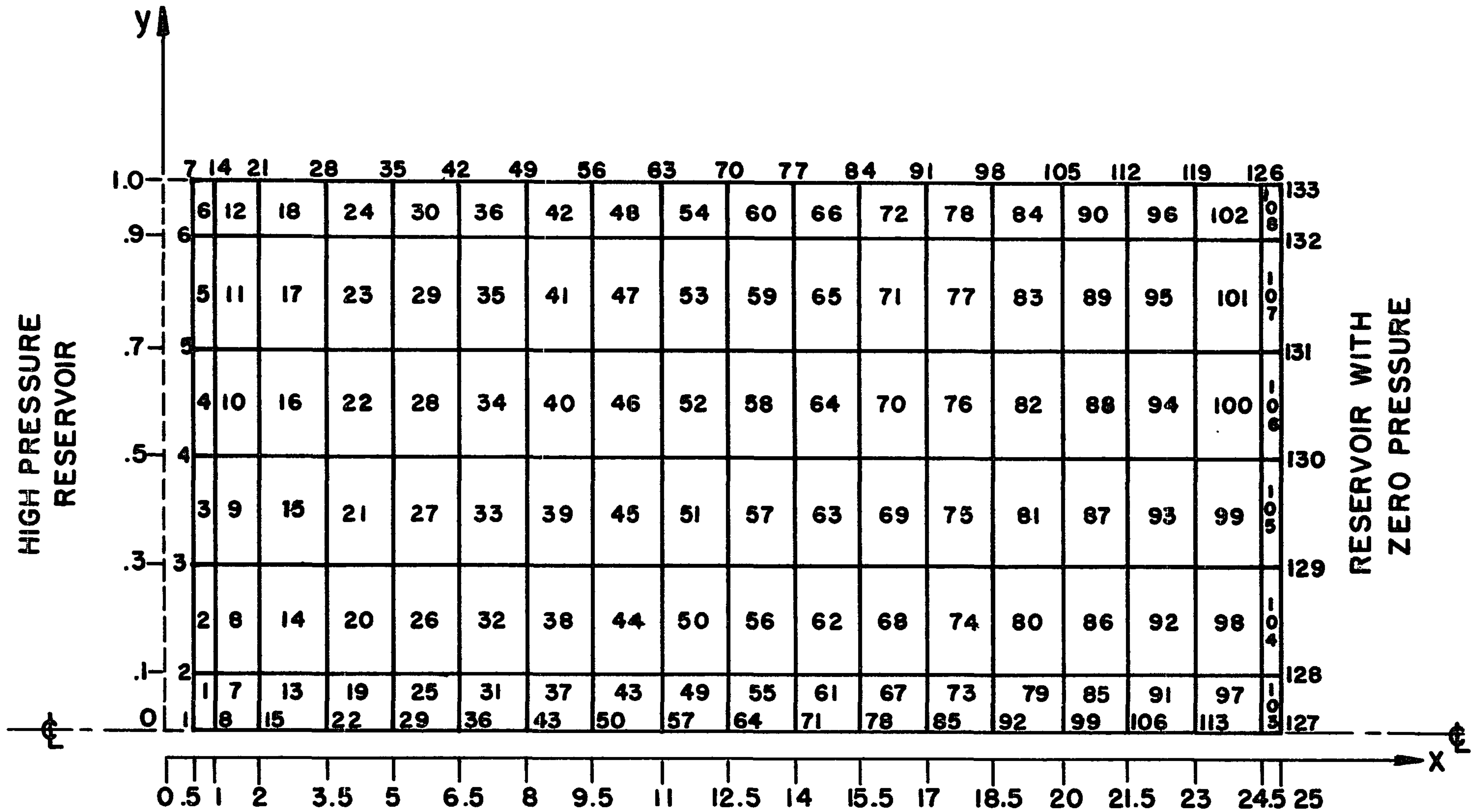


FIG. 4-4 108 QUADRILATERAL ELEMENT, UNIFORM GRID MODEL and compared. In view of the element size difference among the various models, the upstream and downstream positions could not be selected to be the same as seen in Table 1 . For the sake of brevity, only six sets of the above time histories are shown here in Figs. 4-5 through 4-10.

The direct integration method employed by WECAN for this portion of the study is an implicit numerical integration scheme and as such it exhibits high degree of numerical stability. However, if the integration time step exceeds the upper bound required for convergence, the numerical solution will diverge from the true solution. The integration time step for the integration scheme is calculated based on the following considerations.

In view of the symmetry of the flow with respect to the x-axis, pressure at all nodes having the same abscissa are equal and transversal oscillations will not be excited. The minimum period for longitudinal oscillations is equal to the product of 2 π times the ratio of the smallest element length (0.5 ft) to the speed of sound (5000 ft/sec), i.e., 0.000628 sec . Examination of the computer results showed that this period was indeed present in the nodal time histories on the element of the smallest length. The integration time step was then set equal to one-fiftieth of this minimum period, i.e., 0.0000125 sec . This time increment led to a convergent solution for all cases studied. The convergence of the solution was established when doubling the time interval produced no change in the pressure and velocity time histories. A detailed examination of the response time histories reveals the following features.

The inviscid flow pressure and velocity time histories, typically

TABLE 1--UPSTREAM AND DOWNSTREAM

POSITIONS PLOTTED FOR RIGID WALL CASE

Fig. No.	Upstream position Node no.	Upstream position x/L	Downstream position Node no.	Downstream position x/L
4-2	9	0.235	25	0.806
4-3	17	0.204	29	0.653
4-4	29	0.204	92	0.755

TABLE 2--PHYSICAL DATA FOR RIGID WALL CASE

Parameter	Units
Density	9.35521×10^{-5} lbf-sec^2/in^4
Speed of sound	6.0×10^4 in/sec
Pressure at x = 0	0.1 psi
Channel length	294 in
Channel width	24 in
Viscous damping:	
inviscid	0.0 sec^{-1}
slightly viscous	64.0816 sec^{-1}
highly viscous	1046.666 sec^{-1}

48 TRIANGULAR ELEMENT, UNIFORM GRID MODEL, INVISCID FLOW

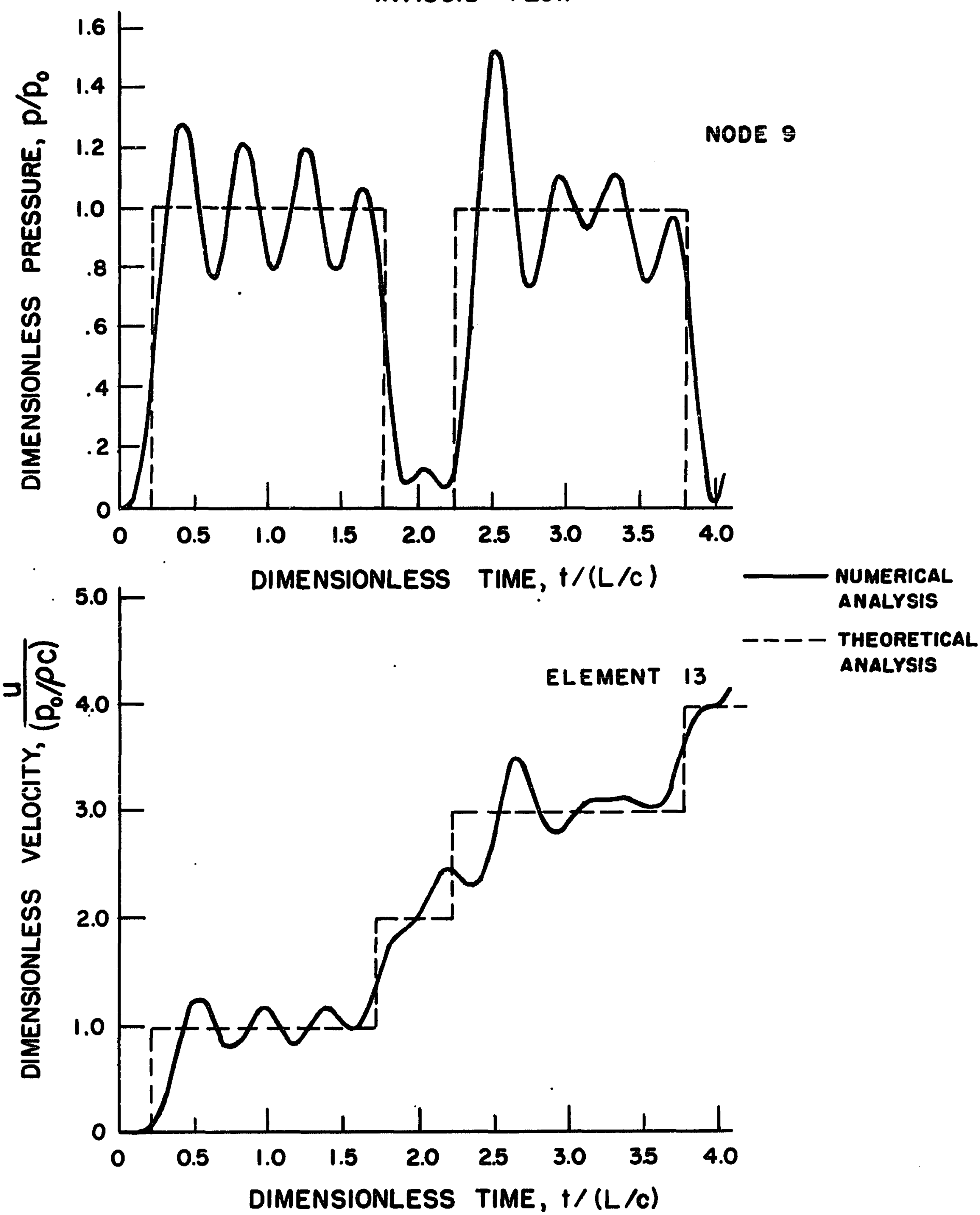


FIG. 4-5 TIME HISTORIES AT X/L=0.235

48 TRIANGULAR ELEMENT, UNIFORM GRID MODEL, INVISCID FLOW

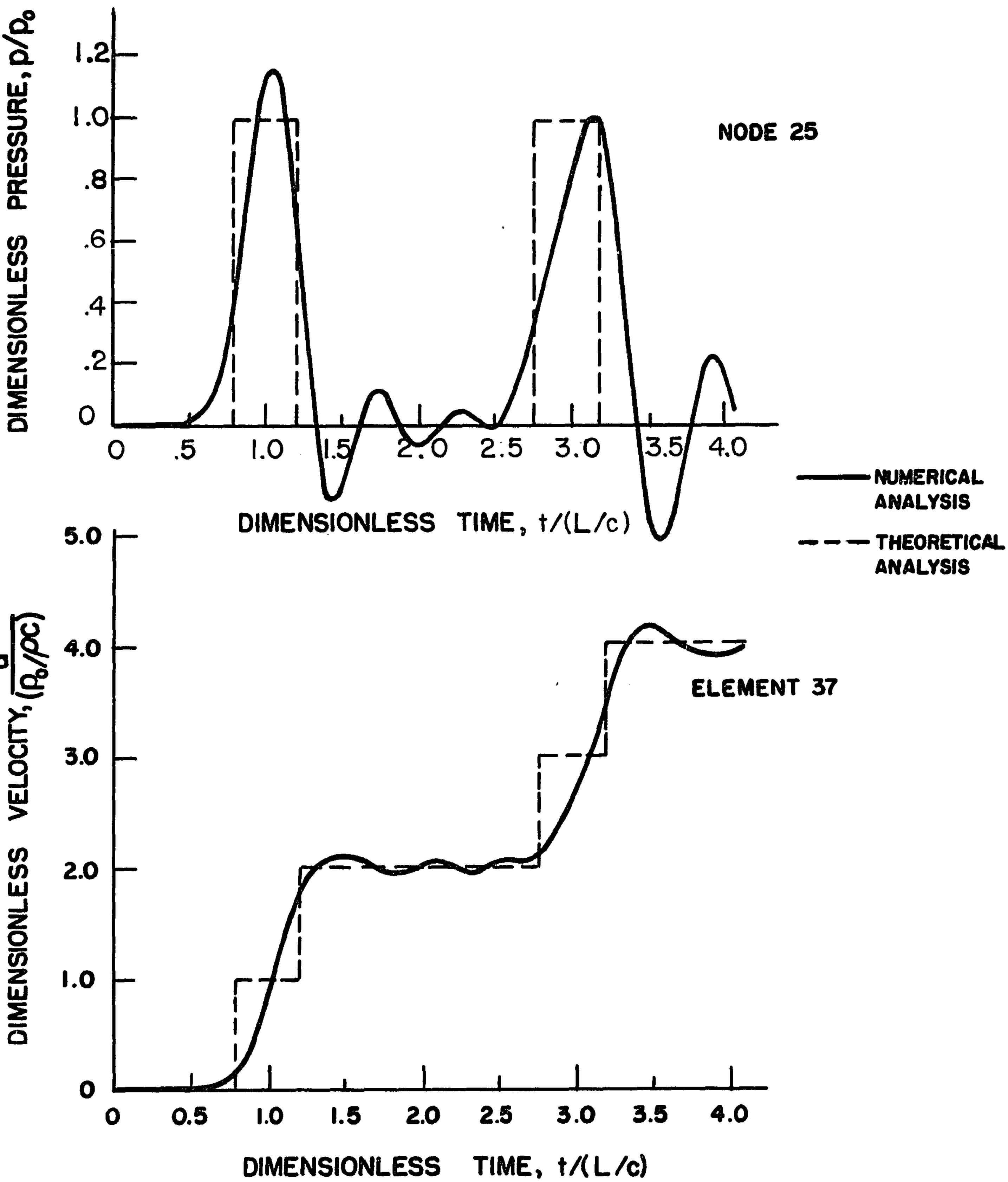


FIG. 4-6 TIME HISTORIES AT X / L=0.806

shown in Figs. 4-5 and 4-6, exhibit an oscillatory behavior about the analytical solution. The pressure and velocity responses for the upstream and downstream nodes oscillate about the analytically-calculated rectangular and staircase wave form, respectively. This oscillatory behavior was also observed by Conway and Jakubowski [6], who investigated wave propagation in axially impacted bars of short length, experimentally and analytically. This problem and the fluid wave propagation problem being examined in this study are governed ideally by the classical one-dimensional wave equation whose solution is the step function presented in Appendix 2. The one-dimensional wave equation allows only axial motion or pressure variation and therefore cannot be expected to be accurate for a finite system. Conway and Jakubowski used Love's theory [13] to obtain a corrected one-dimensional partial differential equation for the motion of the bar that would account for radial motion. Their results agreed closely with experimental results and with the finite element solution presented herein. This indicates that the finite element model for wave propagation yields results which correspond more closely with observed experimental behavior than with the classical solution. This is due to two factors. The first is the discretization of the domain, which allows inter-element response to effect the solution. This type of response is not observed analytically. The second reason is that the wave equation in two-dimensions is being used to develop the finite element model of the fluid. This takes into account any variation of the pressure in the transversal direction, which is what Conway and Jakubowski attempted to accomplish with Love's theory.

The pressure and velocity time histories for slightly viscous flow, typically shown in Figs. 4-7 and 4-8, exhibit a pattern similar to the

48 TRIANGULAR ELEMENT, NON-UNIFORM GRID MODEL, SLIGHTLY VISCOUS FLOW

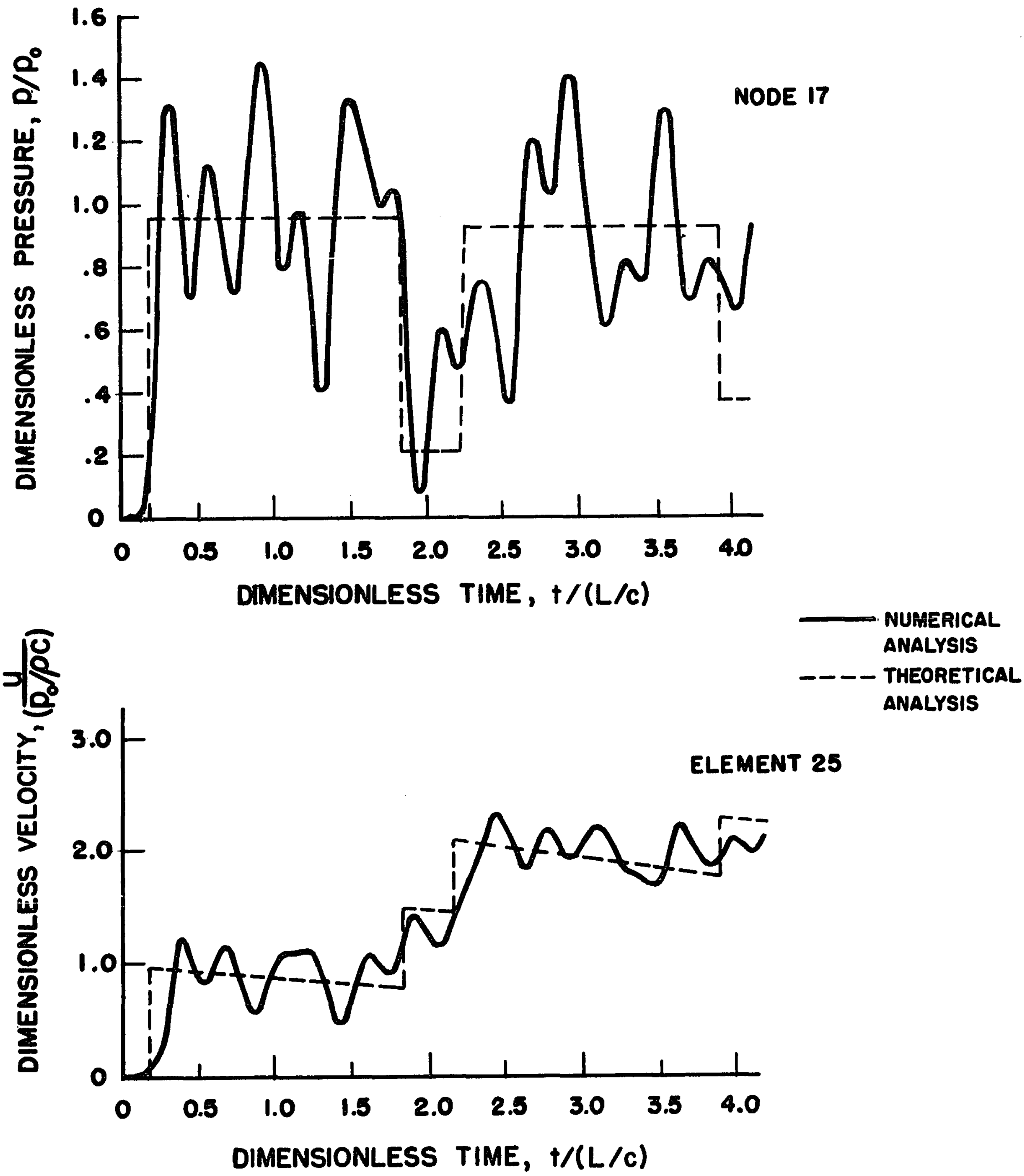


FIG. 4-7 TIME HISTORIES AT X/L = 0.204

48 TRIANGULAR ELEMENT, NON-UNIFORM GRID MODEL, SLIGHTLY VISCOUS FLOW

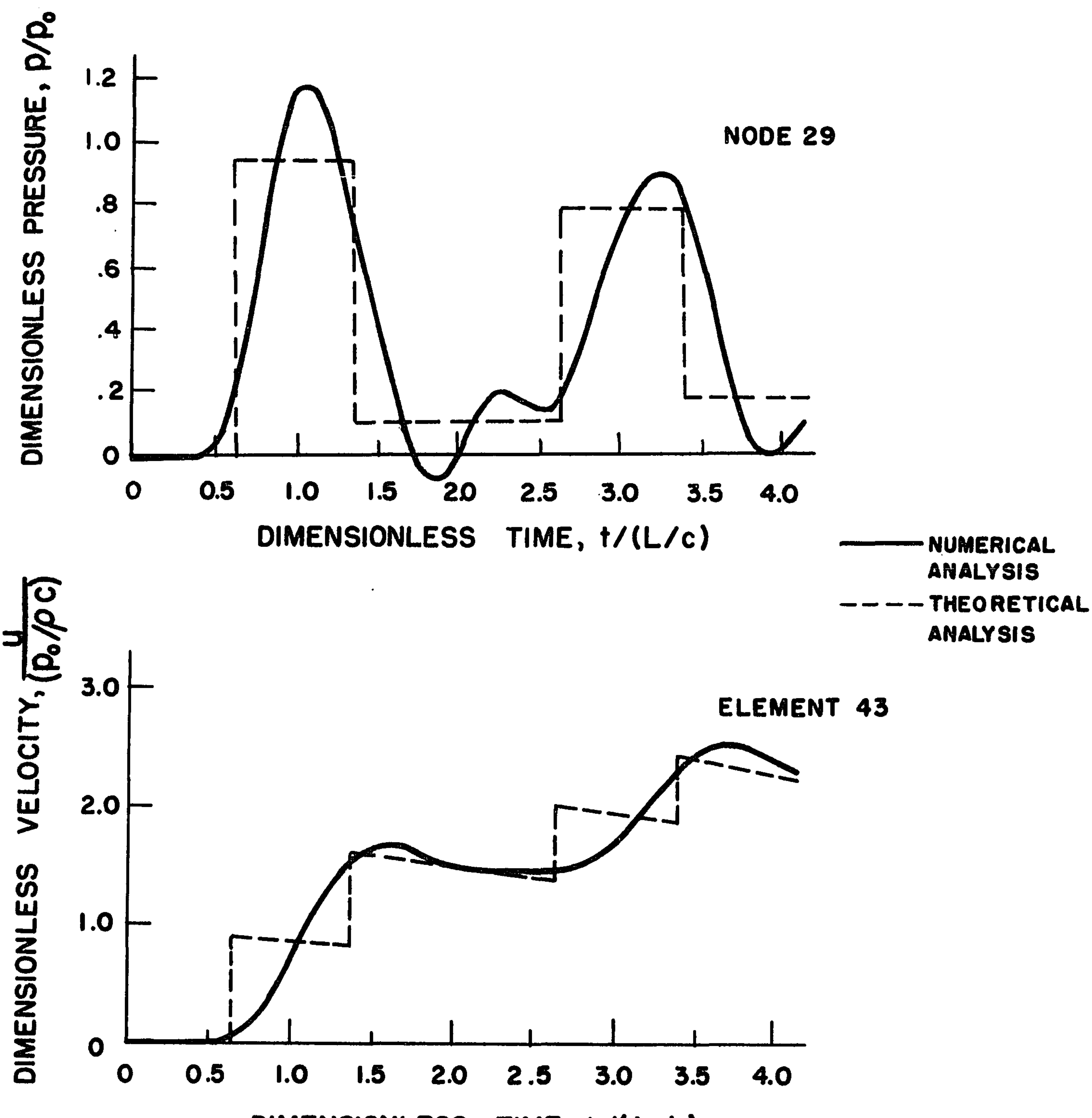


FIG. 4-8 TIME HISTORIES AT X/L=0.653

inviscid flow case and oscillate about the analytical solution. The pressure-time history fluctuations cease to approach closely to the time axis indicating that ultimately a steady state pressure distribution will be reached. Similarly, for the velocity time histories, the staircase wave form changes its shape, the distance between the steps becomes shorter and the slope of each step becomes increasingly negative, again indicating approach toward a fixed steady state value.

The pressure and velocity time histories for highly viscous flow, typically shown in Figs. 4-9 and 4-10, exhibit a highly damped behavior and closely follow the analytical solution. All pressure and velocity responses approach their final steady state values. The physical data for the rigid wall cases studied herein are presented in Table 2.

4.2 Flexible Wall Case

The solid-fluid finite element model developed in this study is verified for a two-dimensional channel, 29.4 inches[1] long, 2 feet wide, and having flexible walls 1/2 inch thick, shown in Fig. 4-11. Water initially at rest is accelerated suddenly by applying a step pressure p_o at the left end while maintaining a zero pressure at the right end. All variables and geometric dimensions have been nondimensionalized so that the final solution is valid for similar problems.

A 72 element grid model, shown in Fig. 4-12, is used in this analysis. Elements 6, 15, 24, 33, 42, 51, 60, and 69 are modeled using solid-fluid quadrilateral finite elements. Elements located below and above these elements are modeled employing fluid and solid triangular finite elements respectively.

[1]The length of the channel in the flexible wall case was reduced from the length used in the rigid wall case in order to reduce computer time.

108 QUADRILATERAL ELEMENT, UNIFORM GRID MODEL, HIGHLY VISCOUS FLOW

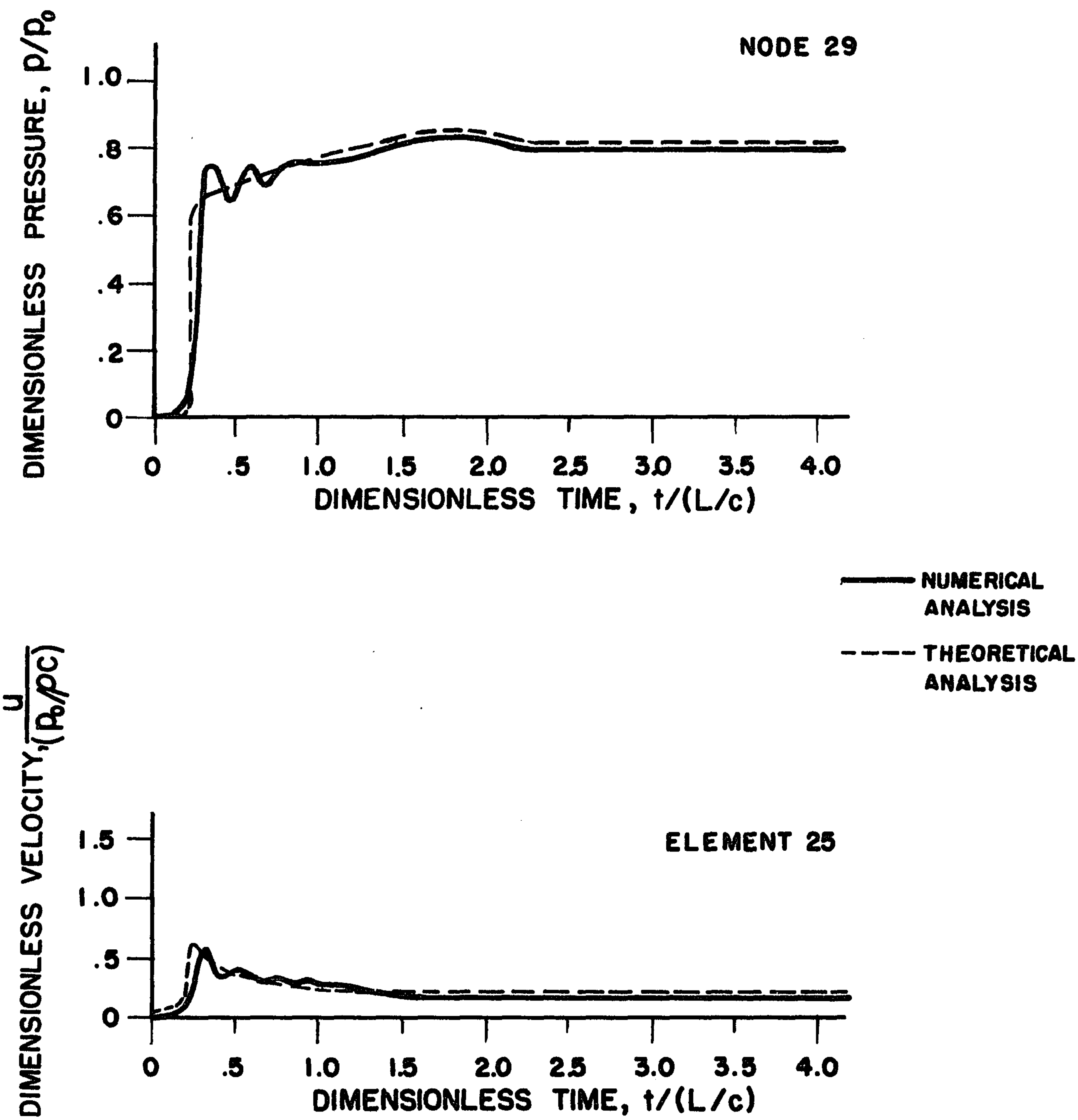
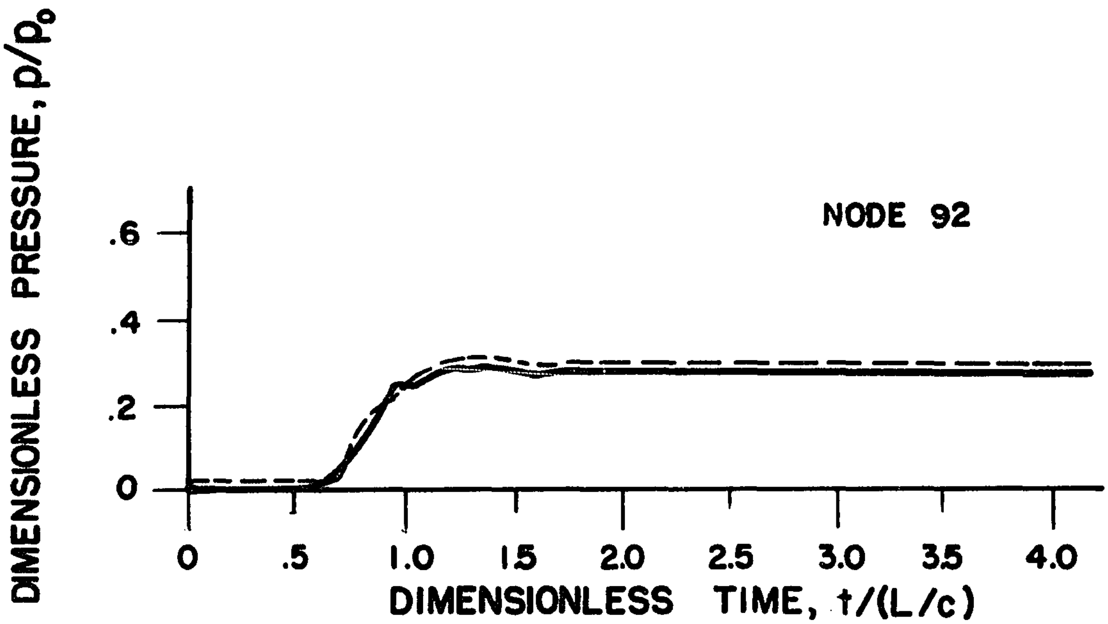


FIG. 4-9 TIME HISTORIES AT X/L=0.204

108 QUADRILATERAL ELEMENT, UNIFORM GRID MODEL,
HIGHLY VISCOUS FLOW



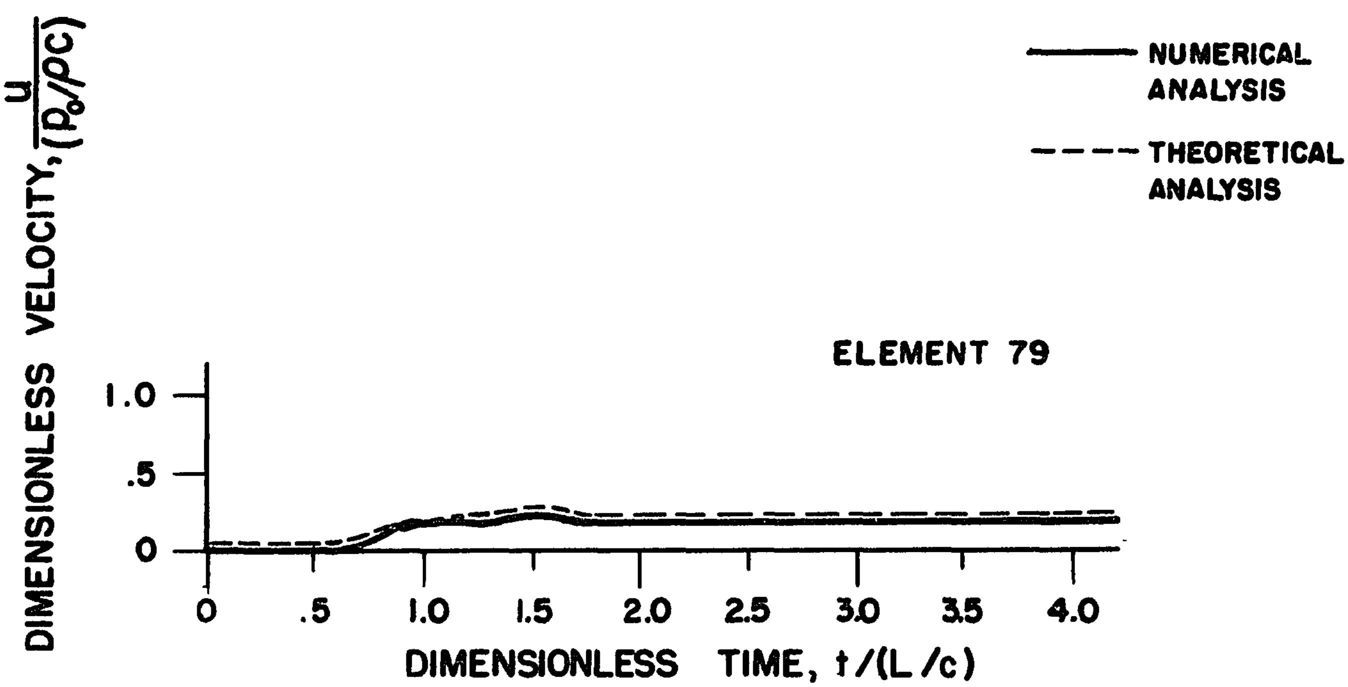


FIG. 4-10 TIME HISTORIES AT X/L=0.755

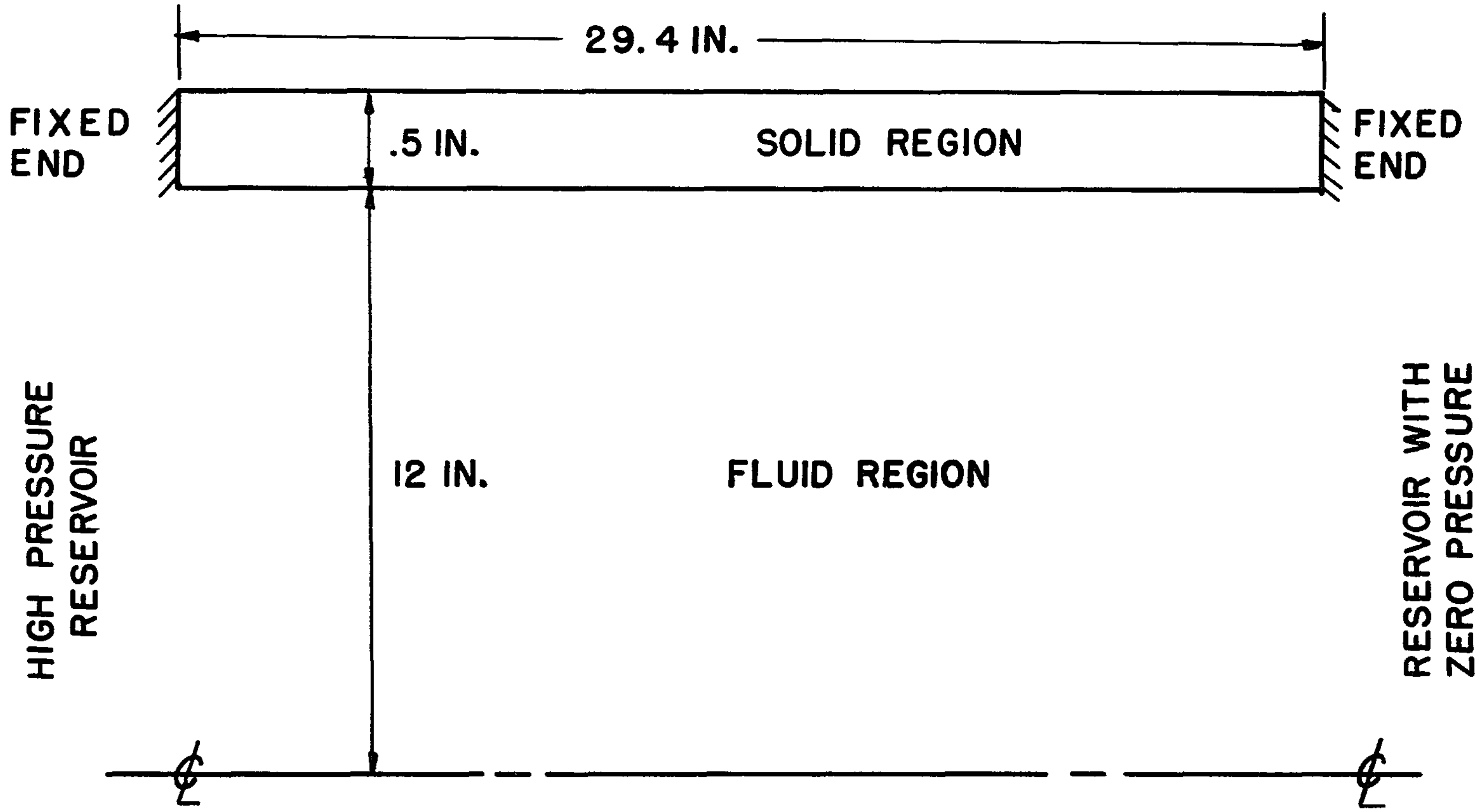


FIG. 4-11 FLEXIBLE WALL FLOW CONFIGURATION

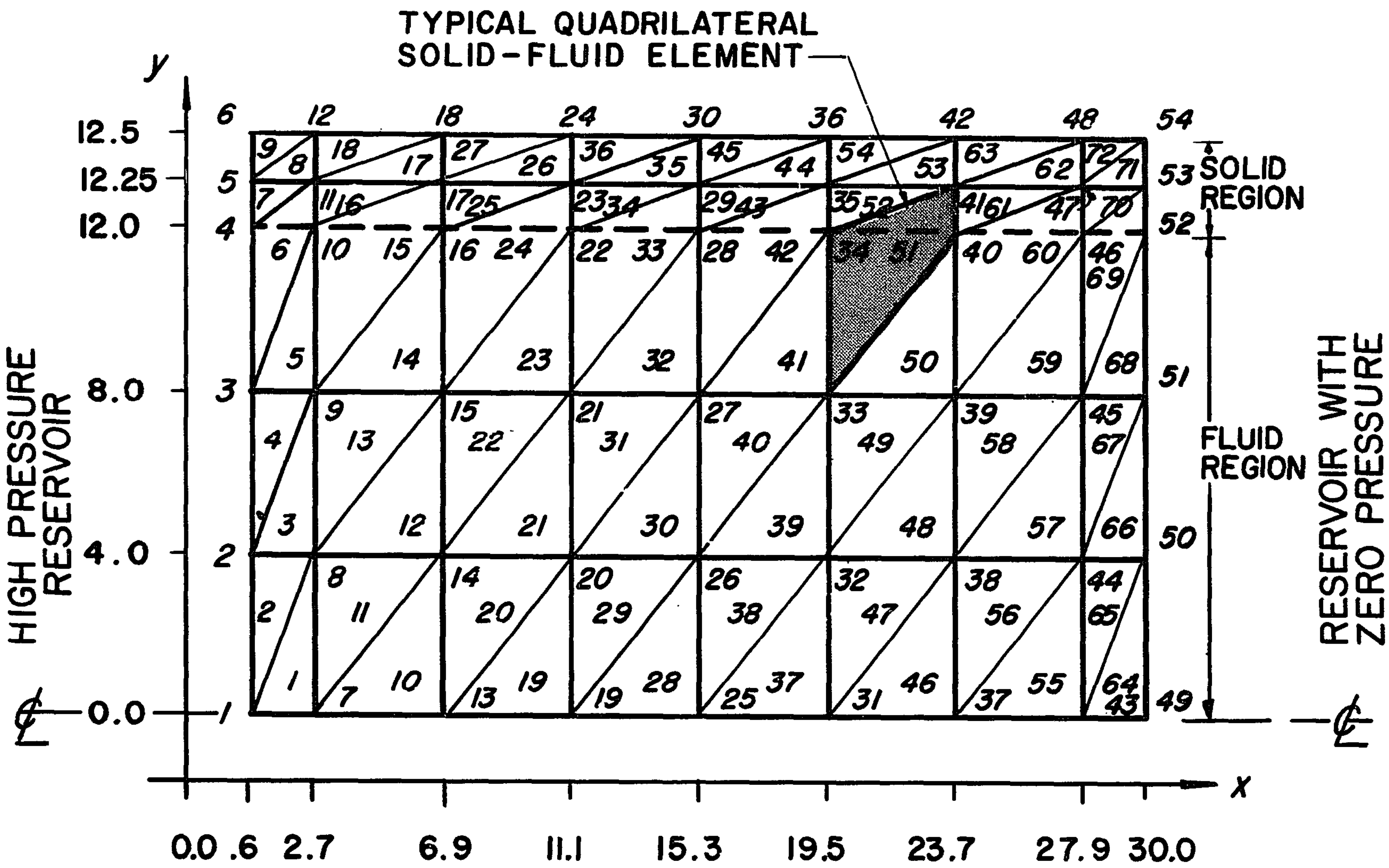


FIG. 4-12 72 ELEMENT SOLID-FLUID MODEL

Two independent computer programs, each having different organizations, algorithms, and methods of solution discussed earlier, are employed to analyze this problem. The first is FLINTS (Fluid Interacting with Solid), a special-purpose finite element program developed herein for solid-fluid interaction studies. The second is WECAN (Westinghouse Electric Computer Analysis), a large general-purpose finite element program to which a library of subroutines for solid-fluid interaction was added based on this study. FLINTS can perform dynamic analysis of solid, fluid, and solid-fluid continua using the modal superposition technique as the method of solution. WECAN is much more diversified in its scope. Its direct integration capability and wave front solution technique are used to solve the matrix differential equation. WECAN is used to confirm the solid-fluid results obtained from FLINTS. Fig. 4-13a shows the undamped solid vertical displacement and fluid pressure time histories from FLINTS and WECAN at an upstream location x = 6.9 inches. Fig. 4-13b shows the fluid velocity components at the same location. The fluid pressure and velocity curves are given at the centerline of the channel and at the wall--the two locations through the channel height which would show the least and most pronounced effects from solid-fluid interaction. The solid vertical displacements did not vary appreciably through the thickness of the solid. An examination of these and other typical results not shown indicates that the FLINTS response characteristics agree very well with those of WECAN. This verification provides the confidence needed to initiate parametric studies. A detailed description of the system response behavior follows.

Specifically, time histories have been obtained using the 72 element

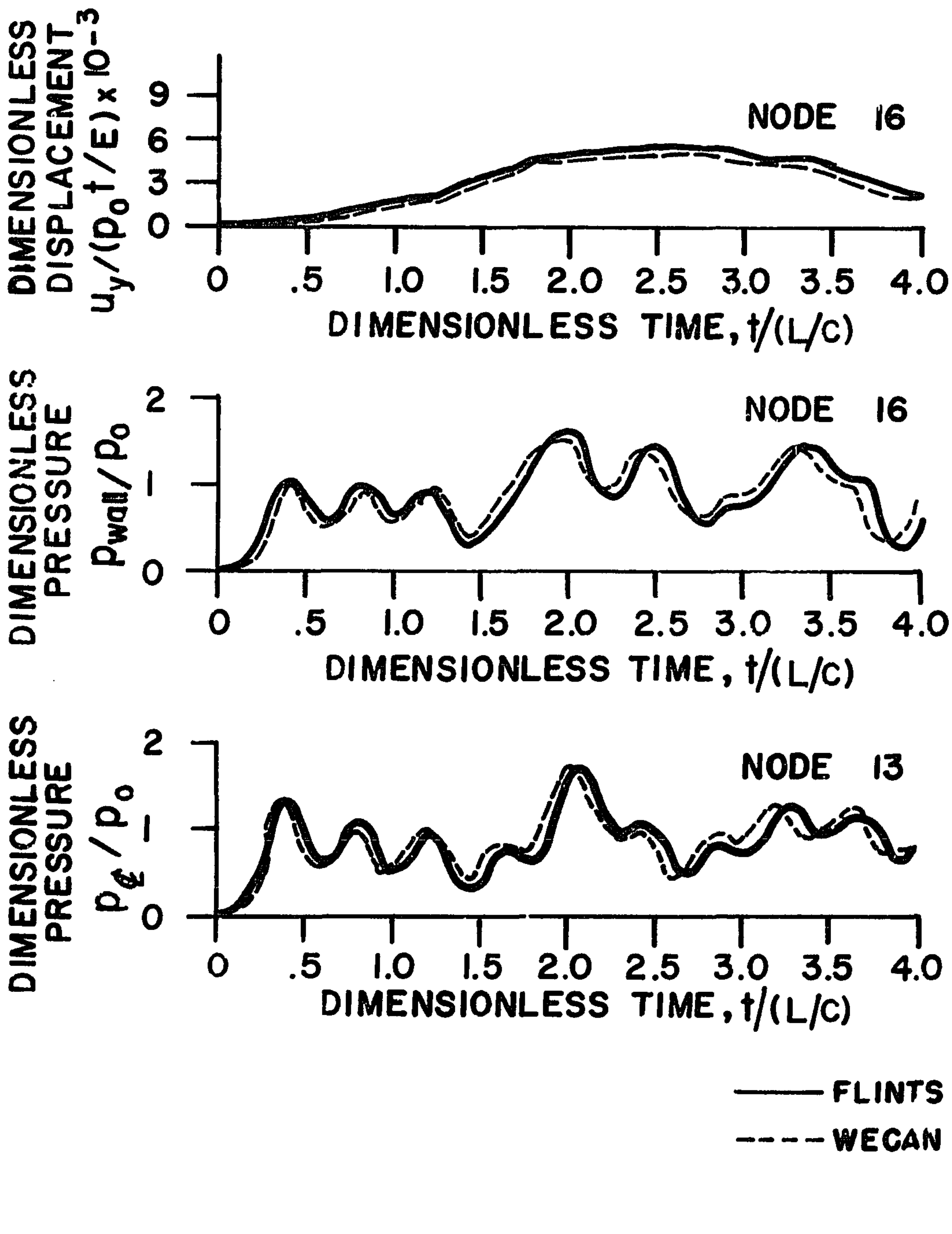


FIG. 4-13a COMPARISON OF RESULTS FROM FLINTS AND WECAN--VERTICAL DISPLACEMENT AND PRESSURES AT X=6.9 INCHES

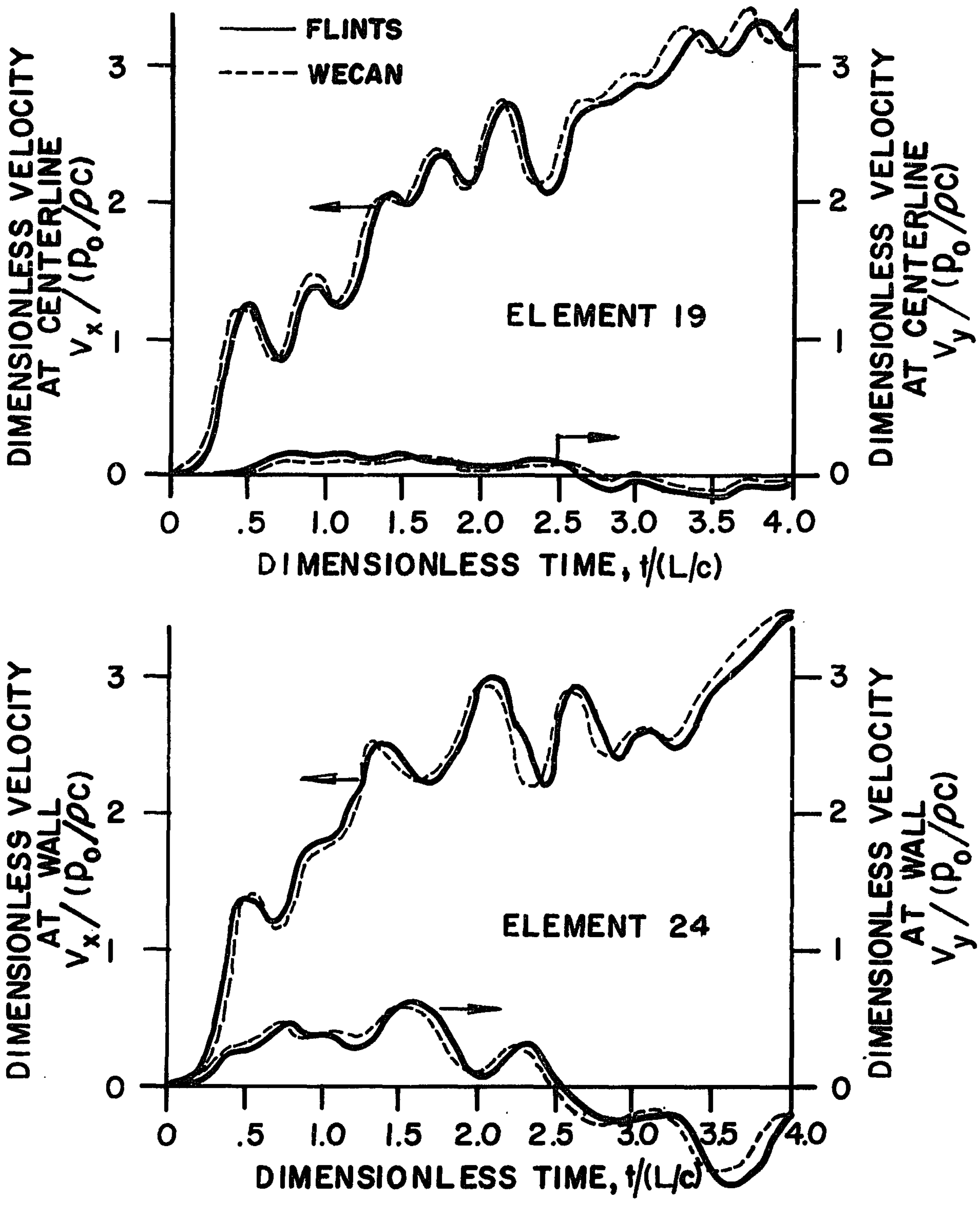


FIG. 4-13b COMPARISON OF RESULTS FROM FLINTS AND WECAN --HORIZONTAL AND VERTICAL COMPONENTS OF FLUID VELOCITY, V_x AND V_y AT X=6.9 INCHES

model for undamped, lightly damped, and heavily damped cases. Results were also found using a finer 88 element mesh. These results were not appreciably different from the solution obtained using the coarser 72 element mesh. This establishes the convergence of the grid size for the 72 element model used throughout this study.

The solid vertical displacement, fluid pressure, and axial and transversal fluid velocity time histories at the centerline of the channel and at the wall are presented at an upstream location ($x = 6.9$ inches) and at a downstream location ($x = 23.7$ inches). These curves are representative of the response of the solid and the fluid at other locations along the channel. The nodes for which pressures and displacements are plotted and the elements for which velocities are presented are given in Table 3. Figures 4-14 through 4-19 are the response characteristics for the undamped ($\alpha = 0.\ \mathrm{sec}^{-1}$), lightly damped ($\alpha = 1046.666\ \mathrm{sec}^{-1}$), and heavily damped ($\alpha = 3062.467\ \mathrm{sec}^{-1}$) cases respectively. Superimposed on these curves are two rigid wall (no solid-fluid interaction) curves for comparison. The first is a finite element solution with rigid walls, similar to the results presented in the previous section, obtained using FLINTS. The second is the analytic solution to the one-dimensional wave equation for the rigid wall problem as given in Appendix 2.

The example which is being used to verify the finite element formulation of solid-fluid interaction is similar to a water hammer problem. The response curves are expected to exhibit some phenomena predicted by water hammer theory [22]. However, the present formulation analyzes the general problem of solid-fluid interaction which encompasses water hammer

TABLE 3--UPSTREAM AND DOWNSTREAM

POSITIONS PLOTTED FOR FLEXIBLE WALL CASE

Location	Upstream position		Downstream position	
	Node no.	x (in)	Node no.	x (in)
Centerline	13	6.9	37	23.7
Wall	16	6.9	40	23.7

TABLE 4--PHYSICAL DATA FOR FLEXIBLE WALL CASE

Parameter	Units
Density	9.35521×10^{-5} lbf-sec^2/in^4 for fluid
	7.297×10^{-4} lbf-sec^2/in^4 for solid
Speed of sound	6.0×10^4 in/sec
Young's modulus	30.0×10^6 psi
Poisson's ratio	.3
Pressure at x = 0	.1 psi
Channel length	29.4 in
Channel width	24 in
Damping coefficients:	
Undamped	0.0 sec^{-1}
Lightly damped	1046.666 sec^{-1}
Heavily damped	3062.467 sec^{-1}

theory. As a result, the time histories show water hammer characteristics as well as substantial effects not predicted by water hammer theory.

According to water hammer theory, a pressure wave surging in a fluid in a rigid channel travels at the speed of sound in the fluid. The presence of flexible walls in the conduit reduces the speed at which the wave travels, thereby reducing the effective speed of sound in the fluid. This in turn increases the system period, which is the time required for the wave to traverse the length of the conduit and be reflected back. Consequently, a phase shift is expected to occur in the response characteristics of the rigid and flexible wall time histories--the rigid wall case responding faster than the interactive case. The pressure time histories of Figures 4-14 through 4-19 show this phase shift. This effect is less pronounced at the centerline (nodes 13 and 37) as compared to the wall (nodes 16 and 40). The nodes at the wall are expected to be more sensitive to the subtler aspects of solid-fluid interaction than the nodes at the centerline as shown in Figures 4-14 through 4-19.

The analytic solution to the rigid wall case indicates that the magnitude of the pressure surge wave caused by the step pressure is equal to the magnitude of the step pressure p_o . The rigid wall response curves in Figures 4-14 through 4-19 exhibit the aforementioned oscillatory behavior about the analytic solution. In the elastic wall solution, it is expected that the deformation of the channel walls as the pressure wave passes will increase the volume of the channel and therefore decrease the magnitude of the pressure wave. Figure 4-14a, the undamped displacement and pressure response curves at the upstream location, is the best illustration of this phenomenon. The remaining displacement and pressure time histories also confirm this. The reduction in the magnitude of the pressure as the wall moves upwards is clearly observed. The nodal pressures rise and oscillate about a continually decreasing magnitude as the wall moves upwards until a transversal water hammer, which will be discussed later, causes an additional surge. As before, these effects are more evident at the wall itself than at the centerline.

The transversal water hammer effect is entirely a result of solid-fluid interaction and is not predicted by conventional water hammer theory and constitutes a distinctive feature of this study. As can be seen in the displacement and pressure time histories of Figs. 4-14 through 4-19, the axial pressure wave causes the wall to move upwards. This in turn reduces the pressure at the wall. The pressure difference between the wall and the centerline causes fluid flow in the transversal direction toward the wall. This flow continues until a major restoring force is built up in the wall due to its deflection leading to a flow deceleration and a pressure buildup at the wall, which then reflects back through the height of the channel. When this surge reaches the centerline of the channel, the axial pressure wave is no longer present at that time to maintain the pressure rise. Then pressure then falls, only to rise again in response to the axial pressure wave in the next cycle. Figs. 4-14a and 4-14b are again the best illustrations of this transversal water hammer phenomenon. The transversal velocity v_y at the wall, in element 24, clearly shows an upward flow until the approximate $t = 1.75\ L/c$, when it exhibits a large drop. At this time, the pressure at the wall, node 16, undergoes the aforementioned surge. At the centerline, the surge in the pressure

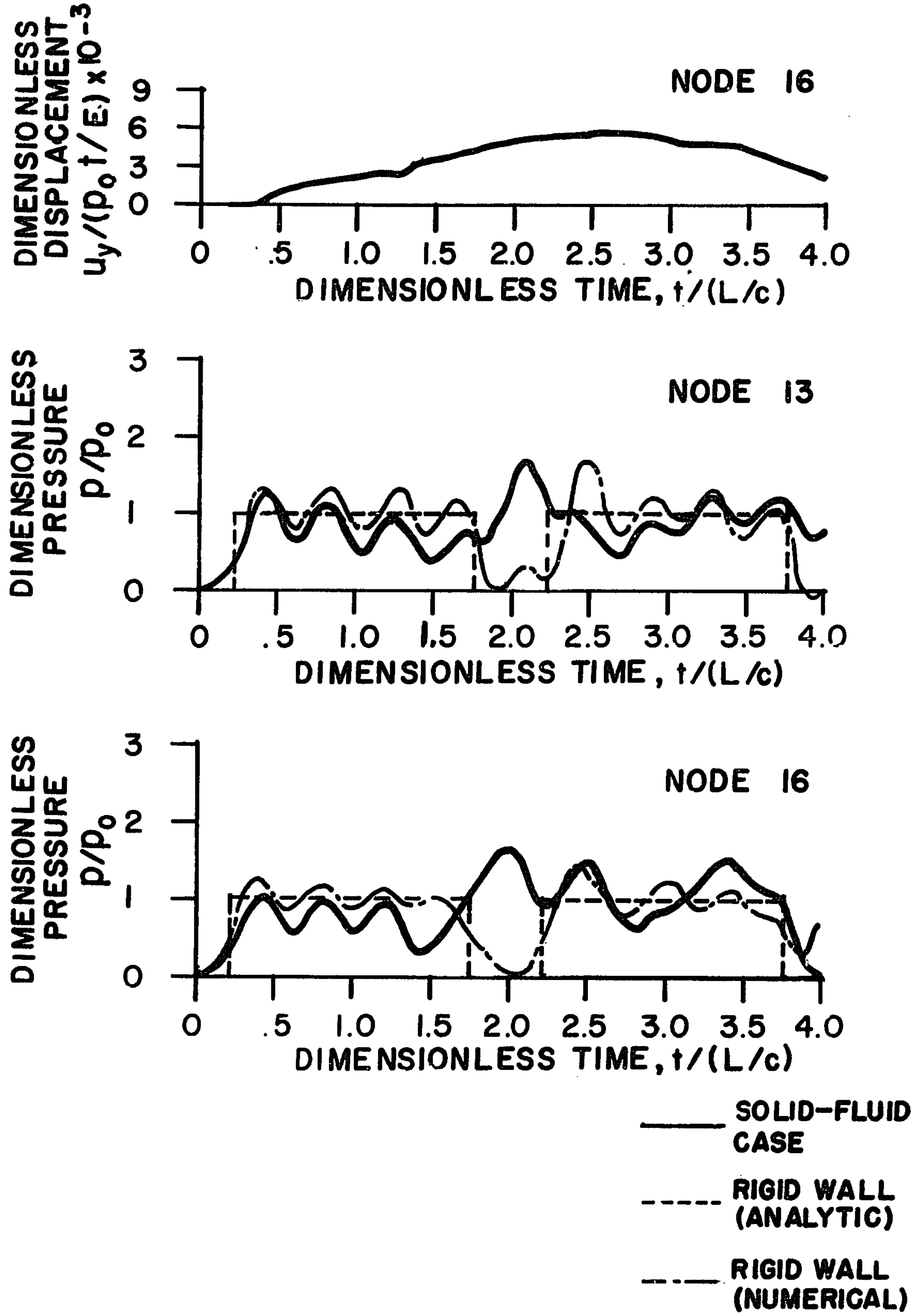


FIG. 4-14a UNDAMPED DISPLACEMENT AND PRESSURE TIME HISTORIES AT X=6.9 INCHES

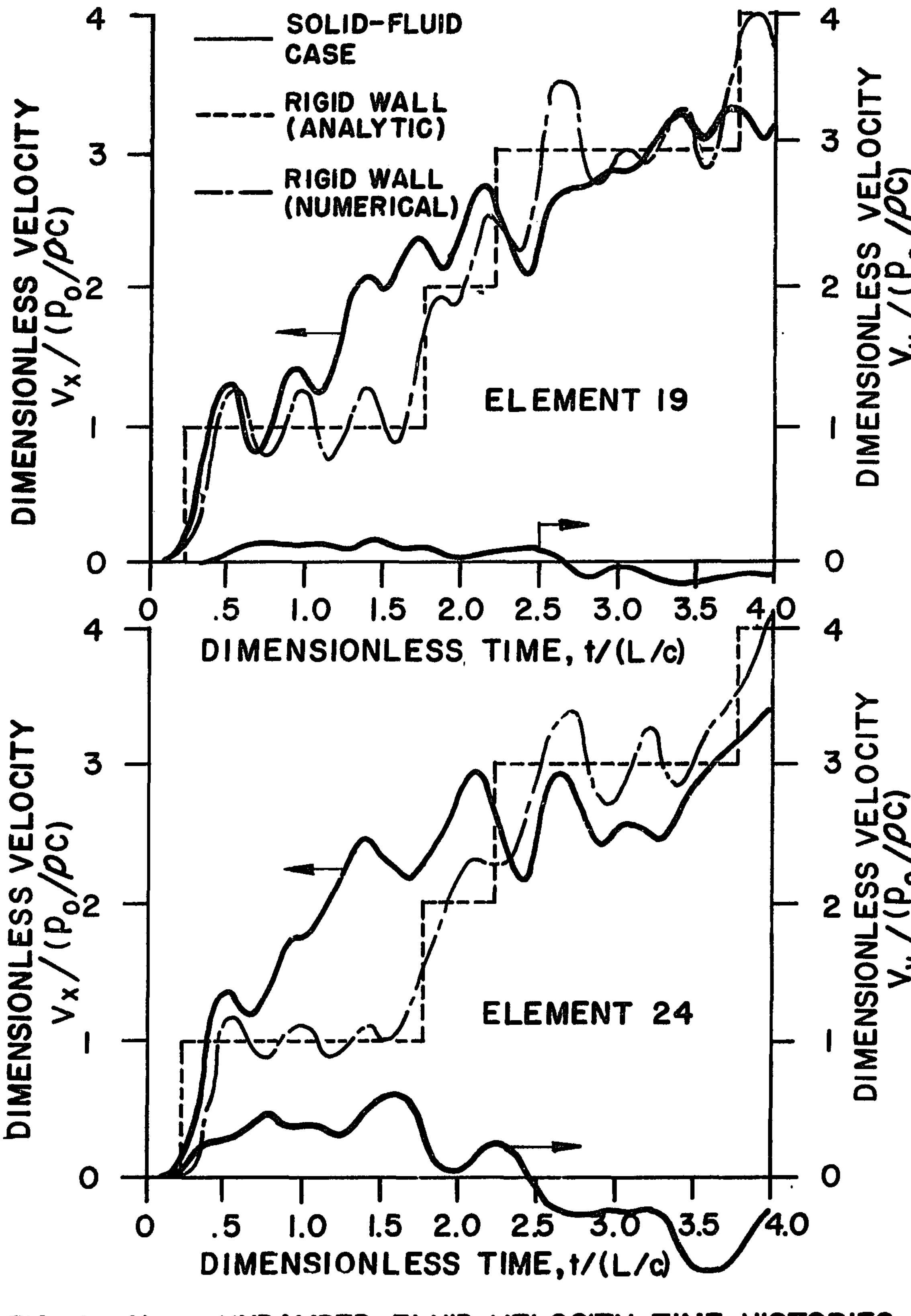


FIG. 4-14b UNDAMPED FLUID VELOCITY TIME HISTORIES AT X = 6.9 INCHES

curve, node 13, is delayed as expected to approximately $t = 1.95\ L/c$. This delay is due to the time required for the surge to reflect back through the height of the channel. Once the peak of the pressure surge is attained, the pressure falls because the axial wave is not present to maintain it. The pressure at the wall falls more gradually because the centerline pressure maintains it to a certain extent. Because of the time delay, the pressure at the centerline is slightly higher than the pressure at the wall, even though it is falling at a faster rate. This causes a re-establishment of flow toward the wall, which accounts for the increase in v_y just after the first drop ($2\ L/c \leq t \leq 2.5\ L/c$). At $t > 2.5\ L/c$, the axial wave has returned and the wall is moving downward with an accompanying increase in the pressure at the wall. This reverses the flow direction toward the centerline, where the pressure is lower. The negative value of v_y at this time confirms this. The transversal velocity at the centerline, in element 19, also follows this pattern, but due to the relative insensitivity of the centerline to solid-fluid interaction, its behavior is not as marked. The downstream time histories for the undamped case, Fig. 4-15, also exhibit the same type of response as the upstream time histories.

The effect of damping is shown in Figures 4-16 through 4-19. Figs. 4-16 and 4-17 are the time histories for the lightly damped case. In comparison with the undamped response, the pressure and displacement amplitudes in the damped case are lower and the curves are smoother as expected. The transversal water hammer surge is not as large, nor are the gradients as steep. The entire solution tends toward a steady-state value. The velocity curves in the lightly damped case follow the same

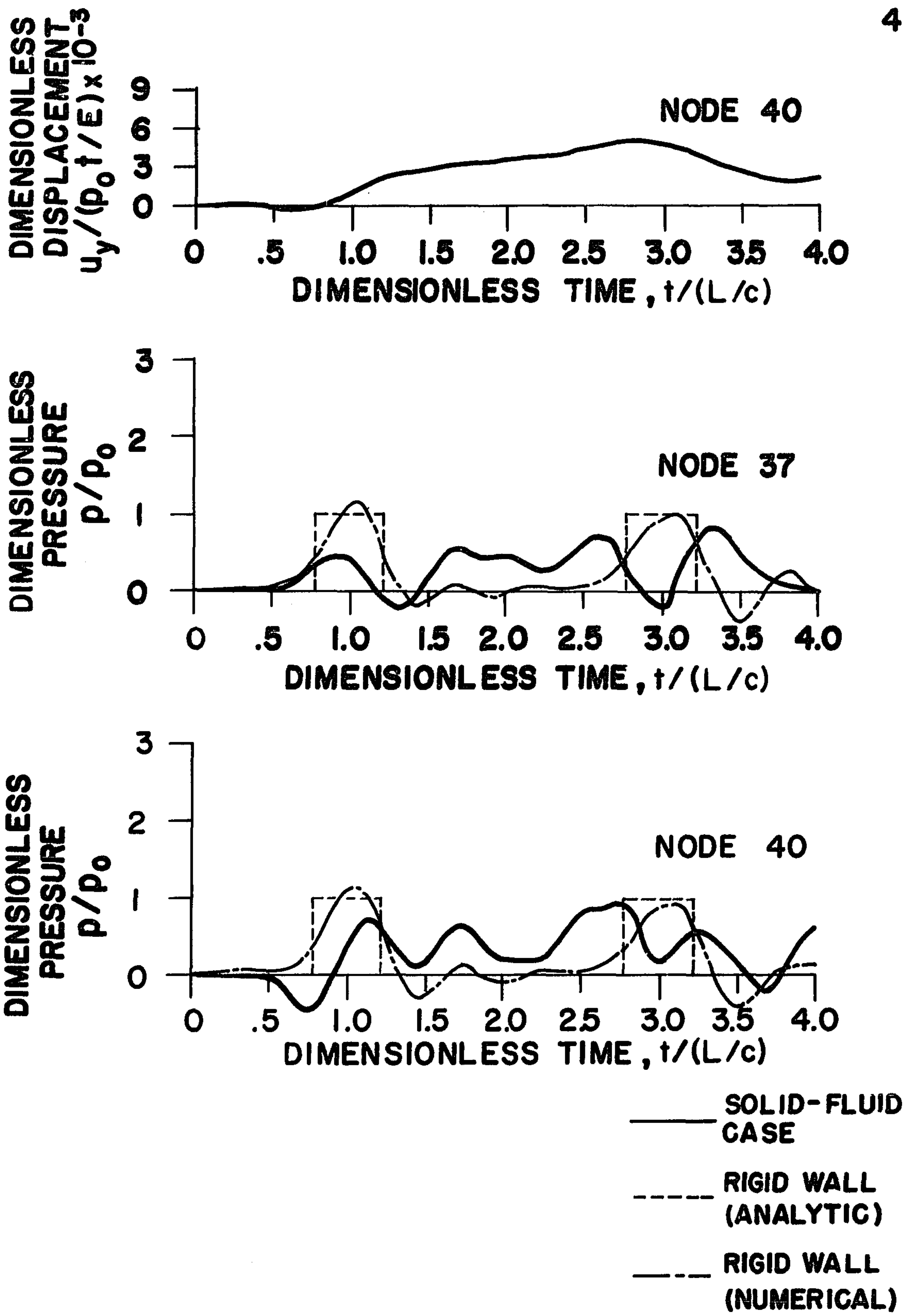


FIG. 4-15a UNDAMPED DISPLACEMENT AND PRESSURE TIME HISTORIES AT X= 23.7 INCHES

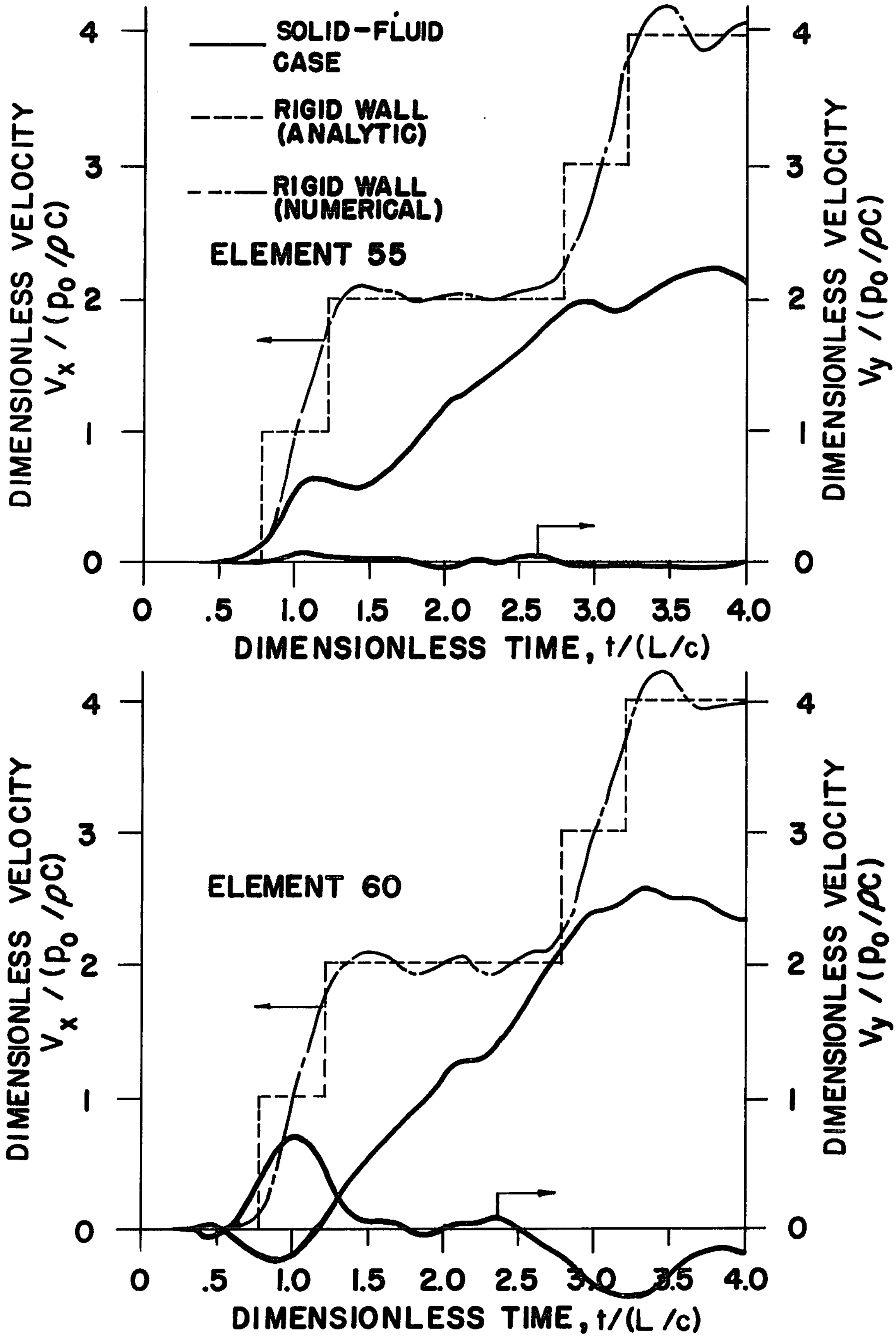


FIG. 4-15b UNDAMPED FLUID VELOCITY TIME HISTORIES AT X = 23.7 INCHES

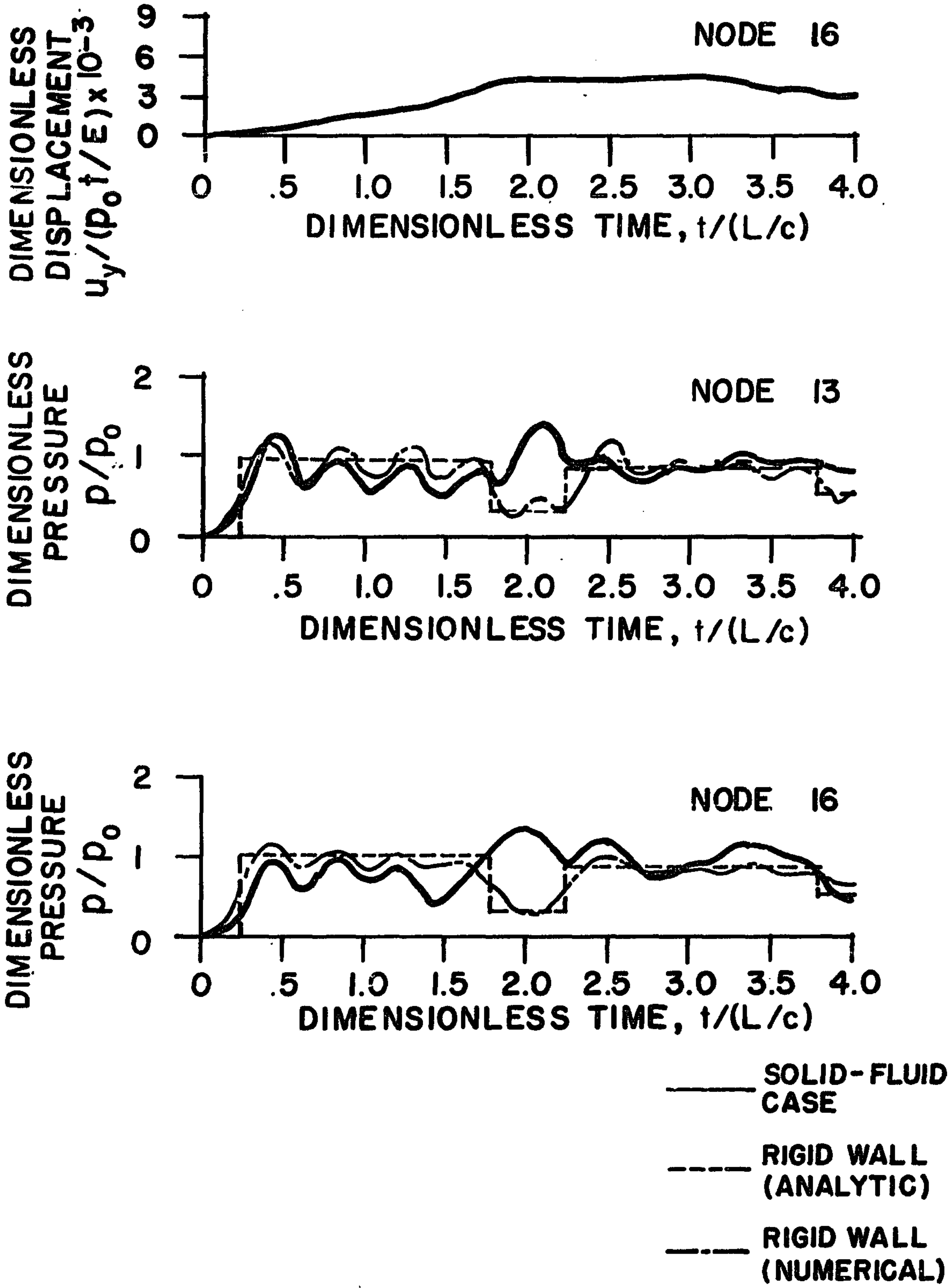
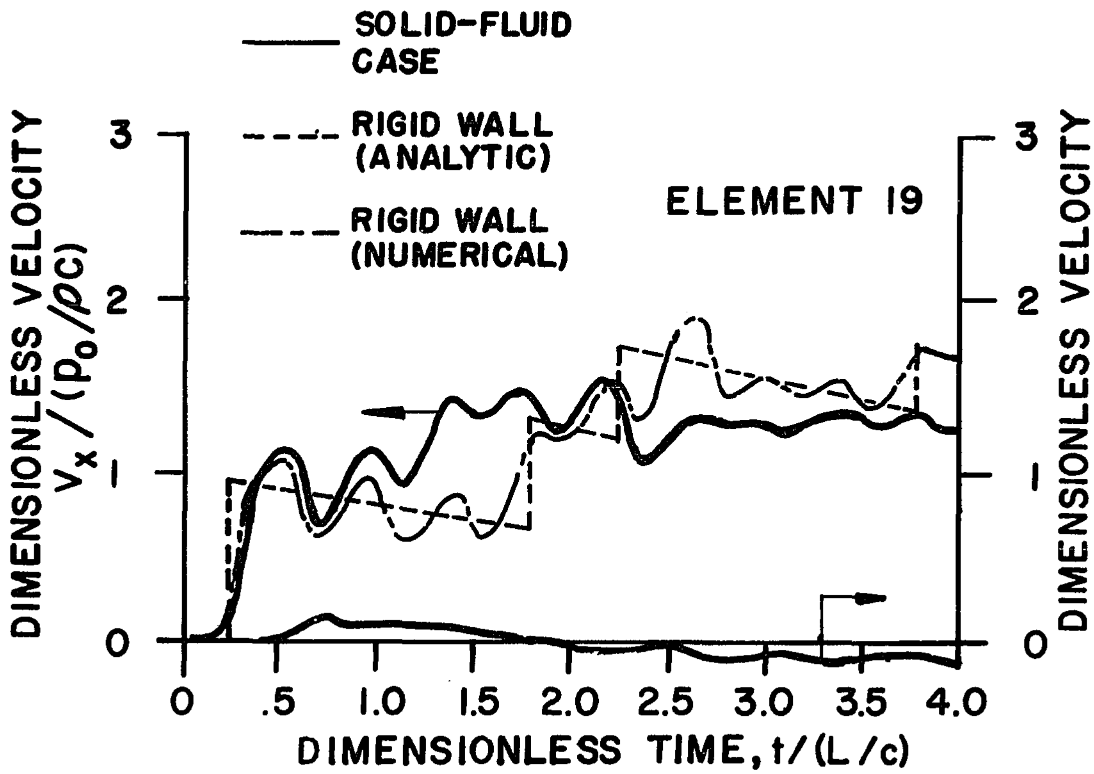


FIG. 4-16a LIGHTLY DAMPED DISPLACEMENT AND PRESSURE TIME HISTORIES AT X=6.9 INCHES



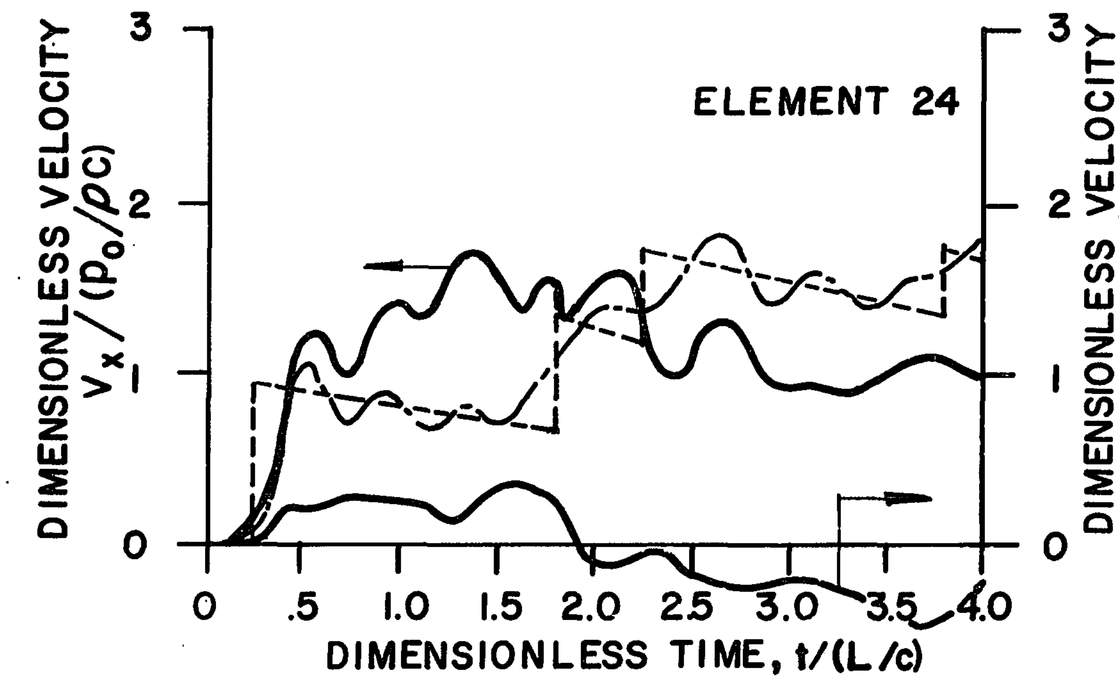


FIG. 4-16b LIGHTLY DAMPED FLUID VELOCITY TIME HISTORIES AT X = 6.9 INCHES

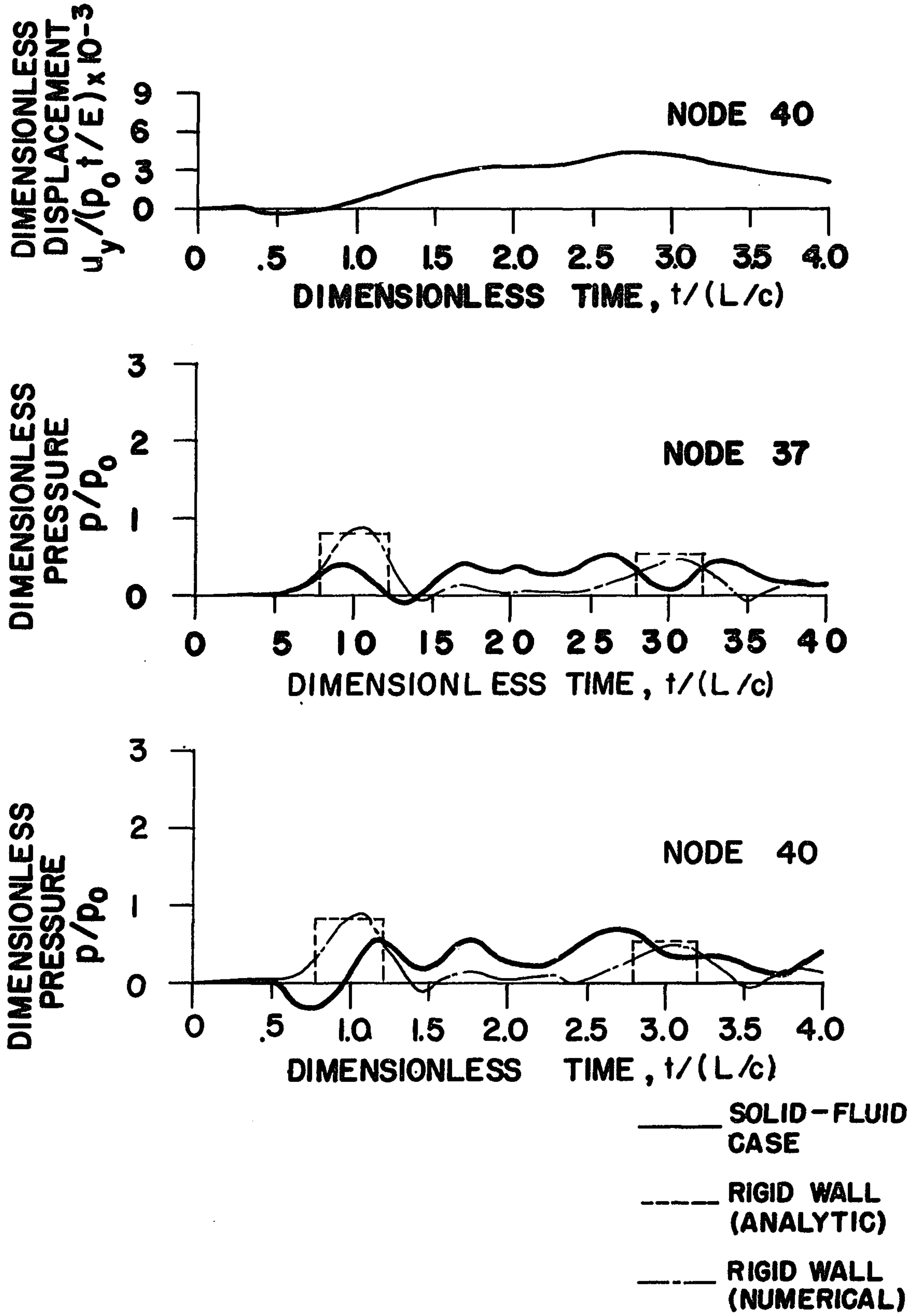


FIG. 4-17a LIGHTLY DAMPED DISPLACEMENT AND PRESSURE TIME HISTORIES AT X = 23.7 INCHES

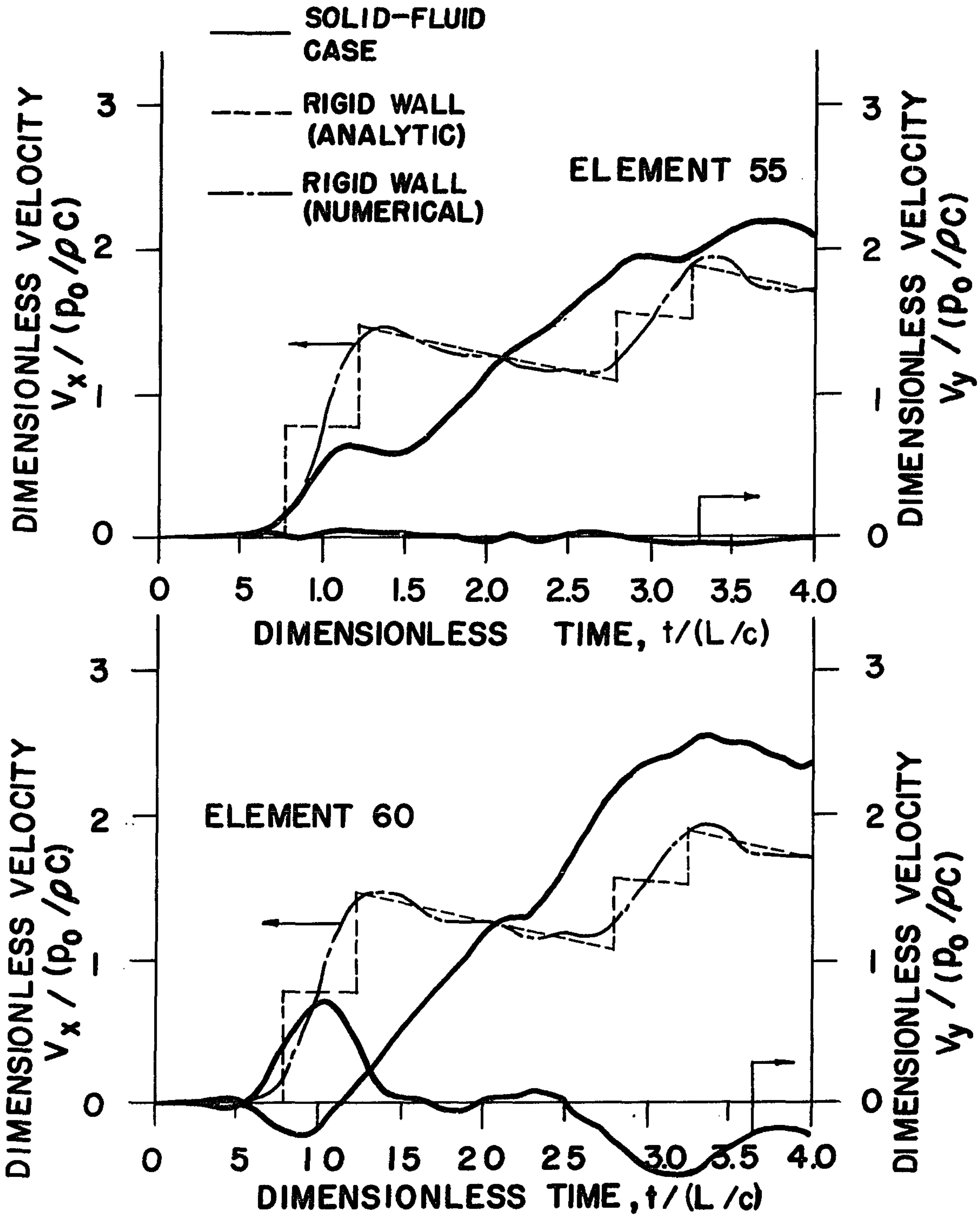


FIG. 4-17b LIGHTLY DAMPED FLUID VELOCITY TIME HISTORIES AT X= 23.7 INCHES

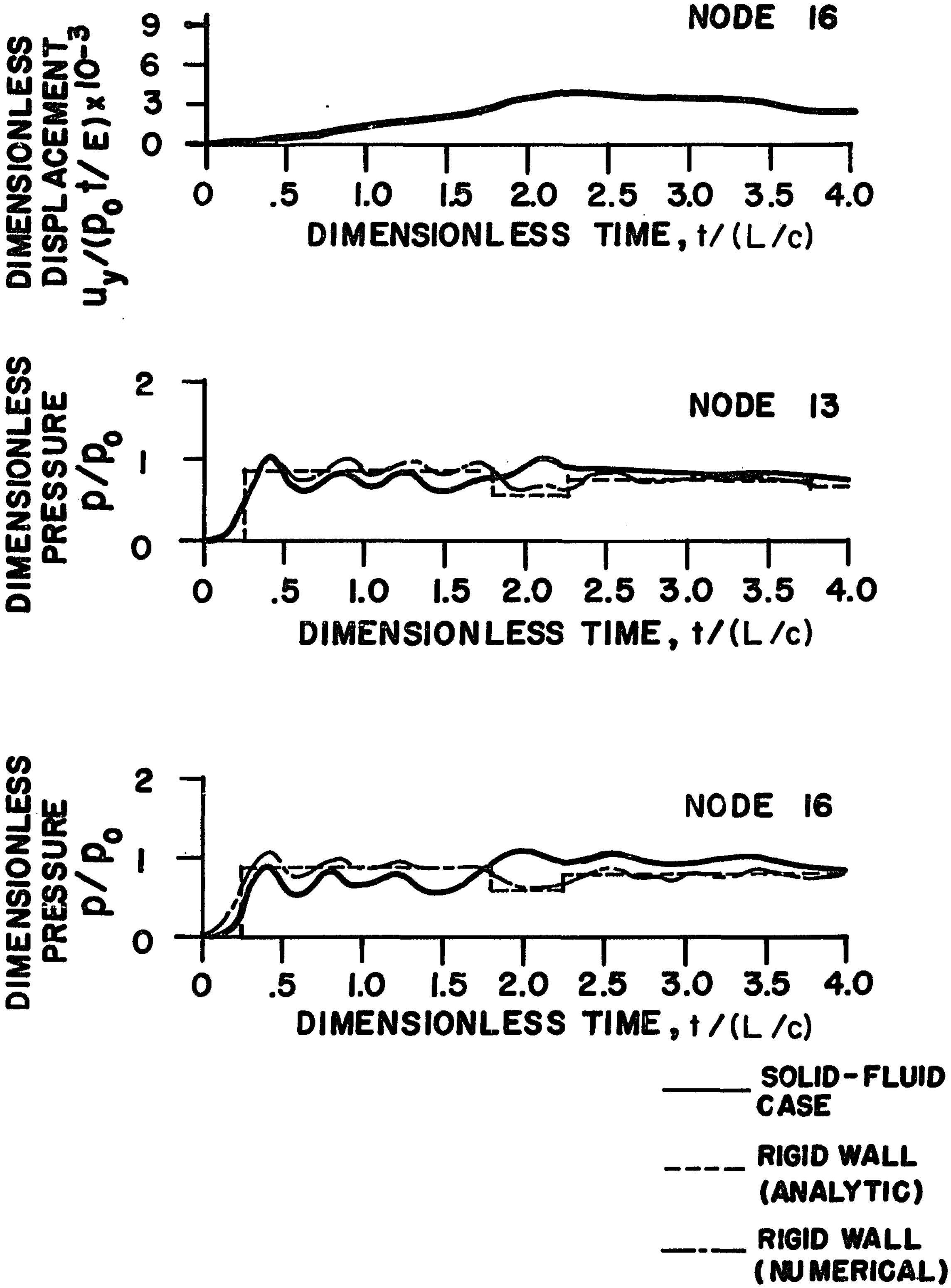


FIG. 4-18a HEAVILY DAMPED DISPLACEMENT AND PRESSURE TIME HISTORIES AT X = 6.9 INCHES

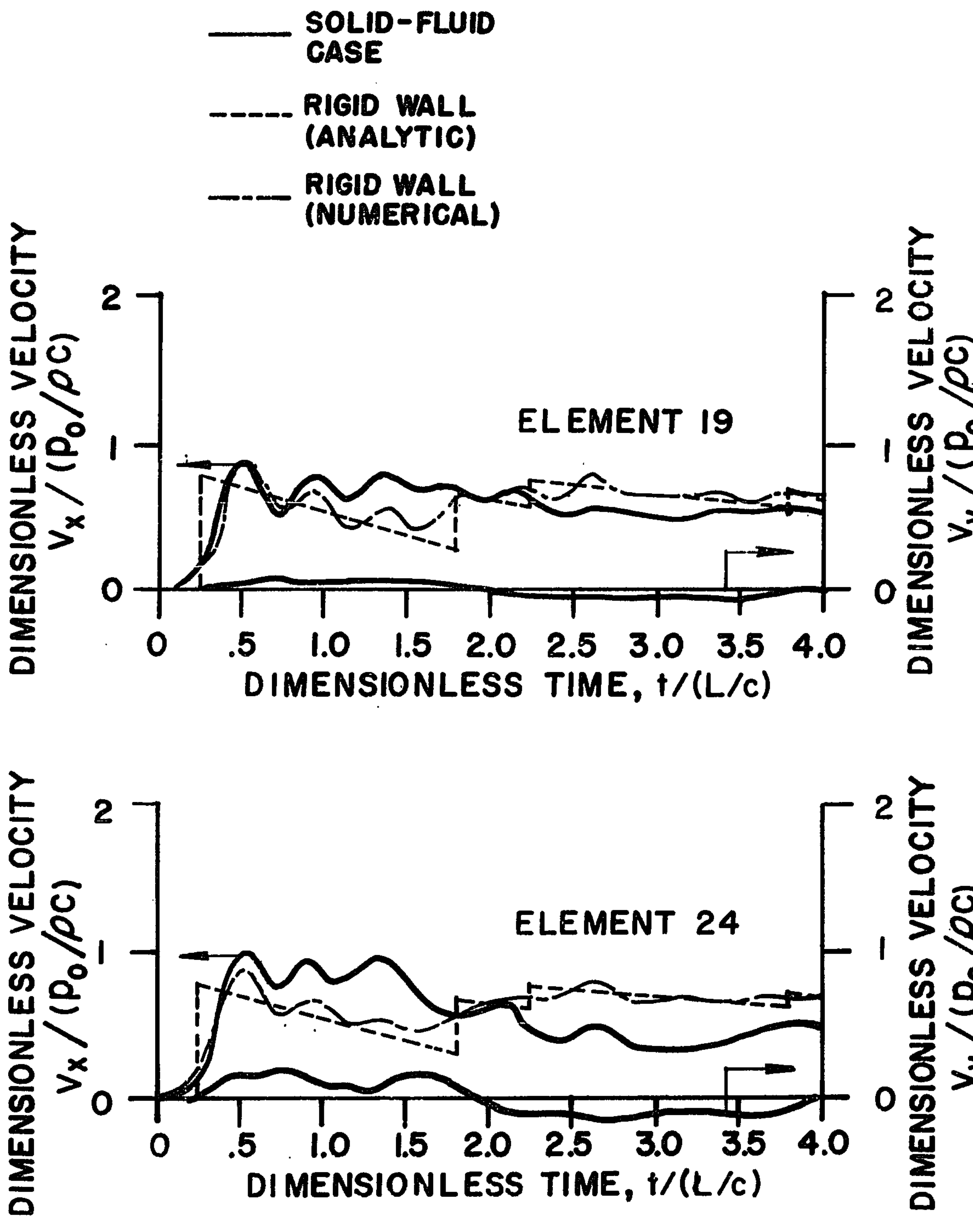


FIG. 4-18b HEAVILY DAMPED FLUID VELOCITY TIME HISTORIES AT X = 6.9 INCHES

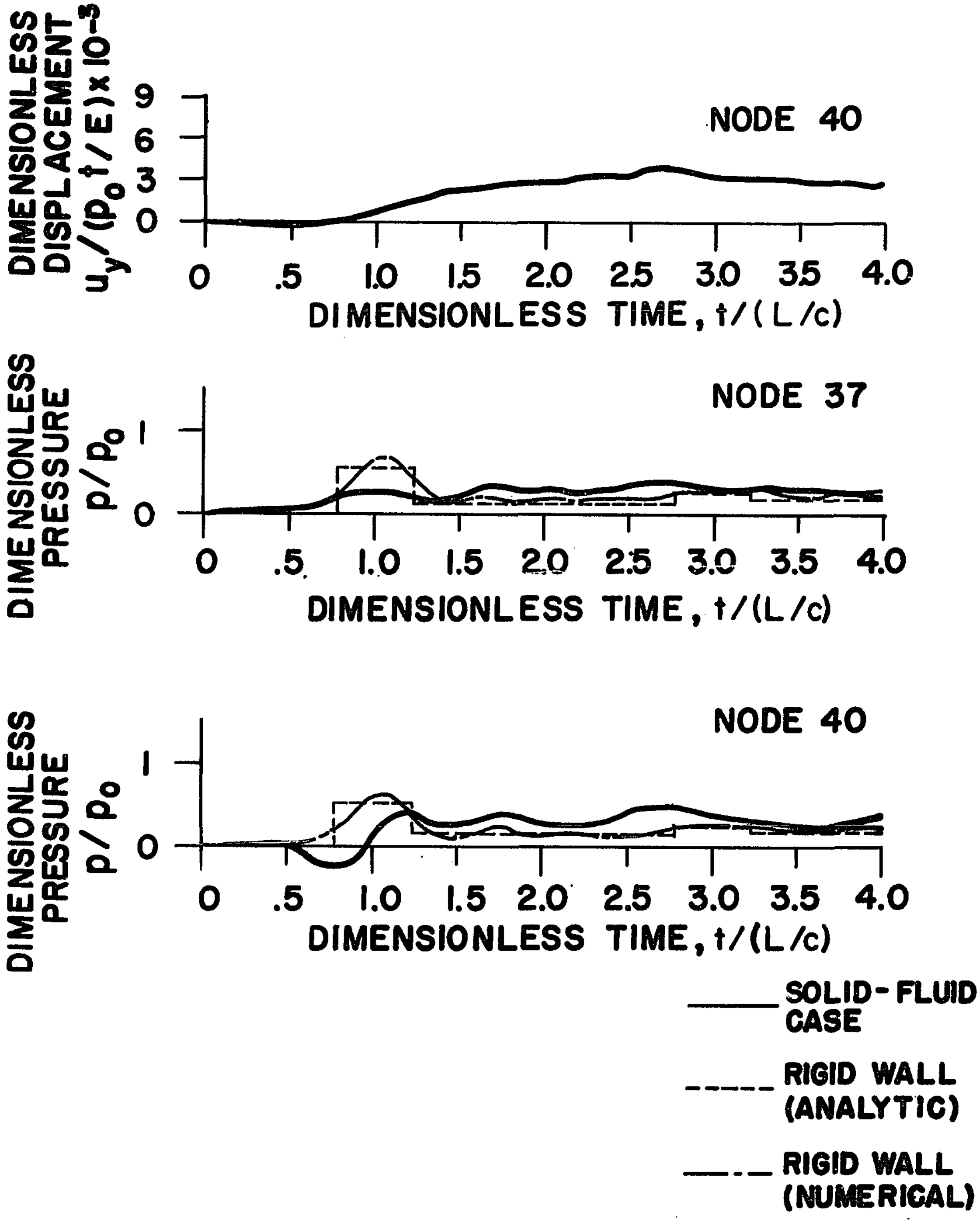


FIG. 4-19a HEAVILY DAMPED DISPLCEMENT AND PRESSURE TIME HISTORIES AT X = 23.7 INCHES

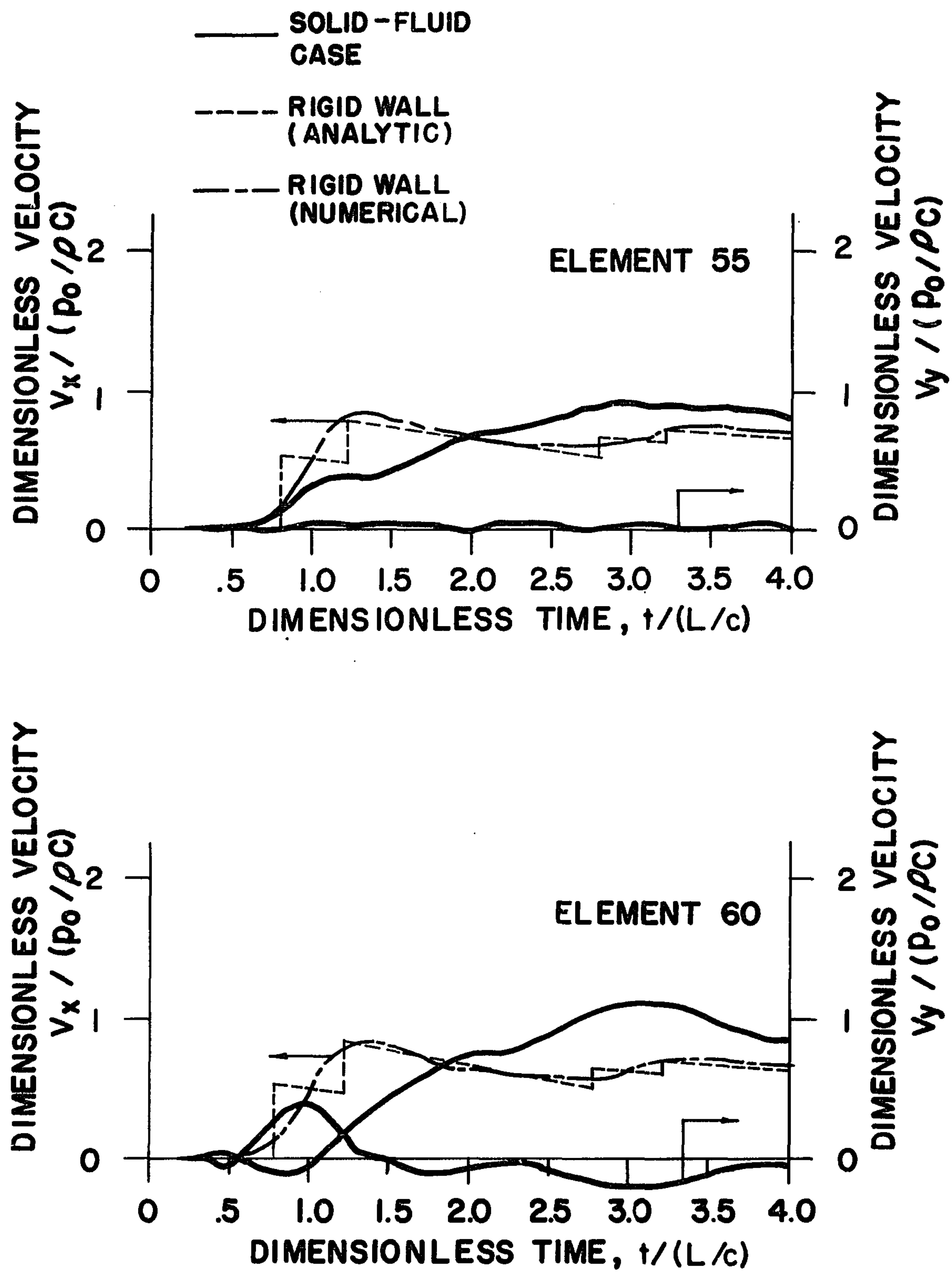


FIG. 4-19b HEAVILY DAMPED FLUID VELOCITY TIME HISTORIES AT X=23.7 INCHES

pattern as in the undamped case, but again the gradients are less steep. Figs. 4-18 and 4-19 are the time histories for the heavily damped case. In comparison with the undamped case, the pressure and displacement amplitudes are much lower. The pressure surge is still evident, but the heavy damping quickly brings the response to steady-state by the end of the time history. The transversal velocities do not exhibit the high amplitudes observed in the undamped case, although the response caused by solid-fluid interaction is still apparent. The physical data for the flexible wall cases studied herein are presented in Table 4.

Initially, the time step used in the rigid wall case was tested for the flexible wall problem and was found to produce non-converged results. This is a direct result of the presence of the time-dependent coupling loads. The time step was then divided by 10, producing a time step of 1.25×10^{-6} seconds. The convergence of the solution was then tested by doubling this time step and observing an insignificant change in the results.

5 CONCLUSION AND RECOMMENDATIONS

A finite element model for solid-fluid interaction has been developed. This finite element model may be used to analyze two-dimensional solid-fluid interaction problems involving complex geometries and loadings. The model was tested for a flow configuration consisting of water, between two flat flexible parallel plates, subjected to a step pressure at one end. Verification of the model was achieved by comparison with an independent general-purpose structural computer program to which a library of subroutines for fluid and solid-fluid interaction was added. Parametric studies to observe the effect of damping were also performed. Results confirmed the predictions of water hammer theory and demonstrated a transversal water hammer phenomenon, which is not predicted by the conventional theory. The transversal water hammer has a substantial effect on the response of the system and cannot be ignored. This study confirms the applicability of this finite element model to generalized solid-fluid interaction problems.

The mathematical formulation of solid-fluid interaction presented herein can treat acousto-structural analysis and other solid-fluid interaction problems in which the boundary conditions on the fluid are or can be expressed in terms of specified nodal pressures. For hydro-thermal analyses, the boundary conditions are often specified in terms of fluid velocity components and temperature rather than pressure. The next step in this study of solid-fluid interaction would be to expand the number of dependent variables in the fluid and include the temperature and fluid velocity components among the dependent variables. The solid-fluid interaction package thereby developed would be capable of analyzing a broader range of solid-fluid interaction problems including thermal effects.

6 NOMENCLATURE

a	Local area coordinates of triangle, in^2, defined by eq. (2.1-5)
A	Area of finite element, in^2; Constant in analytic solution given in Part 1, Section 3, whichever applies
b	Local y coordinates of triangle, in; defined by eq. (2.1-5); function used in analytic solution in Appendix 2, whichever applies
B	Constant in analytic solution given in Part 1, Section 3
[B]	Strain-displacement matrix, in^{-1}
c	Speed of sound, in/sec ; Local x coordinates of triangle, in, defined by eq. (2.1-5), whichever applies
C	Constant in analytic solution given in Part 1, Section 3
[C]	Damping matrix
D	Constant in analytic solution given in Part 1, Section 3
[D]	Stress-strain matrix, psi
E	Young's modulus, psi; Constant in analytic solution given in Part 1, Section 3, whichever applies
f	Element of generalized force array
f(s)	Function of Laplace variable used in analytic solution in Appendix 2
{F}	Condensed global load array; Boundary integral array, whichever applies
[G]	Fluid inertia matrix, $in\text{-}sec^2$
h	Constant value of x or y used in computation of solid-fluid coupling matrix for vertical or horizontal wall, respectively, in

[H]	Fluidity matrix, in
i	Node number; $(-1)^{\frac{1}{2}}$, whichever applies
[I]	Identity matrix
j	Node number
k	Damping parameter
[K]	Stiffness matrix
L	Channel length, in
[L]	Viscous damping matrix, in-sec
m	Slope of solid-fluid boundary
[M]	Mass matrix
n	Term number in analytic solution in Appendix 2
[N]	Finite element shape function
p	Pressure, psi
{p}	Array of nodal pressures, psi
{q}	Array of generalized freedoms
r_1, r_2	Roots of overdamped differential equation in Part 1, Section 3
{R}	Array of applied nodal loads, lbf
{R'}	Coupling force on solid, lbf
s	Laplace transform variable
S	Solid-fluid boundary, in
[S]	Solid-fluid coupling matrix, lbf-sec^2/in^2
t	Time, sec
u	Displacement, in
{u}	Array of nodal displacements, in
v	Velocity, in/sec

W	Work, in-lb
x	Axial coordinate
x_o	Abscissa of first node of solid-fluid boundary, in
x_f	Abscissa of last node of solid-fluid boundary, in
$\{x\}$	Condensed global freedom array
y	Transversal coordinate
y_o	Ordinate of first node of solid-fluid boundary, in
y_f	Ordinate of last node of solid-fluid boundary, in

Greek

α	Damping factor for damping matrix proportional to mass matrix, sec^{-1}; Angle of solid-fluid boundary with x-axis; Polynomial coefficient, whichever applies
α_n	Phase angle in analytic solution in Appendix 2
β	Damping factor for damping matrix proportional to stiffness matrix, sec; Numerical integration parameter; Function used in analytic solution in Appendix 2, whichever applies
γ_{xy}	Shear strain, in/in
δ	Displacement, in
Δt	Time step, sec
ϵ	Strain, in/in
$\{\epsilon\}$	Strain array, in/in
ζ	Modal damping ratio
λ	Eigenvalue
ν	Poisson's ratio
ρ	Density, $lbf\text{-}sec^2/in^4$
σ	Stress, psi

τ_{xy}	Shear stress, psi
$[\phi]$	Modal matrix
$[\omega_n^2]$	Eigenvalue matrix

Subscripts

e	Elemental
f	Fluid
i	Node number
N	Normal
o	Initial
p	Pressure
s	Solid
T	Tangential
x	x-direction
y	y-direction

Superscripts

—	Condensed global or boundary, whichever applies
n	Iteration number
T	Matrix transpose
$\dot{}$, $\ddot{}$	First and second derivatives in time

Operators

d	Virtual

7 APPENDIX 1--EVALUATION OF ELEMENT MATRICES

The matrices defined in equations (2.1-8), (2.2-5), and (2.3-9) are evaluated for linear plane triangular finite elements in this section.

Solid Finite Element

The mass, damping, and stiffness matrices for a two-dimensional solid finite element are defined as

$$\begin{aligned} &\text{The mass matrix} && [M_e] = \iint_A [N_s]^T \rho_s [N_s]\, dx\, dy \\ &\text{The damping matrix}^1 && [C_e] = \alpha [M_e] + \beta [K_e] \\ &\text{The stiffness matrix} && [K_e] = \iint_A [B]^T [D] [B]\, dx\, dy \end{aligned} \qquad \text{(2.2-5) rep.}$$

The mass and damping matrices may be evaluated by directly substituting in the solid shape function (eqs. (2.2-2) and (2.2-3)) and integrating. This yields the so-called consistent or distributed mass matrix [26]

$$[M_e] = \frac{\rho_s A_s}{3} \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{1}{2} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{2} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{2} \end{bmatrix} \qquad \text{(A1-1)}$$

Two types of mass matrices are actually possible. The first is the consistent mass matrix given above. It is termed consistent because it results directly from the finite element formulation. The second is the lumped mass matrix. In early attempts to approach structural dynamics problems, the mass of the element was arbitrarily lumped or concentrated at the nodes. This results in a mass matrix of the form

$$[M_e] = \frac{\rho_s A_s}{3} [I_6] \qquad \text{(A1-2)}$$

in which $[I_6]$ is the 6 x 6 identity matrix. The distributed mass matrix is more appealing from the point of view that it is a direct

---

[1] The damping matrices used throughout this study are proportional to mass only, i.e., $\beta = 0$.

product of the mathematical formulation. However, in the course of this study, it has been observed that the off-diagonal terms in the consistent mass matrix produce sufficient numerical noise to obscure the actual results. For this reason, lumped matrices have been used throughout this study. Both types of matrices are available as options in FLINTS.

The stiffness matrix for the plane triangular linear finite element is evaluated by first considering the matrices which compose it [20].

The strain-displacement matrix [B] is obtained from the strain-displacement relationships. These relationships in two dimensions are

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u_x}{\partial x} \\ \dfrac{\partial u_y}{\partial y} \\ \dfrac{\partial u_x}{\partial y} + \dfrac{\partial u_y}{\partial x} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \qquad \text{(A1-3)}$$

The shape function for the solid is used to relate the element displacements to the nodal displacements (eq. (2.2-2)). This yields

$$\{\epsilon\} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} [N_s] \{u_e\} = [B] \{u_e\} \qquad \text{(A1-4)}$$

Thus

$$[B] = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} [N_s] \qquad \text{(A1-5)}$$

is the strain-displacement matrix for the two-dimensional solid. Using eq. (2.2-3), eq. (A1-5) may now be evaluated for the plane triangular linear finite element, resulting in

$$[B] = \frac{1}{2\,A_s}\begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix} \qquad \text{(A1-6)}$$

Two possible states of stress exist in a two-dimensional body. The first is plane strain in which the strain in the z-direction, ϵ_x , is equal to zero. The second is plane stress, in which all loads lie in the plane of the body, causing the stress in the z-direction, σ_z , to be zero. The problem under consideration is a plane stress problem. The stress-strain matrix is obtained for this case from the constituitive equations of the theory of elasticity. Under the condition of plane stress, these equations are [20]

$$\begin{aligned} \sigma_x &= \frac{E}{1 - \nu^2} (\epsilon_x + \nu\, \epsilon_y) \\ \sigma_y &= \frac{E}{1 - \nu^2} (\epsilon_y + \nu\, \epsilon_x) \\ \tau_{xy} &= \frac{E}{2(1 + \nu)} \gamma_{xy} \end{aligned} \qquad \text{(A1-7)}$$

In matrix form, this may be written as

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = [D]\,\{\epsilon\} \qquad \text{(A1-8)}$$

Thus,

$$[D] = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} \qquad \text{(A1-9)}$$

is the stress-strain matrix for a two-dimensional body in plane stress. This matrix along with the strain-displacement matrix given in eq. (A1-6), is used to evaluate the stiffness matrix in eq. (2.2-5). The integrand in independent of space so that the stiffness matrix may be written as

$$[K_e] = [B]^T [D] [B] A_s \tag{A1-10}$$

which is a 6 x 6 symmetric matrix as follows

$$[K_e] = \frac{E}{4 A_s (1 - \nu^2)} \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ & \text{symmetric} & k_{33} & k_{34} & k_{35} & k_{36} \\ & & & k_{44} & k_{45} & k_{46} \\ & & & & k_{55} & k_{56} \\ & & & & & k_{66} \end{bmatrix}$$

in which

$$k_{11} = b_1^2 + \frac{1 - \nu}{2} c_1^2$$

$$k_{12} = \frac{1 + \nu}{2} b_1 c_1 = k_{21}$$

$$k_{13} = b_1 b_2 + \frac{1 - \nu}{2} c_1 c_2 = k_{31}$$

$$k_{14} = \nu b_1 c_2 + \frac{1 - \nu}{2} b_2 c_1 = k_{41}$$

$$k_{15} = b_1 b_3 + \frac{1 - \nu}{2} c_1 c_3 = k_{51}$$

$$k_{16} = \nu b_1 c_3 + \frac{1 - \nu}{2} b_3 c_1 = k_{61}$$

$$k_{22} = c_1^2 + \frac{1 - \nu}{2} b_1^2$$

$$k_{23} = \nu b_2 c_1 + \frac{1 - \nu}{2} b_1 c_2 = k_{32}$$

$$k_{24} = c_1 c_2 + \frac{1 - \nu}{2} b_1 b_2 = k_{42}$$

$$k_{25} = \nu b_3 c_1 + \frac{1 - \nu}{2} b_1 c_3 = k_{52}$$

$$k_{26} = c_1 c_3 + \frac{1 - \nu}{2} b_1 b_3 = k_{62}$$

$$k_{33} = b_2^2 + \frac{1 - \nu}{2} c_2^2$$

$$k_{34} = \frac{1 + \nu}{2} b_2 c_2 = k_{43}$$

$$k_{35} = b_2 b_3 + \frac{1 - \nu}{2} c_2 c_3 = k_{53}$$

$$k_{36} = \nu b_2 c_3 + \frac{1 - \nu}{2} b_3 c_2 = k_{63}$$

$$k_{44} = c_2^2 + \frac{1 - \nu}{2} b_2^2 \tag{A1-11}$$

$$k_{45} = \nu b_3 c_2 + \frac{1 - \nu}{2} b_2 c_3 = k_{54}$$

$$k_{46} = c_2 c_3 + \frac{1 - \nu}{2} b_2 b_3 = k_{64}$$

$$k_{55} = b_3^2 + \frac{1 - \nu}{2} c_3^2$$

$$k_{56} = \frac{1 + \nu}{2} b_3 c_3 = k_{65}$$

$$k_{66} = c_3^2 + \frac{1 - \nu}{2} b_3^2$$

Fluid Finite Element

The inertia, viscous damping, and fluidity matrices are defined as

The inertia matrix $$[G_e] = \iint_A [N_f]^T \frac{1}{c^2} [N_f] \, dx \, dy$$

(2.1-8) rep.

The viscous damping matrix $$[L_e] = \iint_A [N_f]^T \frac{k_f}{c^2} [N_f] \, dx \, dy = \alpha \, [G_e]$$

The fluidity matrix $$[H_e] = \iint_A \left[\frac{\partial [N_f]^T}{\partial x} \frac{\partial [N_f]}{\partial x} + \frac{\partial [N_f]^T}{\partial y} \frac{\partial [N_f]}{\partial y} \right] dx \, dy$$

The inertia and viscous damping matrices are evaluated as for the solid element by directly substituting the fluid shape function (eq. 2.1-4) and integrating. This yields a consistent inertia matrix

$$[G_e] = \frac{A_f}{3 \, c^2} \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix} \tag{A1-12}$$

A lumped inertia matrix, which is used in this study, also exists for the fluid. It is given by

$$[G_e] = \frac{A_f}{3 \, c^2} [I_3] \tag{A1-13}$$

in which $[I_3]$ is the 3 x 3 identity matrix.

The fluidity matrix is also obtained by substituting in the fluid shape function. The derivatives of the shape function are for the plane triangular linear fluid element

$$\frac{\partial [N_f]}{\partial x} = \frac{1}{2 \, A_f} [b_1 \quad b_2 \quad b_3]$$
$$\frac{\partial [N_f]}{\partial y} = \frac{1}{2 \, A_f} [c_1 \quad c_2 \quad c_3] \tag{A1-14}$$

These expressions may be substituted into the stiffness matrix equation and integrated. The integrand is again independent of space, so that the integral becomes

$$[H_e] = \left[\frac{\partial[N_f]^T}{\partial x} \frac{\partial[N_f]}{\partial x} + \frac{\partial[N_f]^T}{\partial y} \frac{\partial[N_f]}{\partial y} \right] A_f \qquad \text{(A1-15)}$$

which results in

$$[H_e] = \frac{1}{4\ A_f} \begin{bmatrix} b_1^{\ 2} + c_1^{\ 2} & b_1\ b_2 + c_1\ c_2 & b_1\ b_3 + c_1\ c_3 \\ b_1\ b_2 + c_1\ c_2 & b_2^{\ 2} + c_2^{\ 2} & b_2\ b_3 + c_2\ c_3 \\ b_1\ b_3 + c_1\ c_3 & b_2\ b_3 + c_2\ c_3 & b_3^{\ 2} + c_3^{\ 2} \end{bmatrix} \qquad \text{(A1-16)}$$

Solid-Fluid Finite Element

The equivalent mass and stiffness matrices are obtained from the matrices already derived, so that they need not be discussed in detail. The solid-fluid coupling matrix is given by

$$[S_e] = \int_S [N_f]^T \rho_f [\ \overline{N}_s\ ]\ dS \qquad \text{(2.3-9) rep.}$$

in which S is the straight line defining the solid-fluid boundary, and $[\ \overline{N}_s\ ]$ is the solid boundary shape function, given by

$$[\ \overline{N}_s\ ] = [-\sin\alpha \quad \cos\alpha\ ]\ [N_s] \qquad \text{(2.3-5) rep.}$$

This may be substituted in, along with the shape function of the fluid, to yield the coupling matrix for the solid-fluid superelement of Fig. 2-3. Evaluating the integral for the case of a horizontal boundary (y is a constant equal to h, and $\alpha = 0$) results in the 3 x 6 matrix

$$[S_e] = \frac{\rho_f}{4\ A_f\ A_s} \begin{bmatrix} 0 & s_{12} & 0 & s_{14} & 0 & s_{16} \\ 0 & s_{22} & 0 & s_{24} & 0 & s_{26} \\ 0 & s_{32} & 0 & s_{34} & 0 & s_{36} \end{bmatrix} \qquad \text{(A1-17)}$$

in which

$$\begin{aligned} s_{i,2j} = {} & (a_i\ a_j + (a_i\ c_j + a_j\ c_i)\ h + c_i\ c_j\ h^2)(x_f - x_o) + \\ & (a_i\ b_j + a_j\ b_i + (b_i\ c_j + b_j\ c_i)\ h\ )(x_f^{\ 2} - x_o^{\ 2})/2 + \\ & b_i\ b_j\ (x_f^{\ 3} - x_o^{\ 3})/3 \end{aligned} \qquad \text{(A1-18)}$$

where i refers to the i^{th} node of the fluid portion of the solid-fluid element and j refers to the j^{th} node of the solid portion of the solid-fluid element, and x_o and x_f refer to the abscissa of the nodes defining the solid-fluid boundary. The coupling matrix may also be evaluated for a vertical wall and the generalized case of a sloped wall, yielding similar results. For the vertical wall (x is a constant equal to h, and $\alpha = \pi/2$) the coupling matrix is

$$[S_e] = \frac{\rho_f}{4\ A_f\ A_s} \begin{bmatrix} s_{11} & 0 & s_{13} & 0 & s_{15} & 0 \\ s_{21} & 0 & s_{23} & 0 & s_{25} & 0 \\ s_{31} & 0 & s_{33} & 0 & s_{35} & 0 \end{bmatrix} \tag{A1-19}$$

in which

$$\begin{aligned} s_{i,(2j-1)} = {} & (a_i\ a_j + (a_i\ b_j + a_j\ b_i)\ h + b_i\ b_j\ h^2)(y_f - y_o) + \\ & (a_i\ c_j + a_j\ c_i + (b_i\ c_j + b_j\ c_i)\ h\ )(y_f^{\,2} - y_o^{\,2})/2 + \\ & c_i\ c_j\ (y_f^{\,3} - y_o^{\,3})/3 \end{aligned} \tag{A1-20}$$

in which all terms are defined as before and y_o and y_f are the ordinates of the nodes defining the solid-fluid boundary. For the general case of a sloped wall, the relationship along the boundary

$$y = \frac{y_f - y_o}{x_f - x_o}(x - x_o) + y_o \quad = \quad m\ (x - x_o) + y_o \tag{A1-21}$$

is used. This results in the coupling matrix

$$[S_e] = \frac{\rho_f}{4\ A_f\ A_s} \begin{bmatrix} -s_{11}\sin\alpha & s_{11}\cos\alpha & -s_{13}\sin\alpha & s_{13}\cos\alpha & -s_{15}\sin\alpha & s_{15}\cos \\ -s_{21}\sin\alpha & s_{21}\cos\alpha & -s_{23}\sin\alpha & s_{23}\cos\alpha & -s_{25}\sin\alpha & s_{25}\cos \\ -s_{31}\sin\alpha & s_{31}\cos\alpha & -s_{33}\sin\alpha & s_{33}\cos\alpha & -s_{35}\sin\alpha & s_{35}\cos \end{bmatrix} \tag{A1-22}$$

in which

$$s_{i,(2j-1)} = (a_i\, a_j - (a_i\, c_j + a_j\, c_i)(m\, x_o - y_o) + c_i\, c_j\, (m\, x_o - y_o)^2)(x_f - x_o) + (a_i\, b_j + a_j\, b_i + (a_i\, c_j + a_j\, c_i)\, m - (b_i\, c_j + b_j\, c_i)(m\, x_o - y_o) - 2\, c_i\, c_j\, m\, (m\, x_o - y_o)\,)(x_f^2 - x_o^2)/2 + (b_i\, b_j + (b_i\, c_j + b_j\, c_i)\, m + c_i\, c_j\, m\,)(x_f^3 - x_o^3)/3 \tag{A1-23}$$

$$\alpha = \tan^{-1} \frac{y_f - y_o}{x_f - x_o}$$

and all other quantities are defined as before.

8 APPENDIX 2--ANALYTIC SOLUTION

The analytic solution to the one-dimensional wave equation, representing the rigid wall case, is presented in this section. This equation is given by

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} + \frac{k_f}{c^2}\frac{\partial p}{\partial t} \tag{A2-1}$$

Applying a Laplace transformation to the above equation yields

$$\frac{d^2 p(x,s)}{dx^2} - \frac{s^2 + sk_f}{c^2}\, p(x,s) = 0 \tag{A2-2}$$

Employing the boundary conditions $p(0,x) = p_o/s$ and $p(L,s) = 0$ results in

$$\frac{p(x,s)}{p_o} = f(s) = \frac{[\sinh(s^2 + sk_f)^{\frac{1}{2}}\,(L/c)][1 - (x/L)]}{s\,\sinh(s^2 + sk_f)^{\frac{1}{2}}\,(L/c)} \tag{A2-3}$$

Eq. (A2-3) has poles at

$$s = 0\ ,\ s = -b \pm i\,\beta_n$$

where

$$b = k_f/2\ ,\ \beta_n = \frac{1}{2}\,(4\,n^2\,\pi^2\,c^2/L^2 - k_f^{\,2})^{\frac{1}{2}}\quad ,\ n = 1, 2, 3, \ldots \tag{A2-4}$$

The inverse transform of eq. (A2-3) is found by the application of the residue theorem, yielding

$$\frac{p(x,t)}{p_o} = 1 - \frac{x}{L} + \sum_{n=1}^{\infty} \frac{(-1)^n\,2\,e^{-bt}}{\beta_n\,L/c}\,[\sin(\beta_n t + \alpha_n)]\,\sin n\,\pi(1 - \frac{x}{L}) \tag{A2-5}$$

where

$$\alpha_n = \tan^{-1}(\beta_n/b)$$

The first term on the right hand side of eq. (A2-5) corresponds to the residue at $s = 0$ and represents the steady state solution, while the series term corresponds to residues at $s = s_n$ and represents the transient solution which is damped out as time progresses.

Employing the pressure distribution from eq. (A2-5), the velocity distribution can be found by direct integration of the momentum equation (eq. (2.1-11))

$$\frac{v_x}{p_o/\rho_f c} = \frac{c(1 - e^{-2bt})}{2\ L\ b} + \sum_{n=1}^{\infty} \frac{2\ (-1)^n}{n\ \pi\ \beta_n} \{e^{-bt}[b\ \sin(\beta_n t + \alpha_n) - \beta_n \cos(\beta_n t + \alpha_n)] - e^{-2bt}[b\ \sin\alpha_n - \beta_n \cos\alpha_n]\} \cdot \cos n\ \pi\ (1 - \frac{x}{L}) \tag{A2-6}$$

REFERENCES

[1] Baker, A.J., "Numerical Solution to the Dynamics of Viscous Fluid Flow by a Finite-Element Algorithm: A First Step Towards Computational Continuum Mechanics", presented at Int. Assoc. for Shell Structures, Pacific Symposium on Hydrodynamically Loaded Shells, Honolulu, Hawaii, 1971.

[2] Baker, A.J., "Finite Element Solution Algorithm for Viscous Incompressible Fluid Dynamics", Int. J. for Numerical Methods in Engineering, vol. 6, 1973, pp. 89-101.

[3] Baker, A.J., "A Finite Element Solution Algorithm for the Navier-Stokes Equations", NASA CR-2391, May, 1973.

[4] Chan, S.P., Cox, H.L., Benfield, W.A., "Transient Analysis of Forced Vibrations of Complex Structural-Mechanical Systems, J. Roy. Aeronaut. Soc., July, 1962.

[5] Chen, S.S., Rosenberg, G.S., "Free Vibrations of Fluid-Conveying Cylindrical Shellf", ASME 73-DET-96, 1973.

[6] Conway, H.D., Jakubowski, M., "Axial Impact of Short Cylindrical Bars", ASME 69-WA/APM-6, 1969.

[7] Craggs, A., "The Transient Response of Coupled Acousto-Mechanical Systems", NASA CR-1421, 1969.

[8] Feng, G.C., Kiefling, L., "Fluid-Structure Finite-Element Vibrational Analysis", AIAA 74-102, AIAA 12th Aerospace Sciences Meeting, Washington, D.C., 1974.

[9] Fox, G.L. Jr., Stepnewski, D.D., "Pressure Wave Transmission in a Fluid Contained in a Plastically Deforming Pipe", ASME 74-PVP-26, 1974.

[10] Greenspon, J.E., Fluid-Solid Interaction, New York: The American Society of Mechanical Engineers, 1967.

[11] Hine, M.J., "Acoustically Induced Vibrations of Slender Rods in a Cylindrical Duct", ASME 72-WA/APM-8, 1972.

[12] Kuo, S.S., Numerical Methods and Computers, Reading, Mass.: Addison-Wesley, 1965.

[13] Love, A.E.H., A Treatise on the Mathematical Theory of Elasticity, New York: Dover Publications, 1944.

[14] MacNeal, R.H., The NASTRAN Theoretical Manual, NASA SP-221(01), December, 1972, 11.3-1 to 11.3-13.

[15] Martin, R.S., Wilkinson, J.H., "Reduction of the Symmetric Eigenproblem $A x = \lambda B x$ and Related Problems to Standard Form", Numerische Mathematik, vol. 11, 1968, pp. 99-110.

[16] Nahavandi, A.N., Sun, B.C., Ball, W.H., "A Simple Solution of Sound Transmission Through an Elastic Wall to a Rectangular Enclosure", Proc. Internoise 76, Washington, D.C., 1976, pp. 251-254.

[17] Nahavandi, A.N., Sun, B.C., Ball, W.H., "Sound Transmission Through an Elastic Wall to a Rectangular Enclosure", Advances in Engineering Science, vol. 3, 13th Annual Meeting of the Society of Engineering Science, Hampton, Va., 1976, pp. 907-916, NASA CP-2001.

[18] Newmark, N.M., "A Method of Computation for Structural Dynamics", J. Eng. Mech. Div., ASCE, vol. 85 (EM3), July, 1959, p. 67.
[19] Paidoussis, M.P., "Vibration of Cylindrical Structures Induced by Axial Flow", ASME 73-DET-117, 1973.
[20] Przemieniecki, J.S., Theory of Matrix Structural Analysis, New York: McGraw-Hill, 1968.
[21] Sharp, G.R., Wenzel, W.A., "Hydrodynamic Mass Matrix for a Multi-bodied System", ASME 73-DET-121, 1973.
[22] Streeter, V.J., Wylie, E.B., Hydraulic Transients, 1st ed., New York: McGraw-Hill, 1967.
[23] Szilard, R., Hydrodynamically Loaded Shells, Honolulu: The University Press of Hawaii, 1973.
[24] Tong, P., "The Finite Element Method for Fouid Flow", Finite Element Methods in Flow Problems, Huntsville, Ala.: UAH Press, The University of Alabama in Huntsville, 1974.
[25] Wylie, C.R. Jr., Advanced Engineering Mathematics, 3rd ed., New York: McGraw-Hill, 1966.
[26] Zienkiewicz, O.C., The Finite Element Method in Engineering Science, London: McGraw-Hill, 1971.

PART TWO

USER'S MANUAL

1 DESCRIPTION OF FLINTS PROGRAM

The program FLINTS (Fluid Interacting with Solid) consists of a main program and several subroutines which were written to solve the problem of interaction between a solid and a fluid in a two-dimensional medium. At the user's option, the program may also be used to analyze totally fluid and totally solid continua. Briefly, the program and its subroutines perform the following functions:

PROGRAM OR SUBROUTINE NAME	FUNCTIONS
MAIN	Controls the calling sequence of the other subroutines and the normal termination of the program.
CLEAR	Initializes all labelled common blocks to zero.
GDATA	Reads and prints the input data and initializes time and iteration number.
LOAD	Reads in and assembles applied loads and specified freedoms and initializes the problem.
FORMK	Forms global mass and stiffness matrices by calling STIFT1, STIFT2, and STIFT3 subroutines and reduces them to condensed form for modal analysis.
STIFT1(N)	Finds mass and stiffness matrices for solid element N. In stress pass, calculates stresses in the solid element.

PROGRAM OR SUBROUTINE NAME	FUNCTIONS
STIFT2(N)	Finds the inertia and fluidity matrices for fluid element N. In stress pass calculates velocities in the fluid element.
STIFT3(N)	Finds the mass, inertia, stiffness, fluidity and coupling matrices for solid-fluid superelement N. In stress pass, calculates solid stresses and fluid velocities for solid-fluid element.
SF(I,J)	Function subroutine used to calculate solid-fluid coupling matrix.
MODAL	Computes the eigenvalues and eigenvectors for the system and from them obtains the generalized mass and stiffness matrices. Transforms the load array, initial displacements, and initial velocities to generalized form. At this point, the entire problem is ready for solution.
REDUC1(A,B,N)	Reduces the matrix equation $[A]\{x\} = \lambda[B]\{x\}$ to $[K]\{z\} = \lambda\{z\}$ as described in Section 1.11 .
JACOBI(N,Q,JVEC,M,V)	Finds the eigenvalues and eigenvectors.

PROGRAM OR SUBROUTINE NAME	FUNCTIONS
REBAKA	Transforms the eigenvectors, which are in terms of $\{z\}$, to the original coordinates $\{x\}$ as described in Section 1.13 .
DIS	Finds analytic solution to uncoupled matrix differential equation.
MODMAK	Expands the freedom and acceleration arrays to global form for use in calculating the updated coupling loads and for back-substitution in the stress pass. Also modifies nodal coordinate array for large displacements.
STRESS	Controls the calling sequence of the stiffness subroutines for the calculation of the elemental stresses and velocities

Although some logic is available in the program for time dependent applied loads and specified freedoms, this option is not fully developed in this version of FLINTS and use of it would require an excessive and inconvenient amount of input data. The prospective user is urged to explore this option for himself.

A more detailed description of each subroutine follows.

1.1 MAIN Program

The MAIN program controls the calling sequence of other subroutines and the normal termination of the program. A flowchart of MAIN is given in Figs. F-1a through F-1d . Fig. F-1a is the flowchart convention. As the program begins, the input data describing the geometry of the problem are read in. The formation of the mass and stiffness matrices and the calculation of the eigenvalues and eigenvectors are performed only once unless it is desired that these matrices be updated due to large solid displacements. The solution is then obtained in an iterative loop based on the elapsed time of the problem. If a solid-fluid interaction problem is being solved, or if the stresses are desired, additional subroutines for the manipulation of the output data are called. The program normally terminates when the elapsed problem time is equal to or greater than the final time.

1.2 CLEAR Subroutine

CLEAR is a subroutine which sets all variables in a labelled COMMON block equal to zero. Its calling sequence is

```
CALL CLEAR (AMEMBR, LENGTH)
```

in which AMEMBR is the name of the first variable in the COMMON block and LENGTH is the total number of variables occupying the COMMON block. For example, if a COMMON block is of the form

```
COMMON/XTEST/X,Y,A(10)
```

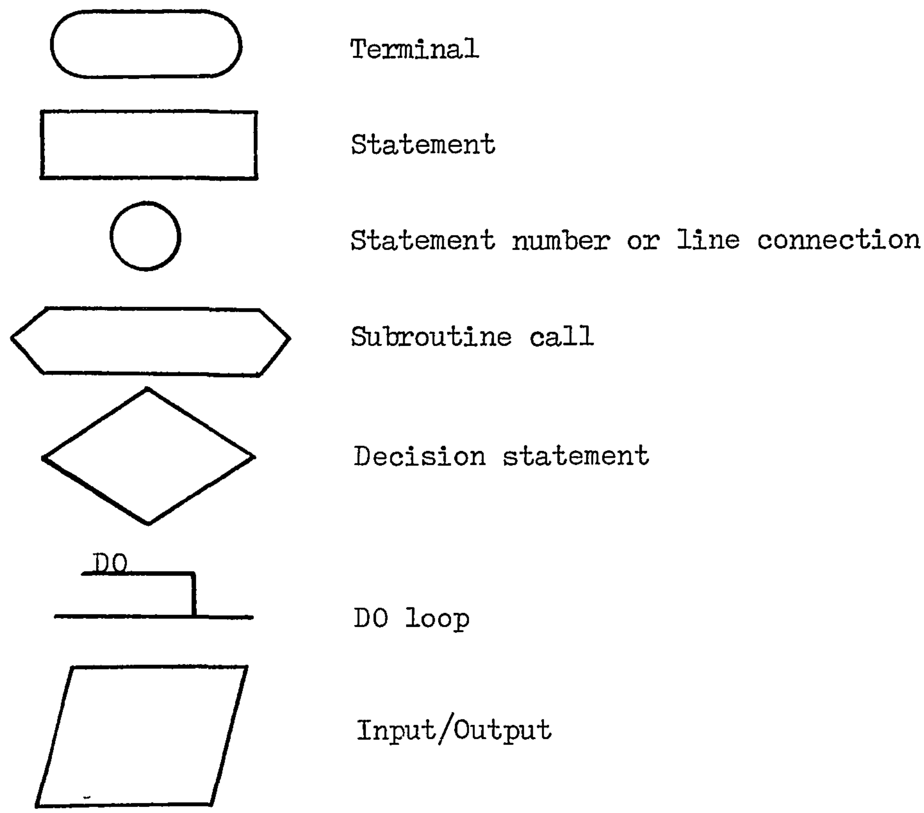
the calling sequence for CLEAR would be

```
CALL CLEAR(X,12)
```

A flowchart of CLEAR is in Fig. F-2.

Fig. F-1a Flowchart Convention

All flowcharts are to be read from left to right unless indicated otherwise. The following symbols are used throughout these Figures.



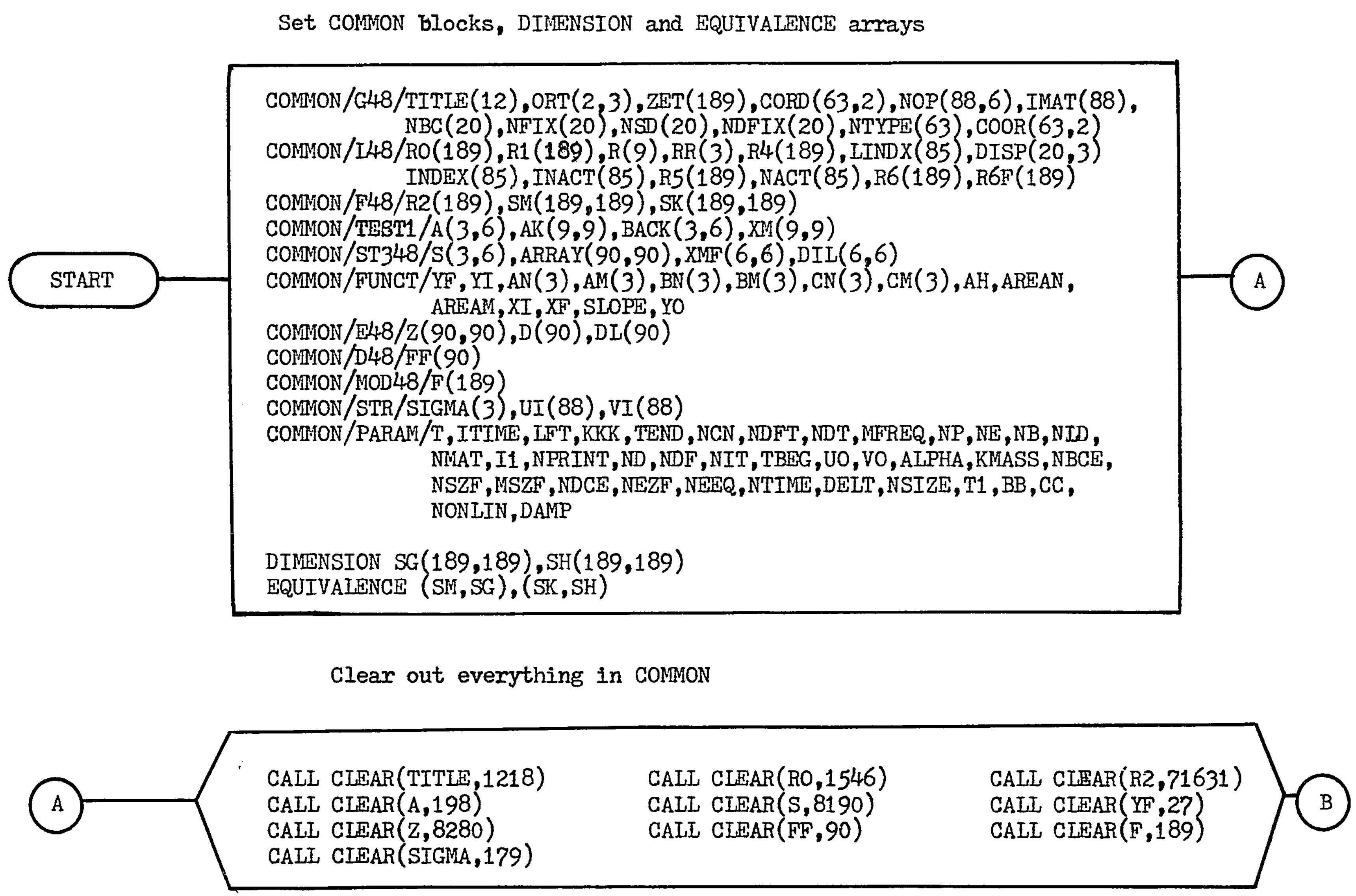


Fig. F-1b Flowchart of MAIN Program

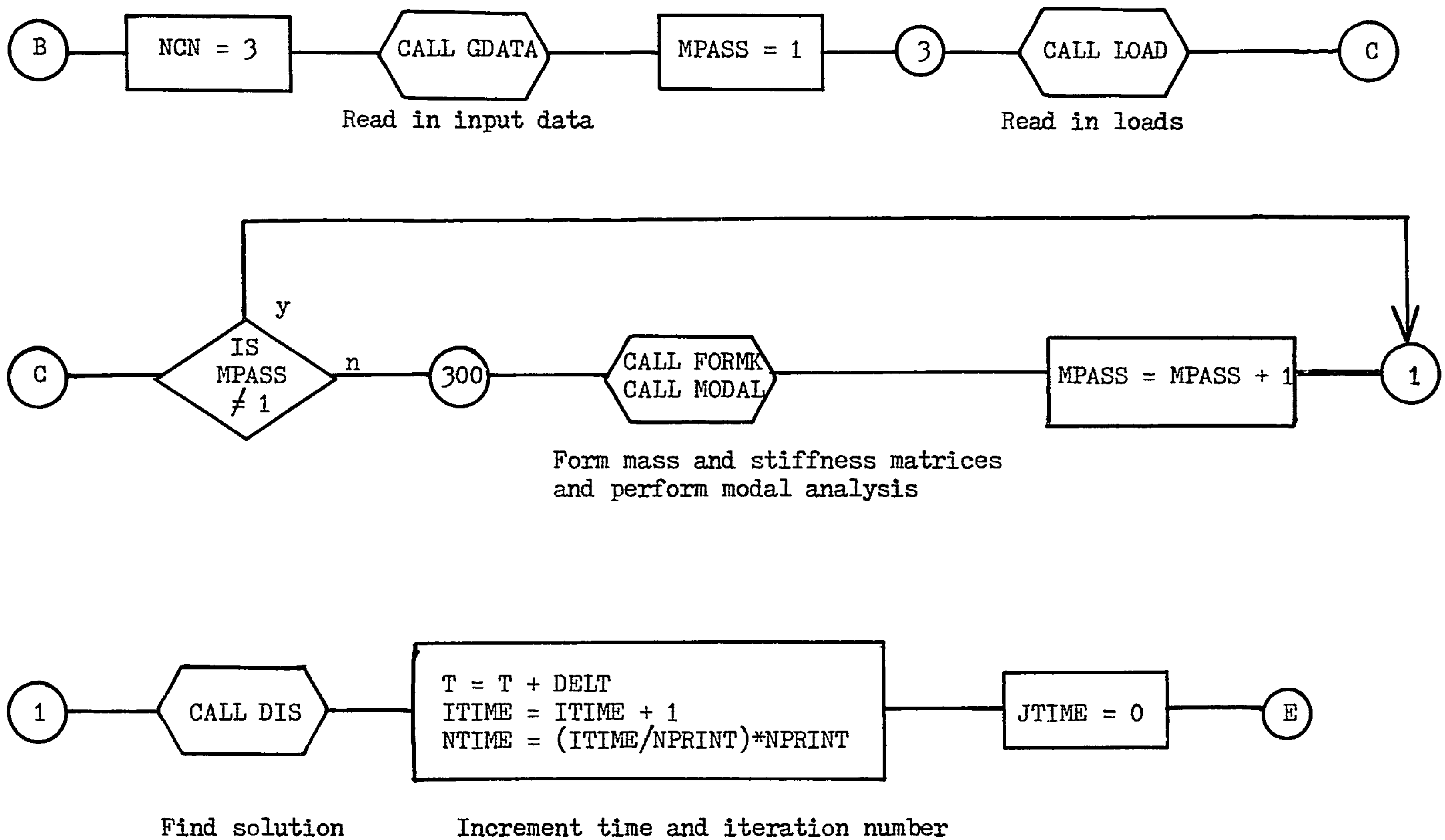


Fig. F-1c Flowchart of MAIN Program

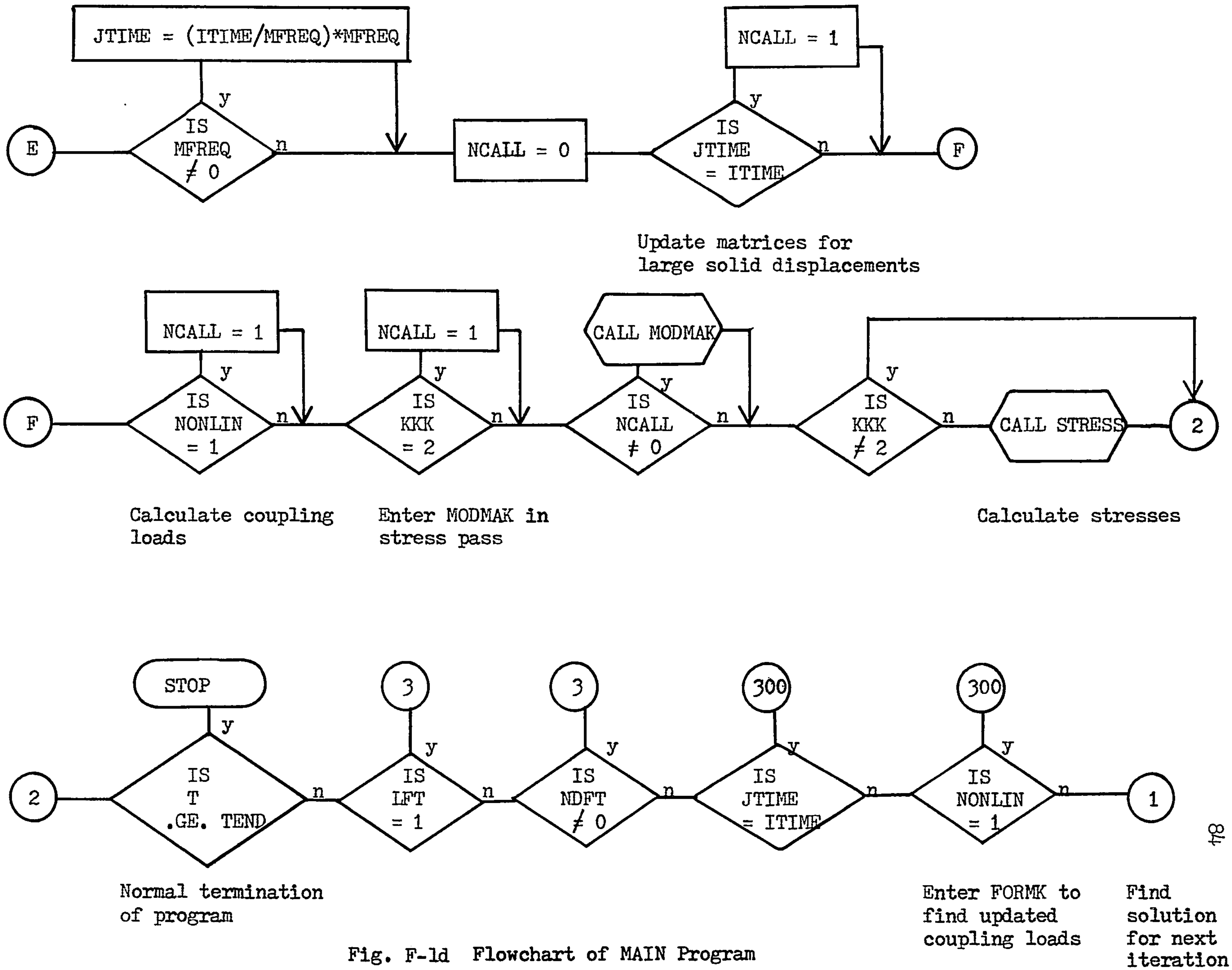


Fig. F-1d Flowchart of MAIN Program

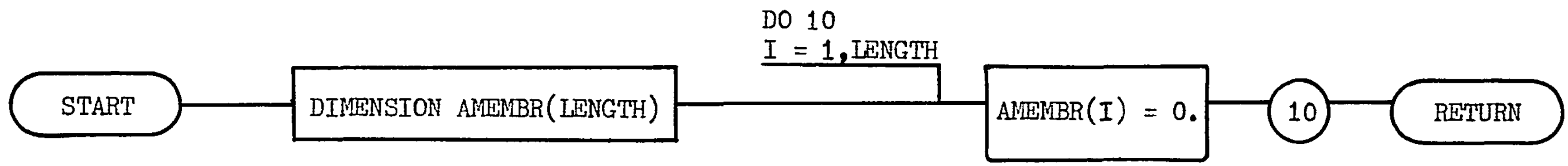


Fig. F-2 Flowchart of Subroutine CLEAR

1.3 GDATA Subroutine

GDATA reads in and prints out all the input data describing the finite element problem. The data which are read in are described in Section 2, Description of Input Data. GDATA also initializes the elapsed time and iteration number. Its calling sequence is

CALL GDATA

A flowchart of this subroutine is given in Fig. F-3.

1.4 LOAD Subroutine

Aside from reading in and assembling the global applied load array and reading in the specified displacements and pressures, LOAD also functions as the system initialization routine. It determines the global system parameters, such as the size of the unrestrained global matrix, etc. Further, it sets up indicial arrays of row numbers indicating the location in the global matrix of specified freedoms, inactive freedoms, pressure freedoms, and active row numbers for modal analysis. The calling statement for LOAD is

CALL LOAD

Its flowchart is given in Fig. F-4a through F-4j.

1.5 FORMK Subroutine

Subroutine FORMK is entered generally in the first iteration. If a solid-fluid problem is being analyzed or if updating the matrices is required because of large solid displacements, FORMK is entered also in subsequent iterations. Its function is to find and assemble the mass and stiffness matrices and the interactive load array for the finite element model. FORMK calls the library of stiffness

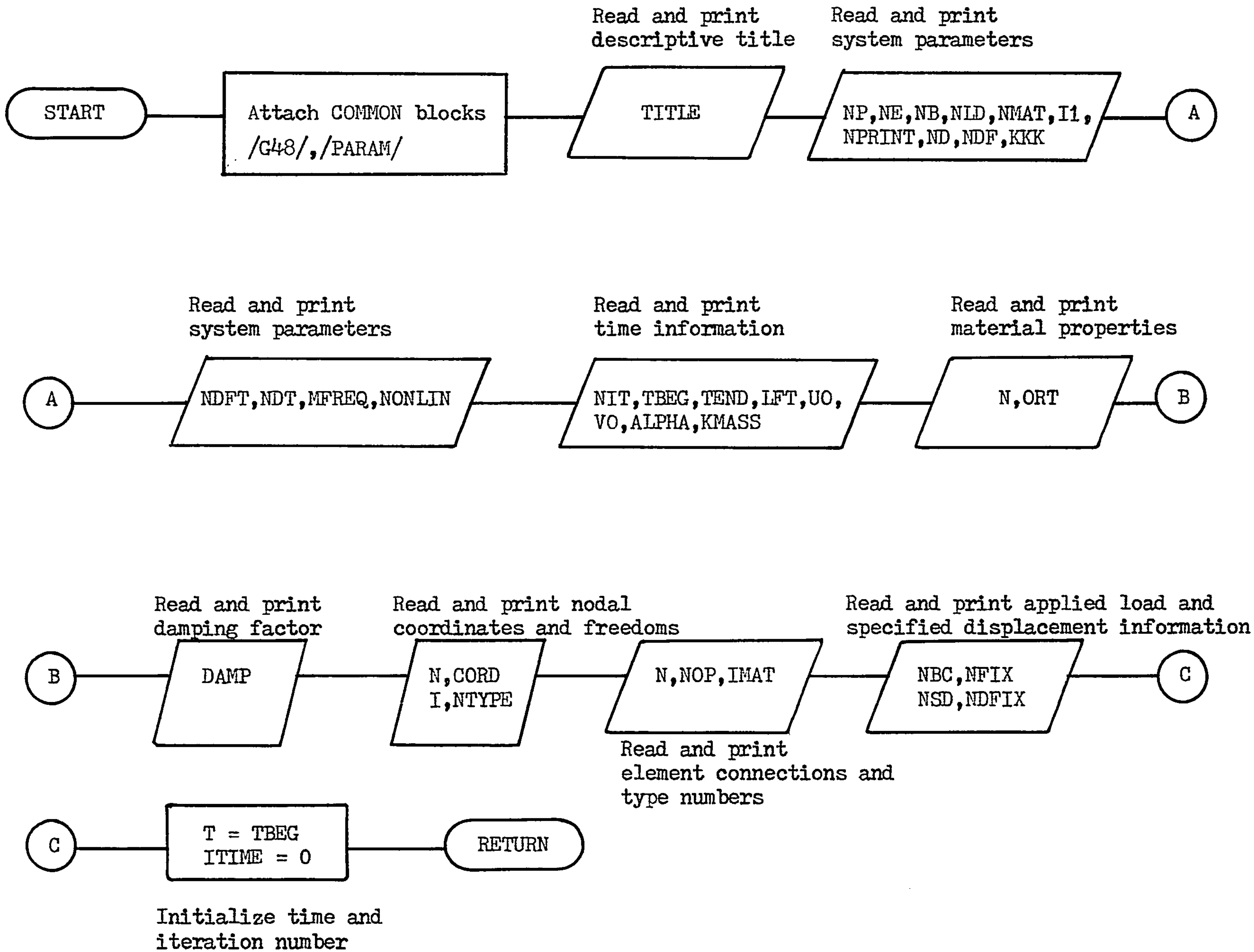


Fig. F-3 Flowchart of Subroutine GDATA

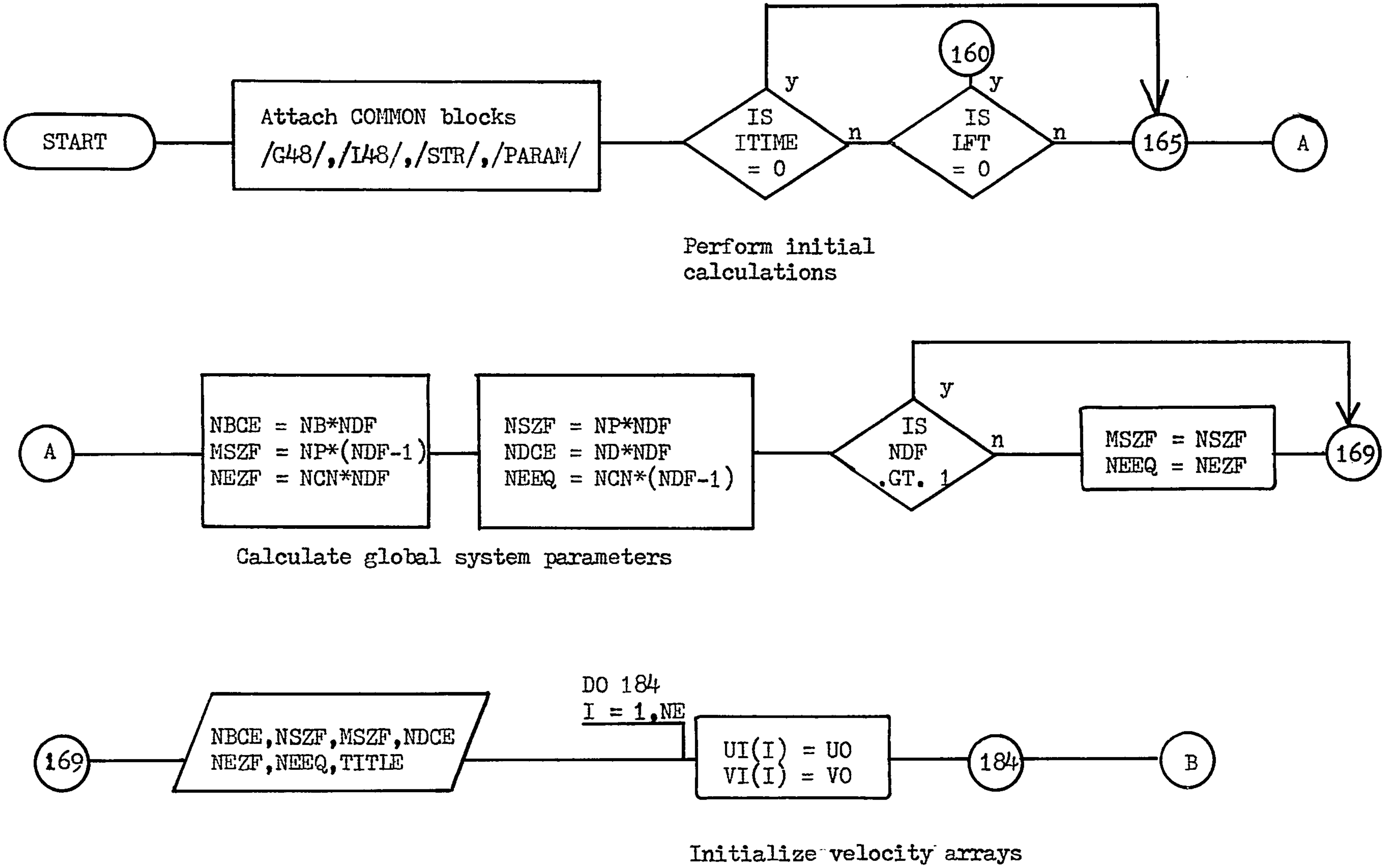


Fig. F-4a Flowchart of Subroutine LOAD

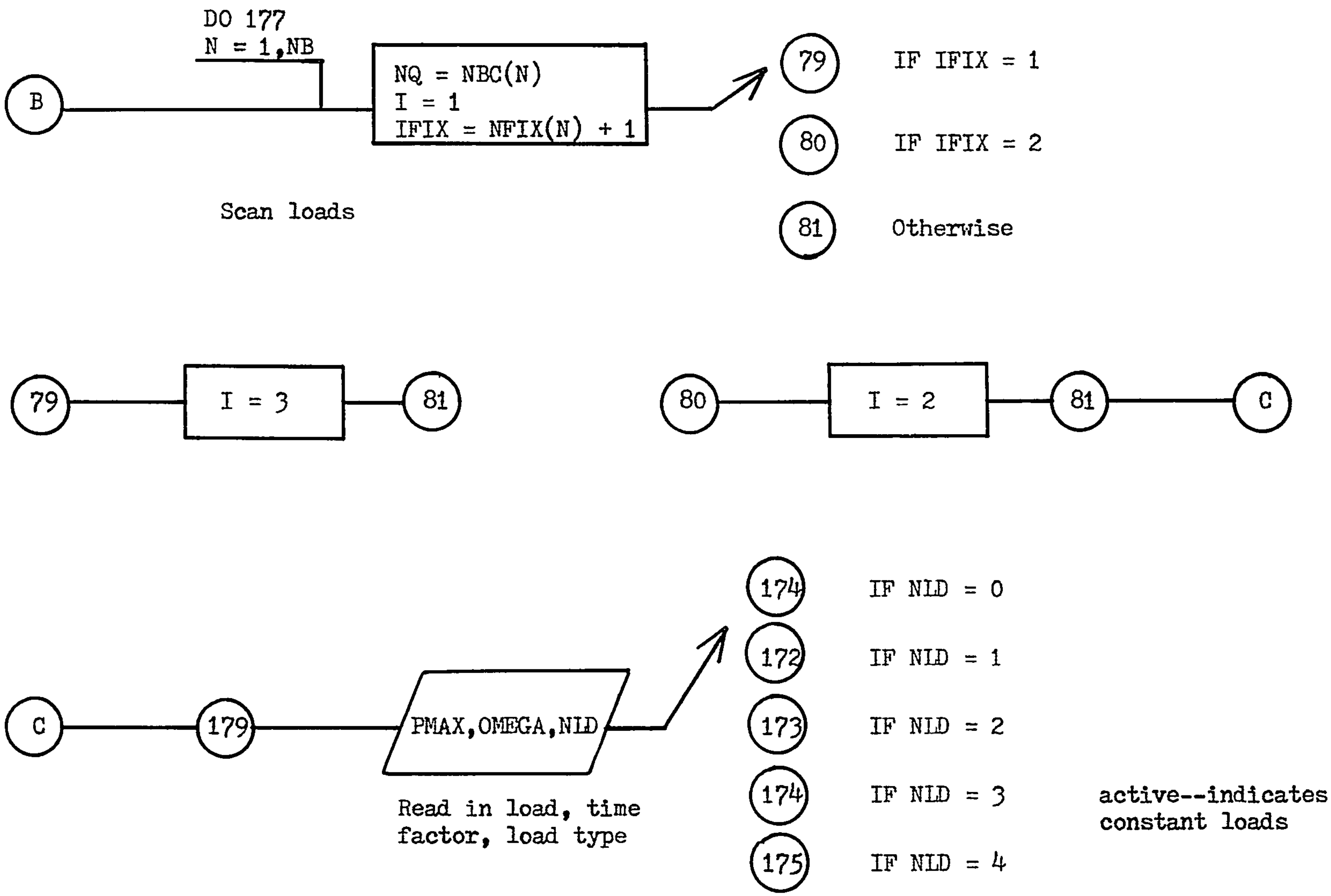


Fig. F-4b Flowchart of Subroutine LOAD

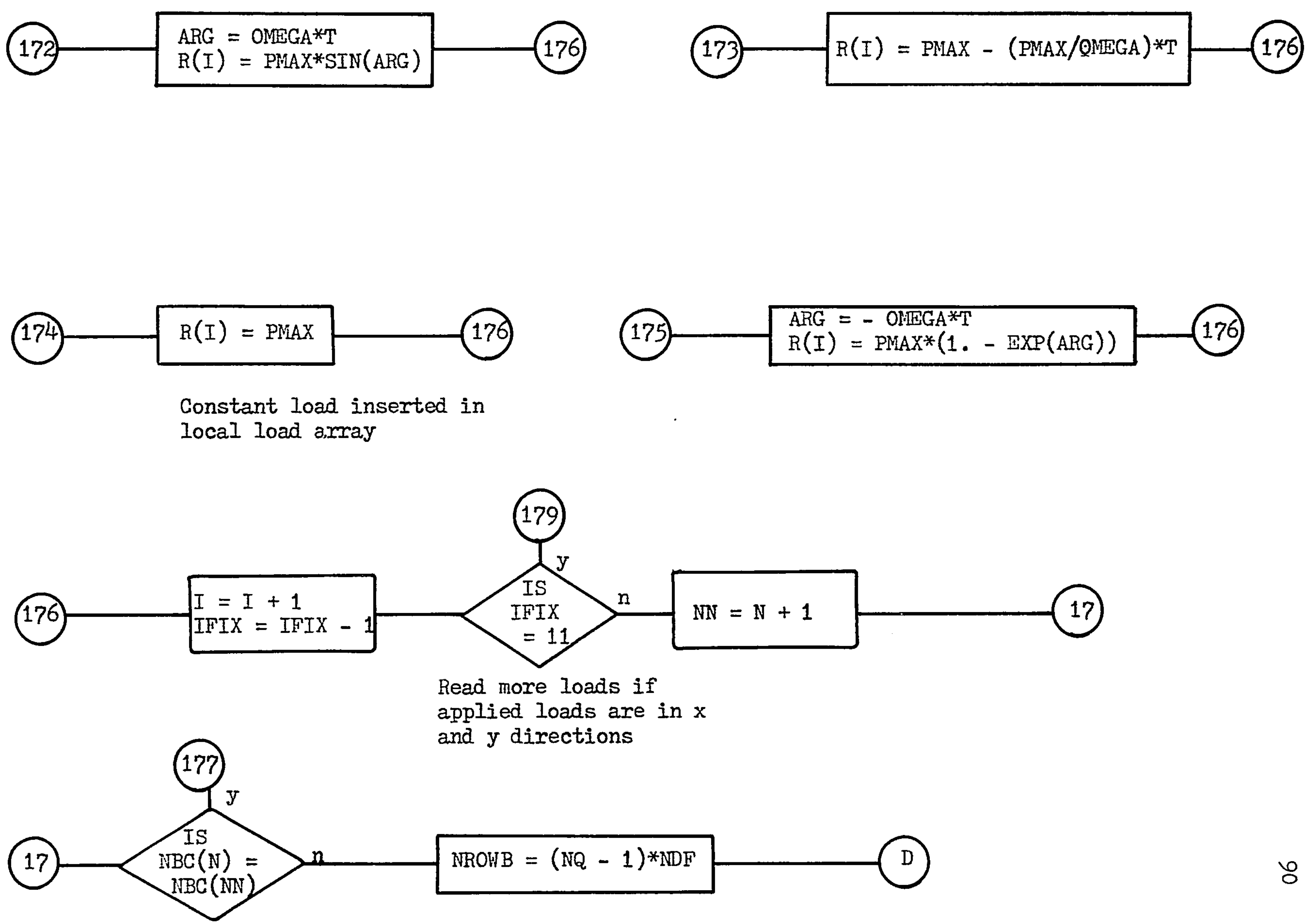


Fig. F-4c Flowchart of Subroutine LOAD

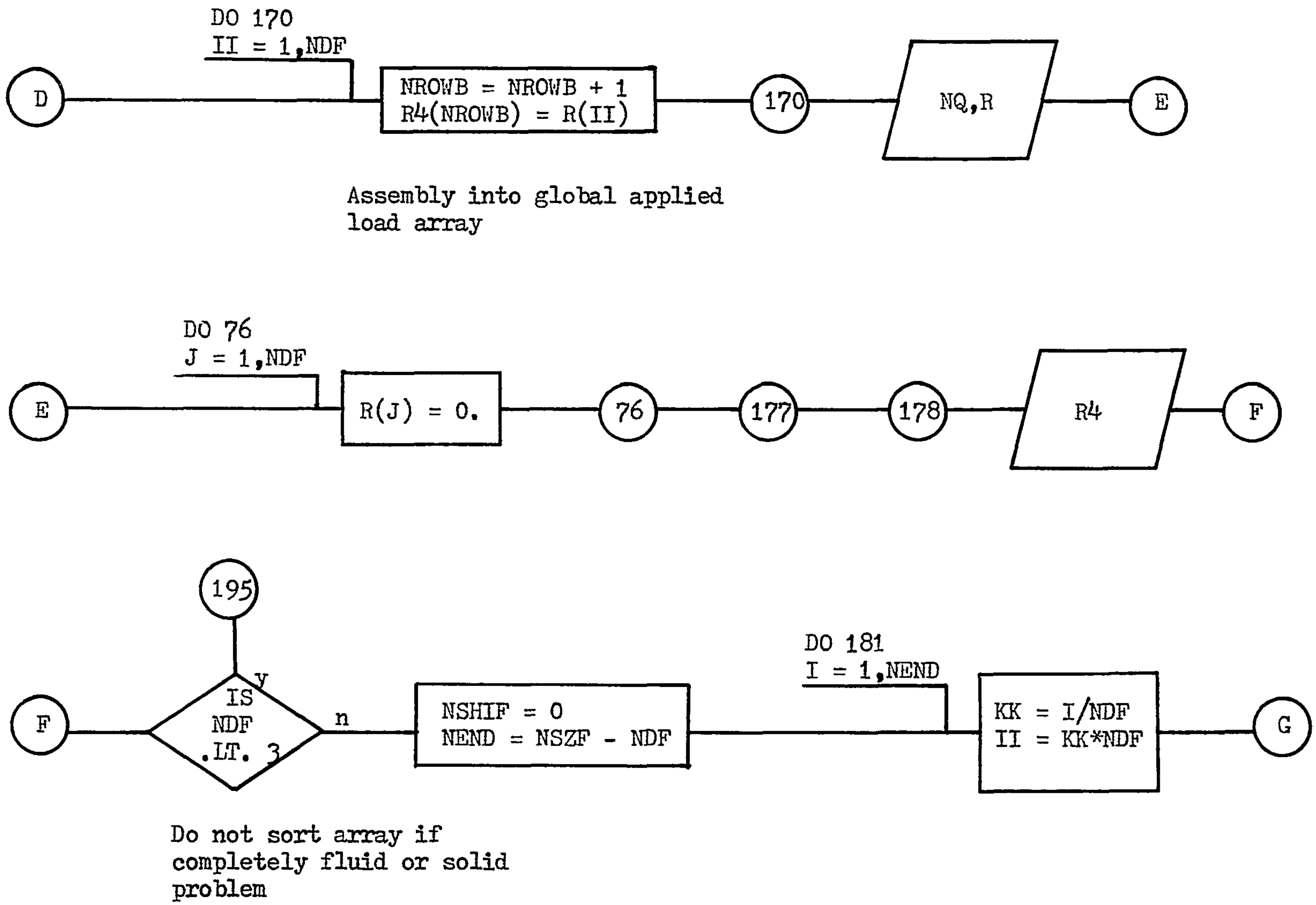


Fig. F-4d Flowchart of Subroutine LOAD

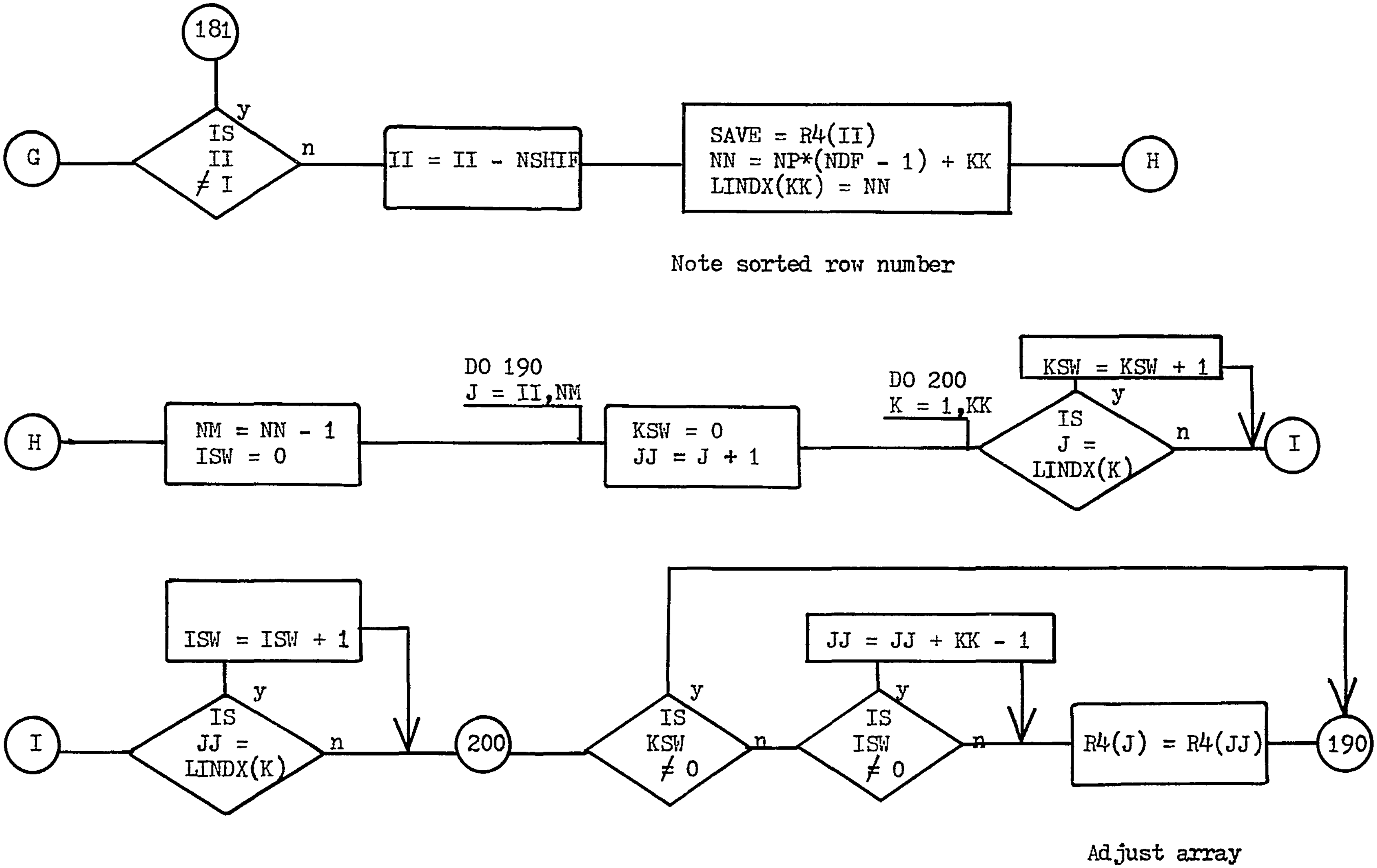


Fig. F-4e Flowchart of Subroutine LOAD

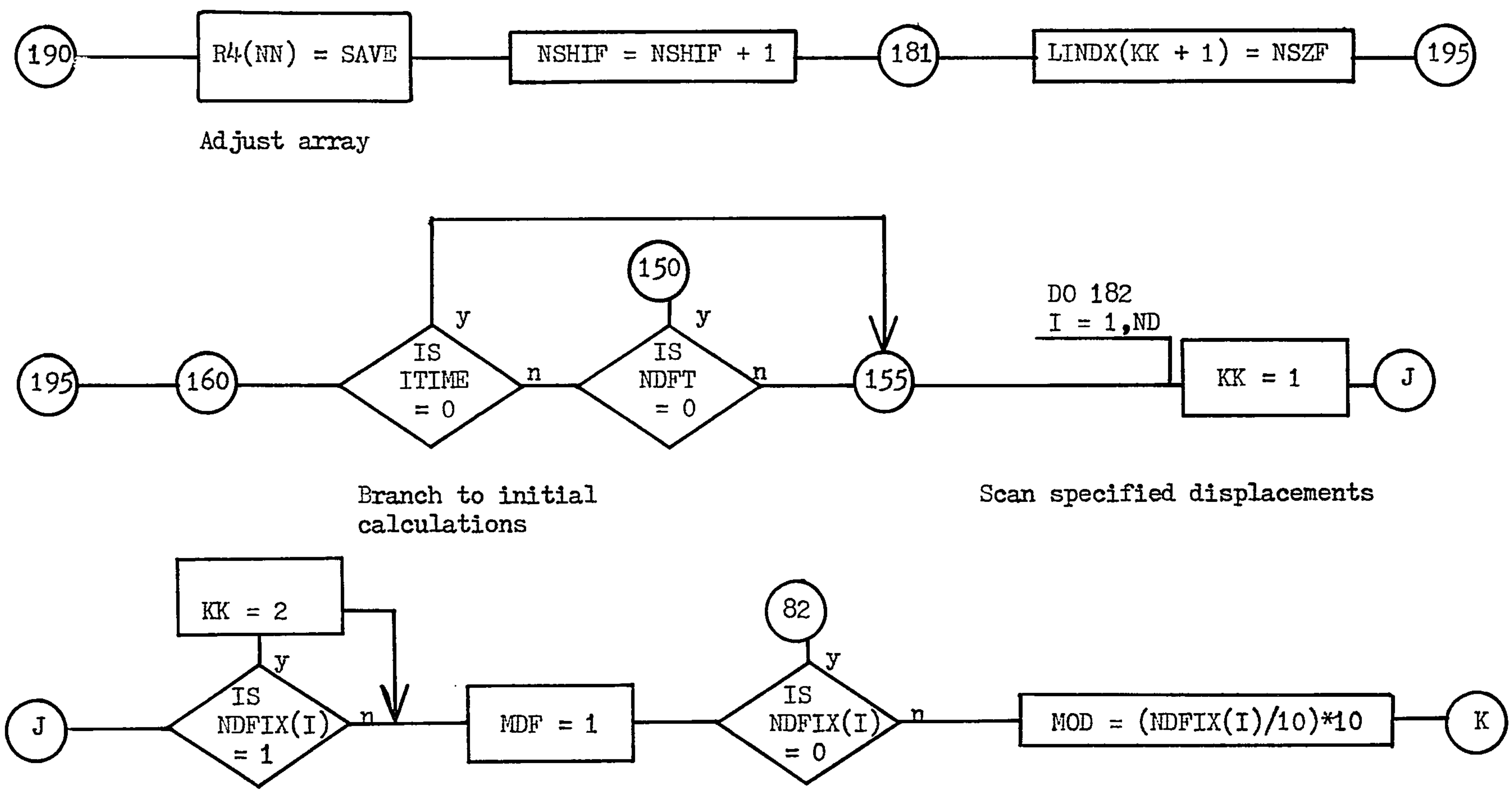


Fig. F-4f Flowchart of Subroutine LOAD

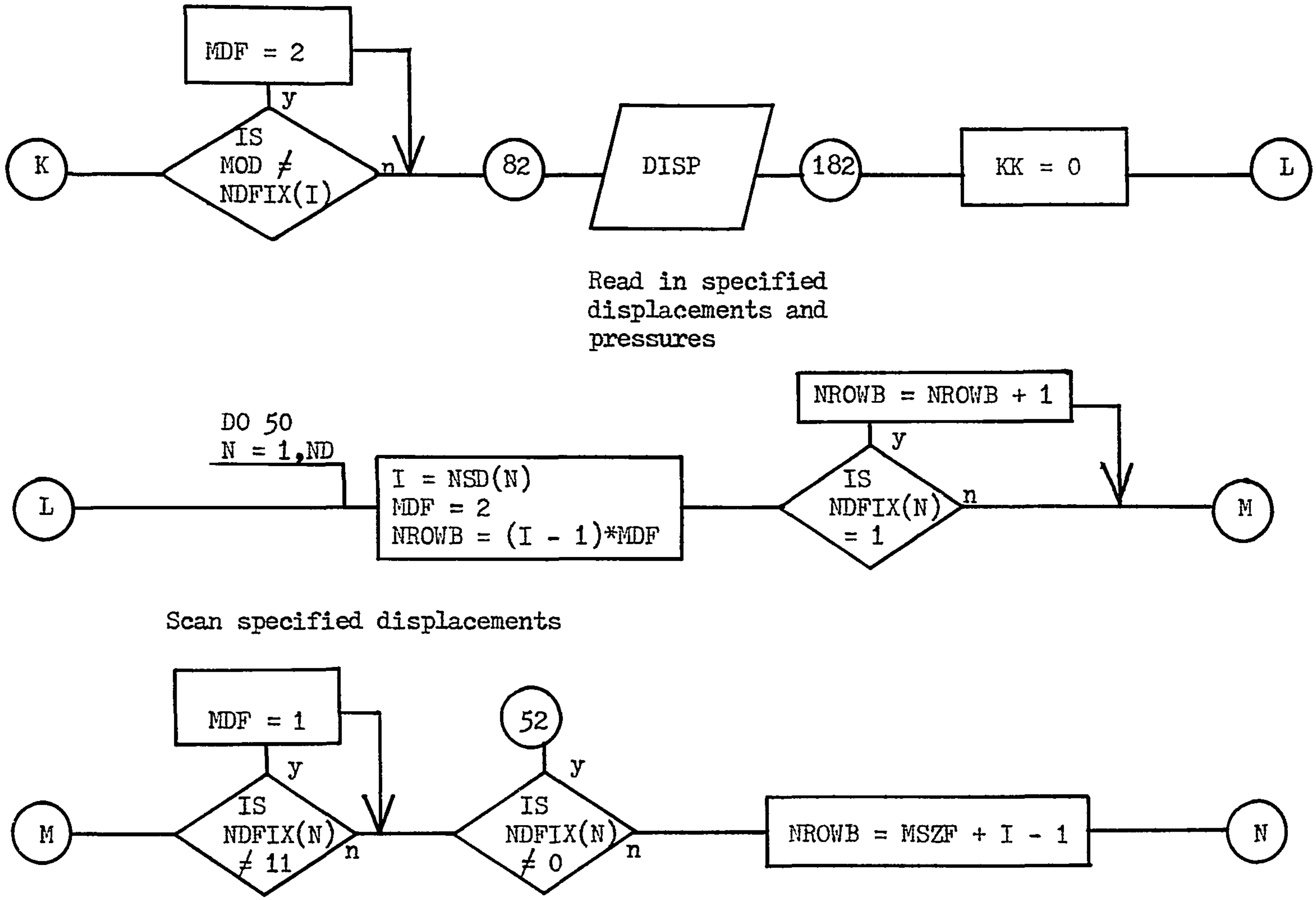


Fig. F-4g Flowchart of Subroutine LOAD

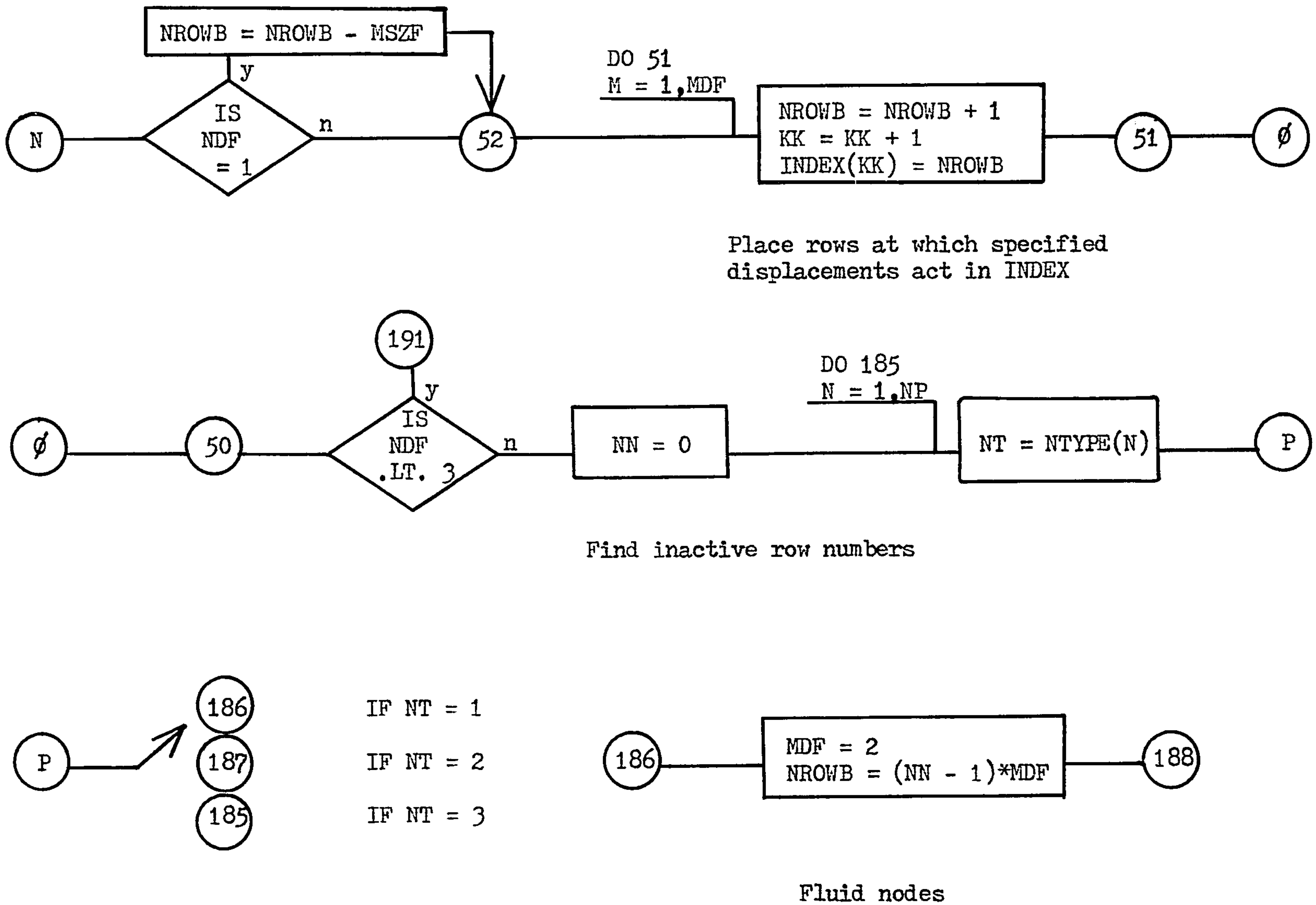


Fig. F-4h Flowchart of Subroutine LOAD

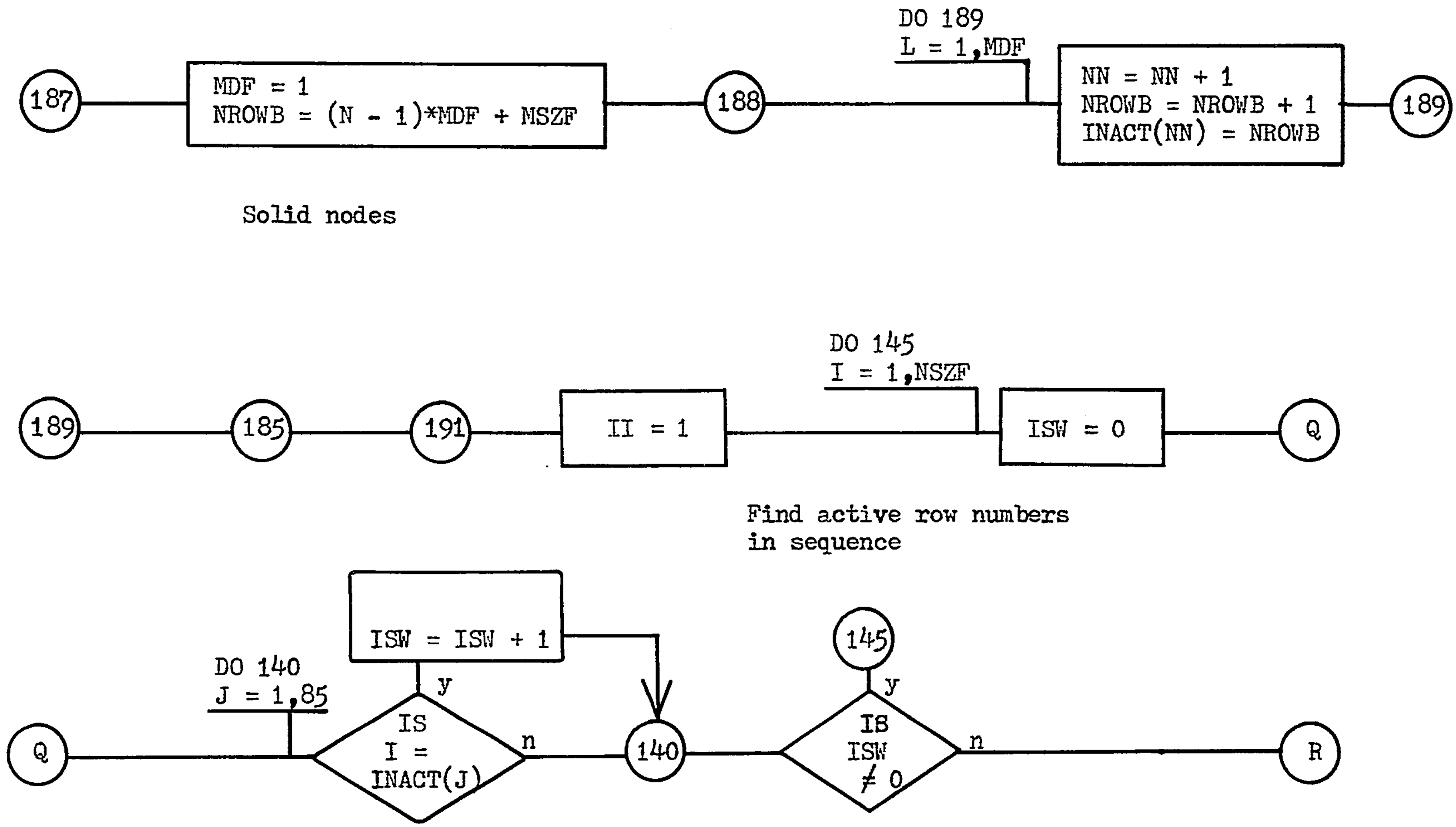


Fig. F-4i Flowchart of Subroutine LOAD

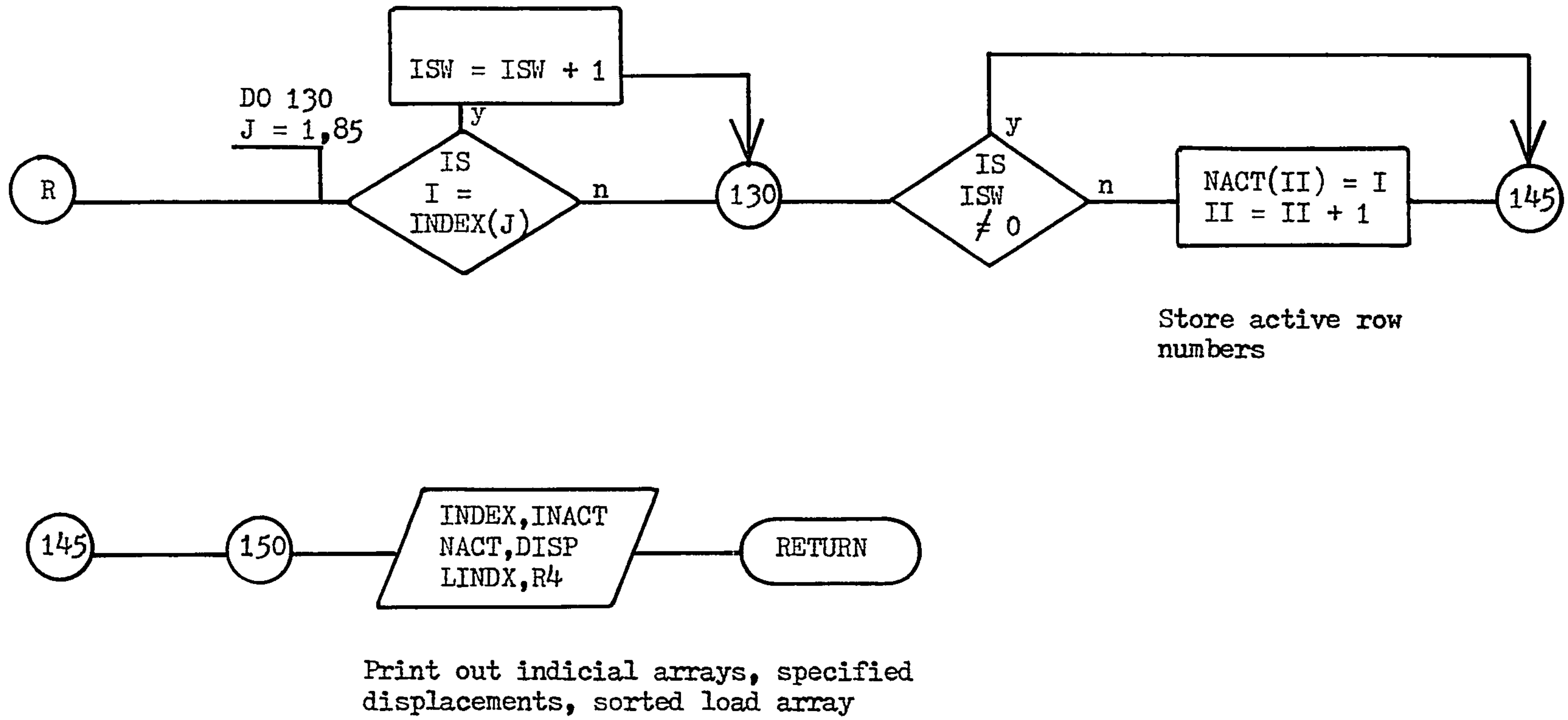


Fig. F-4j Flowchart of Subroutine LOAD

subroutines, STIFT1, STIFT2, and STIFT3 to accomplish this. FORMK then inserts the specified displacements into the system and condenses the matrices and arrays so that at the end of the subroutine, the system is ready for modal analysis. If FORMK is entered because the coupling loads are to be calculated, FORMK calls STIFT3 to obtain these loads and then assembles them into the updated coupling load array. The calling sequence for this subroutine is

CALL FORMK

Its flowchart is given in Fig. F-5a through F-5o .

1.6 STIFT1(N) Subroutine

Subroutine STIFT1(N) is called in the displacement pass and in the stress pass. In the displacement pass, STIFT1(N) is called by FORMK to calculate the mass and stiffness matrices for the solid plane linear triangular finite element N. In the stress pass, STIFT1(N) is called by STRESS. The stress back substitution matrix is calculated and used in conjunction with the elemental nodal displacements to determine the elemental stresses. The calling statement for this subroutine is

CALL STIFT1(N)

in which the argument N is the solid element number. The flowchart for STIFT1(N) is given in Fig. F-6a through F-6h .

1.7 STIFT2(N) Subroutine

Subroutine STIFT2(N) is called in the displacement pass and in the stress pass. In the displacement pass, STIFT2(N) is called by FORMK to calculate the inertia and fluidity matrices for the fluid plane linear triangular finite element N. In the stress

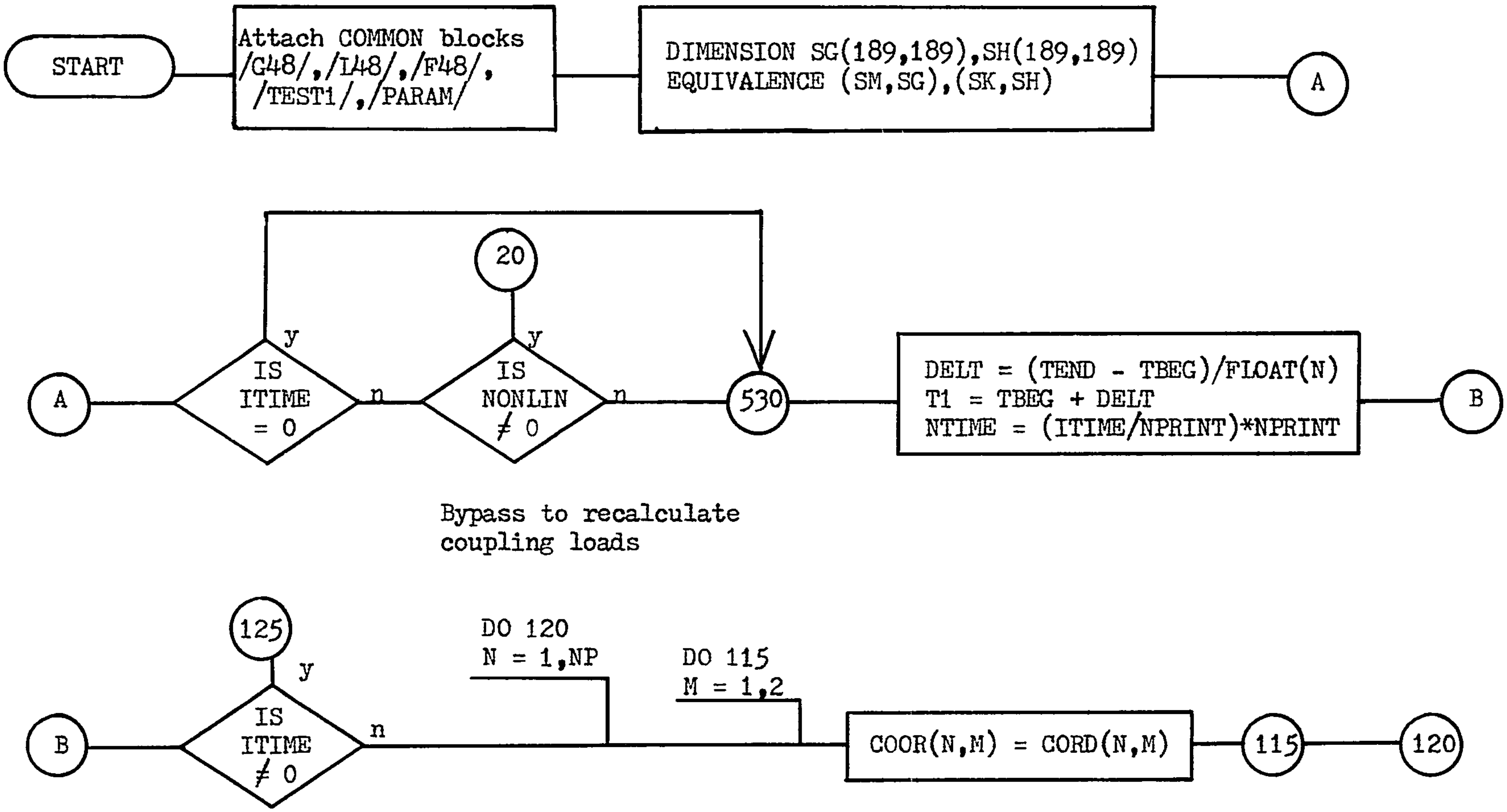


Fig. F-5a Flowchart of Subroutine FORMK

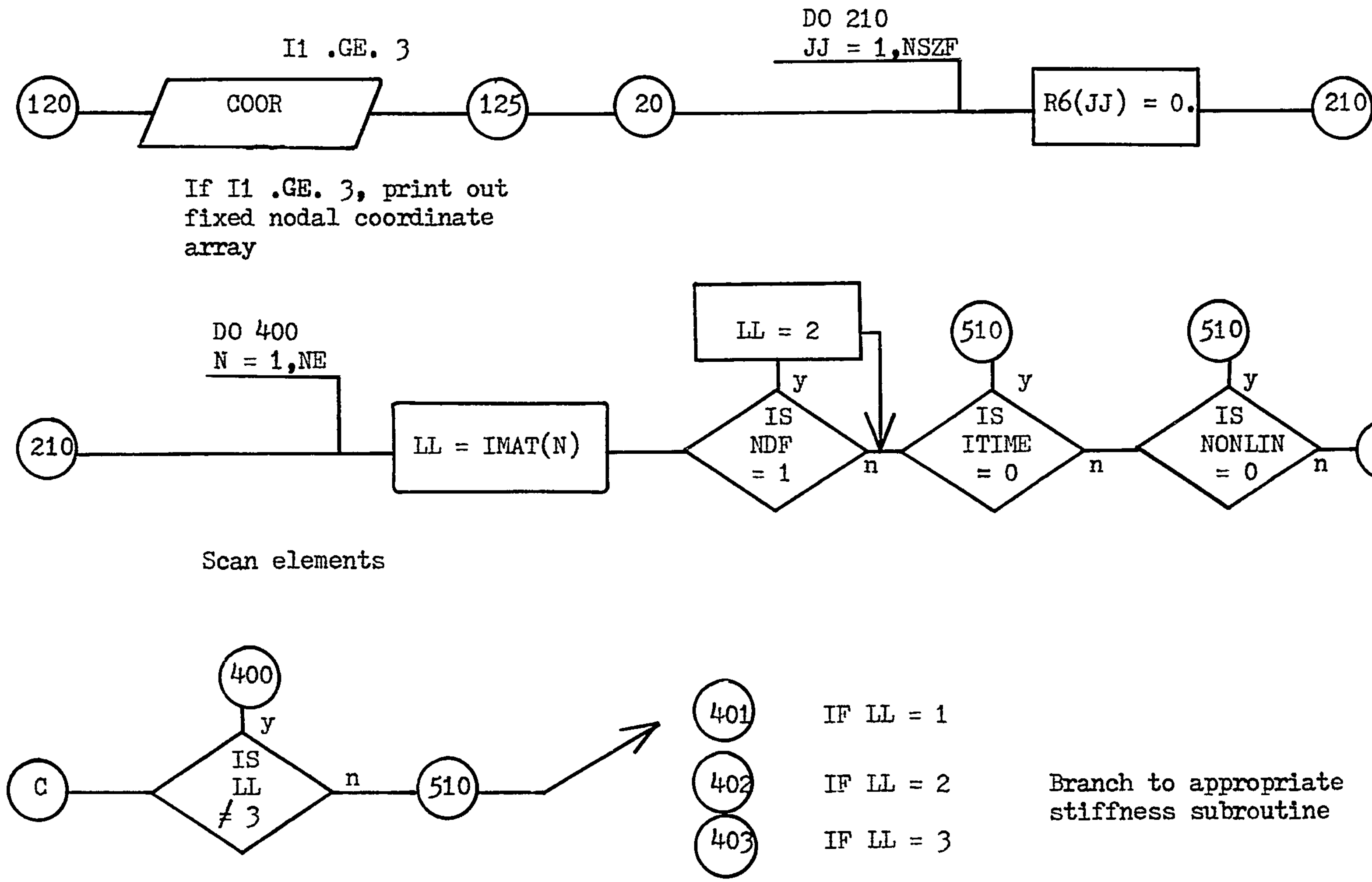


Fig. F-5b Flowchart of Subroutine FORMK

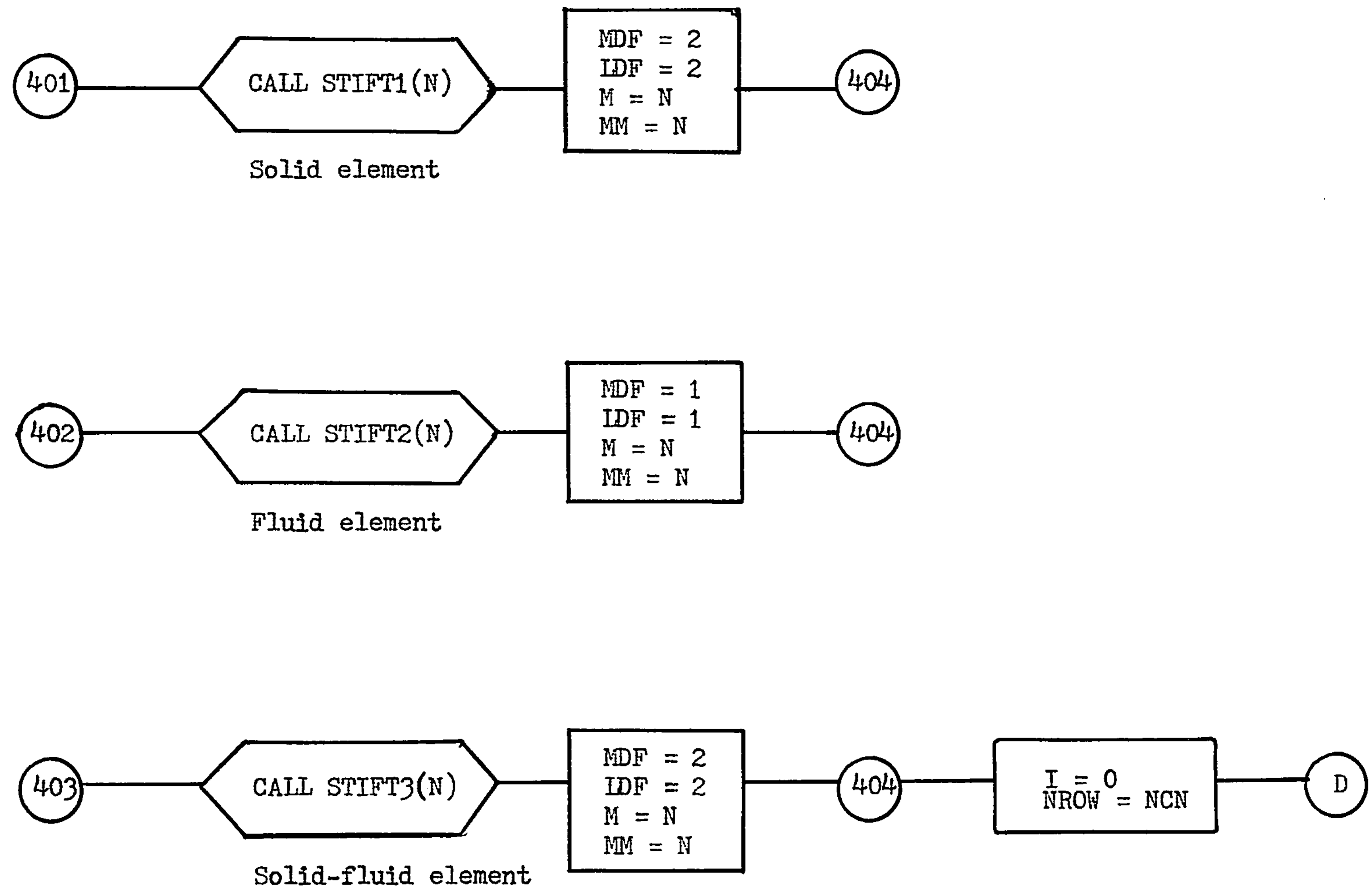


Fig. F-5c Flowchart of Subroutine FORMK

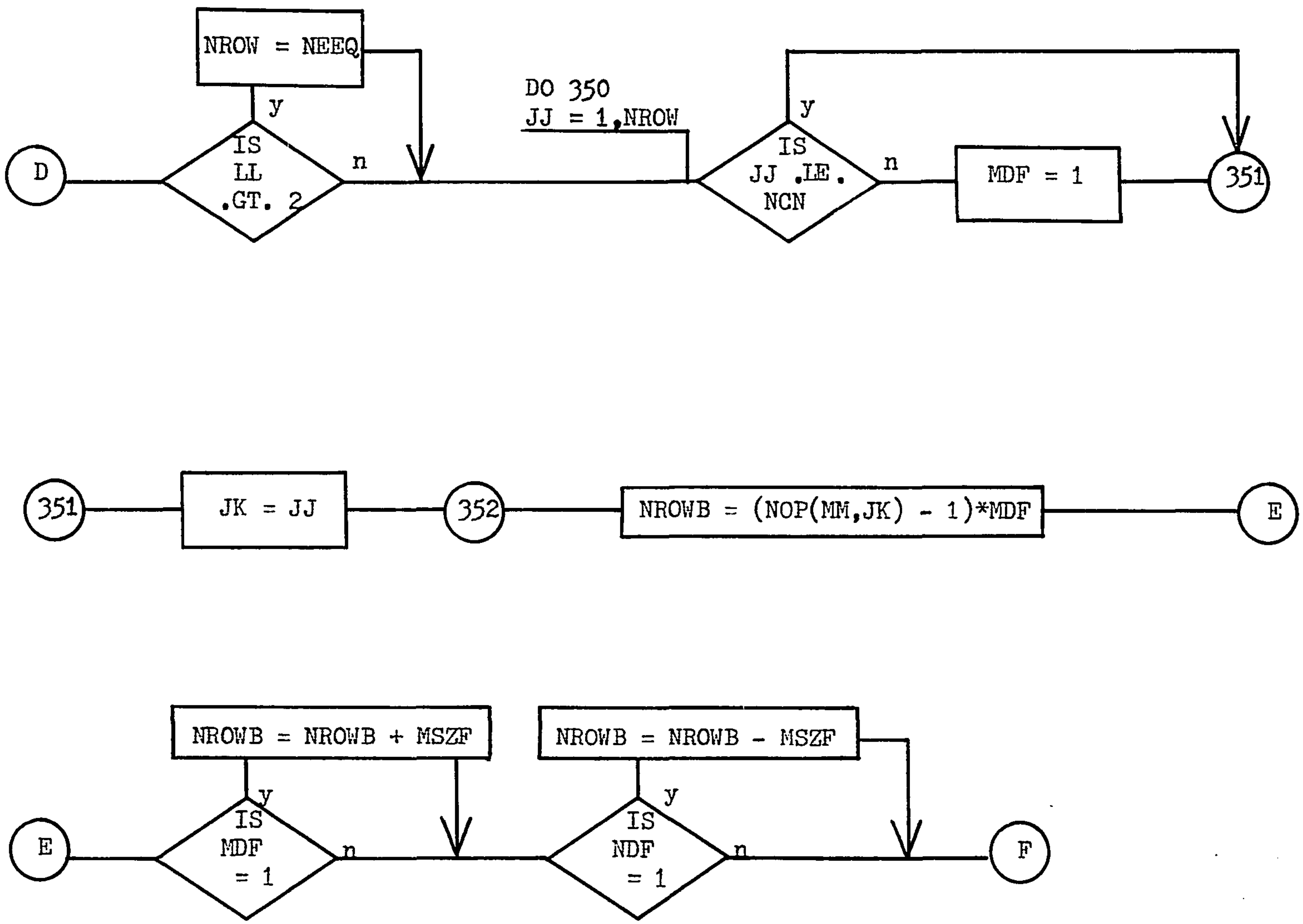


Fig. F-5d Flowchart of Subroutine FORMK

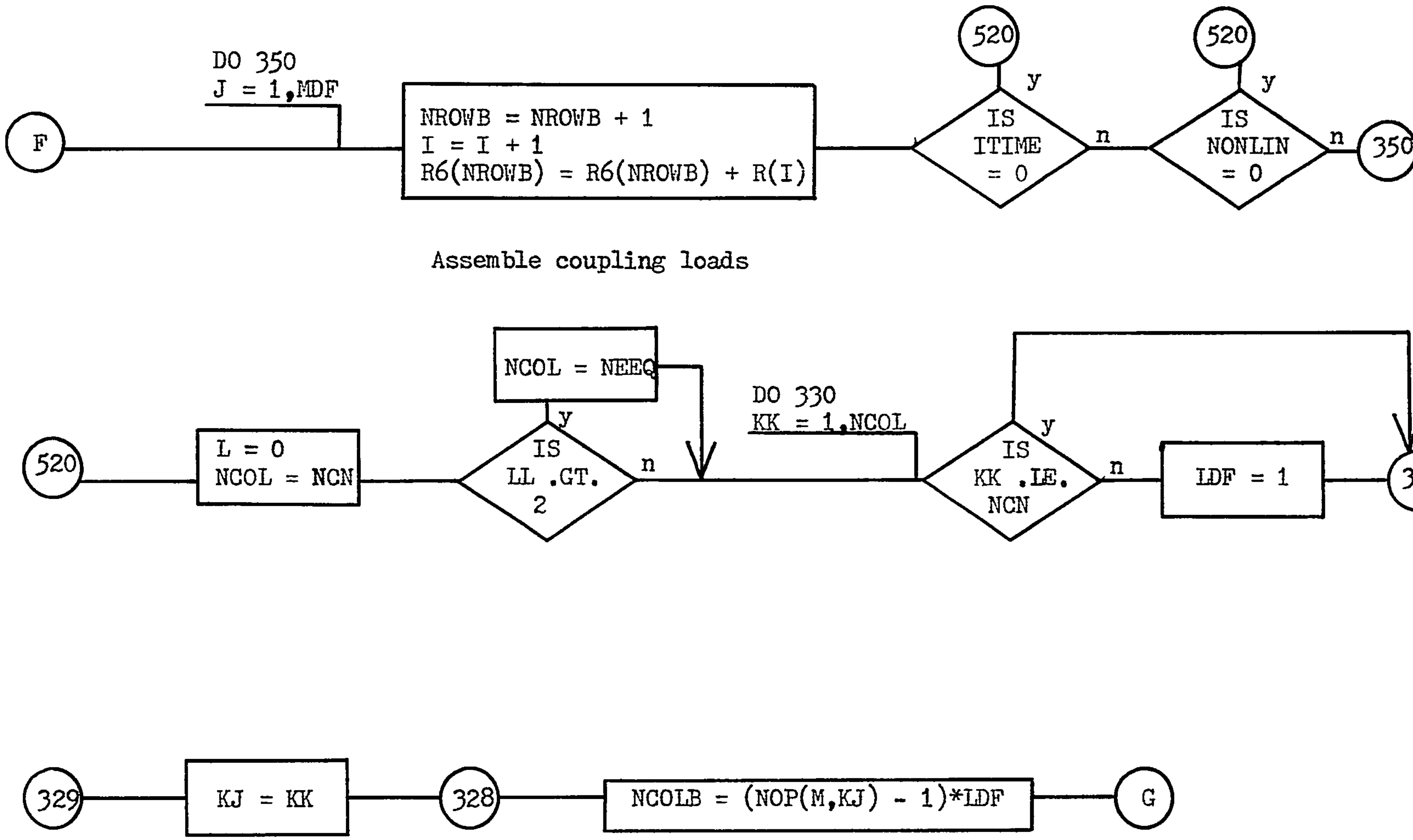


Fig. F-5e Flowchart of Subroutine FORMK

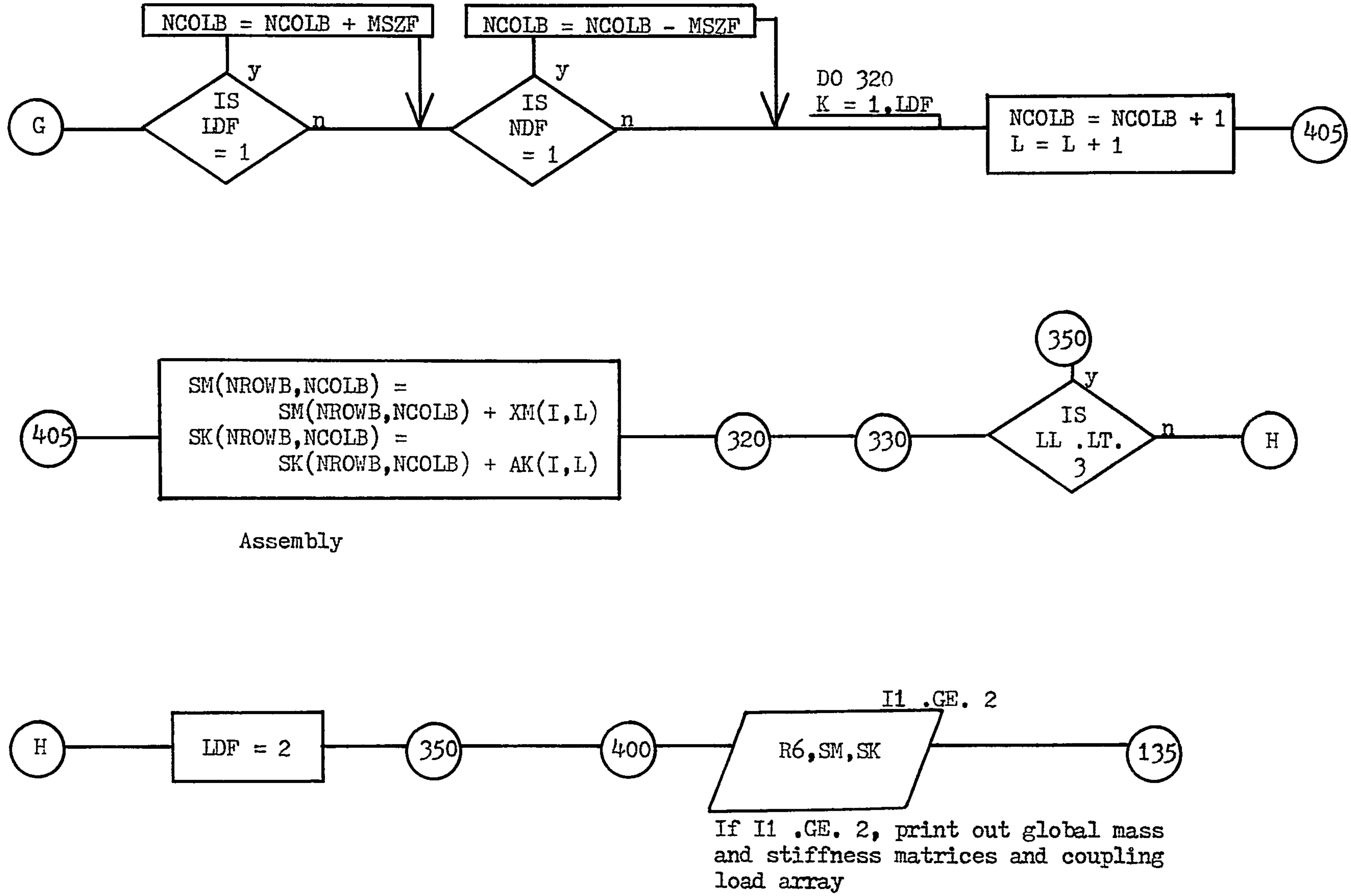


Fig. F-5f Flowchart of Subroutine FORMK

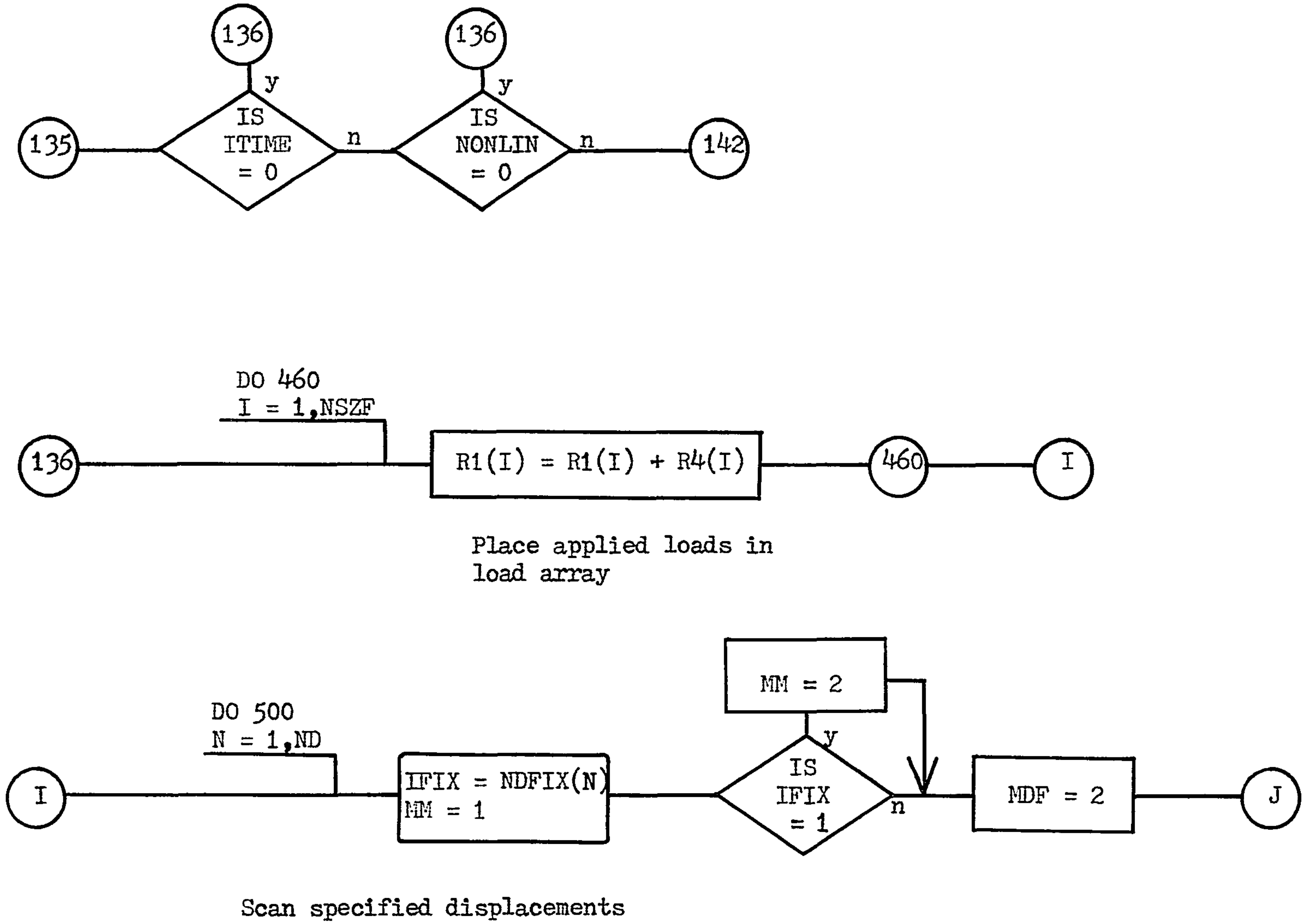
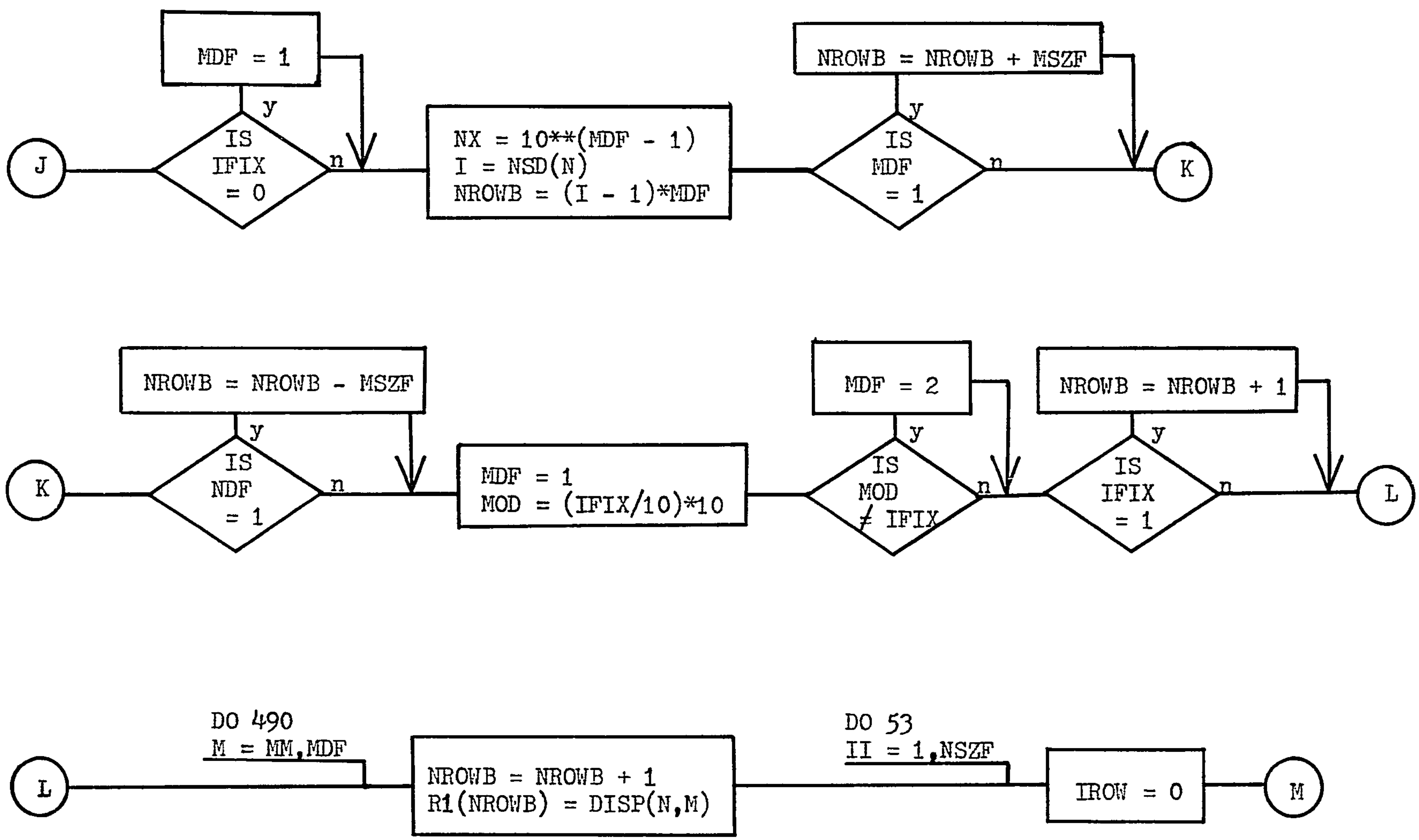


Fig. F-5g Flowchart of Subroutine FORMK



Place specified displacement into load array

Fig. F-5h Flowchart of Subroutine FORMK

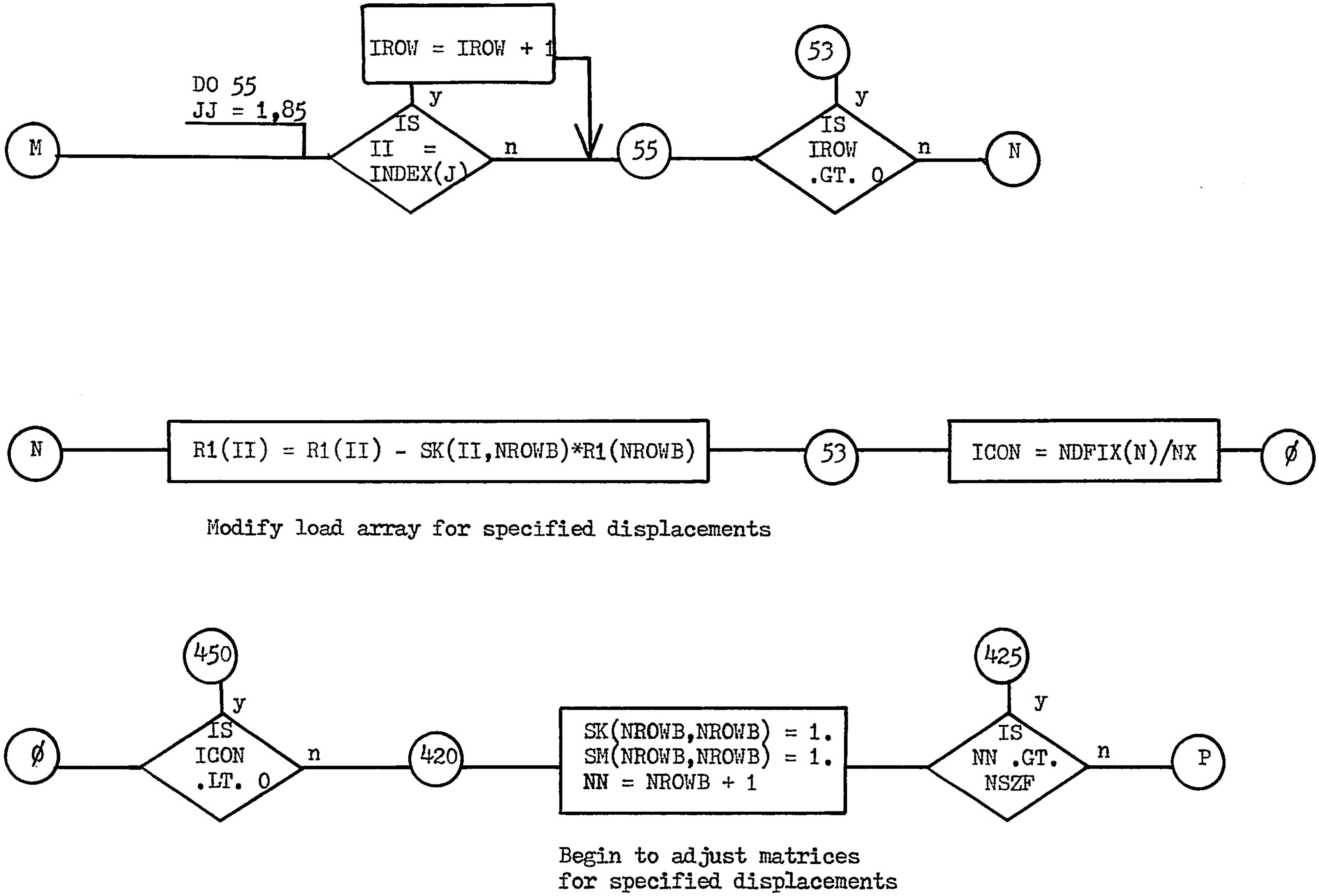


Fig. F-5i Flowchart of Subroutine FORMK

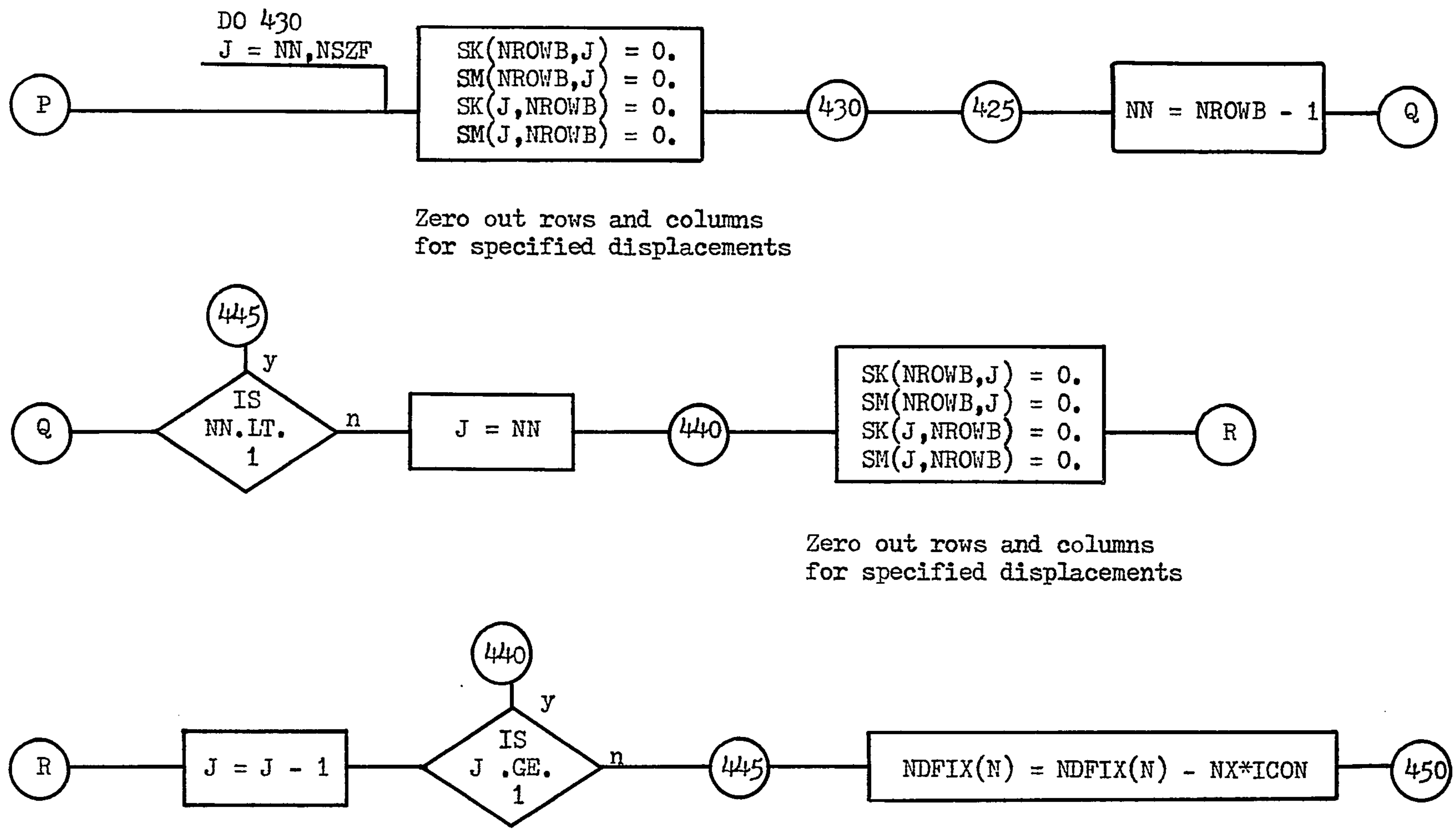


Fig. F-5j Flowchart of Subroutine FORMK

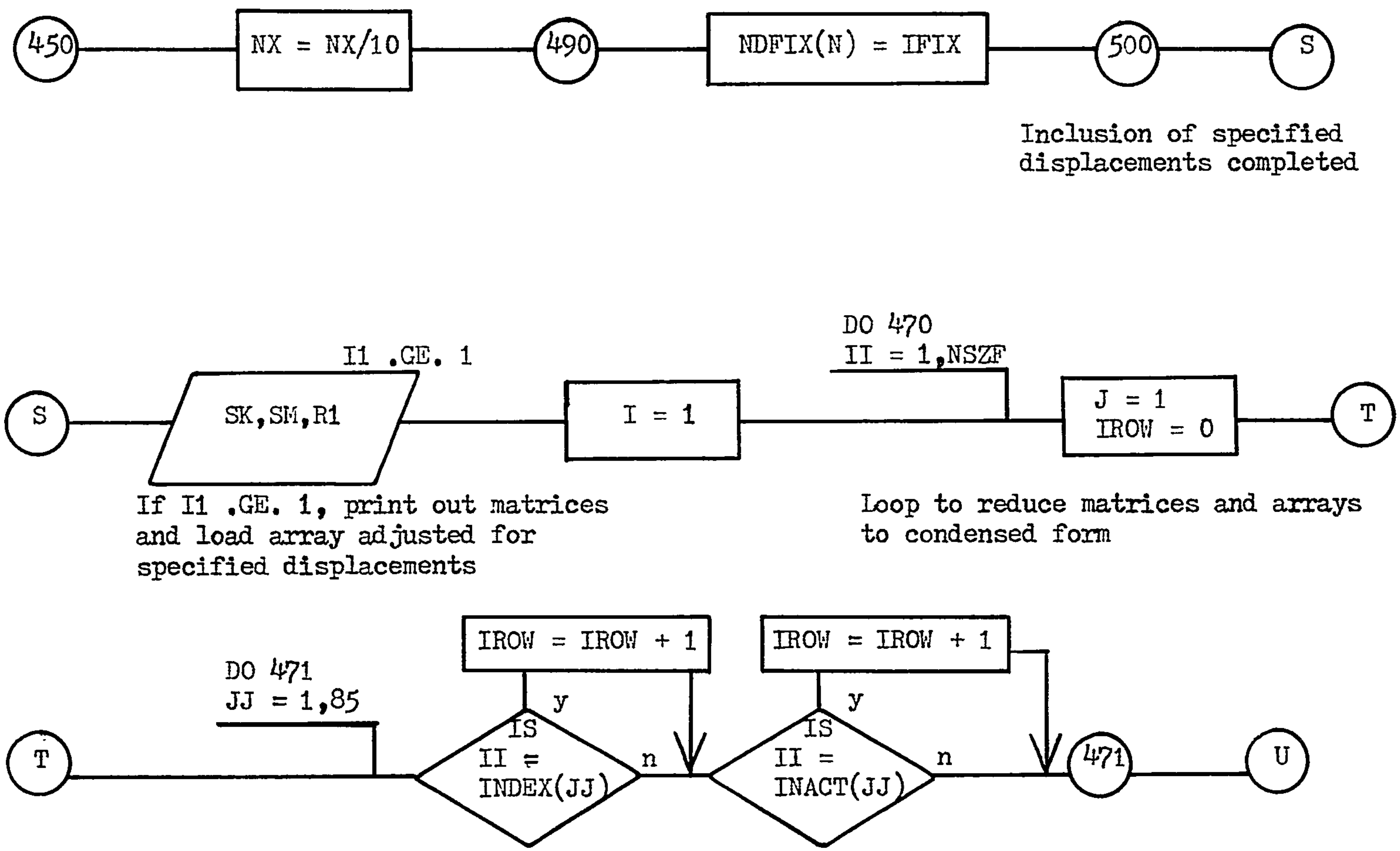


Fig. F-5k Flowchart of Subroutine FORMK

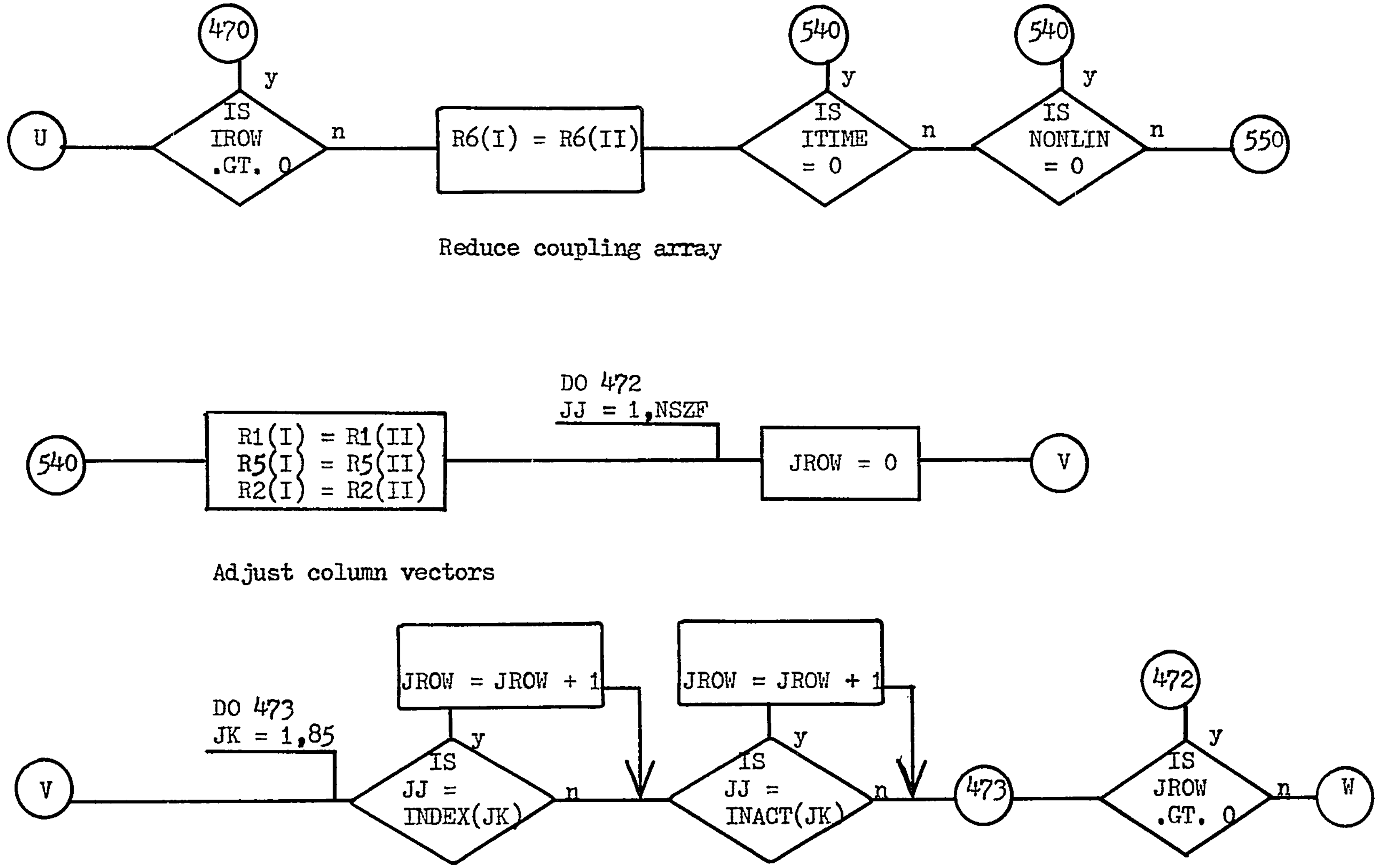


Fig. F-51 Flowchart of Subroutine FORMK

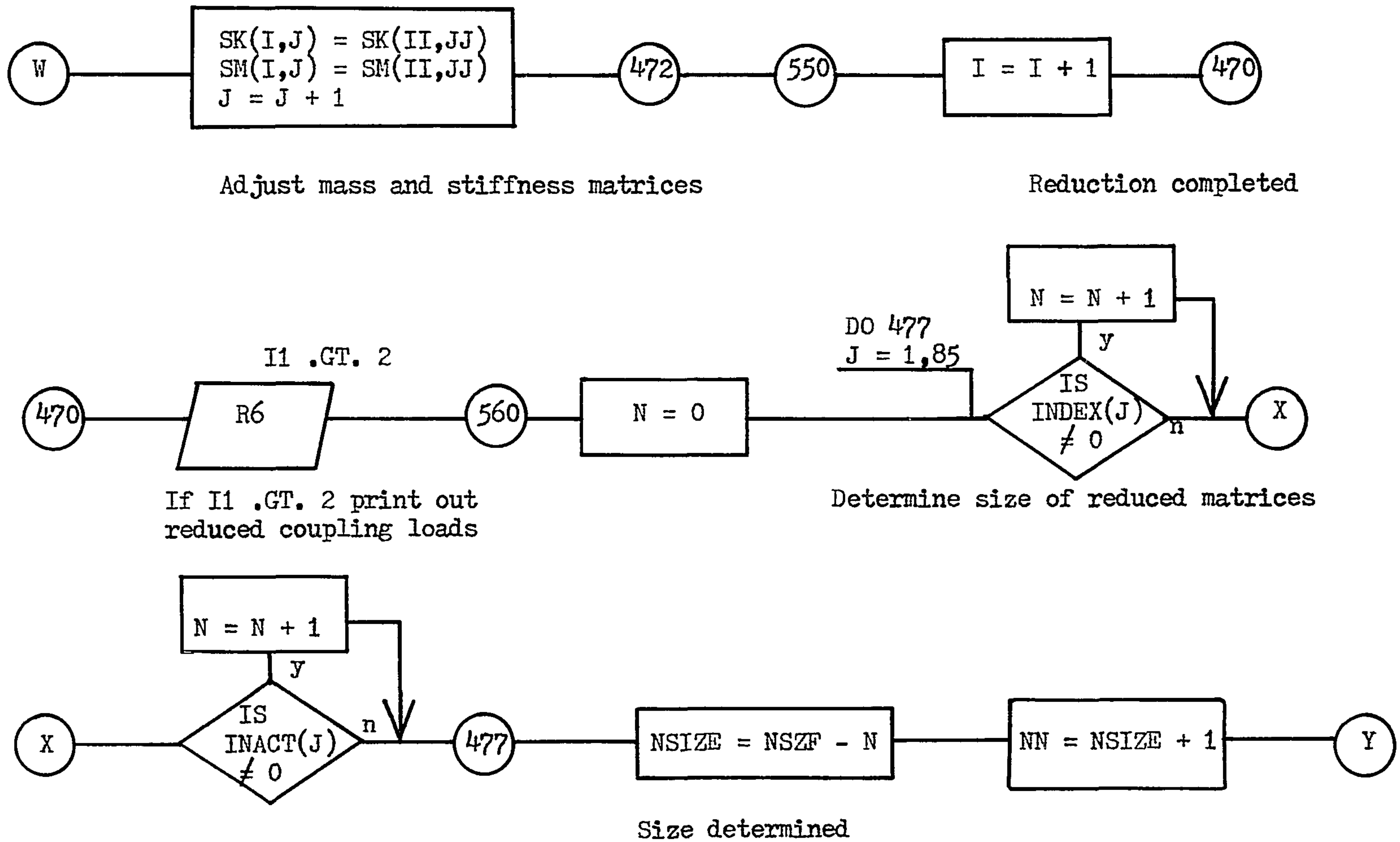


Fig. F-5m Flowchart of Subroutine FORMK

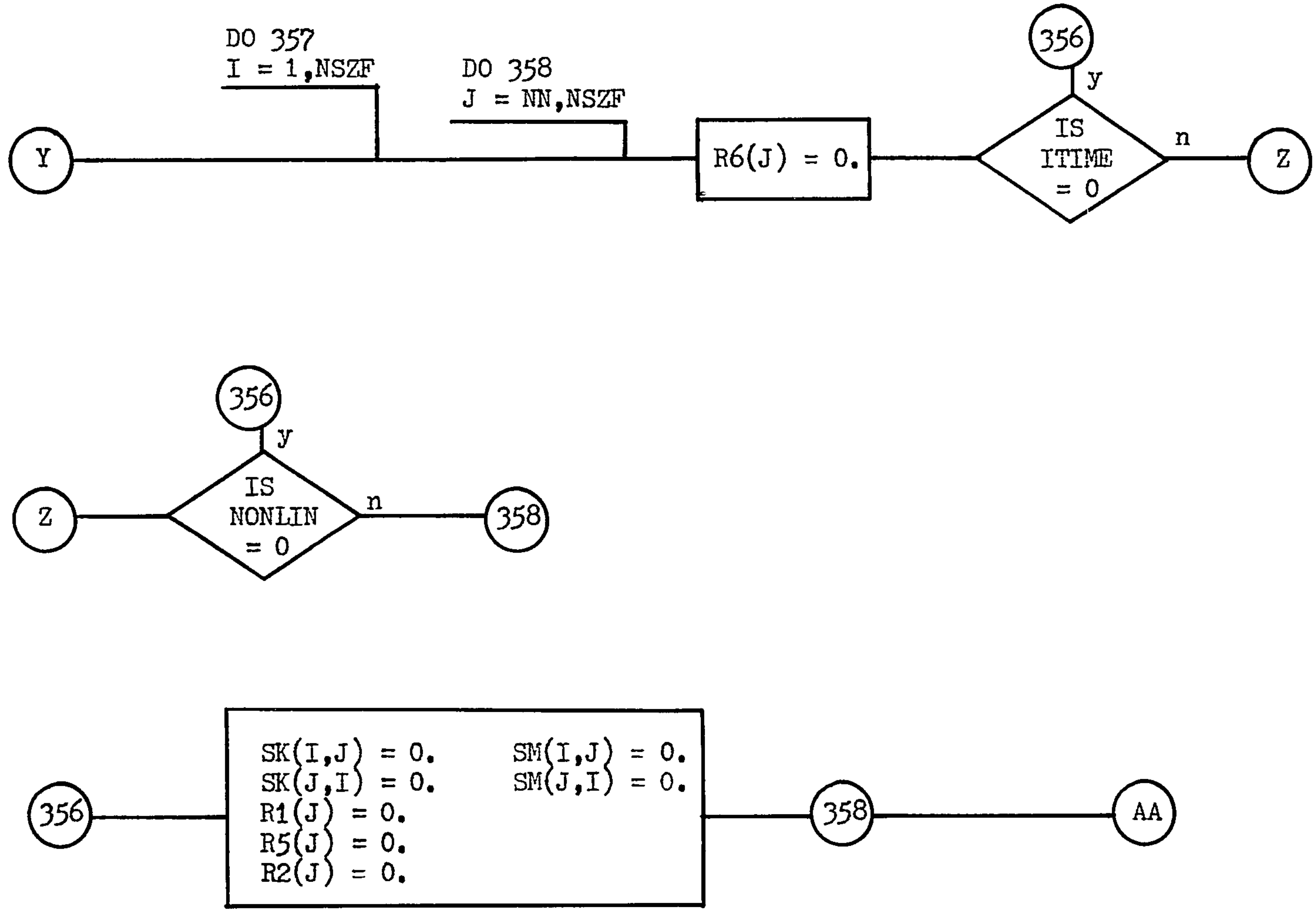


Fig. F-5n Flowchart of Subroutine FORMK

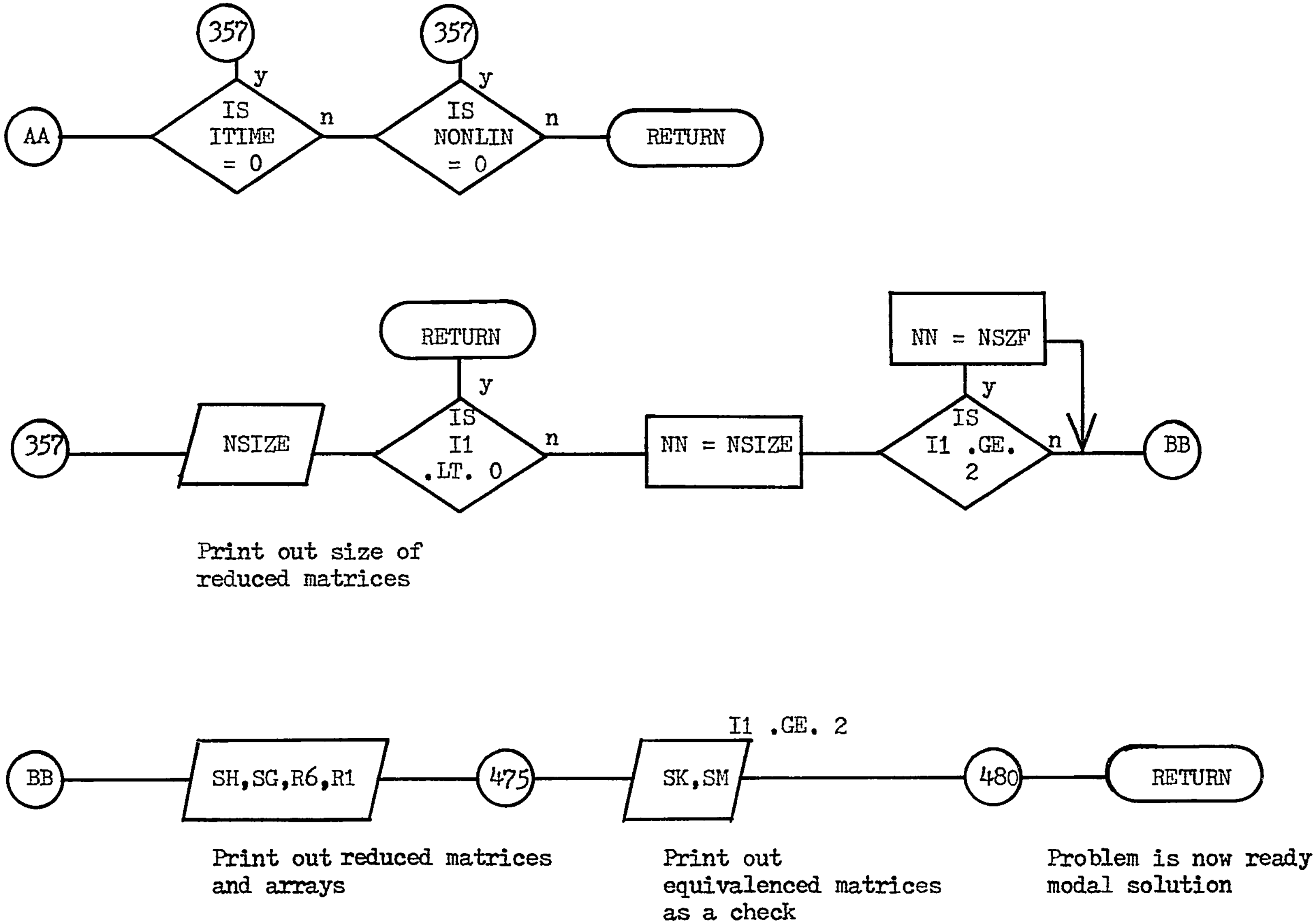


Fig. F-50 Flowchart of Subroutine FORMK

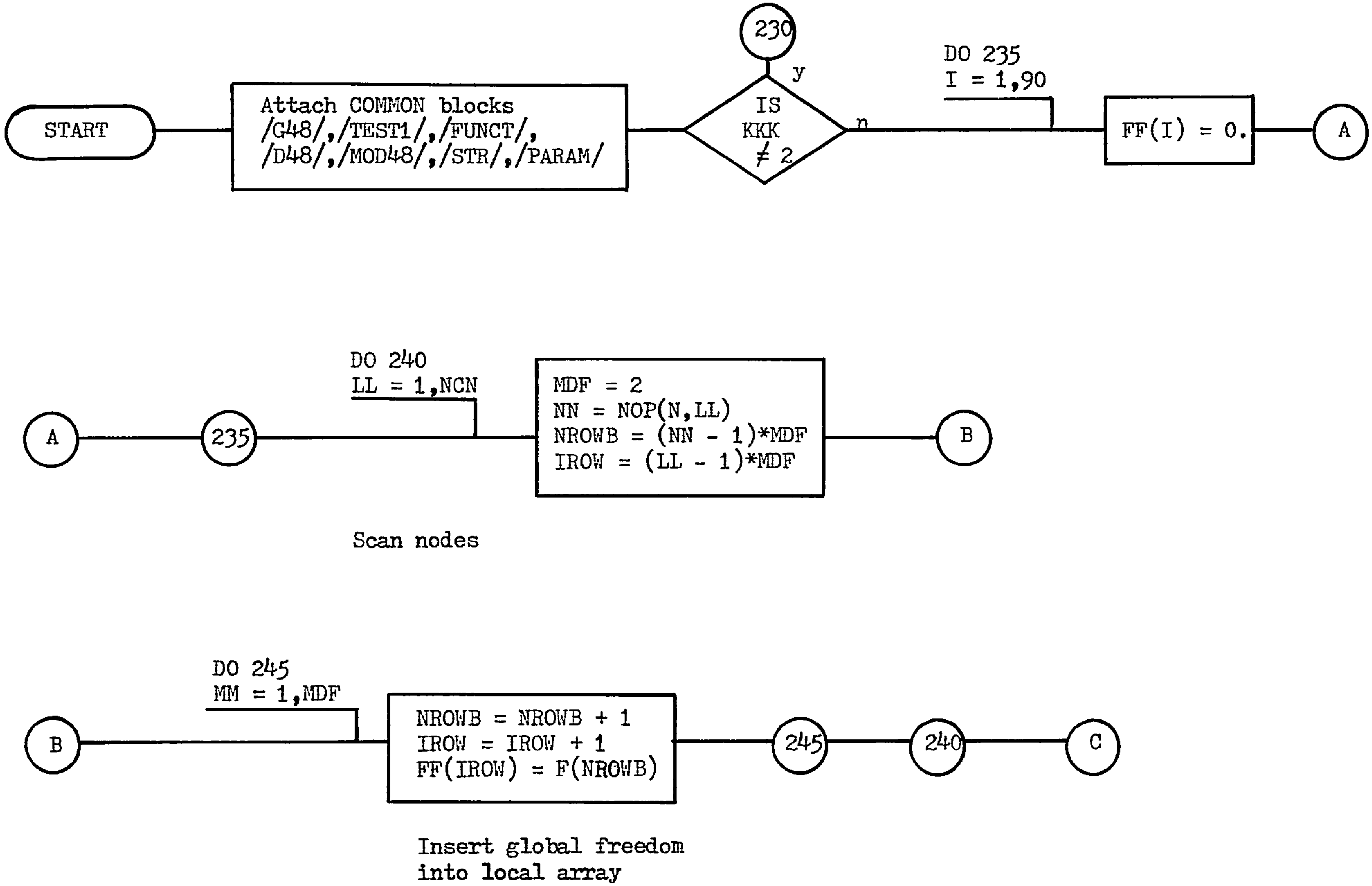


Fig. F-6a Flowchart of Subroutine STIFT1

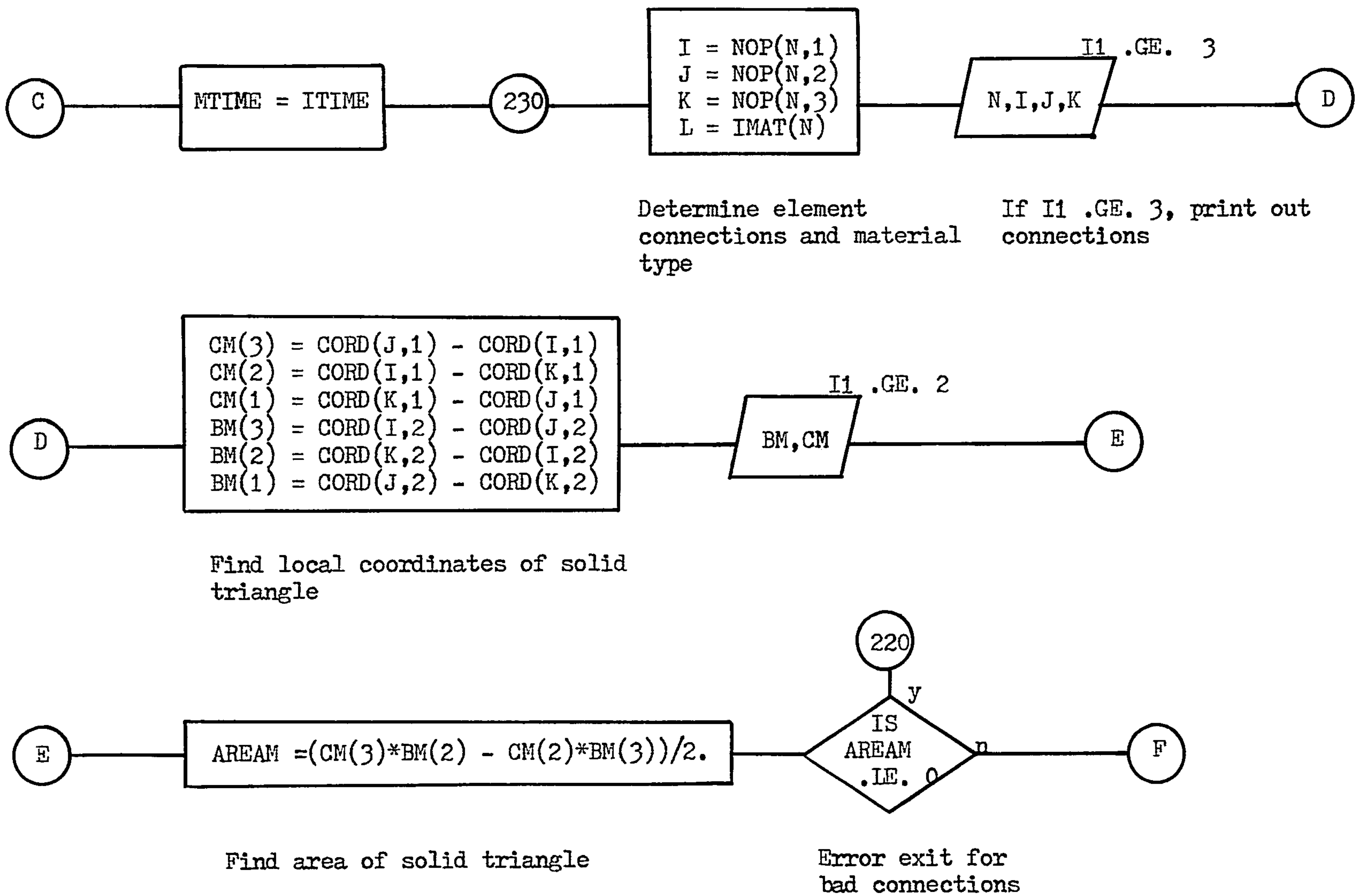


Fig. F-6b Flowchart of Subroutine STIFT1

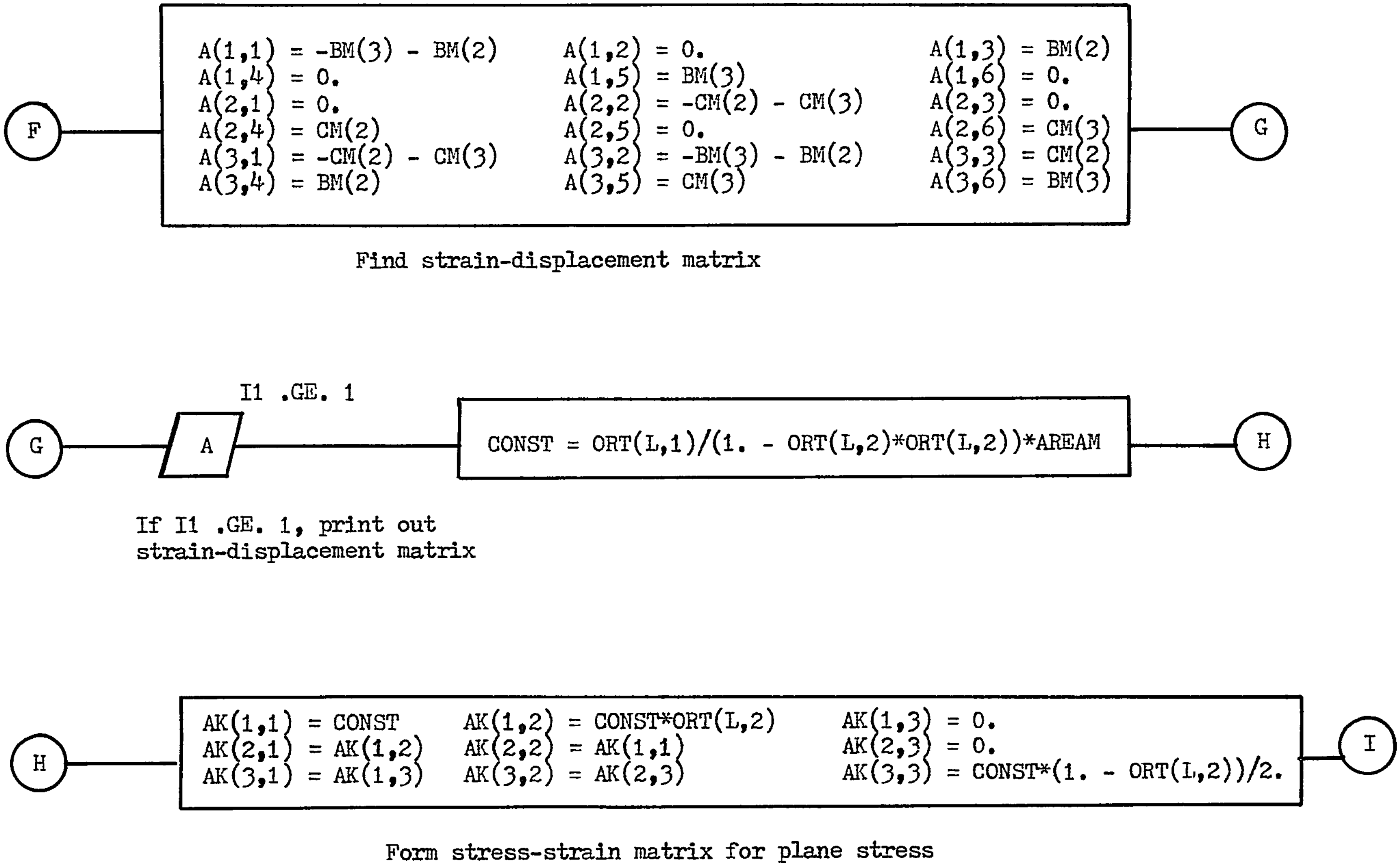


Fig. F-6c Flowchart of Subroutine STIFT1

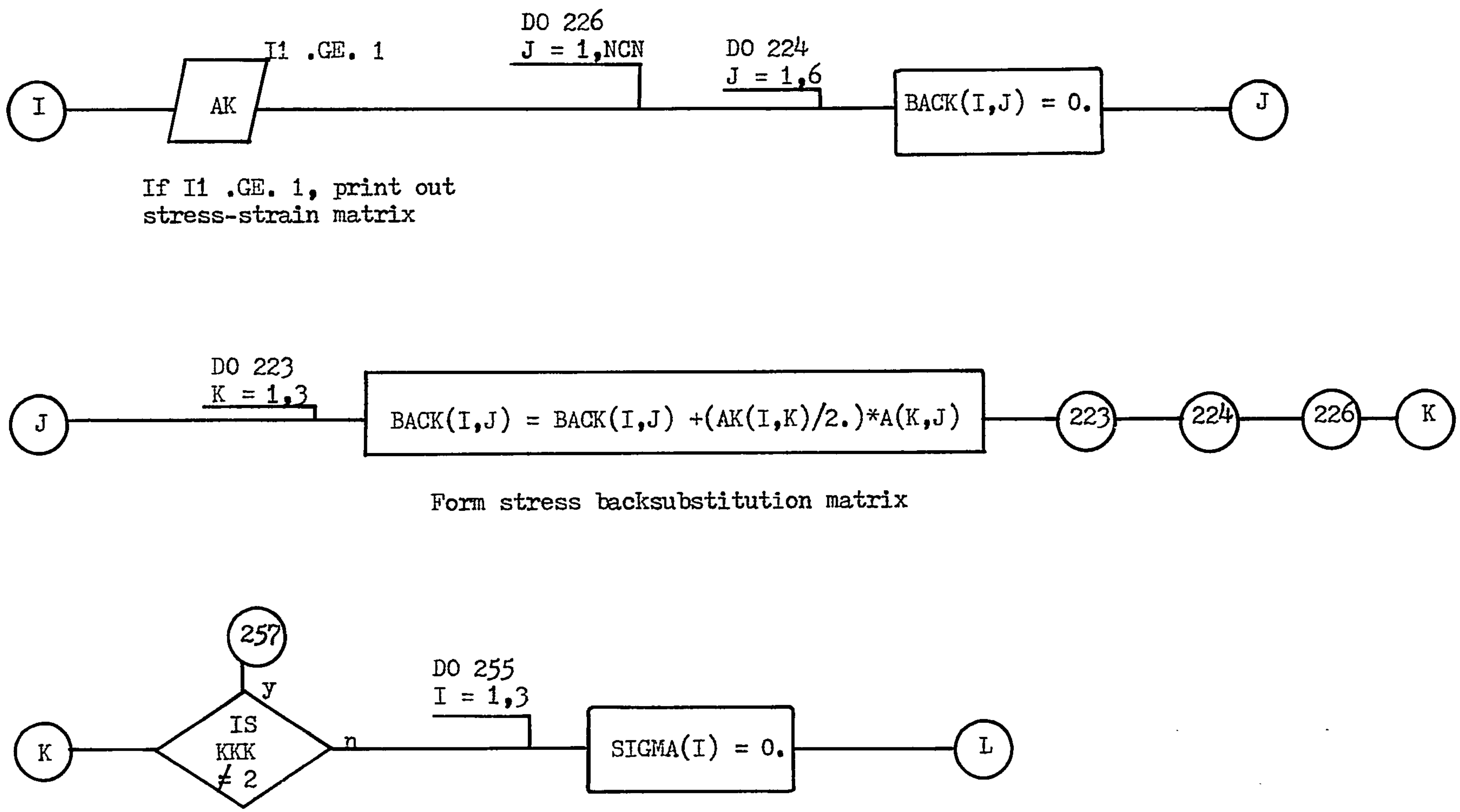


Fig. F-6d Flowchart of Subroutine STIFT1

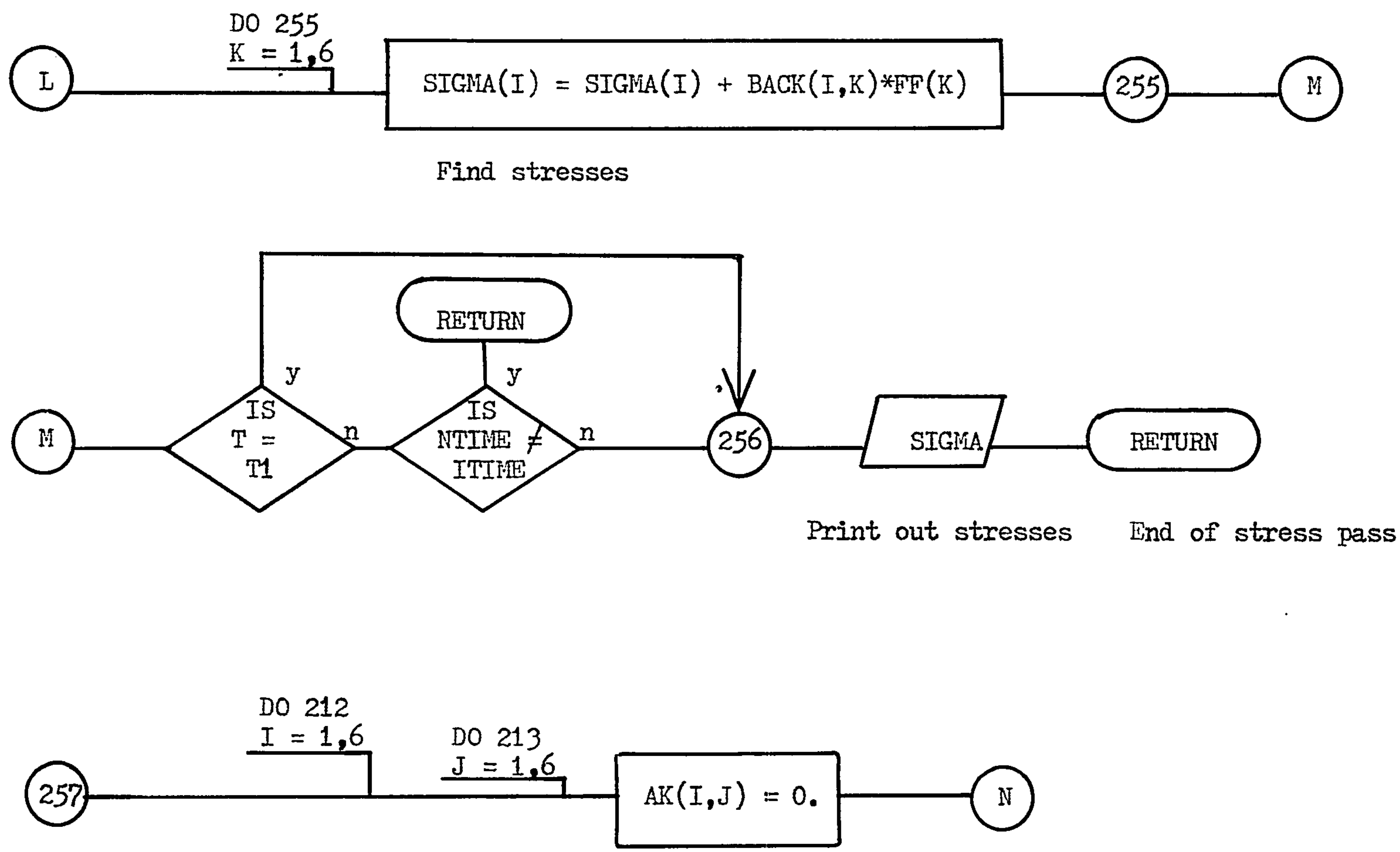


Fig. F-6e Flowchart of Subroutine STIFT1

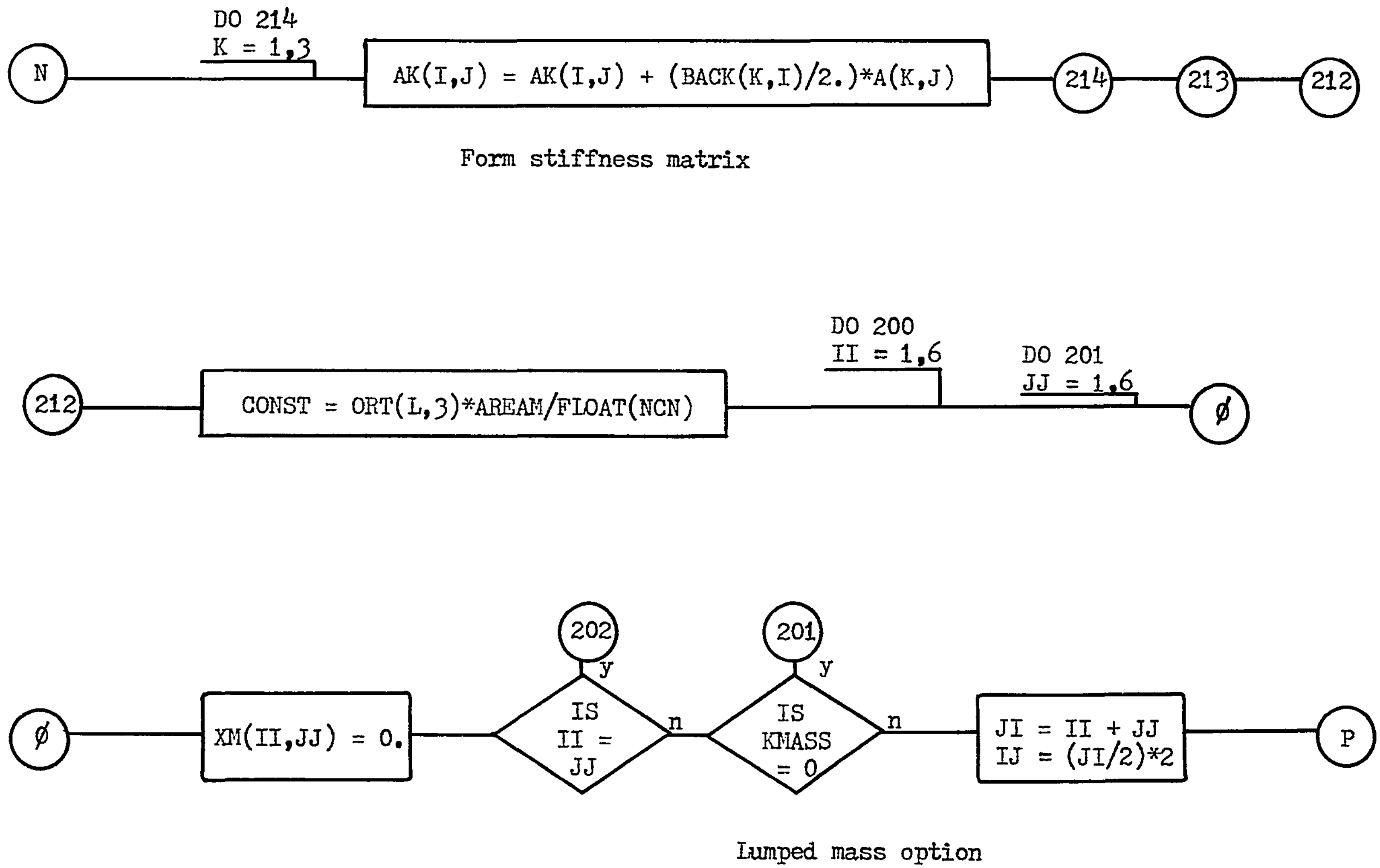


Fig. F-6f Flowchart of Subroutine STIFT1

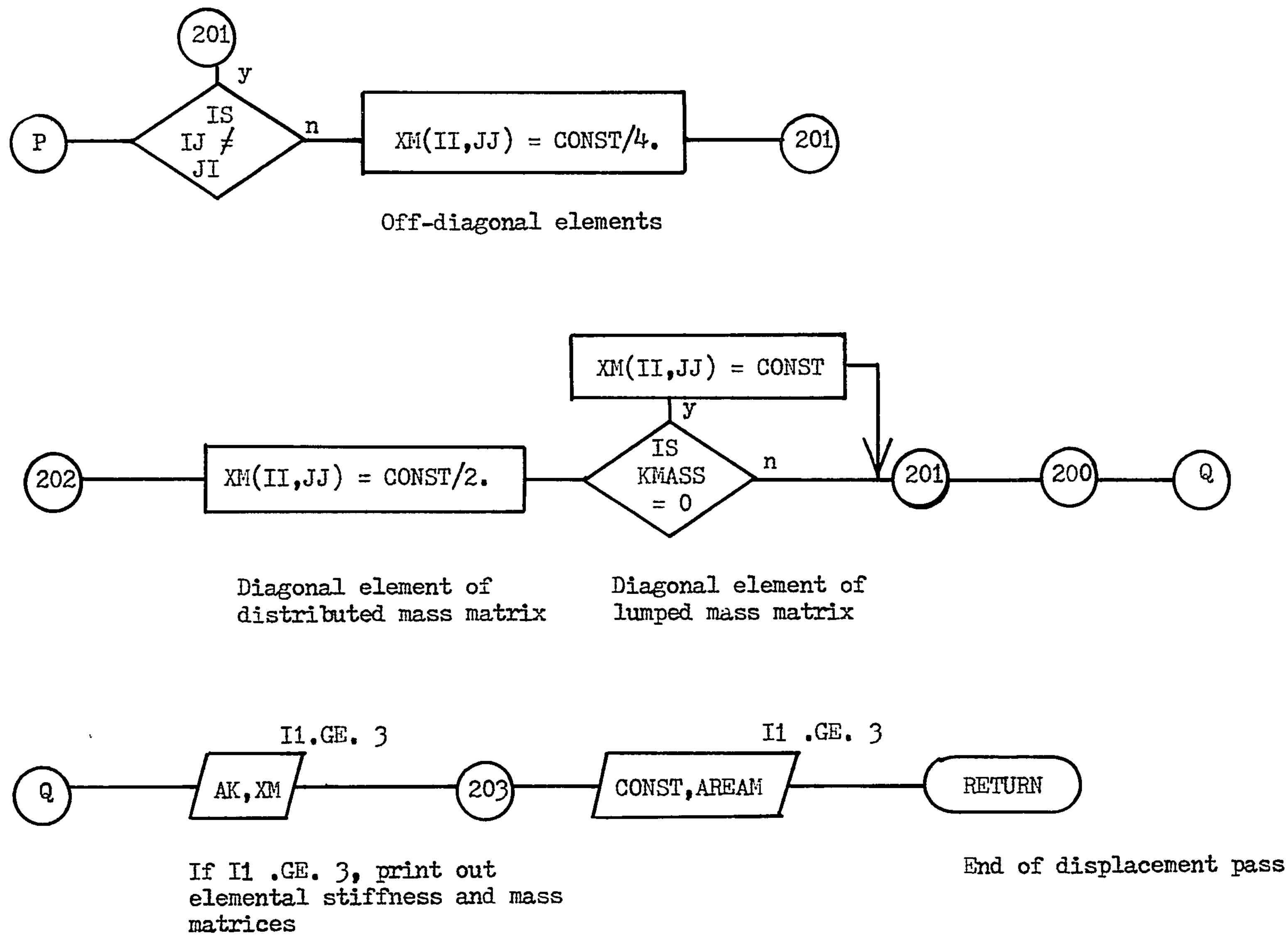


Fig. F-6g Flowchart of Subroutine STIFT1

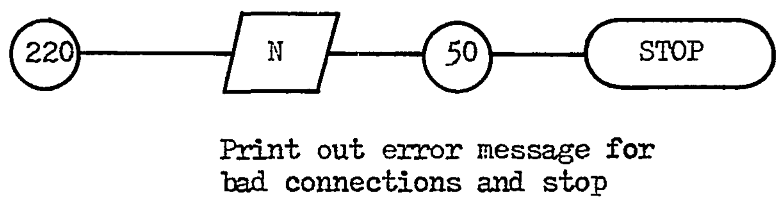


Fig. F-6h Flowchart of Subroutine STIFT1 pass, STIFT2(N) is called by STRESS. The elemental velocities are calculated from the elemental nodal pressures and the elemental velocities from the previous iteration by a temporal finite difference scheme. The calling statement for this subroutine is

CALL STIFT2(N)

in which the argument N is the fluid element number. The flowchart for STIFT2(N) is given in Fig. F-7a through F-7f .

1.8 STIFT3(N) Subroutine

Subroutine STIFT3(N) is called in the displacement pass and in the stress pass. In this displacement pass, STIFT3(N) is called by FORMK to calculate the equivalent mass and stiffness matrices for the superelement plane linear quadrilateral solid-fluid finite element N. In the stress pass, STIFT3(N) is called by STRESS. The stress back substitution matrix for the solid part of the solid-fluid element is calculated and is used in conjunction with the elemental nodal displacements to determine the elemental stresses. The elemental velocities for the fluid part of the solid-fluid element are calculated from the elemental nodal pressures and the elemental velocities from the previous iteration by a temporal finite difference scheme. STIFT3(N) is also entered from FORMK in every iteration after the first iteration to calculate the solid-fluid interaction loads. The calling statement for this subroutine is

CALL STIFT3(N)

in which the argument N is the solid-fluid superelement number. The flowchart for STIFT3(N) is given in Fig. F-8a through F-8q .

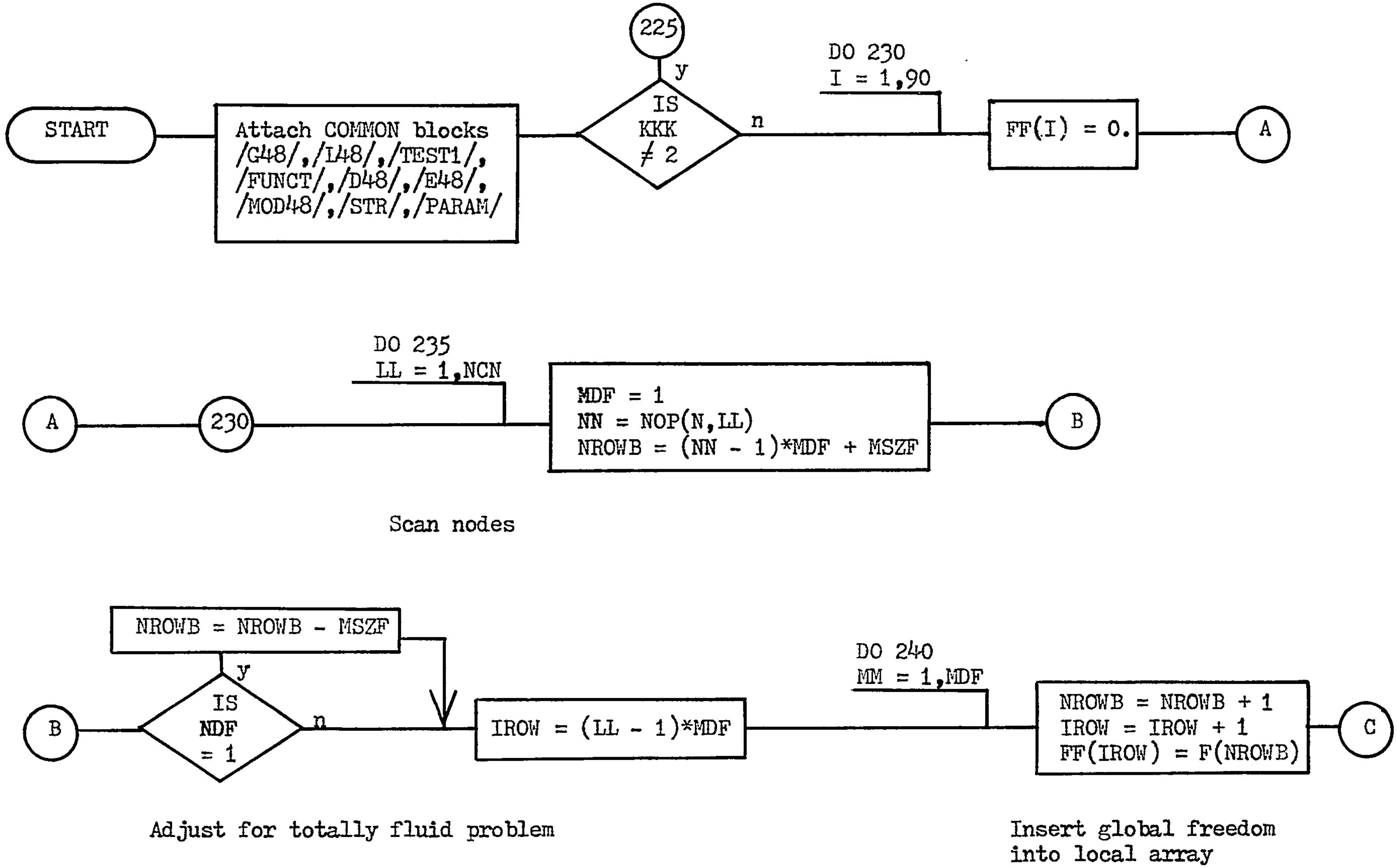


Fig. F-7a Flowchart of Subroutine STIFT2

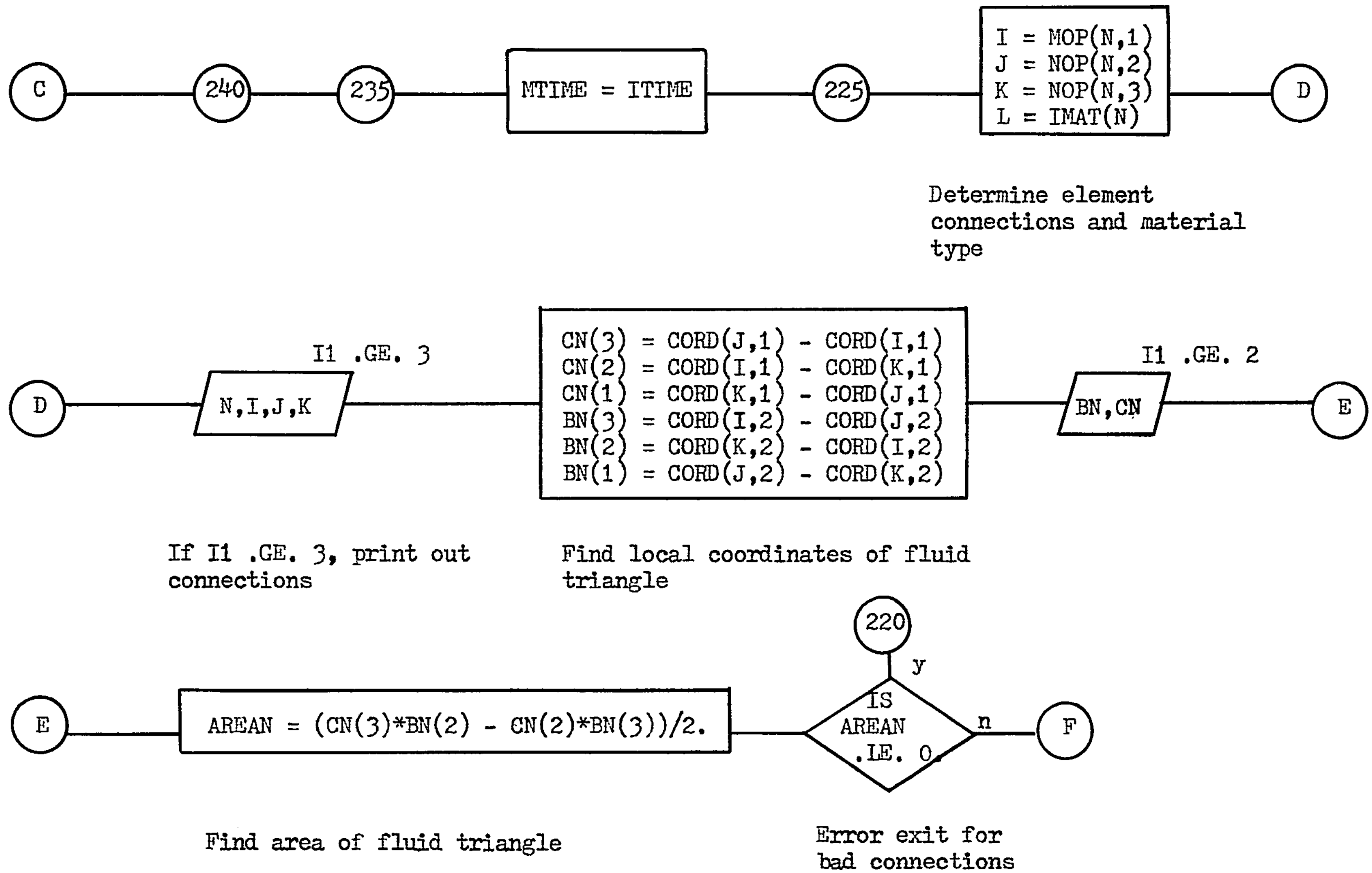


Fig. F-7b Flowchart of Subroutine STIFT2

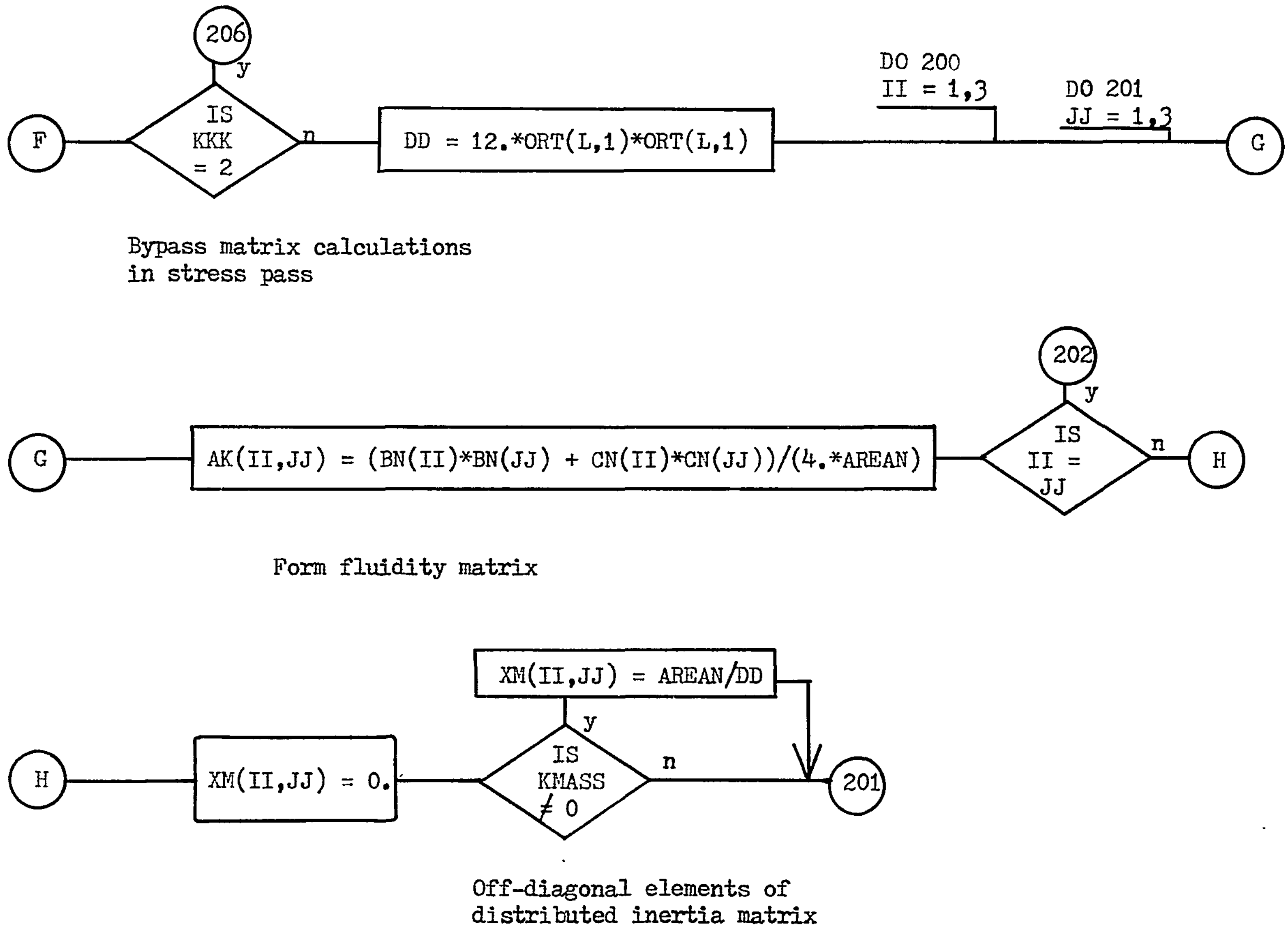


Fig. F-7c Flowchart of Subroutine STIFT2

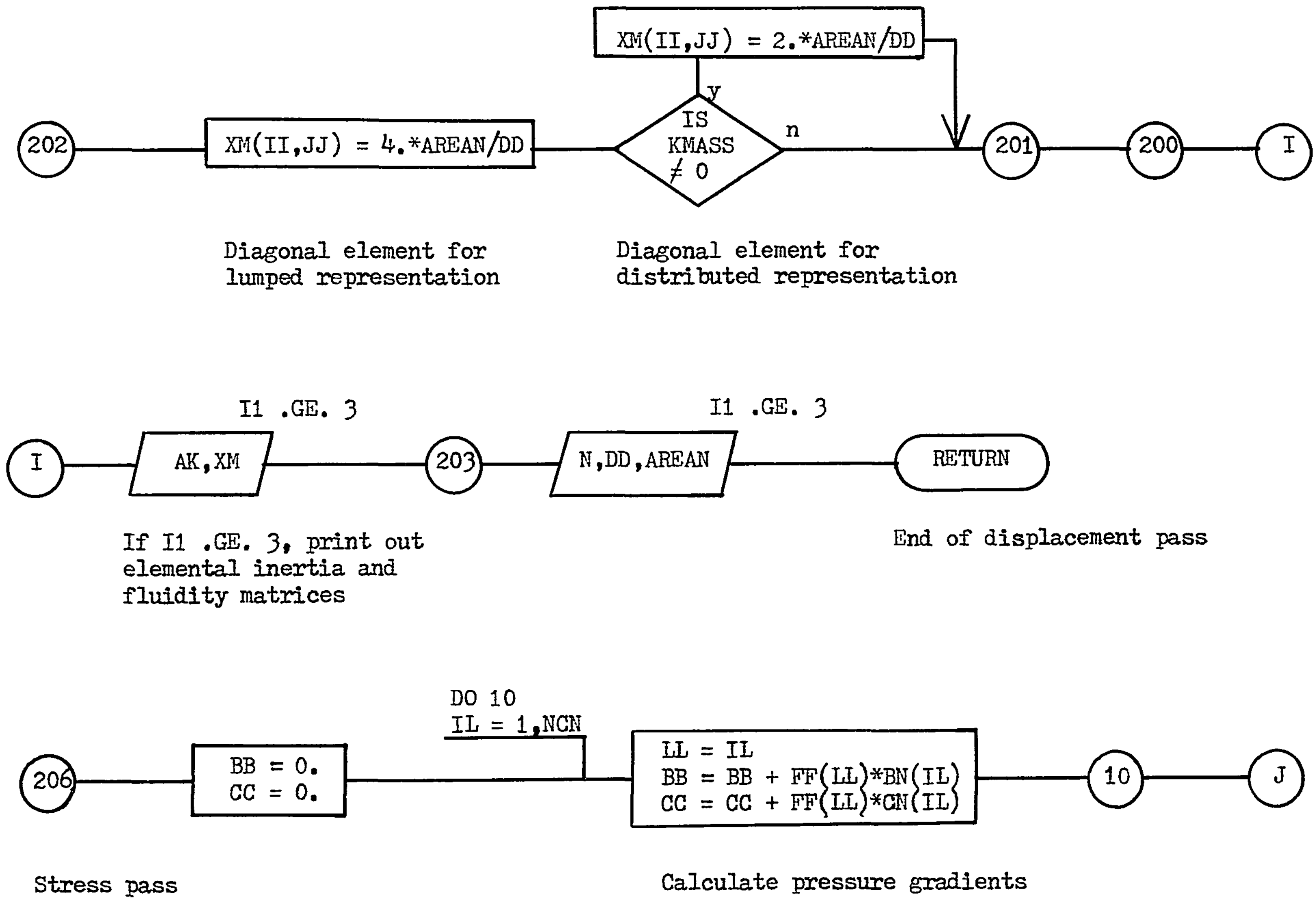


Fig. F-7d Flowchart of Subroutine STIFT2

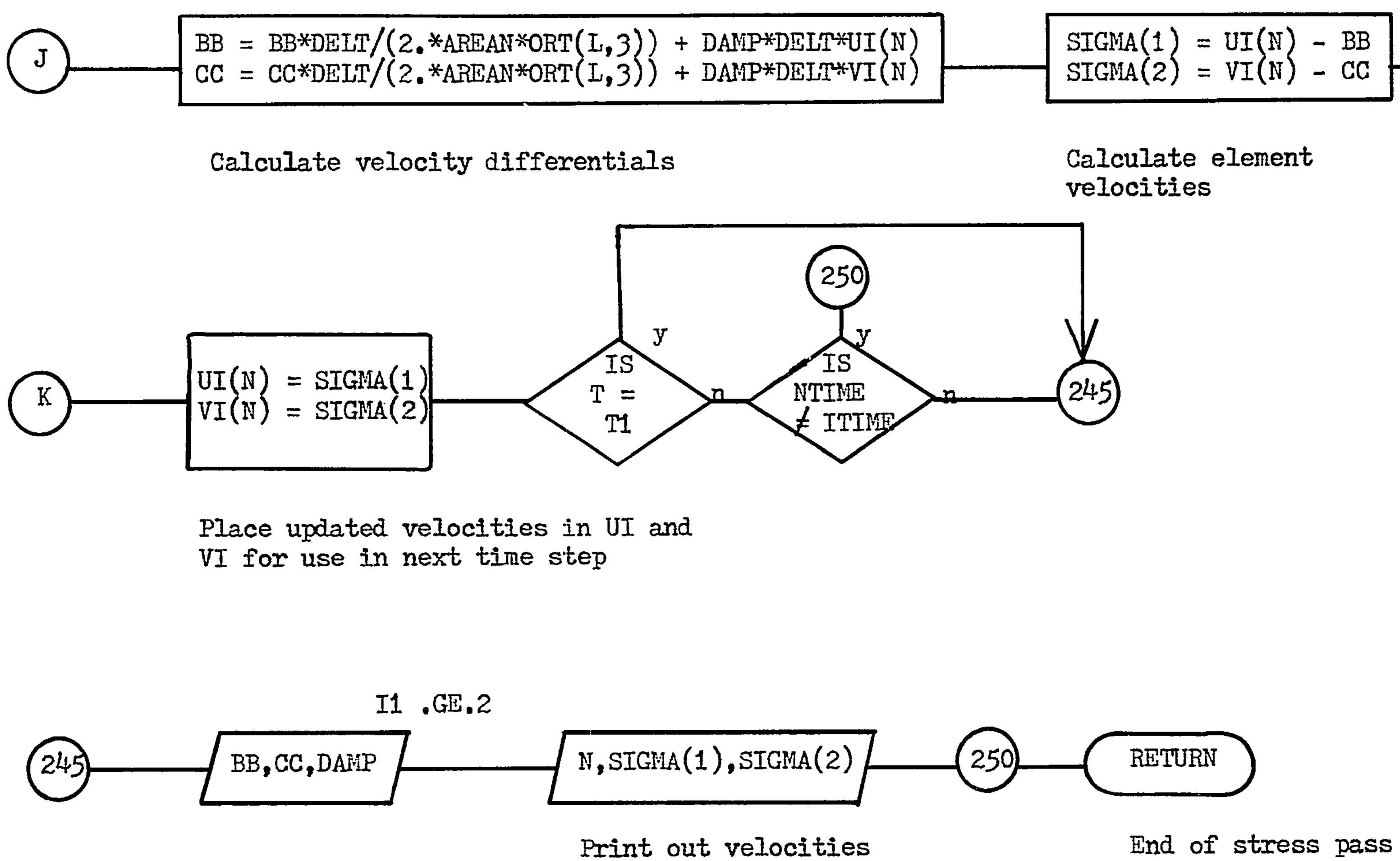


Fig. F-7e Flowchart of Subroutine STIFT2

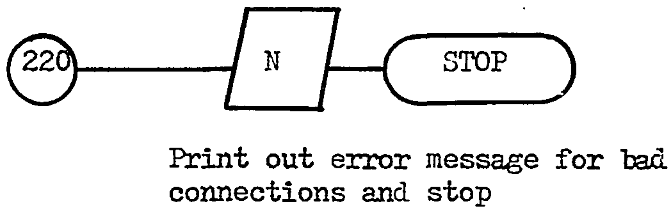


Fig. F-7f Flowchart of Subroutine STIFT2

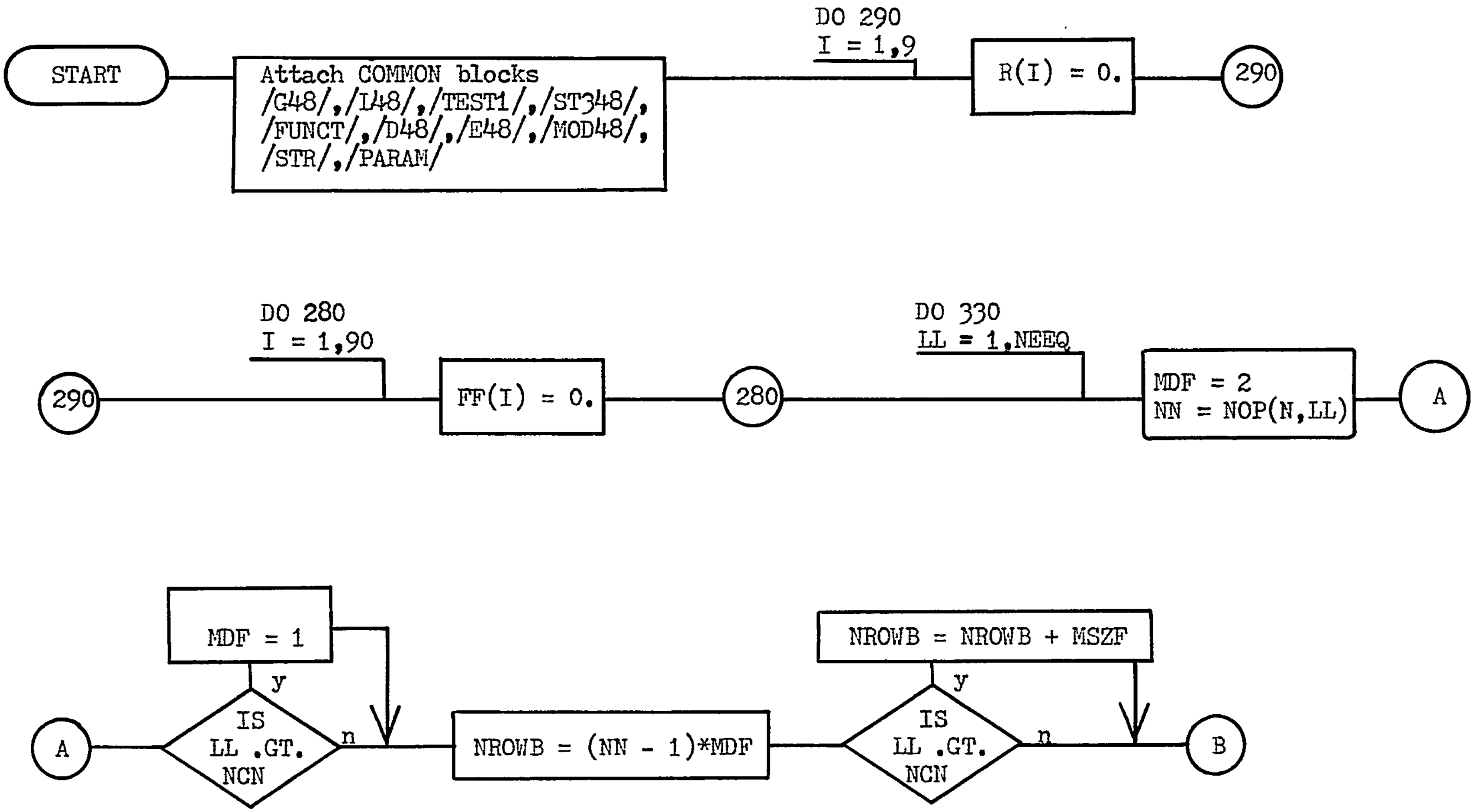


Fig. F-8a Flowchart of Subroutine STIFT3

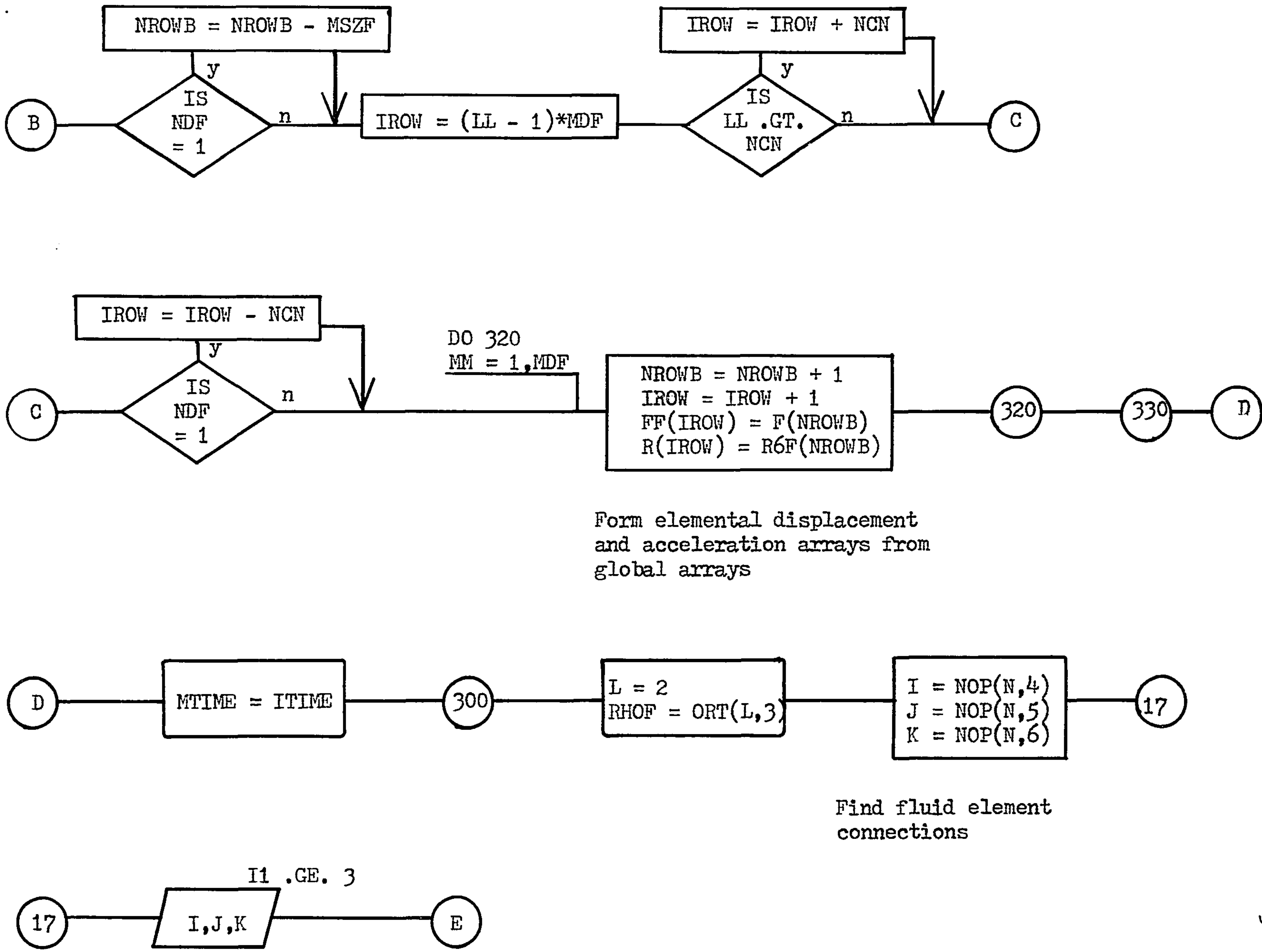


Fig. F-8b Flowchart of Subroutine STIFT3

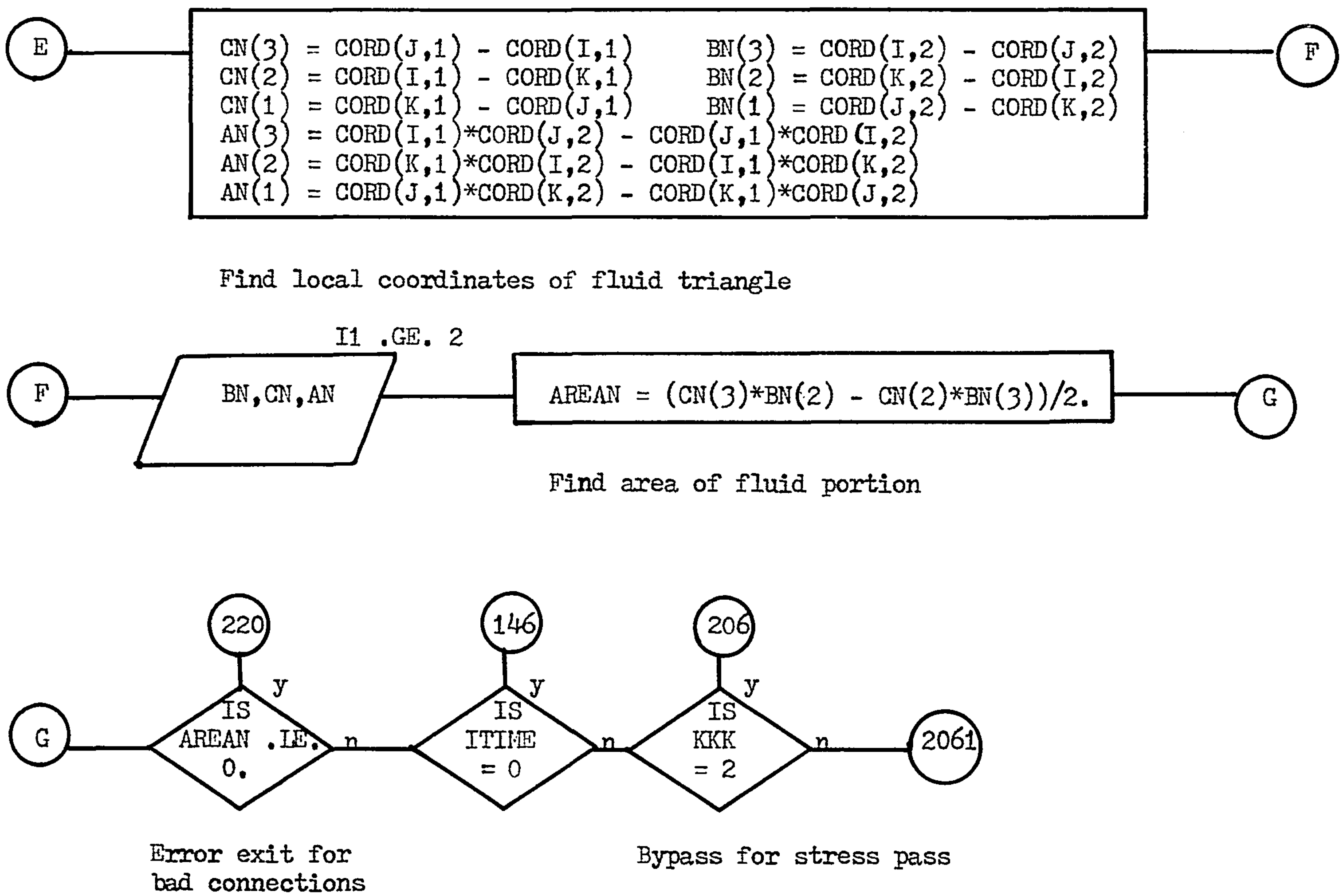


Fig. F-8c Flowchart of Subroutine STIFT3

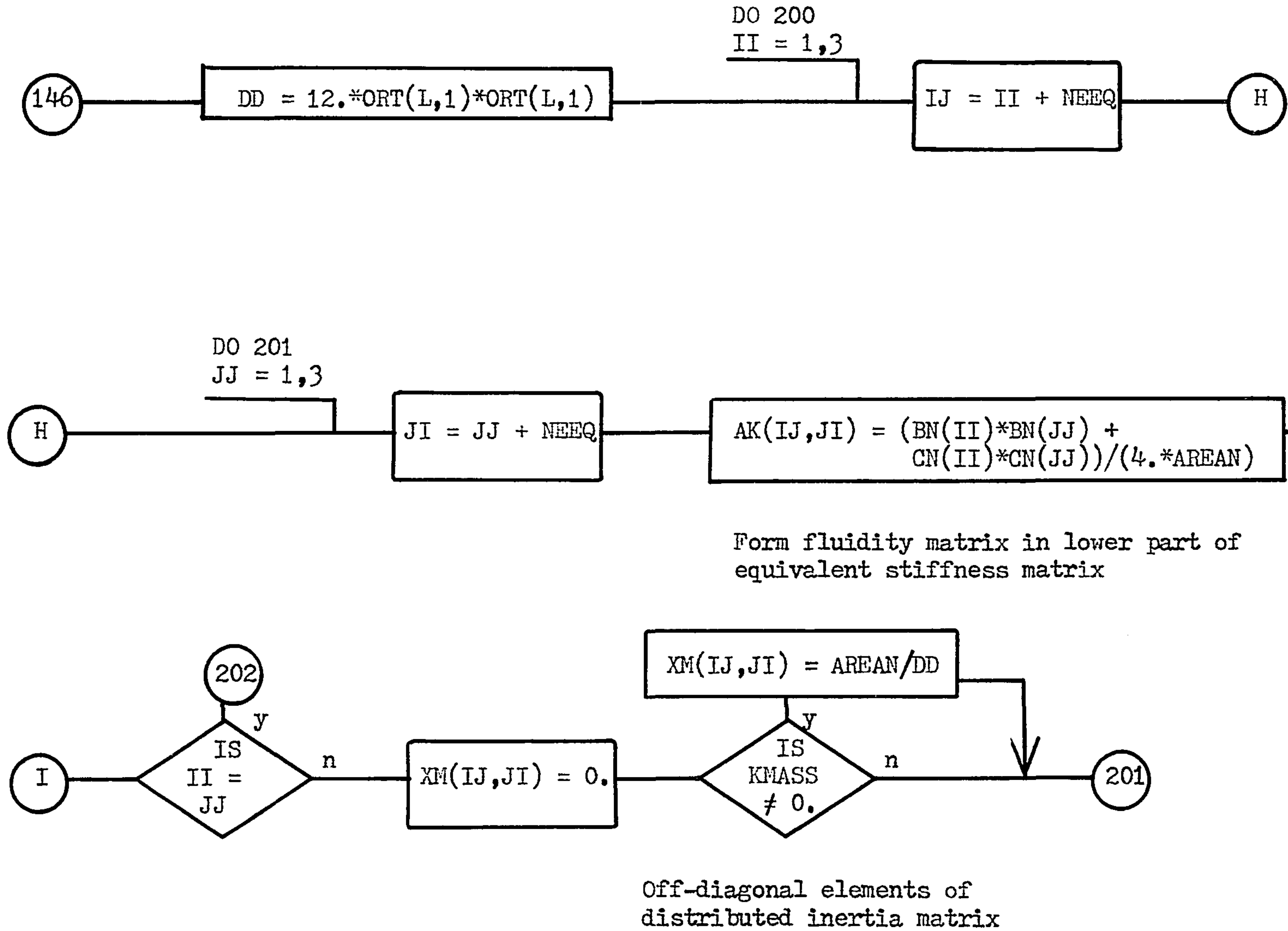


Fig. F-8d Flowchart of Subroutine STIFT3

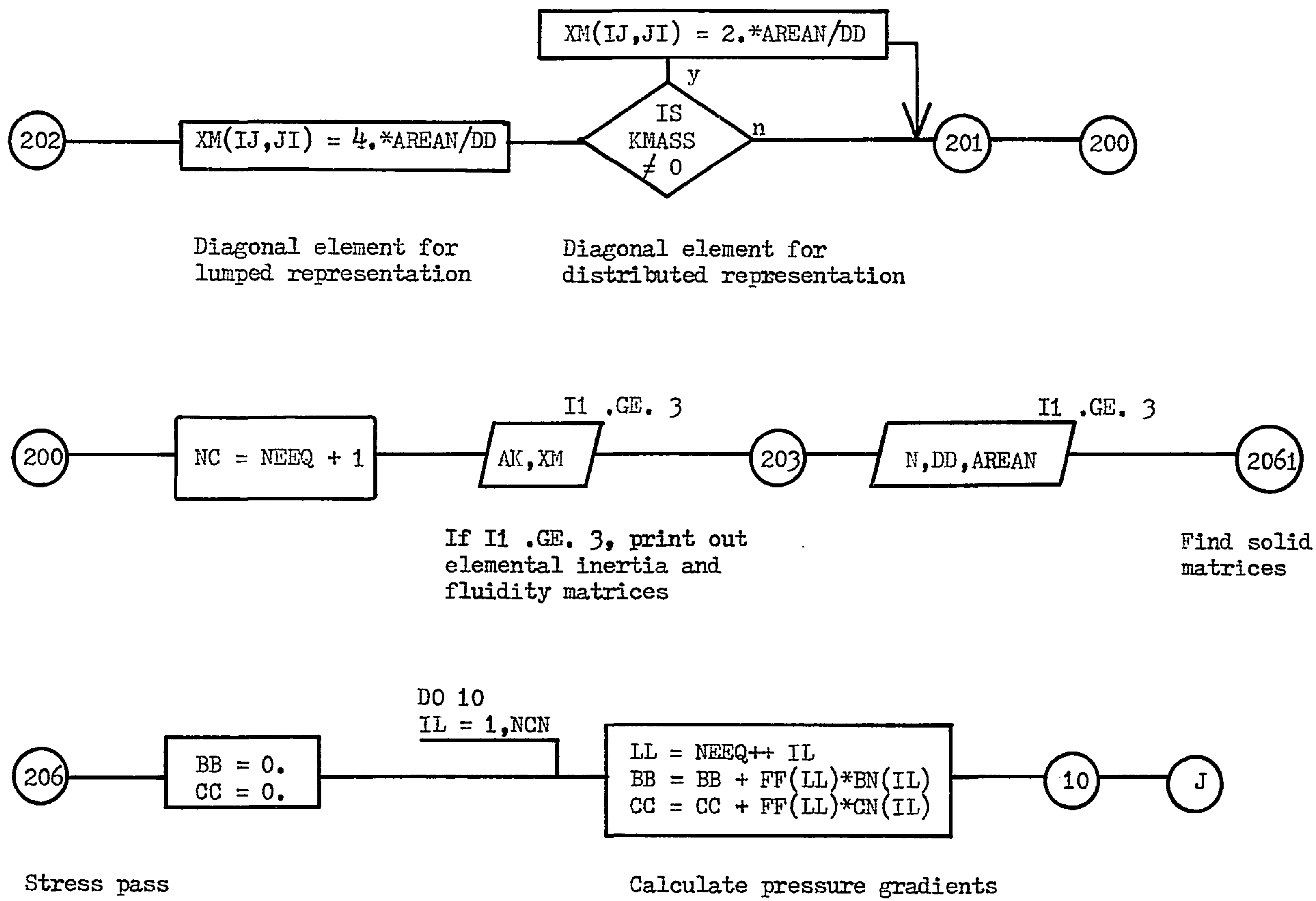


Fig. F-8e Flowchart of Subroutine STIFT3

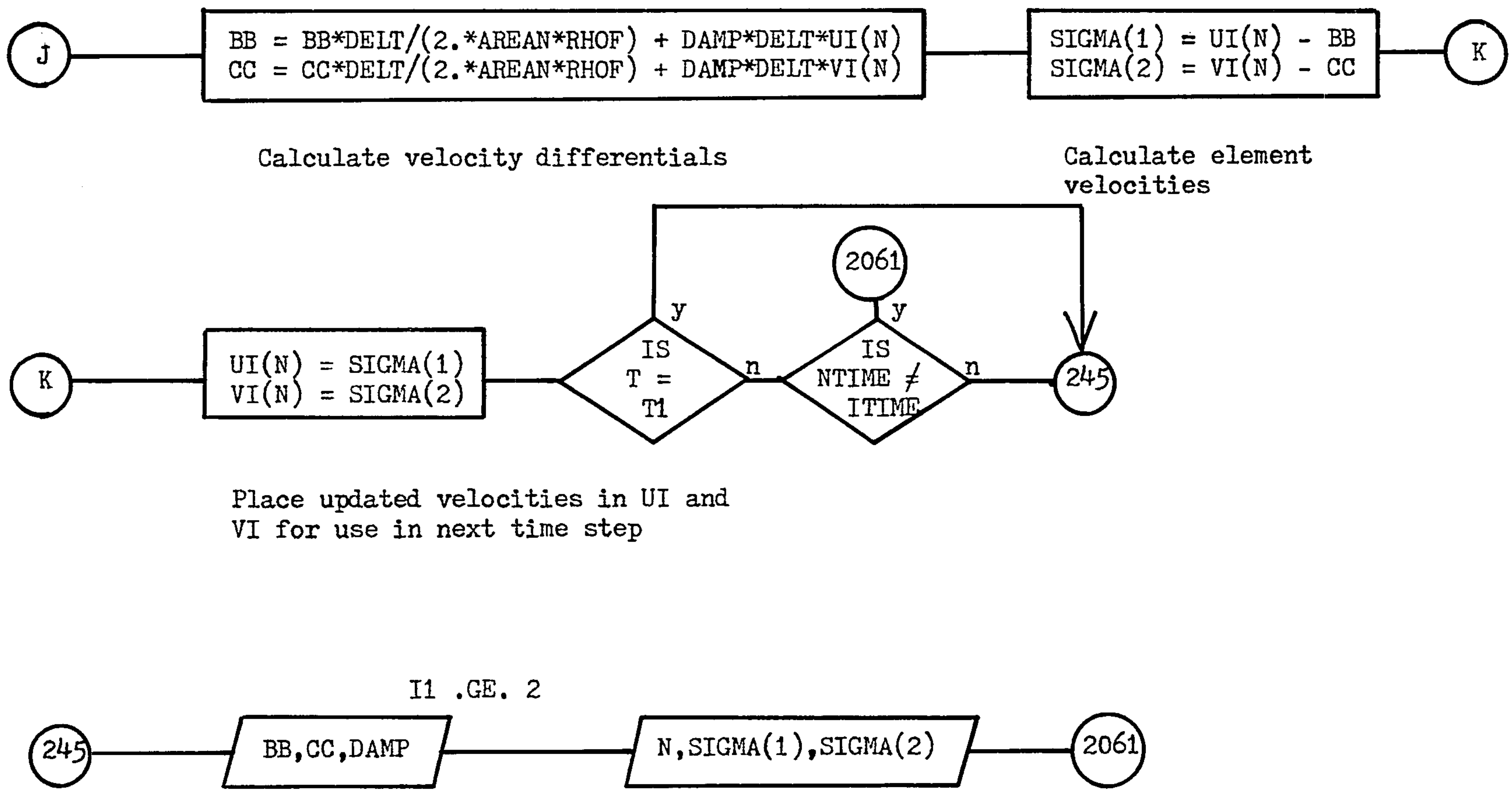


Fig. F-8f Flowchart of Subroutine STIFT3

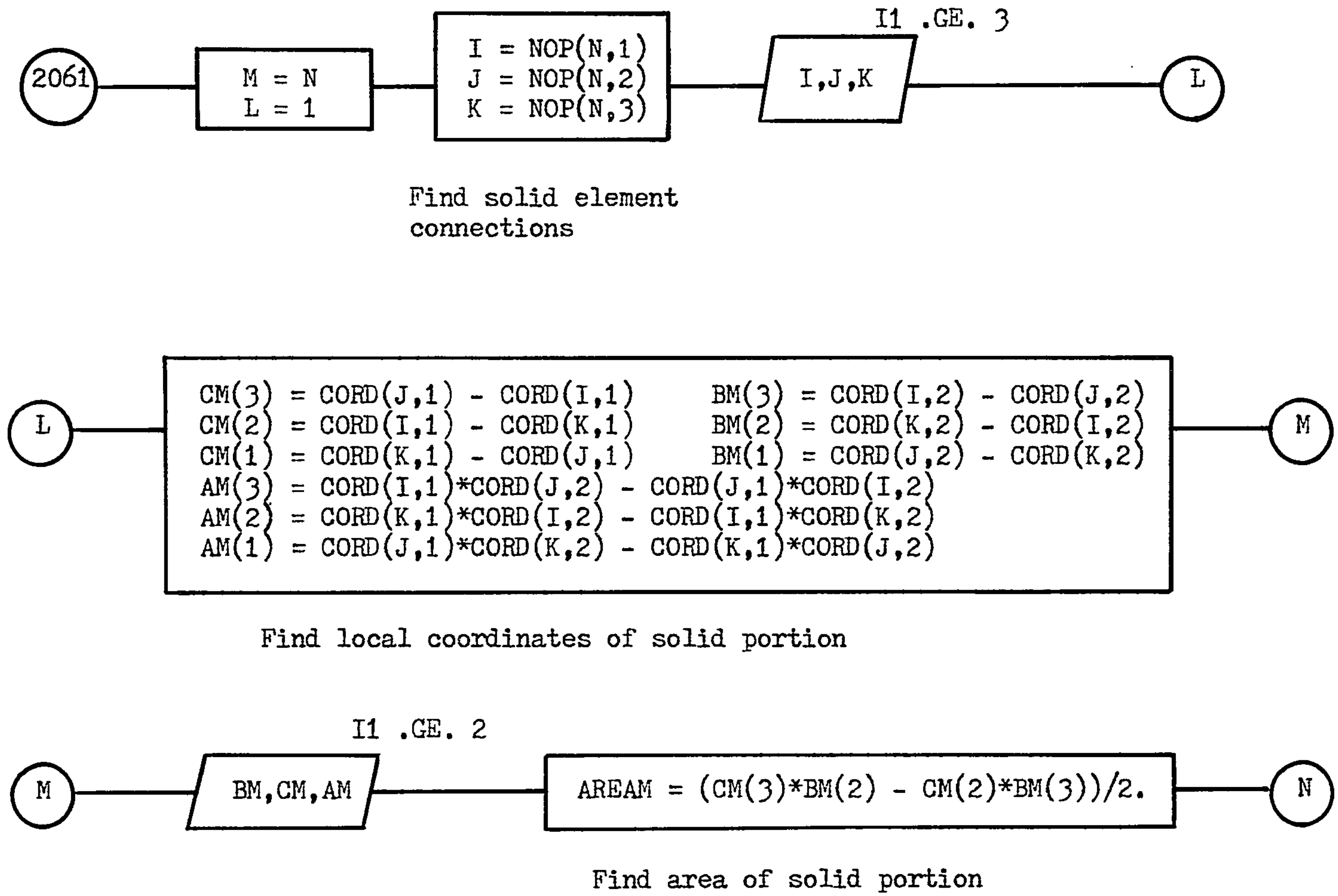


Fig. F-8g Flowchart of Subroutine STIFT3

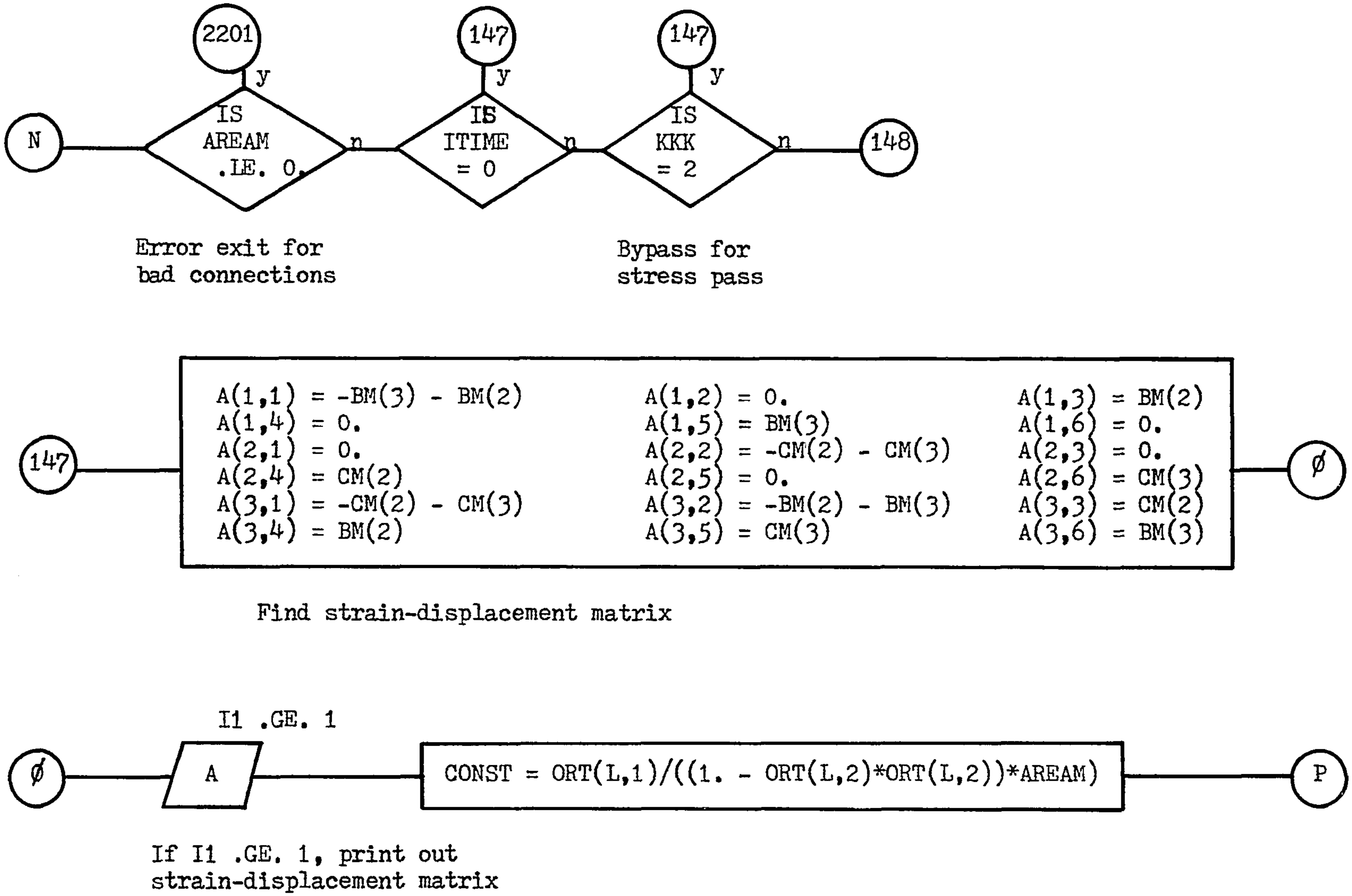


Fig. F-8h Flowchart of Subroutine STIFT3

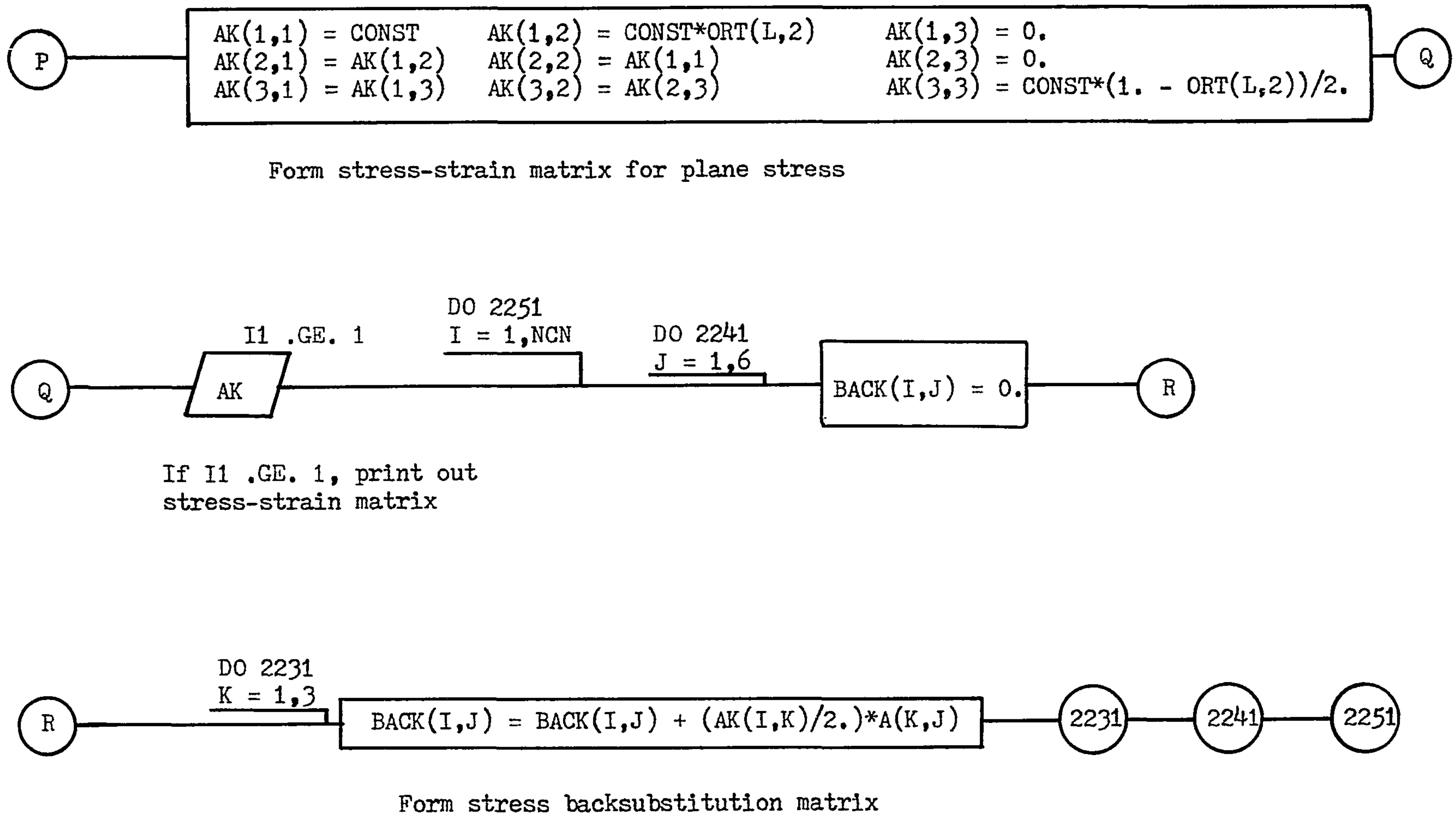


Fig. F-81 Flowchart of Subroutine STIFT3

ISSUE ENGR DRAWN TITLE NO. OF SHEETS PER SET SHEET

PRINTED IN U.S.A.
E-1812-A-1 (10-71)

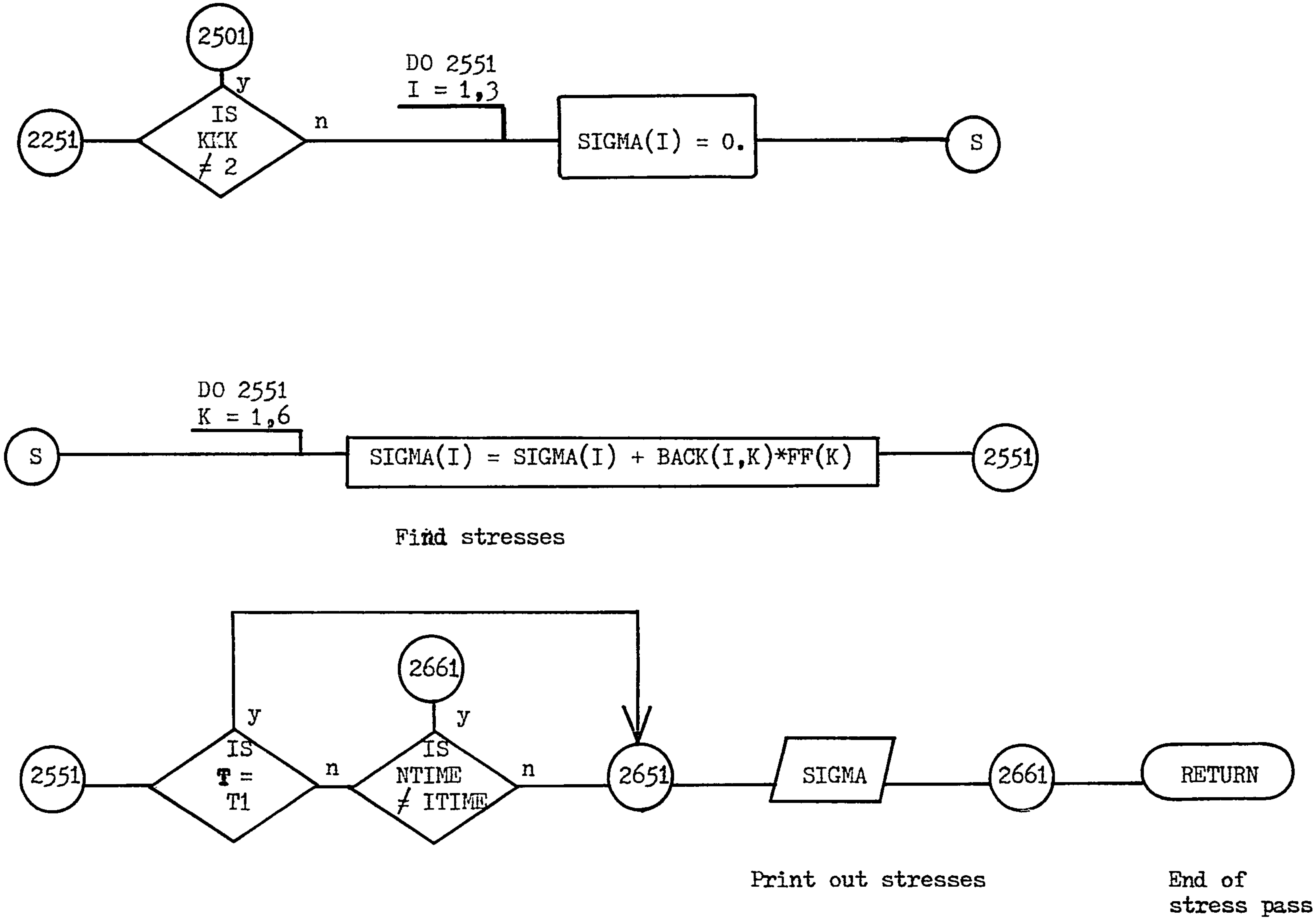


Fig. F-8j Flowchart of Subroutine STIFT3

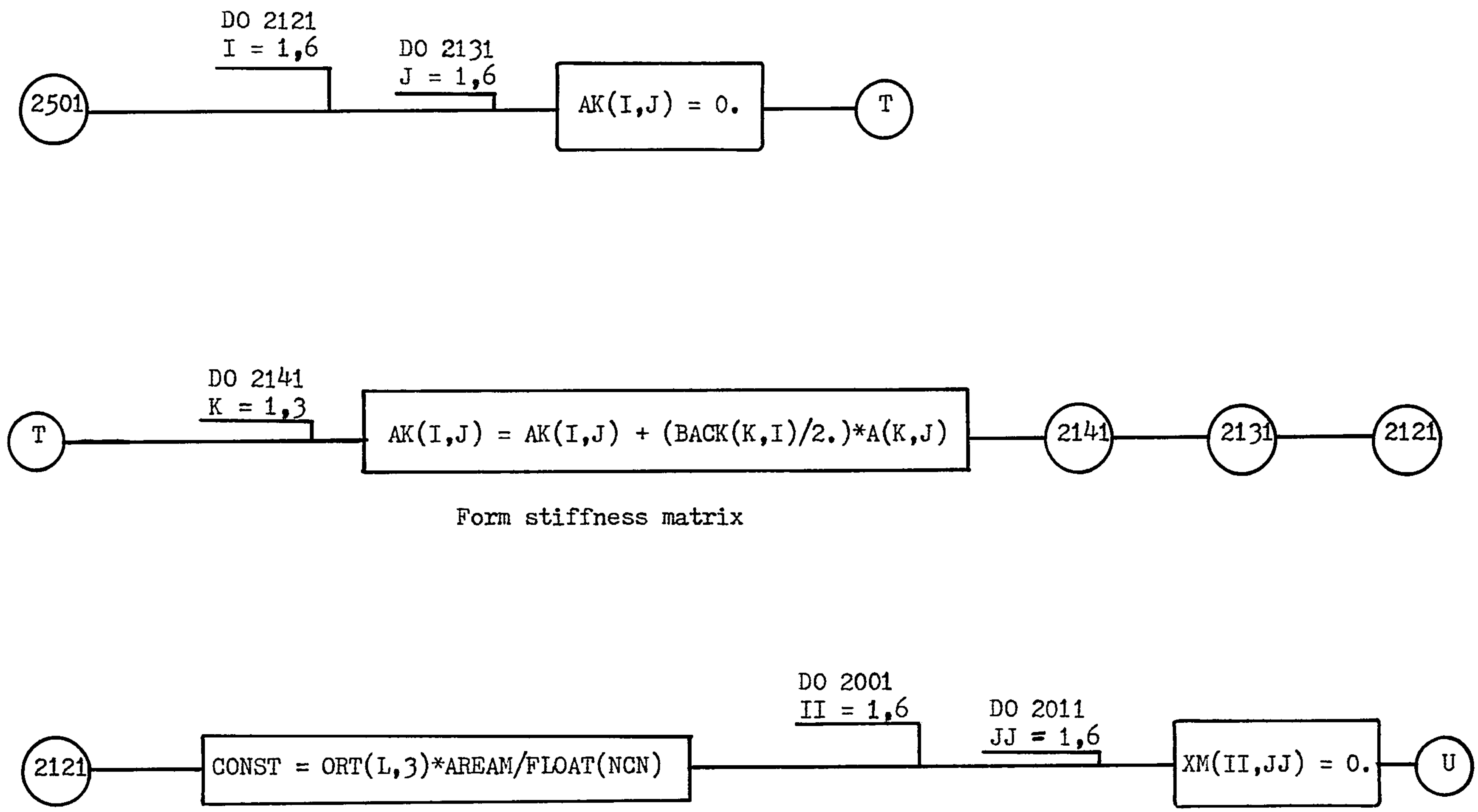


Fig. F-8k Flowchart of Subroutine STIFT3

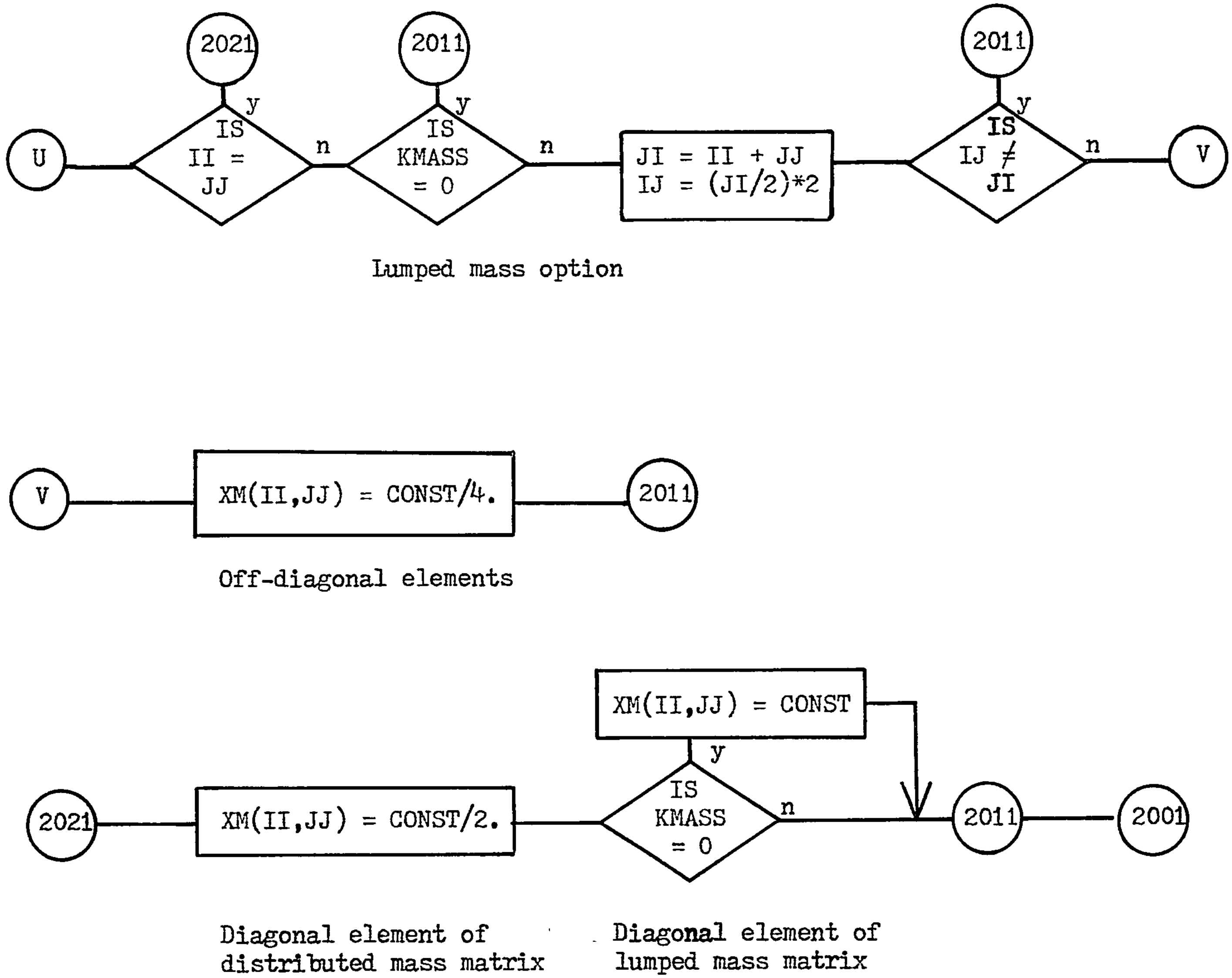


Fig. F-81 Flowchart of Subroutine STIFT3

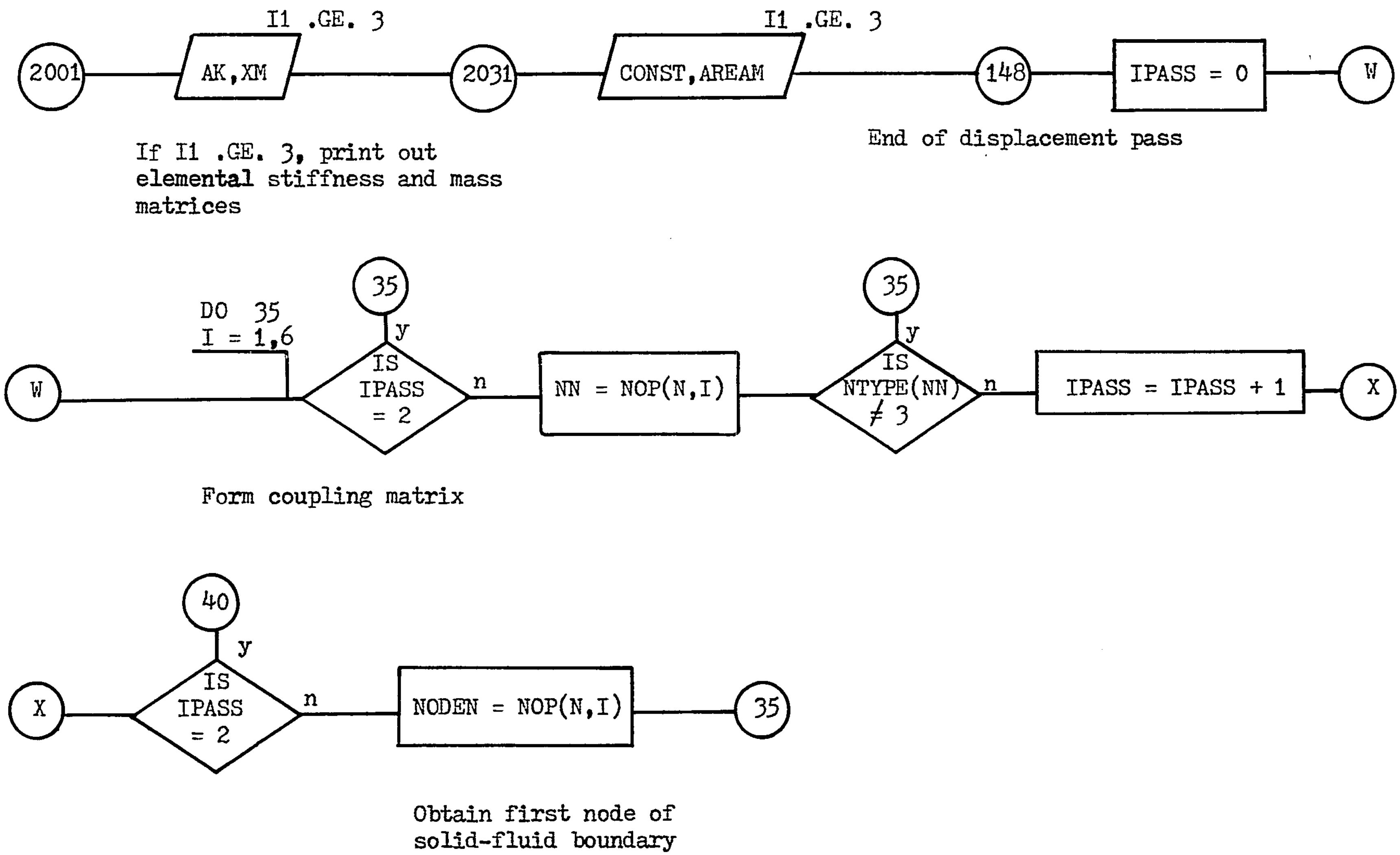


Fig. F-8m Flowchart of Subroutine STIFT3

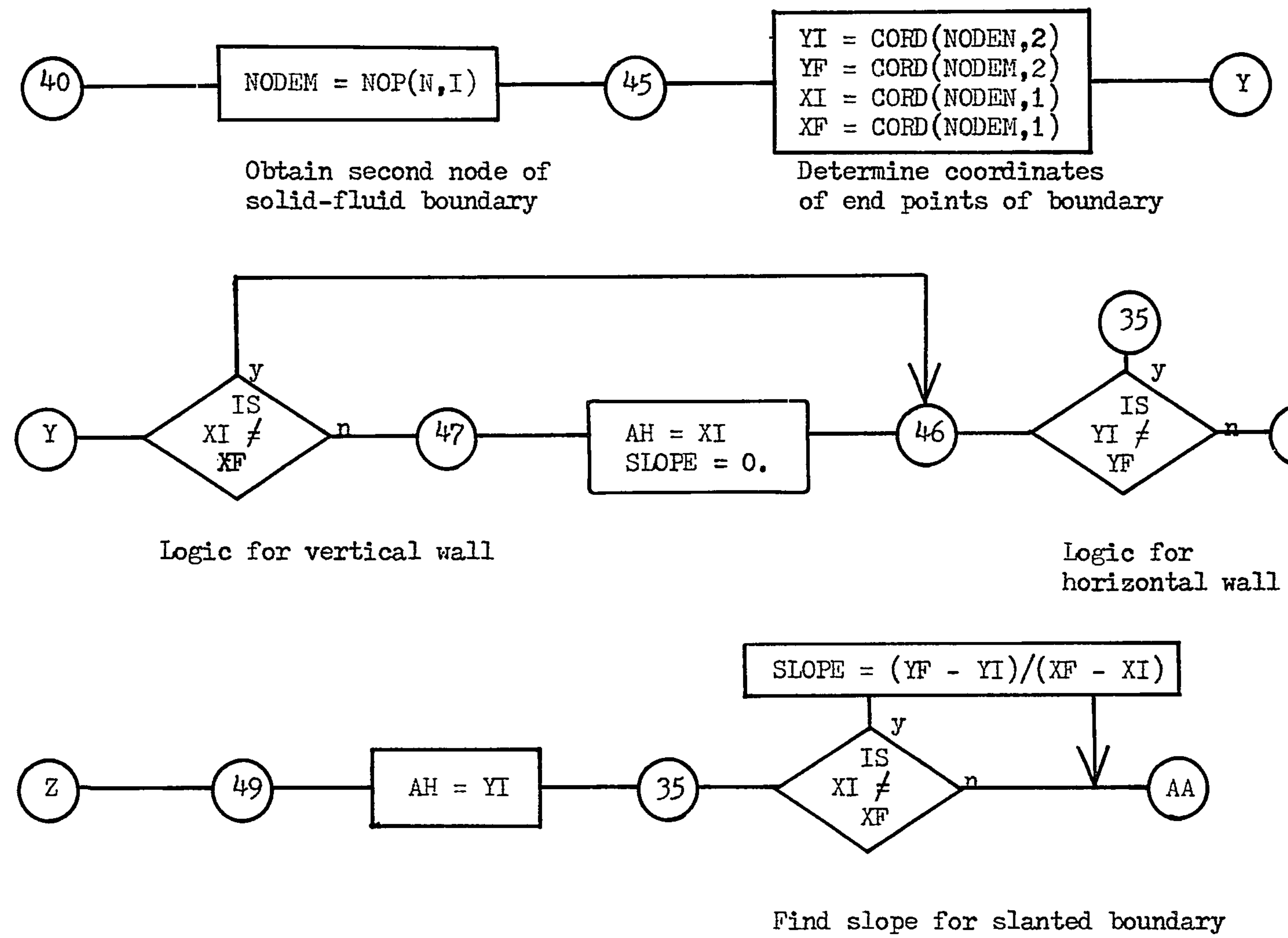


Fig. F-8n Flowchart of Subroutine STIFT3

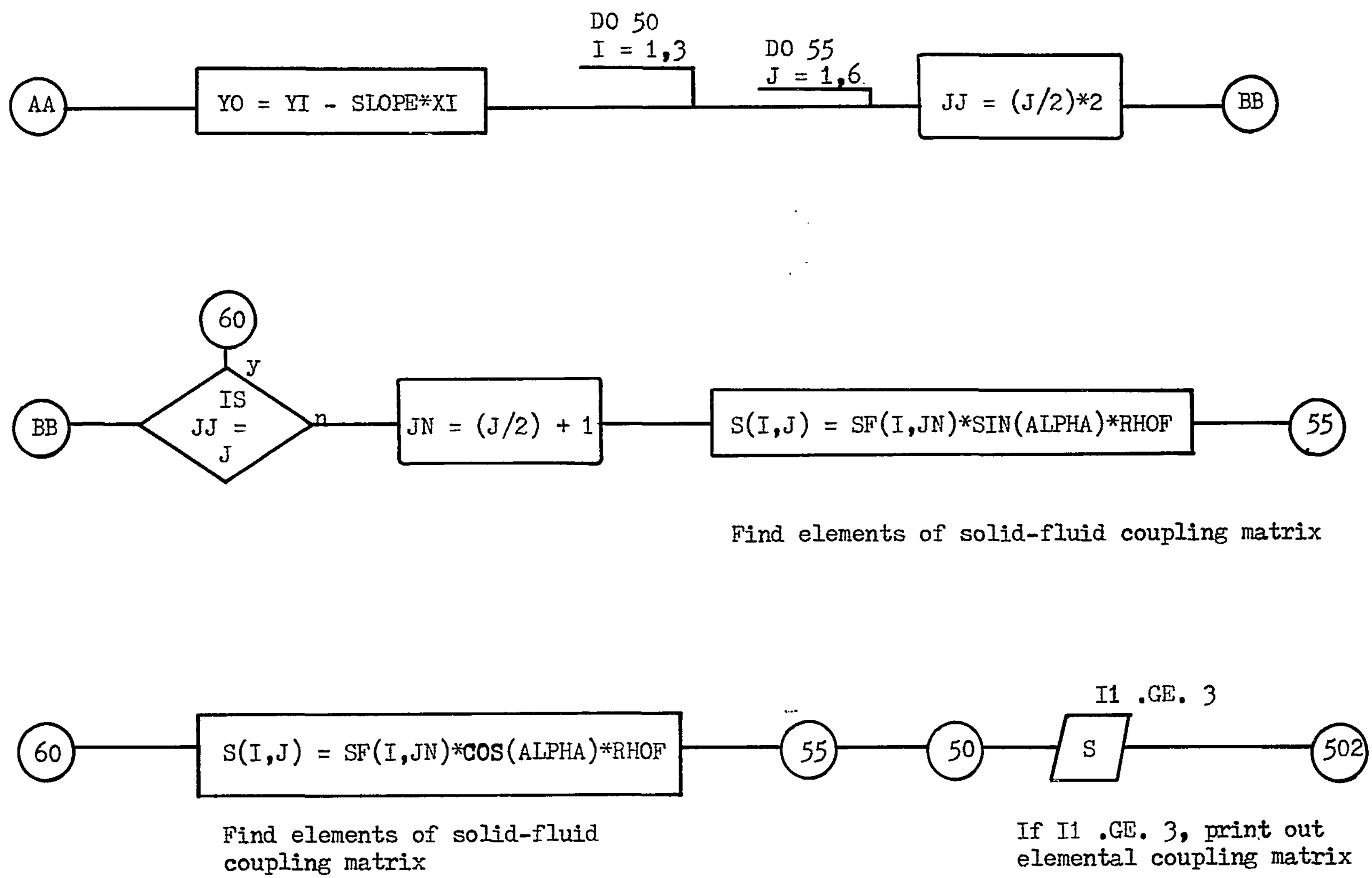


Fig. F-8o Flowchart of Subroutine STIFT3

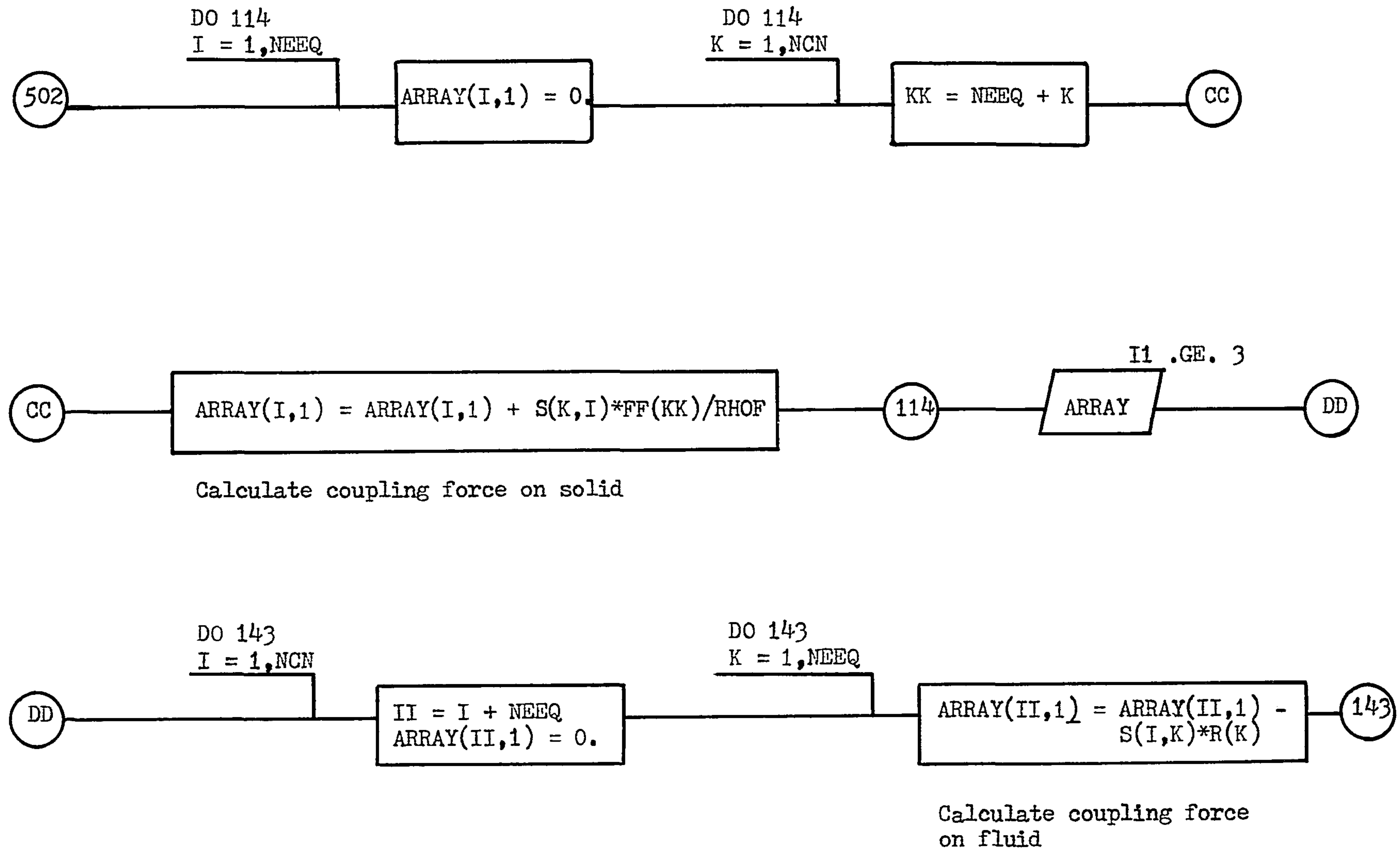


Fig. F-8p Flowchart of Subroutine STIFT3

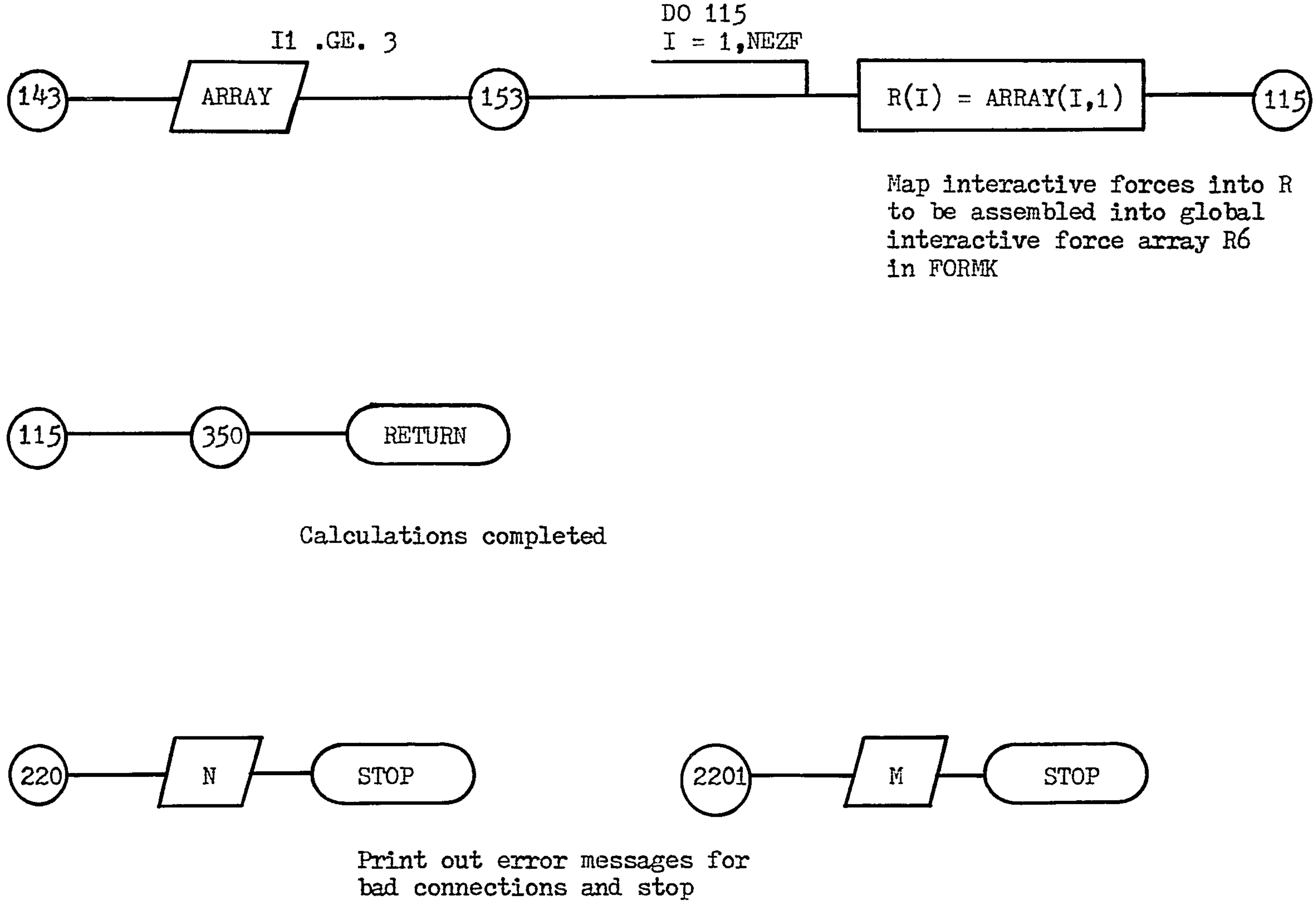


Fig. F-8q Flowchart of Subroutine STIFT3

1.9 SF Function Subroutine

Function SF(I,J) calculates the surface integral required to form the terms of the 3 x 6 solid-fluid coupling matrix. Because plane triangular elements compose the solid-fluid element, the integral, which was obtained analytically (Appendix 1) and coded into this functional subroutine, is calculated over a straight line segment (XI,YI) to (XF,YF). I and J are indicial node numbers which refer to the node of the element being considered, i.e., first, second, or third. I refers to the fluid nodes of the solid-fluid superelement, and J refers to the solid nodes. The flowchart for the function SF(I,J) is given in Fig. F-9a through F-9b .

1.10 MODAL Subroutine

Subroutine MODAL is entered generally in the first iteration. If solid-fluid problem is being analyzed or if updating the matrices is specified because of large solid displacements, MODAL is entered also in subsequent iterations. MODAL prepares the matrix differential equation for analytic solution. Its function is to find the eigenvalues and eigenvectors of the reduced matrix differential equation, and to uncouple the equation by forming the generalized mass and stiffness matrices. In order to accomplish this, MODAL calls the subroutines REDUC1, which transforms the eigenvalue equation $[A]\{x\} = \lambda[B]\{x\}$ into the form $[K]\{z\} = \lambda\{z\}$, JACOBI, which calculates the eigenvalues and eigenvectors of the transformed equation by the Jacobi method, and REBAKA, which transforms the eigenvectors from z back to x. The initial displacements and velocities which are needed to solve the uncoupled matrix differential equation, as well as the load arrays are converted to generalized form by this subroutine. If MODAL

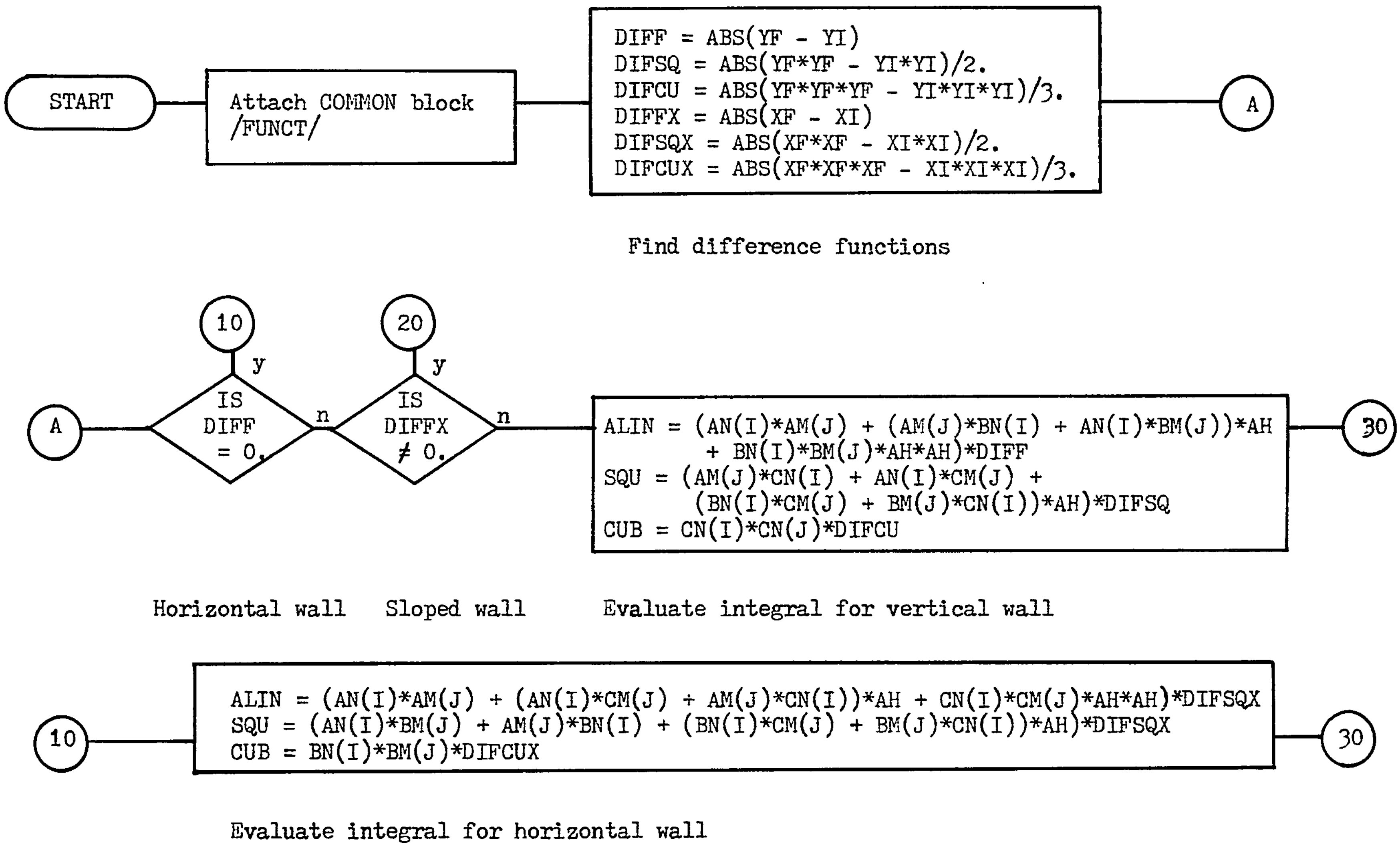


Fig. F-9a Flowchart Of Function SF

(20)

```
ALIN = (AN(I)*AM(J) + (AN(I)*CM(J) + AM(J)*CN(I))*YO + CN(I)*CM(J)*YO*YO)*DIFFX
SQU = (AN(I)*BM(J) + AM(J)*BN(I) + (AN(I)*CM(J) + AM(J)*CN(I))*SLOPE +
      (BN(I)*CM(J) + BM(J)*CN(I))*YO + CN(I)*CM(J)*2.*YO*SLOPE)*DIFSQX
CUB = (BN(I)*BM(J) + (BN(I)*CM(J) + BM(J)*CN(I))*SLOPE
      +CN(I)*CM(J)*SLOPE*SLOPE)*DIFCUX
```

(30)

Evaluate integral for sloped wall

(30)

```
SF = (ALIN + SQU + CUB)/(4.*AREAN*AREAM)
```

RETURN

Find function

Fig. F-9b Flowchart of Function SF

is entered because the coupling loads only were calculated in FORMK, MODAL only converts these loads to generalized form and adds them to the constant generalized load array. The calling sequence for this subroutine is

CALL MODAL

Its flowchart is given in Fig. F-10a through F-10g .

1.11 REDUC1 Subroutine

Subroutine REDUC1(A,B,N) transforms the general eigenvalue problem[1]

$$[A]\{x\} = \lambda[B]\{x\}$$

to

$$[K]\{z\} = \lambda\{z\}$$

in which

$$[K] = [L]^{-1}[A][L]^{-T}$$

$$\{z\} = [L]^{T}\{x\}$$

$$[B] = [L][L]^{T}$$

[L] is a lower triangular matrix. Symmetric matrices are assumed. The transformed matrix [K] is formed in [A], whose original contents are destroyed. The off-diagonal elements of [L] are formed in [B], however, the diagonal of the [L] matrix is retained in the storage array {DL}. The calling sequence for this subroutine is

CALL REDUC1(A,B,N)

in which A and B are defined above, and N is the size of the matrices. The flowchart for REDUC1 is given in Fig. F-11a through F-11d . The flowchart is not commented because details on this subroutine are already available [15] .

[1] [A] corresponds to $[\ \bar{K}\ ]$, and [B] corresponds to $[\ \bar{M}\ ]$.

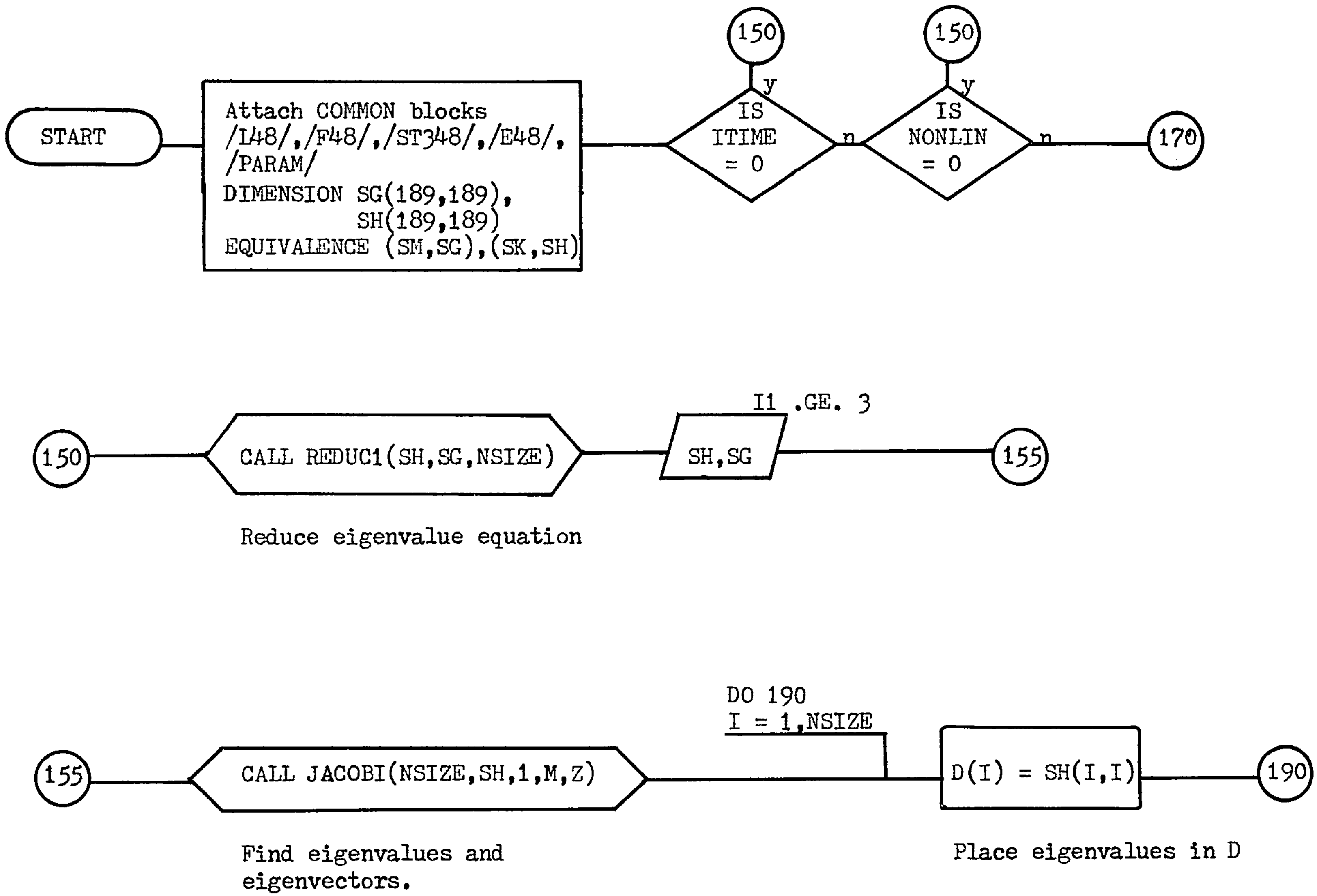


Fig. F-10a Flowchart of Subroutine MODAL

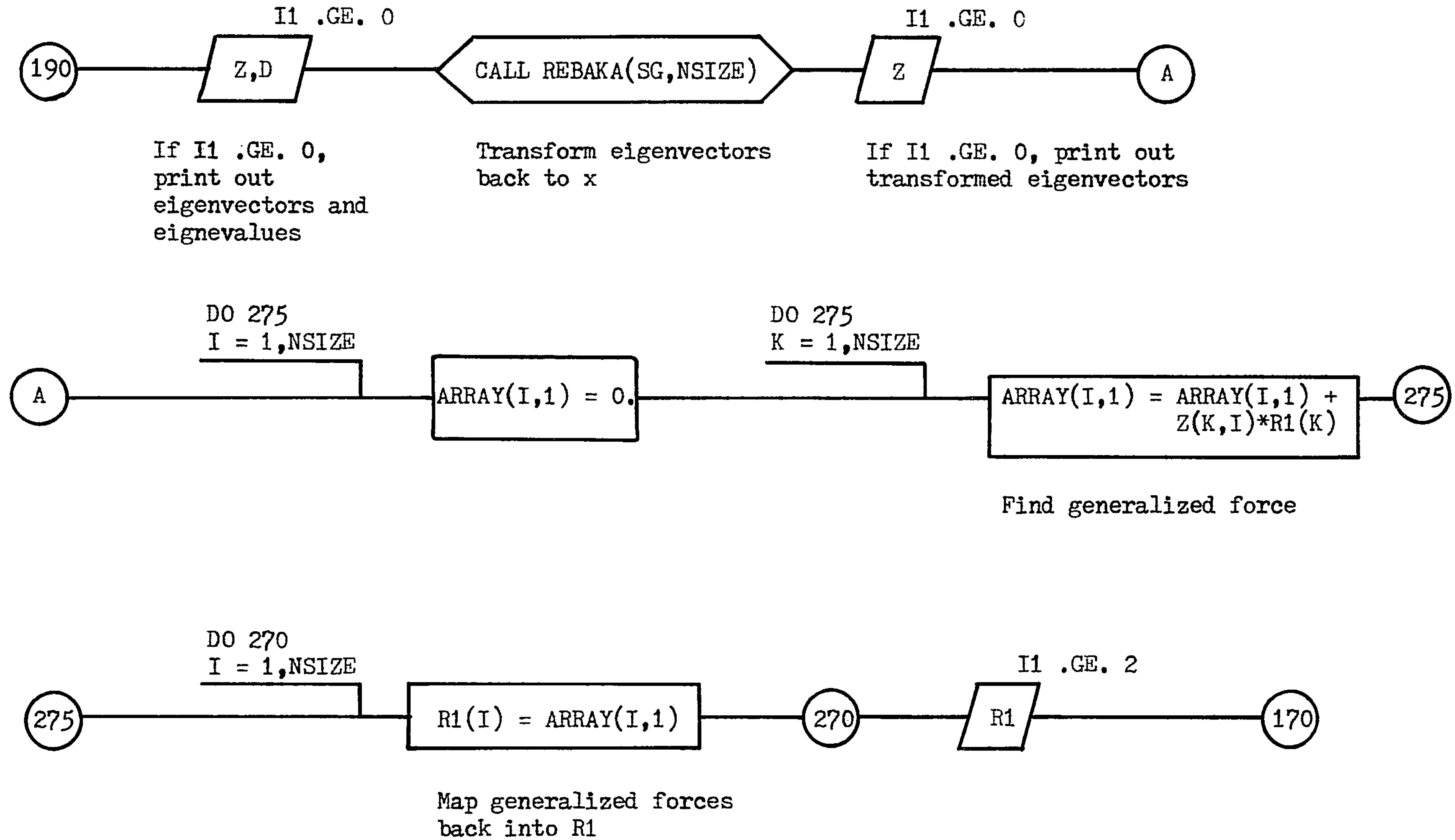


Fig. F-10b Flowchart of Subroutine MODAL

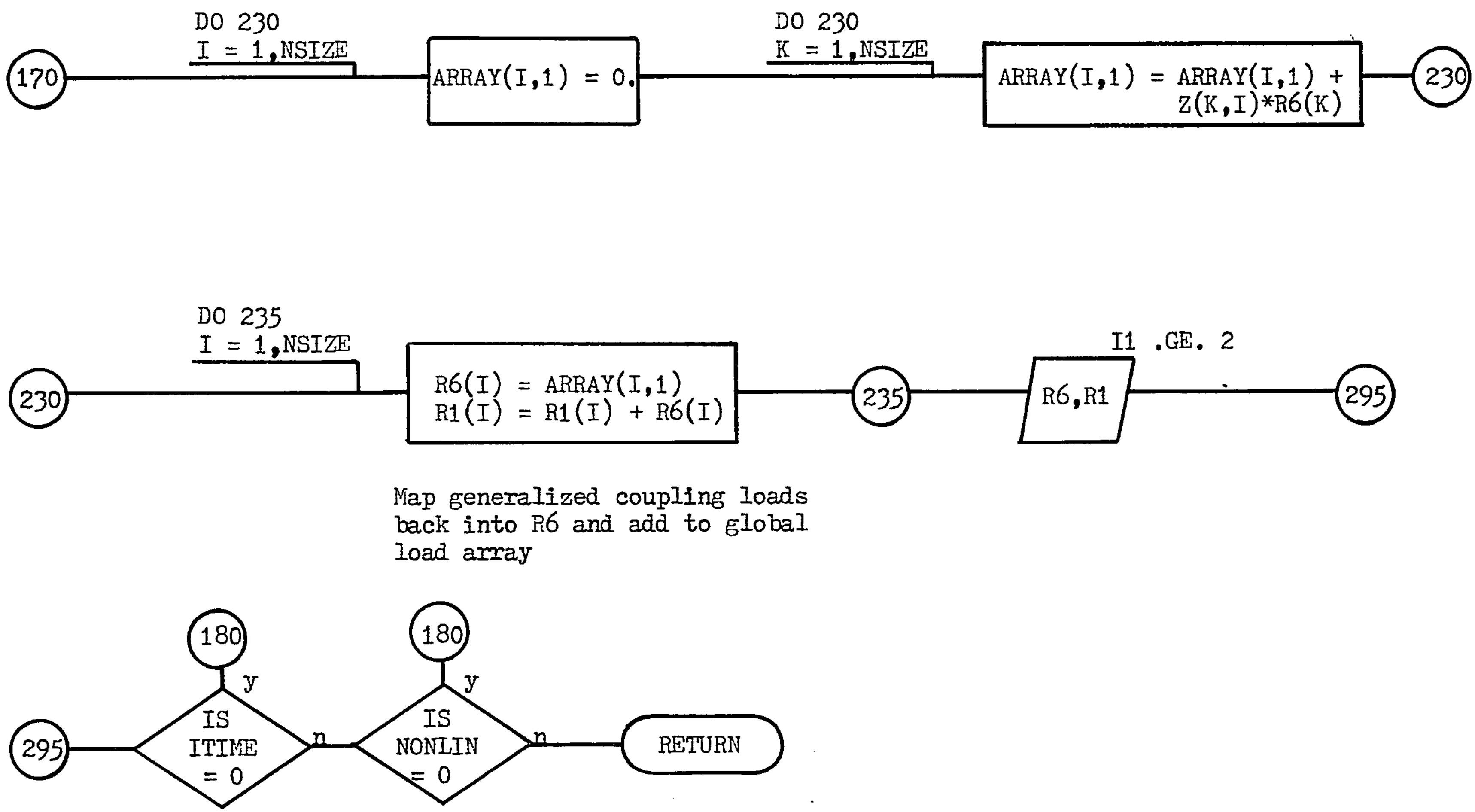


Fig. F-10c Flowchart of Subroutine MODAL

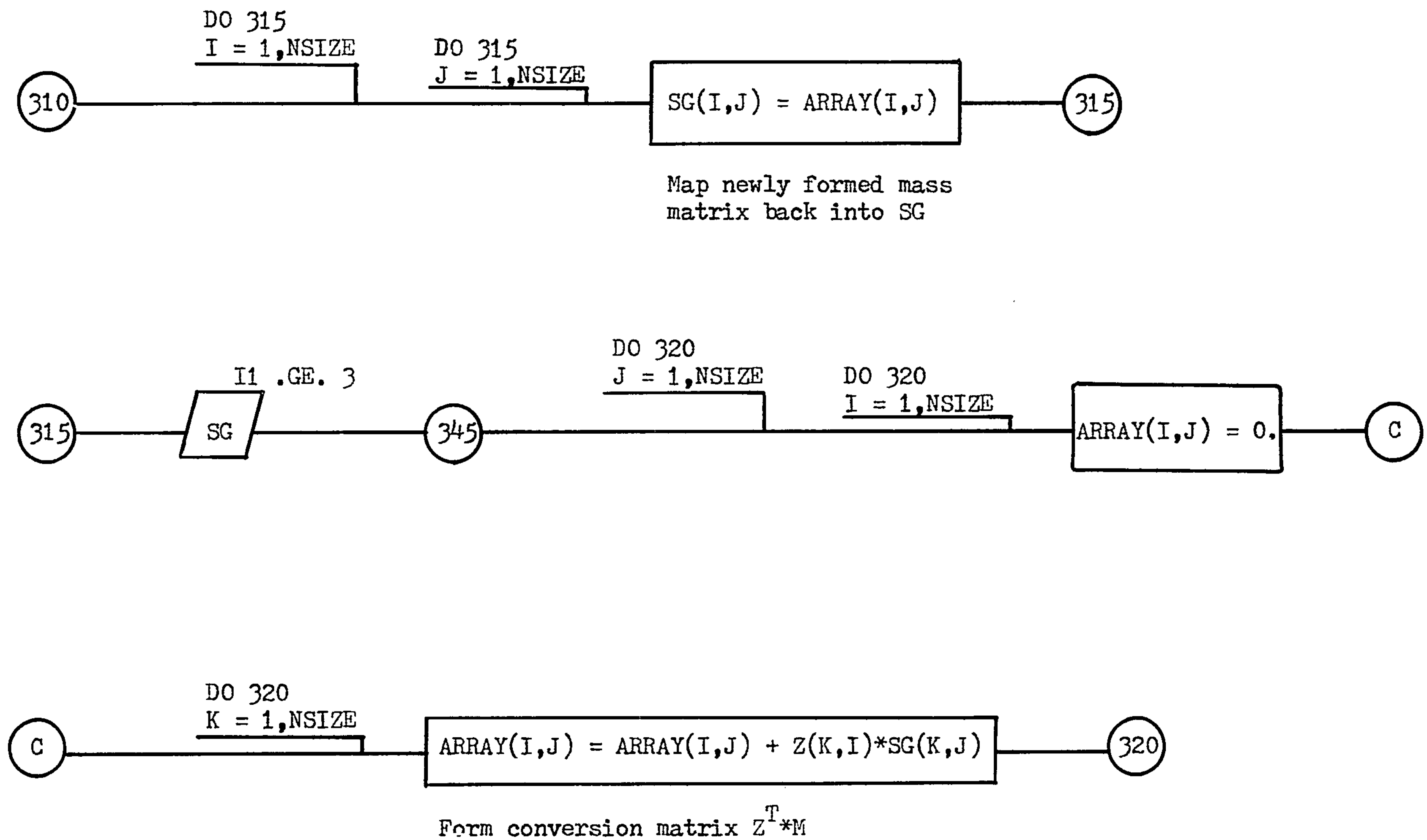


Fig. F-10d Flowchart of Subroutine MODAL

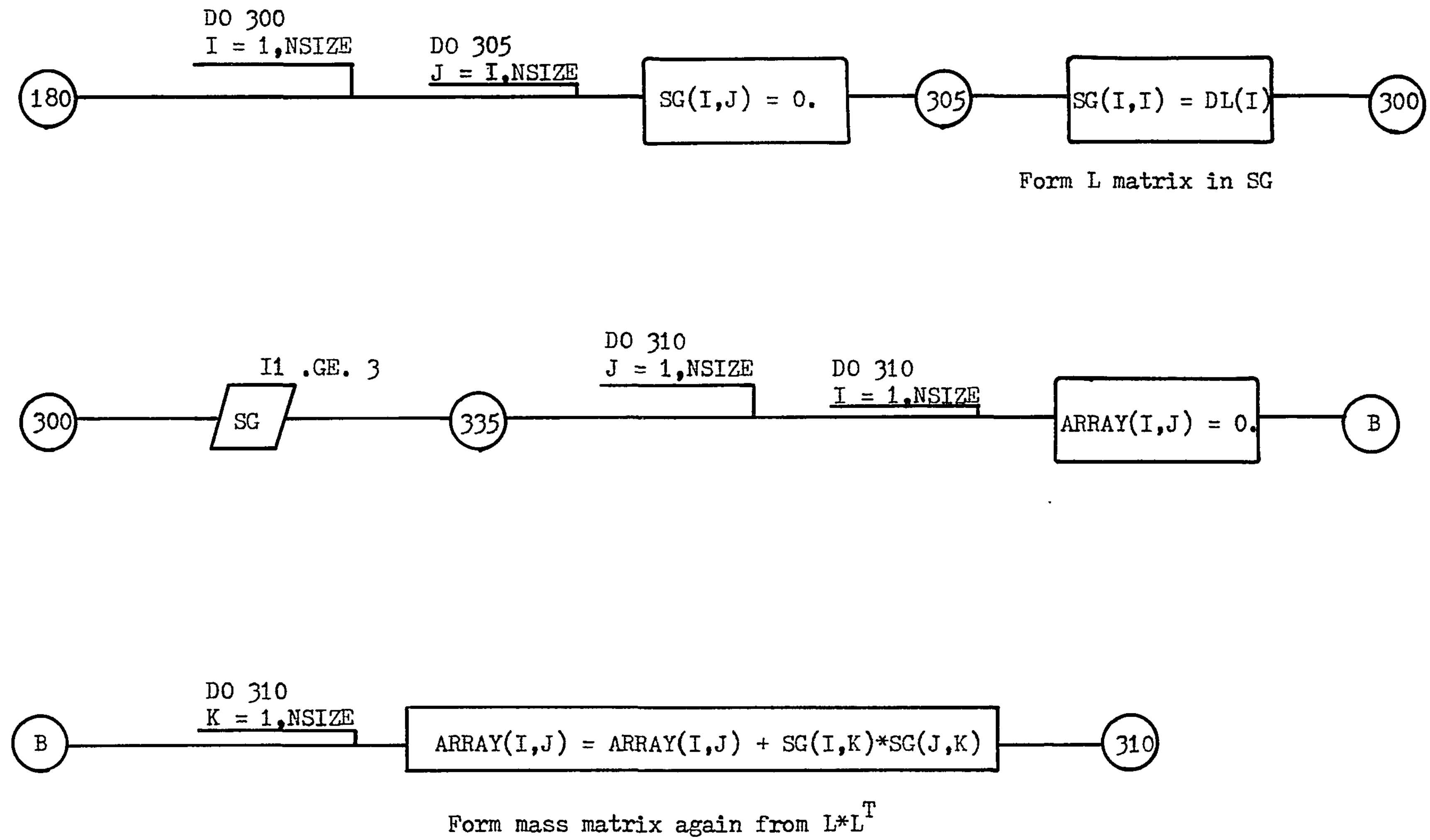


Fig. F-10e Flowchart of Subroutine MODAL

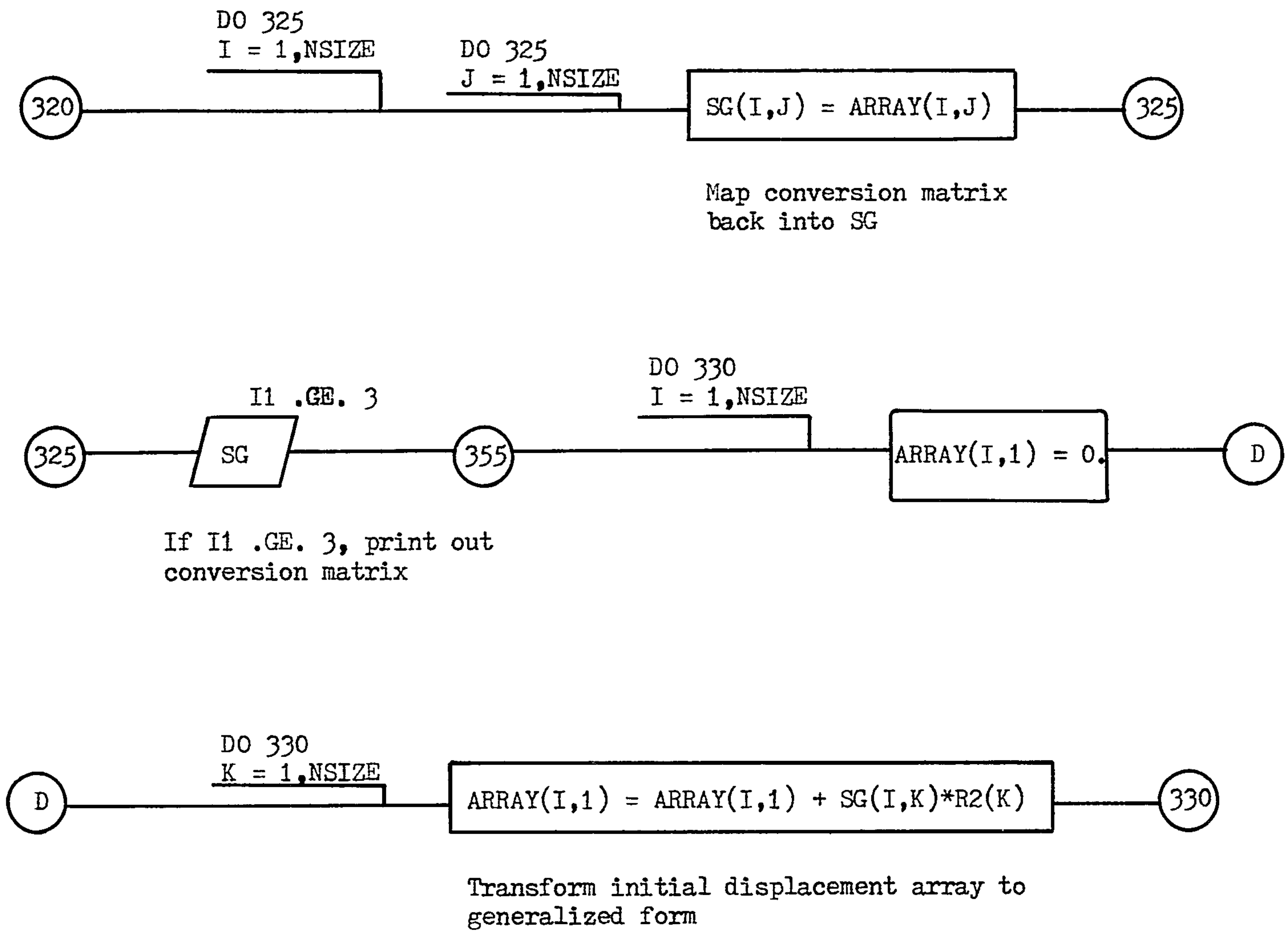


Fig. F-10f Flowchart of Subroutine MODAL

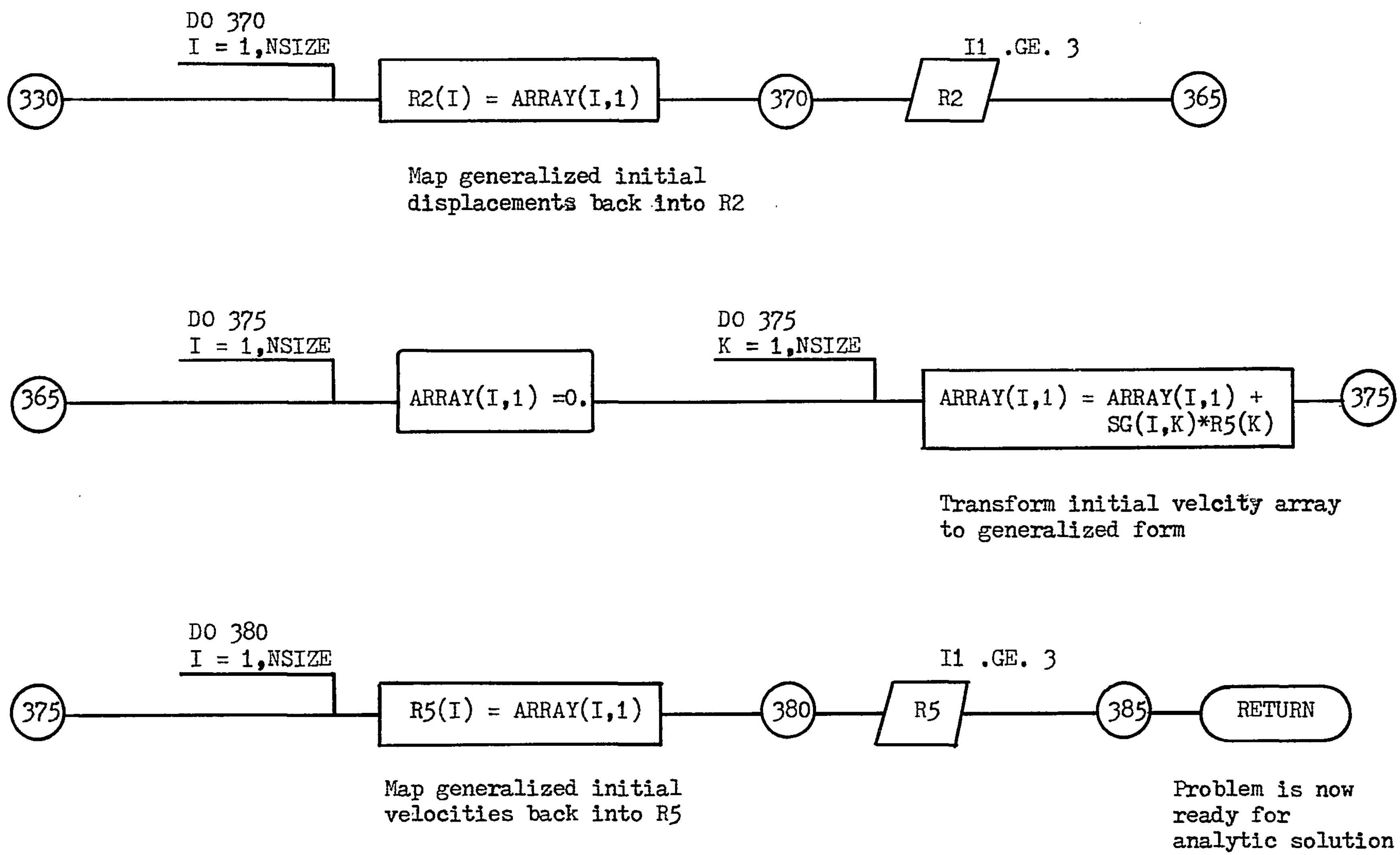


Fig. F-10g Flowchart of Subroutine MODAL

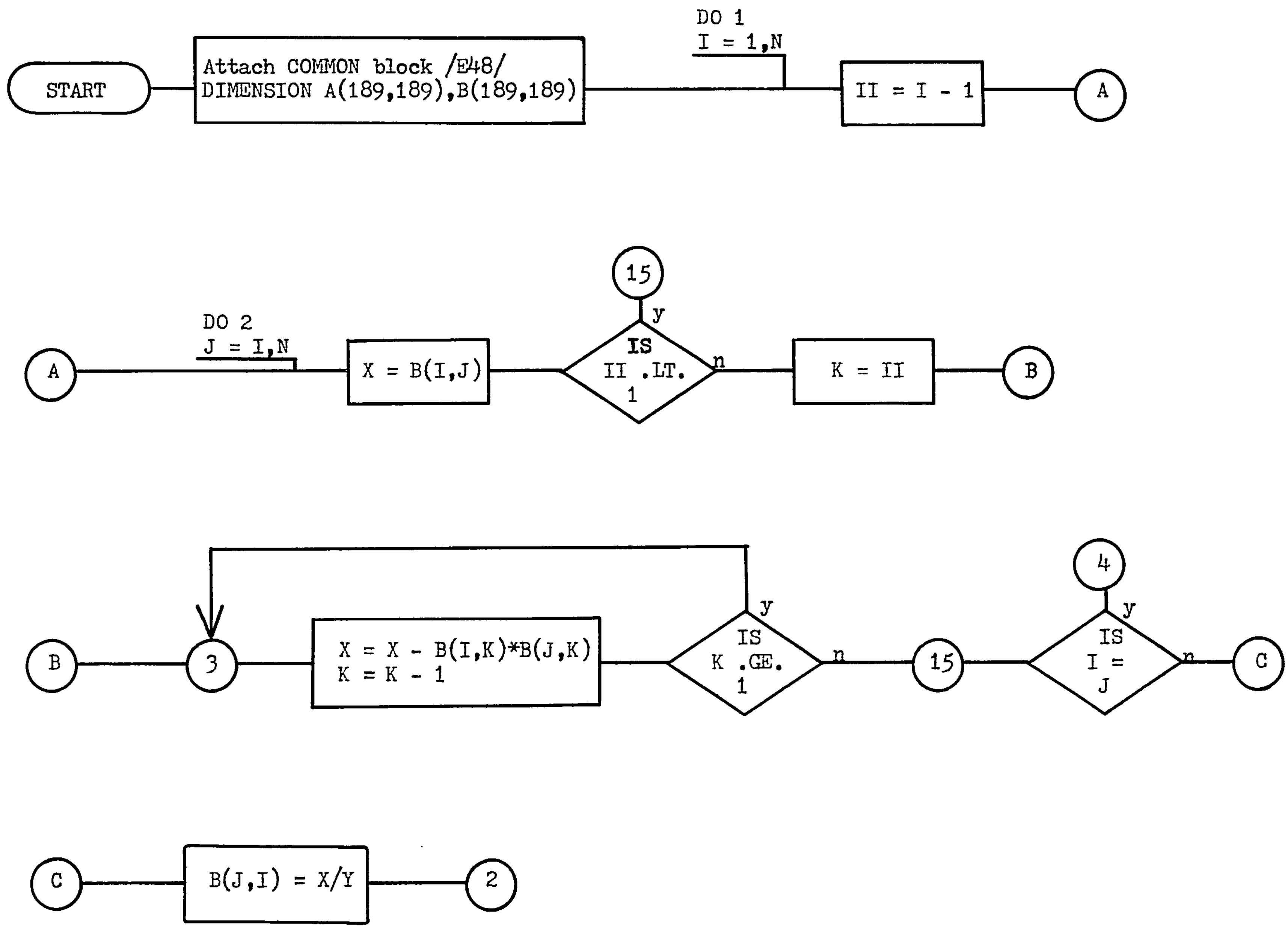


Fig. F-11a Flowchart of Subroutine REDUC1

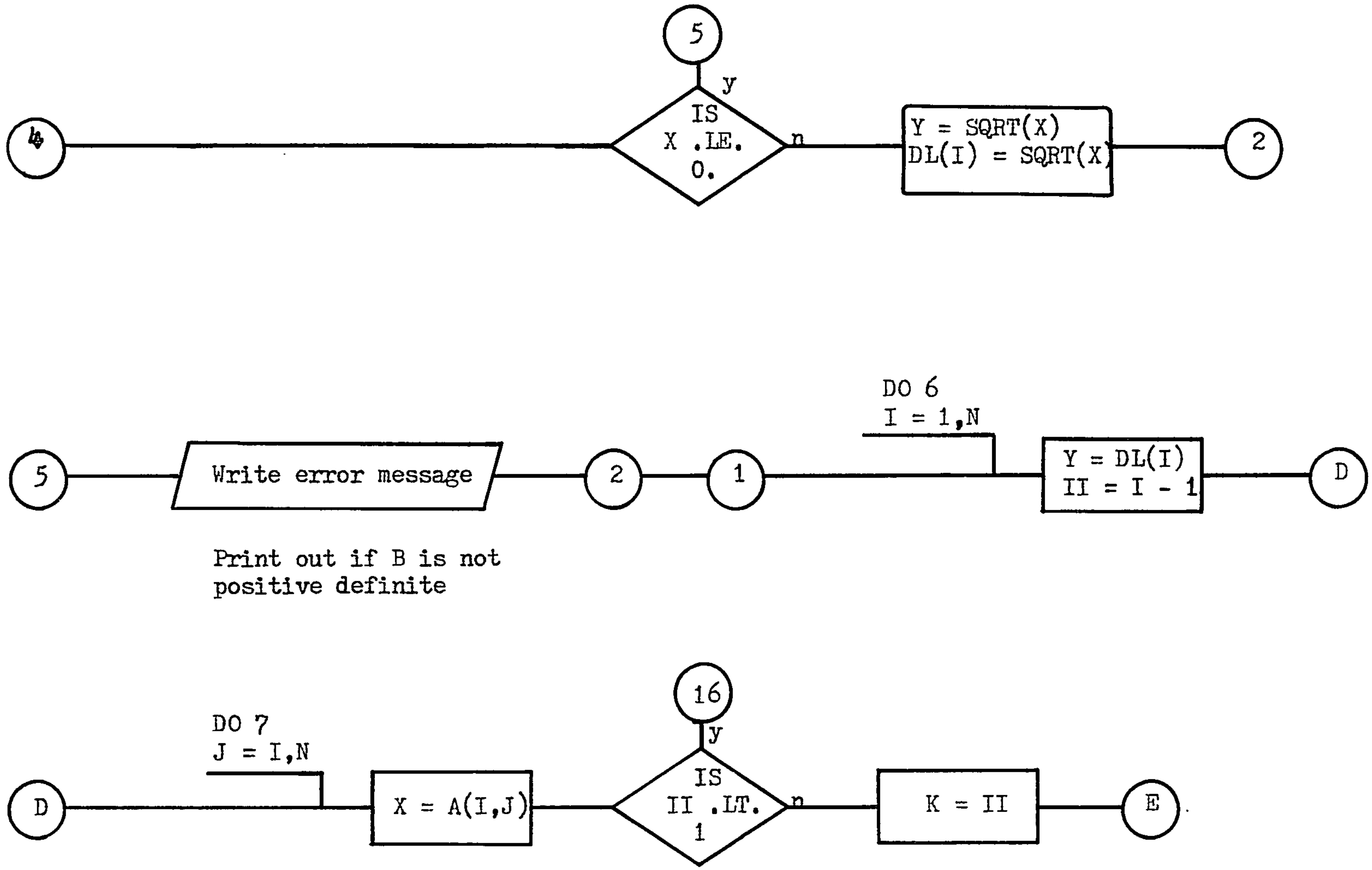


Fig. F-11b Flowchart of Subroutine REDUC1

PRINTED IN U.S.A.
E-1812-A-1 (10-71)

ISSUE	ENGR	TITLE		SHEET
	DRAWN		NO. OF SHEETS PER SET	

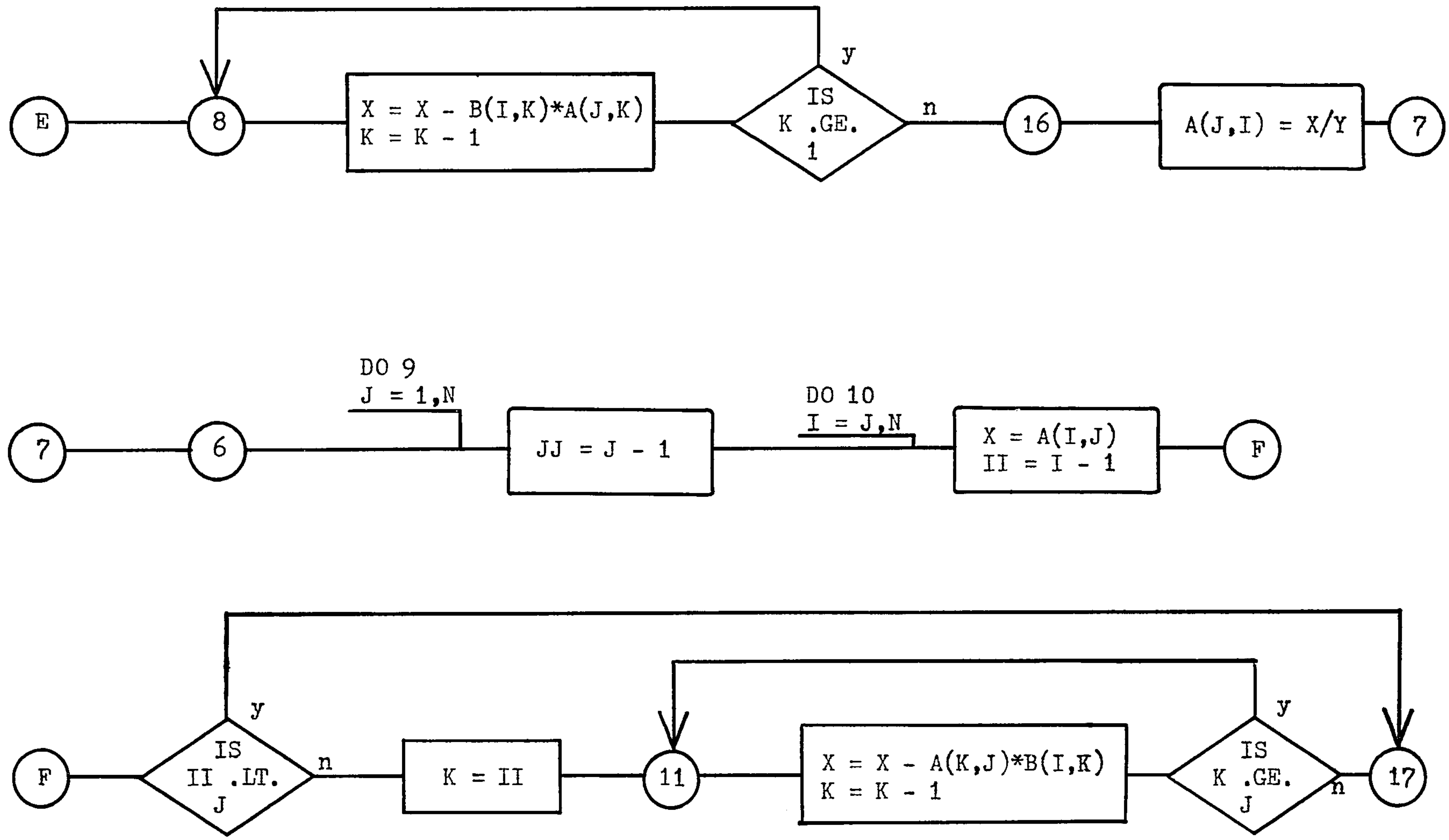


Fig. F-11c Flowchart of Subroutine REDUC1

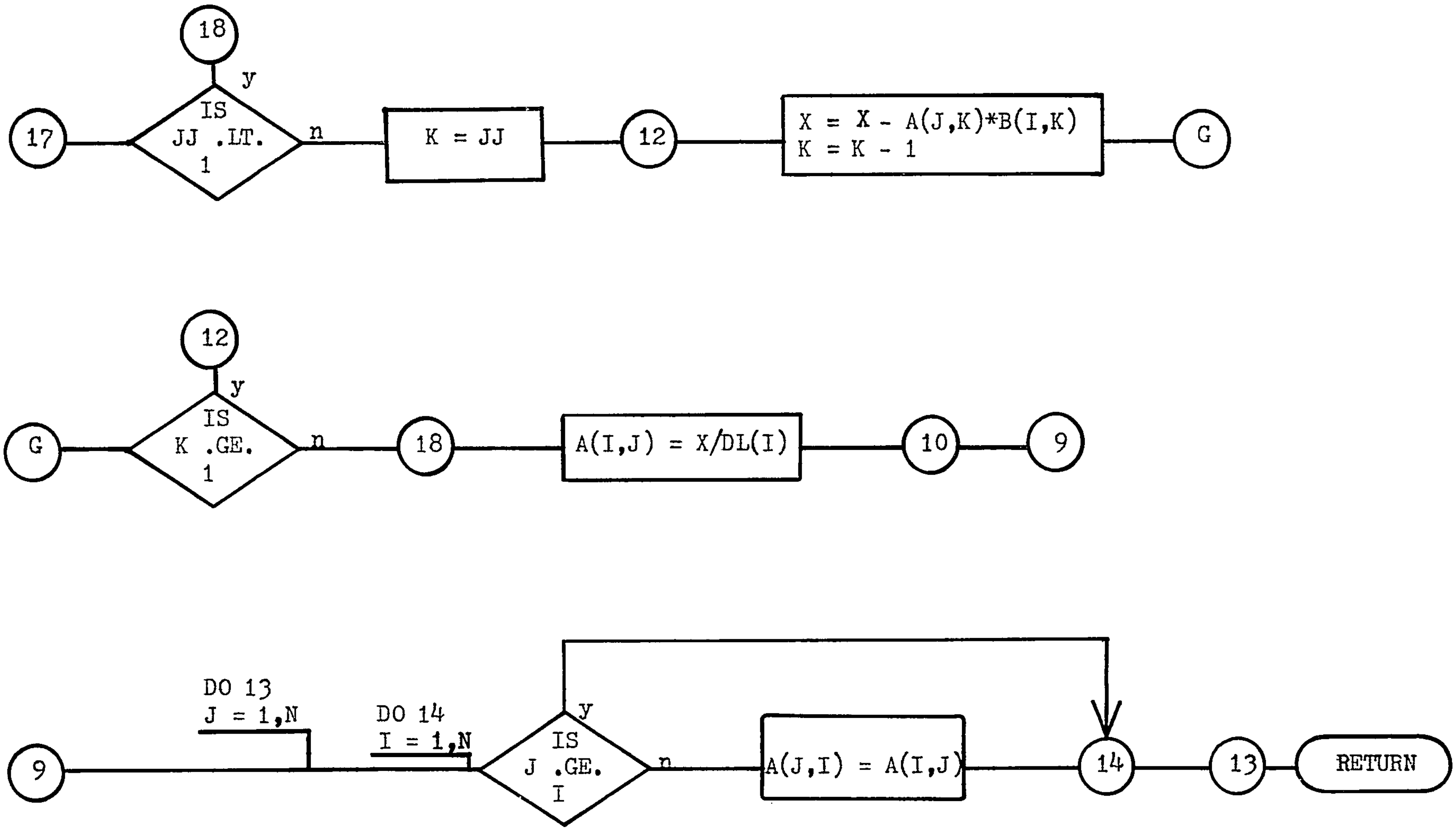


Fig. F-11d Flowchart of Subroutine REDUC1

1.12 JACOBI Subroutine

Subroutine JACOBI computes the eigenvalues and eignevectors by the Jacobi method for the matrix equation

$$[K] \{z\} = \lambda \{z\}$$

in which $[K]$ is real and symmetric. It is called by MODAL in the first iteration and in subsequent iterations if the mass and stiffness matrices are updated because of large solid displacements. The calling sequence for this subroutine is

CALL JACOBI(N,Q,JVEC,M,V)

in which N is the order of the matrix to be diagonalized, ≥ 2; Q is the matrix to be diagonalized, into whose diagonal the eigenvalues are placed; JVEC is the eigenvector flag described later; M is the number of rotations performed to achieve diagonalization; V is the temporary storage array for the eigenvectors. If JVEC = 0, only eigenvalues are to be found; if JVEC = 1, both eigenvalues and eigenvectors are to be found. The flowchart for JACOBI if given in Fig. F-12a through F-12j . It is also not commented because details on this subroutine are already available [12].

1.13 REBAKA Subroutine

Once the eigenvectors are found, subroutine REBAKA performs the operation

$$\{x\} = [L]^{-T} \{z\}$$

to transform them back to the original coordinates $\{x\}$. This is accomplished using the off-diagonal elements of $[L]$, located in the original mass matrix, and the diagonal elements of $[L]$, stored in DL . The calling sequence for this subroutine is

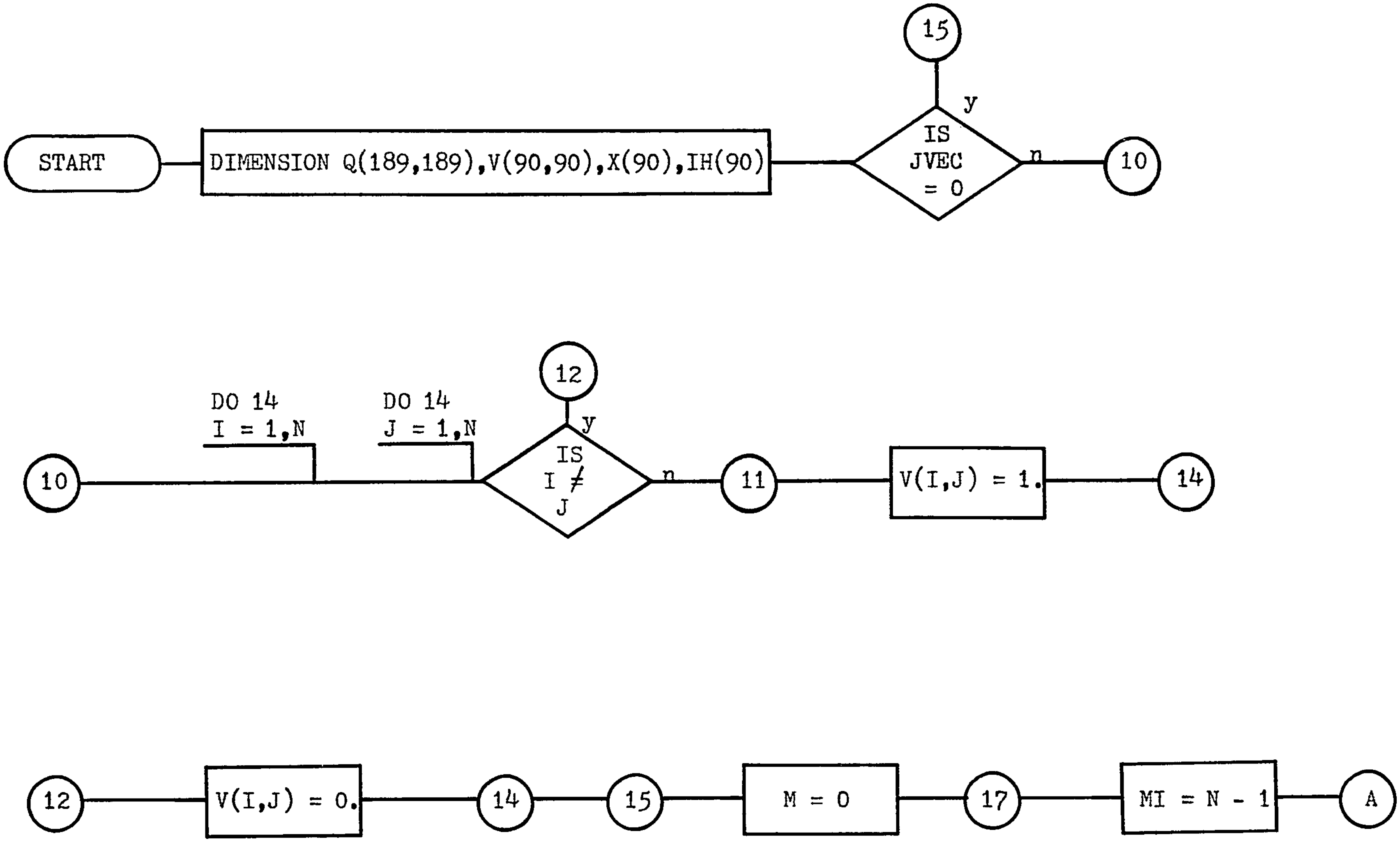


Fig. F-12a Flowchart of Subroutine JACOBI

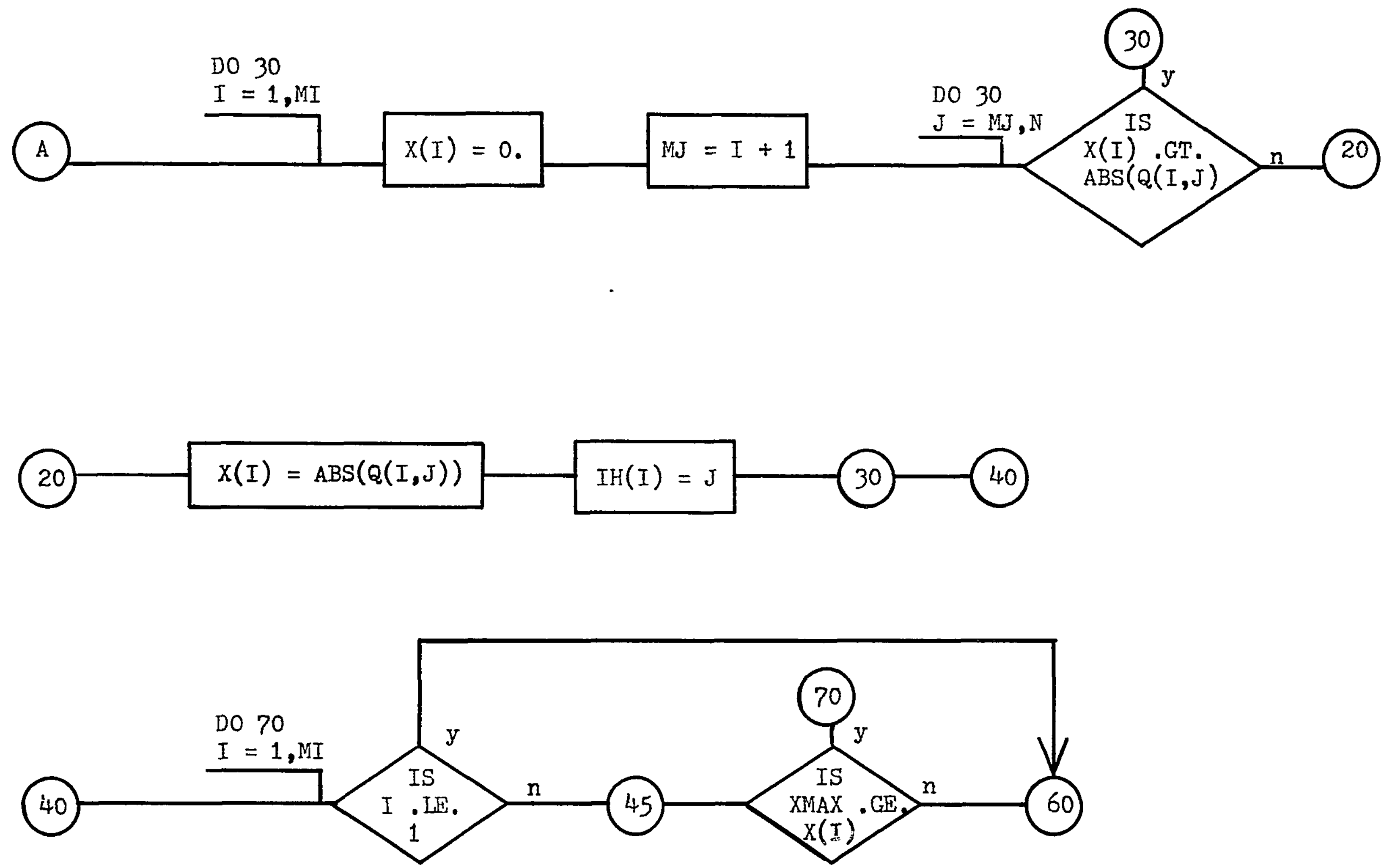


Fig. F-12b Flowchart of Subroutine JACOBI

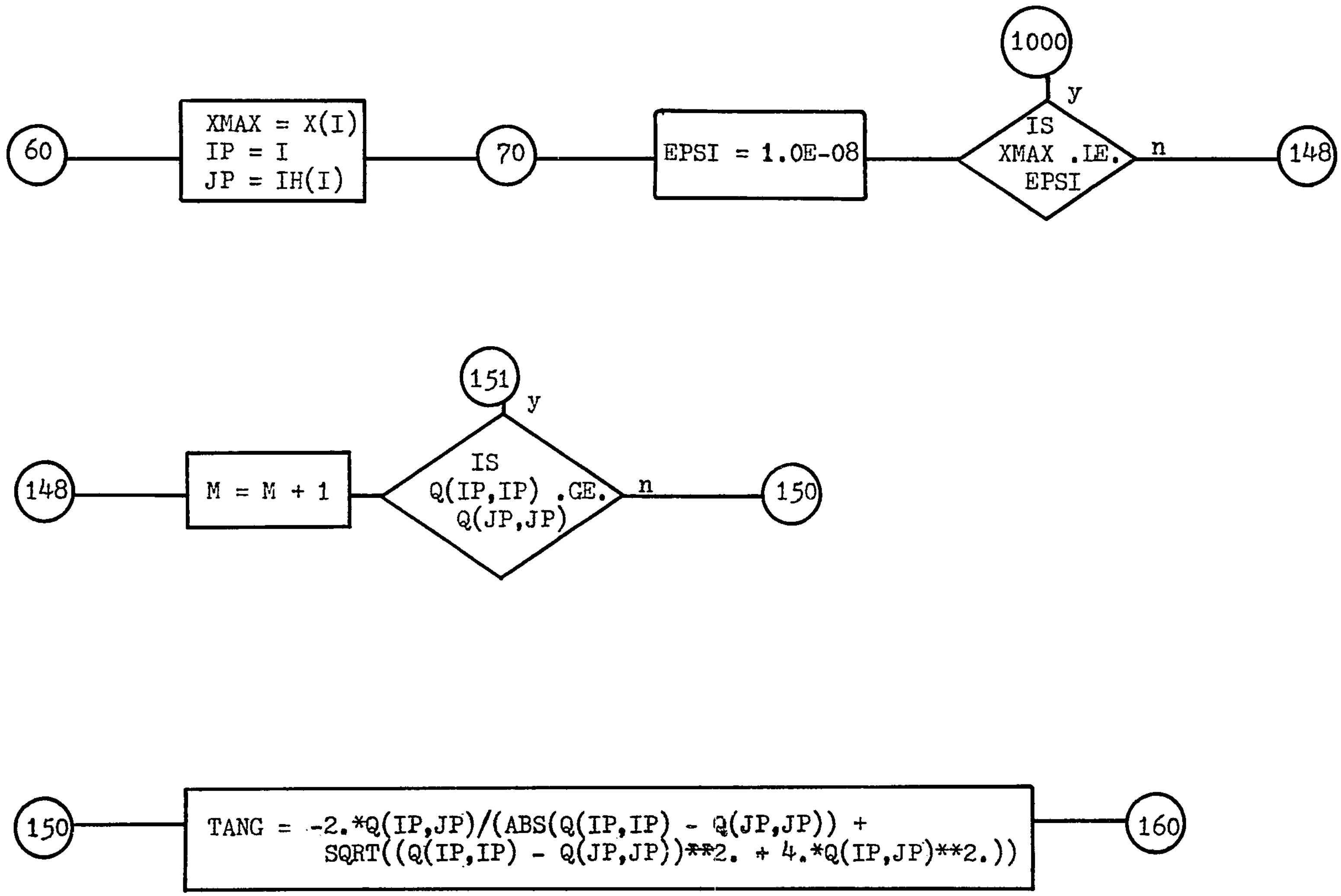


Fig. F-12c Flowchart of Subroutine JACOBI

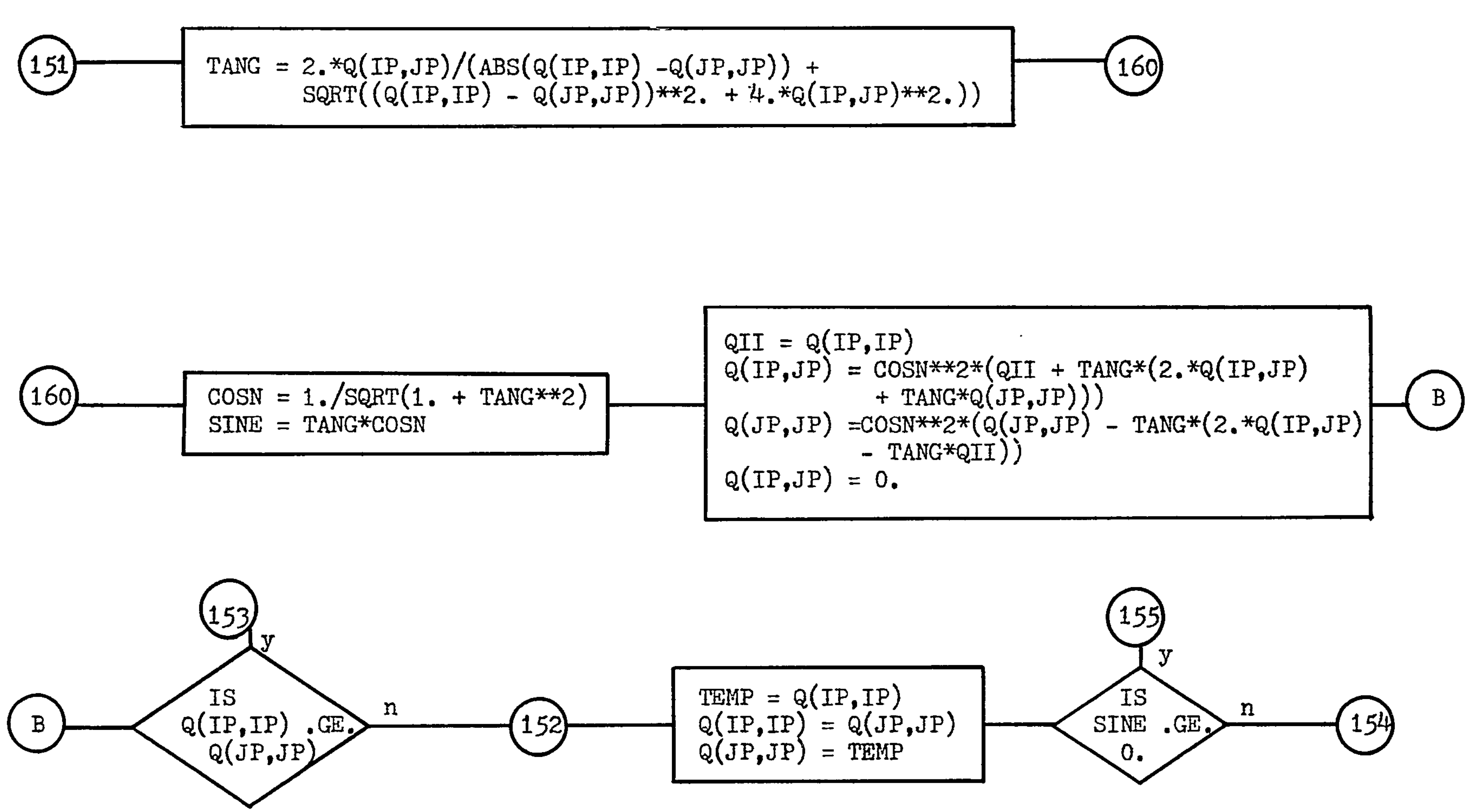


Fig. F-12d Flowchart of Subroutine JACOBI

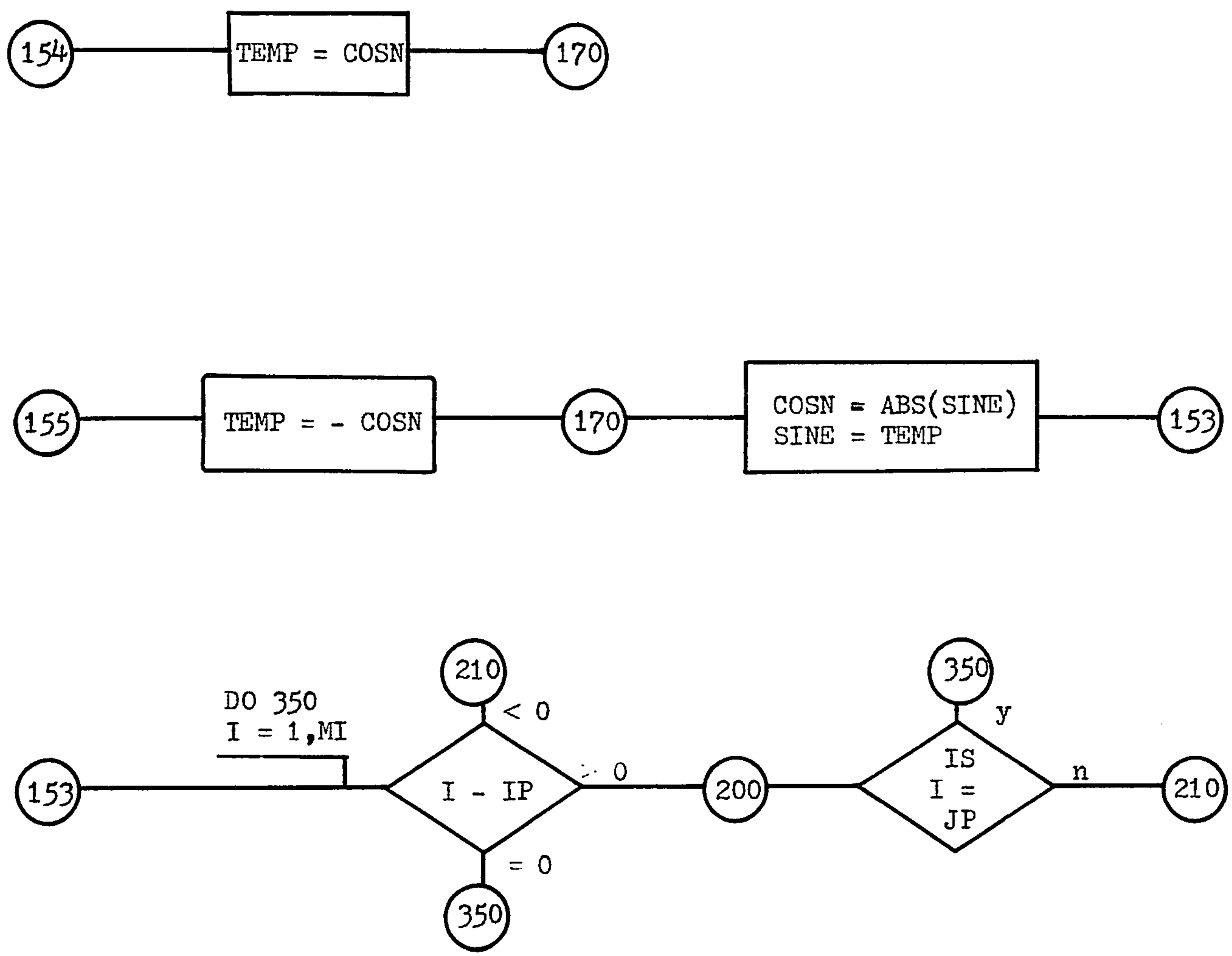


Fig. F-12e Flowchart of Subroutine JACOBI

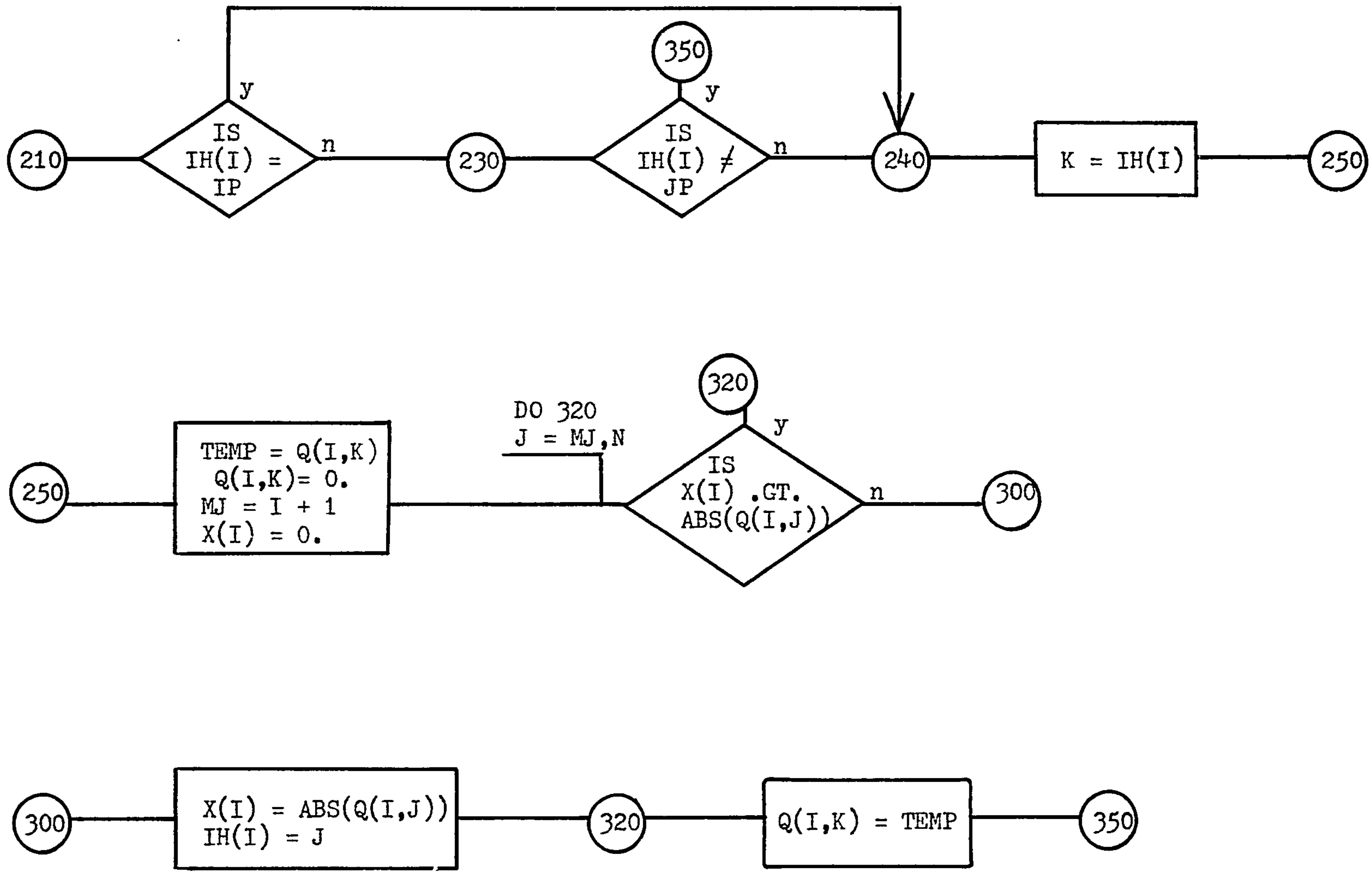


Fig. F-12f Flowchart of Subroutine JACOBI

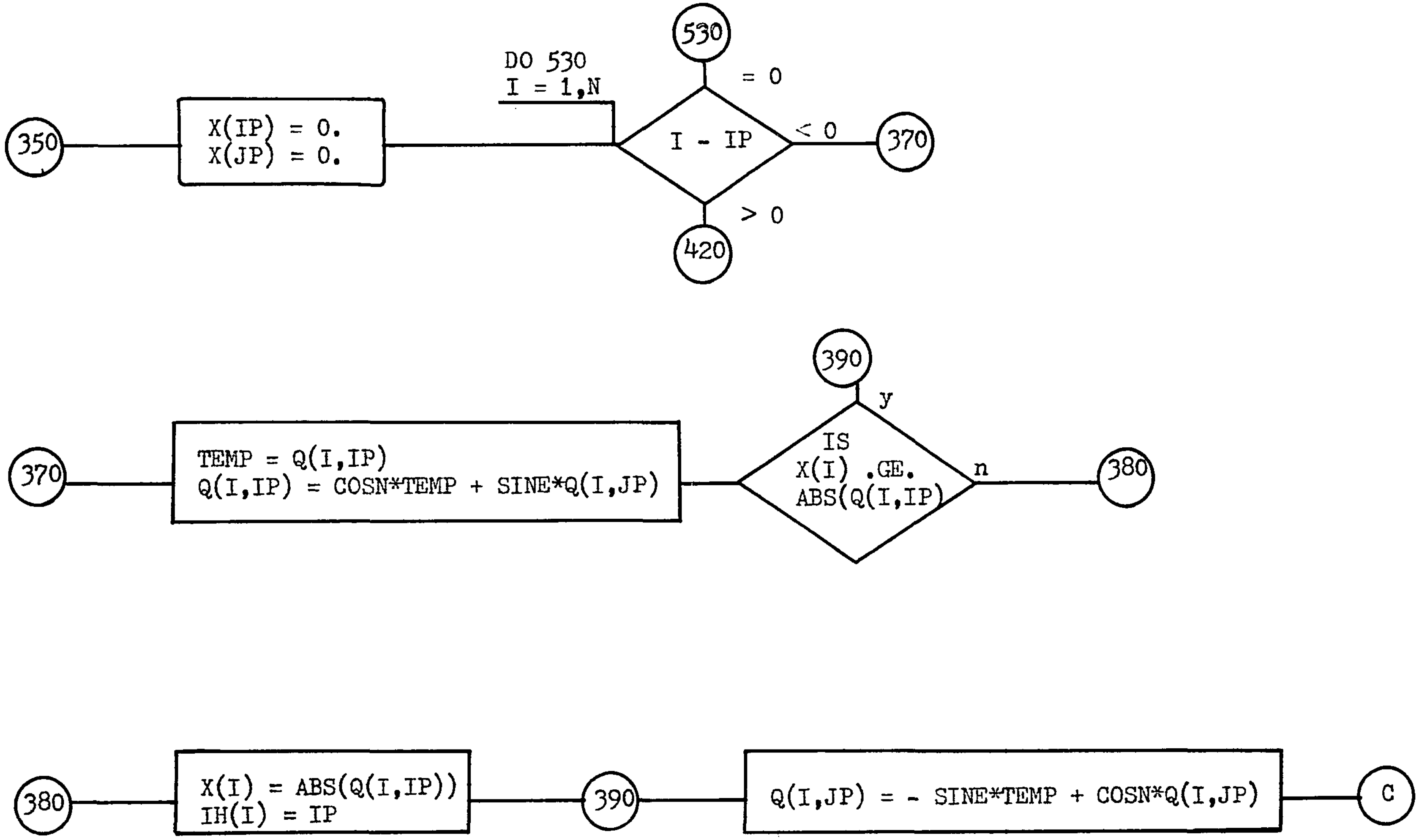


Fig. F-12g Flowchart of Subroutine JACOBI

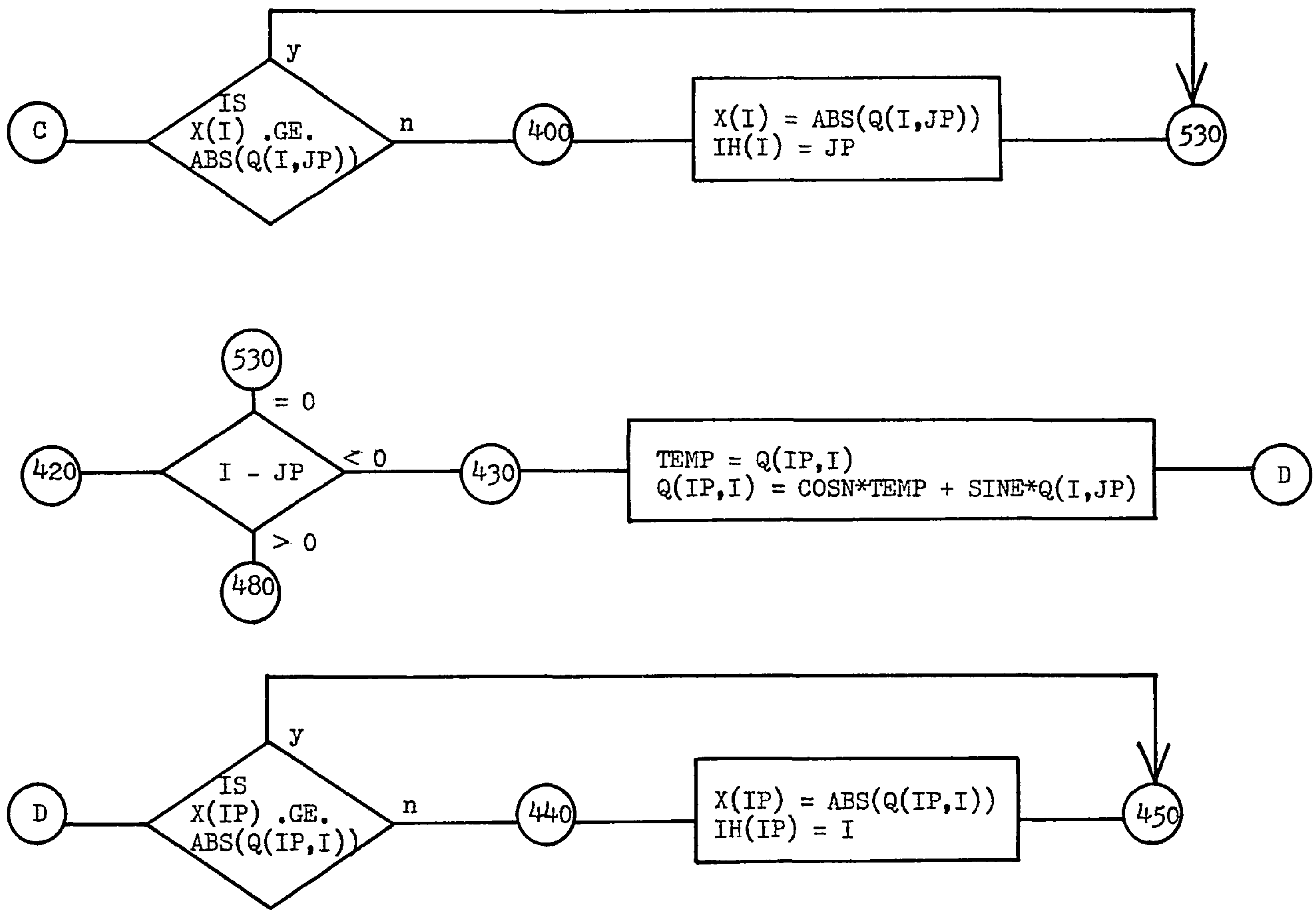


Fig. F-12h Flowchart of Subroutine JACOBI

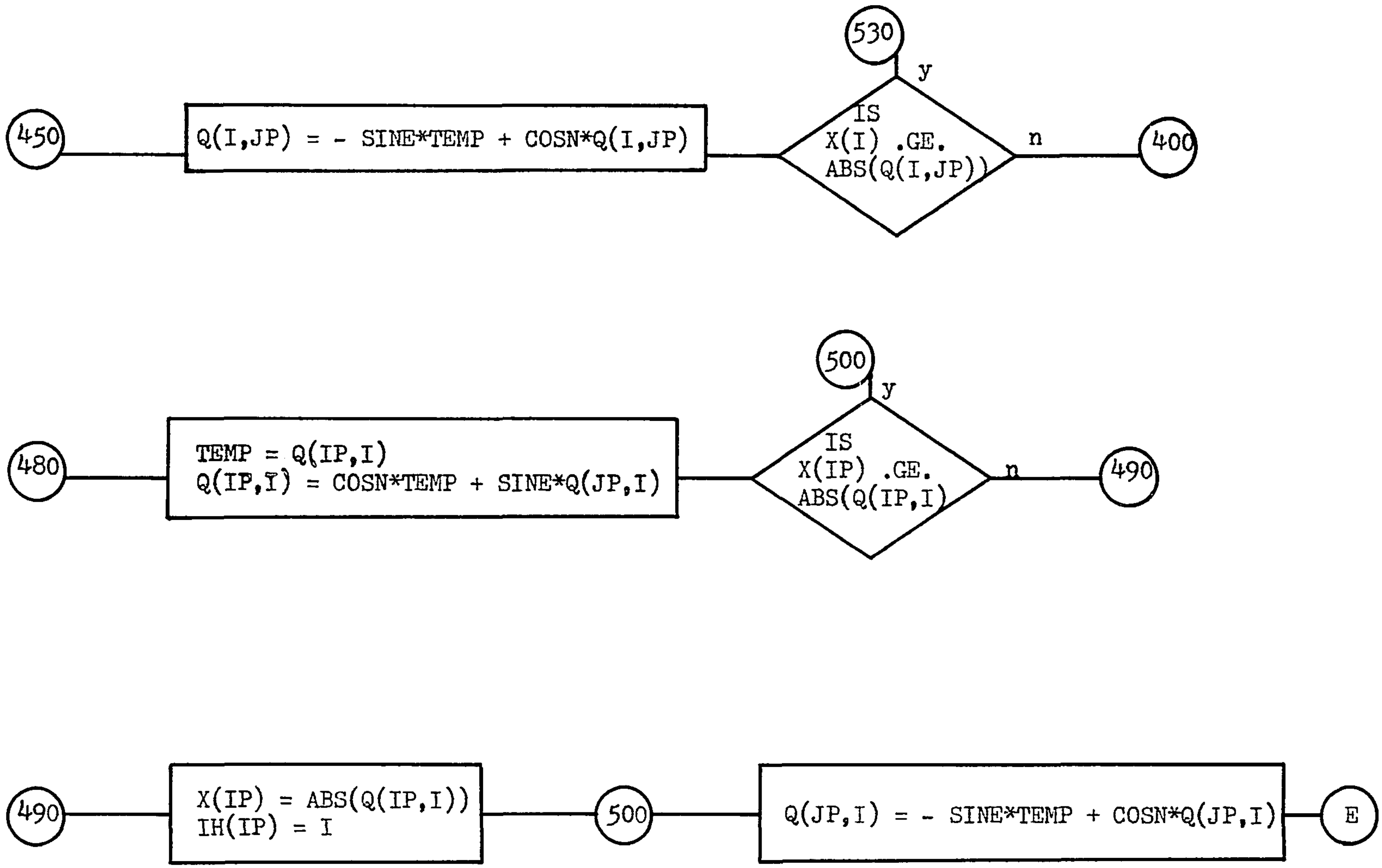


Fig. F-12i Flowchart of Subroutine JACOBI

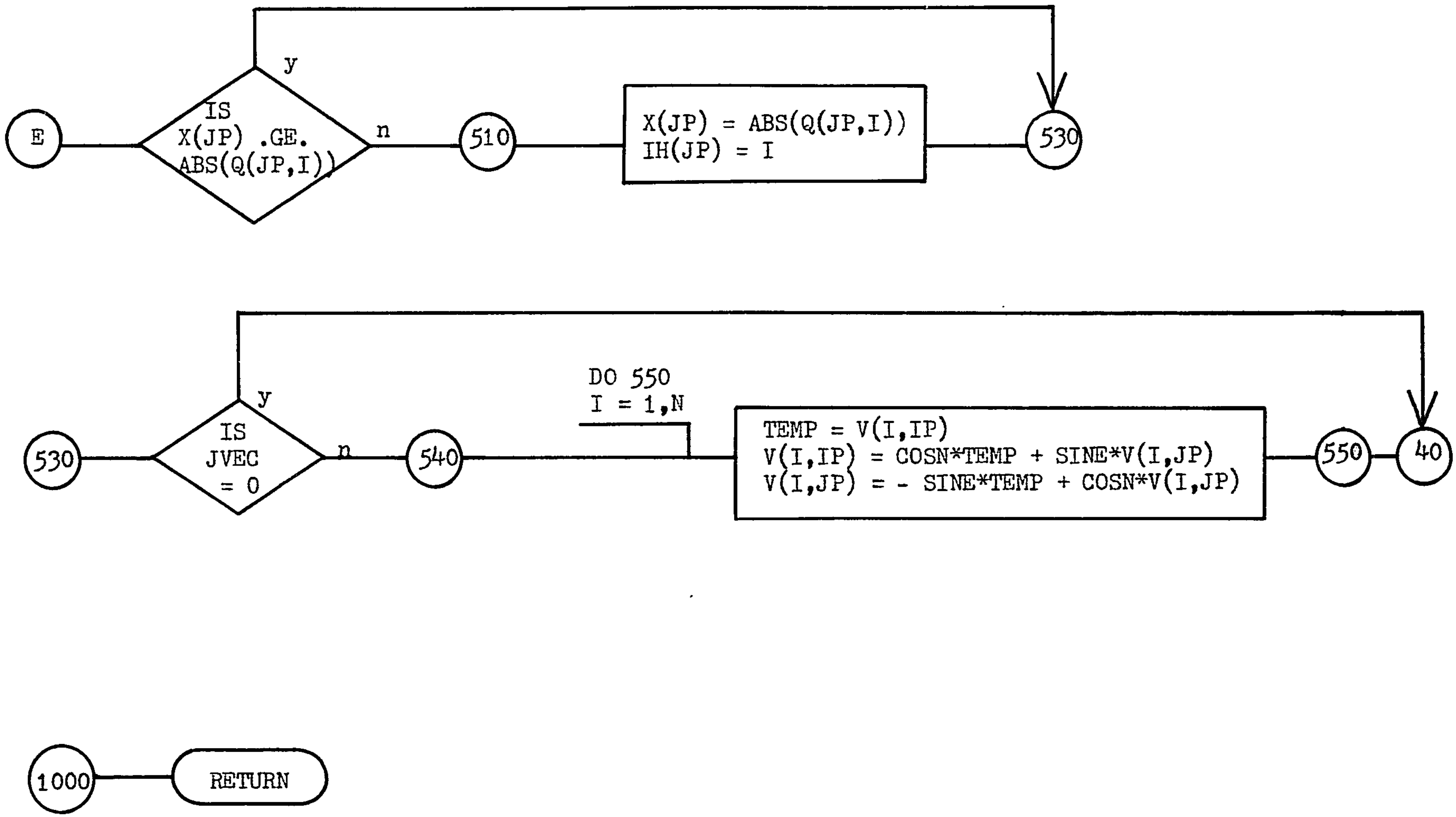


Fig. F-12j Flowchart of Subroutine JACOBI

CALL REBAKA(B,N)

in which B is the mass matrix which now contains the off-diagonal elements of [L] and N is the order of the system. A flowchart for this subroutine is given in Fig. F-13. It is not commented because details on this subroutine are already available [15].

1.14 DIS Subroutine

DIS solves the uncoupled matrix differential equation which results from forming the generalized mass and stiffness matrices. Nodal displacements, velocities, and accelerations are determined analytically in time. The solution can be obtained for under-damped, lightly damped, and heavily damped cases. The solution and its derivatives are found in terms of the generalized coordinates and are multiplied by the modal matrix to transform them to actual coordinates. If the program is solving a coupled solid-fluid problem, the coupling loads are removed from the load array so that the former may be updated in the next iteration. The old coupling loads are destroyed as they are no longer needed. If stresses and velocities are desired, provision for entering the STRESS subroutine is made at the end of the subroutine. The calling statement for DIS is

CALL DIS

Its flowchart is given in Fig. F-14a through F-14h .

1.15 MODMAK Subroutine

Subroutine MODMAK performs two functions. For solid-fluid coupling problems, it expands the solution and acceleration arrays to global form and inserts the specified freedoms into the

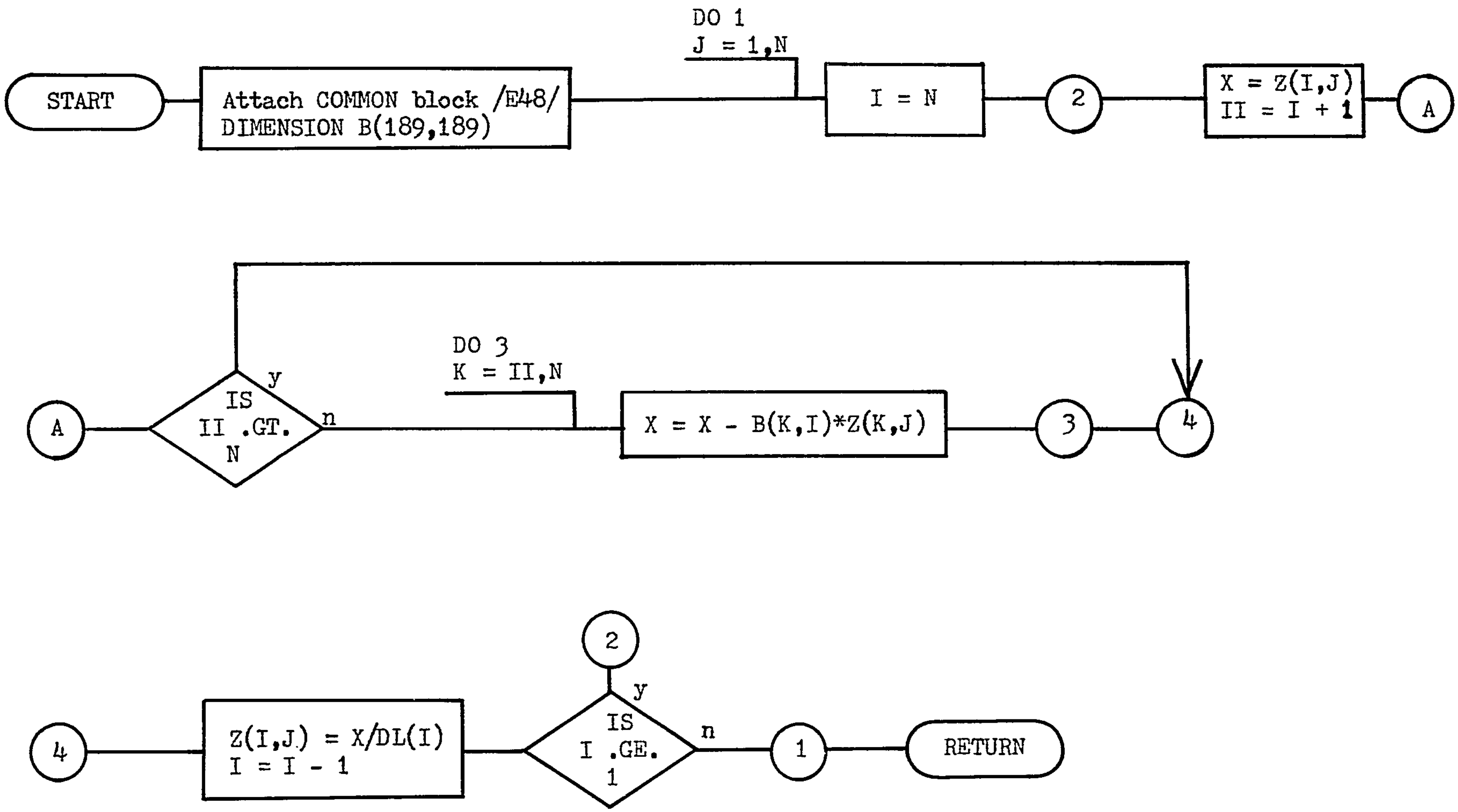


Fig. F-13 Flowchart of Subroutine REBAKA

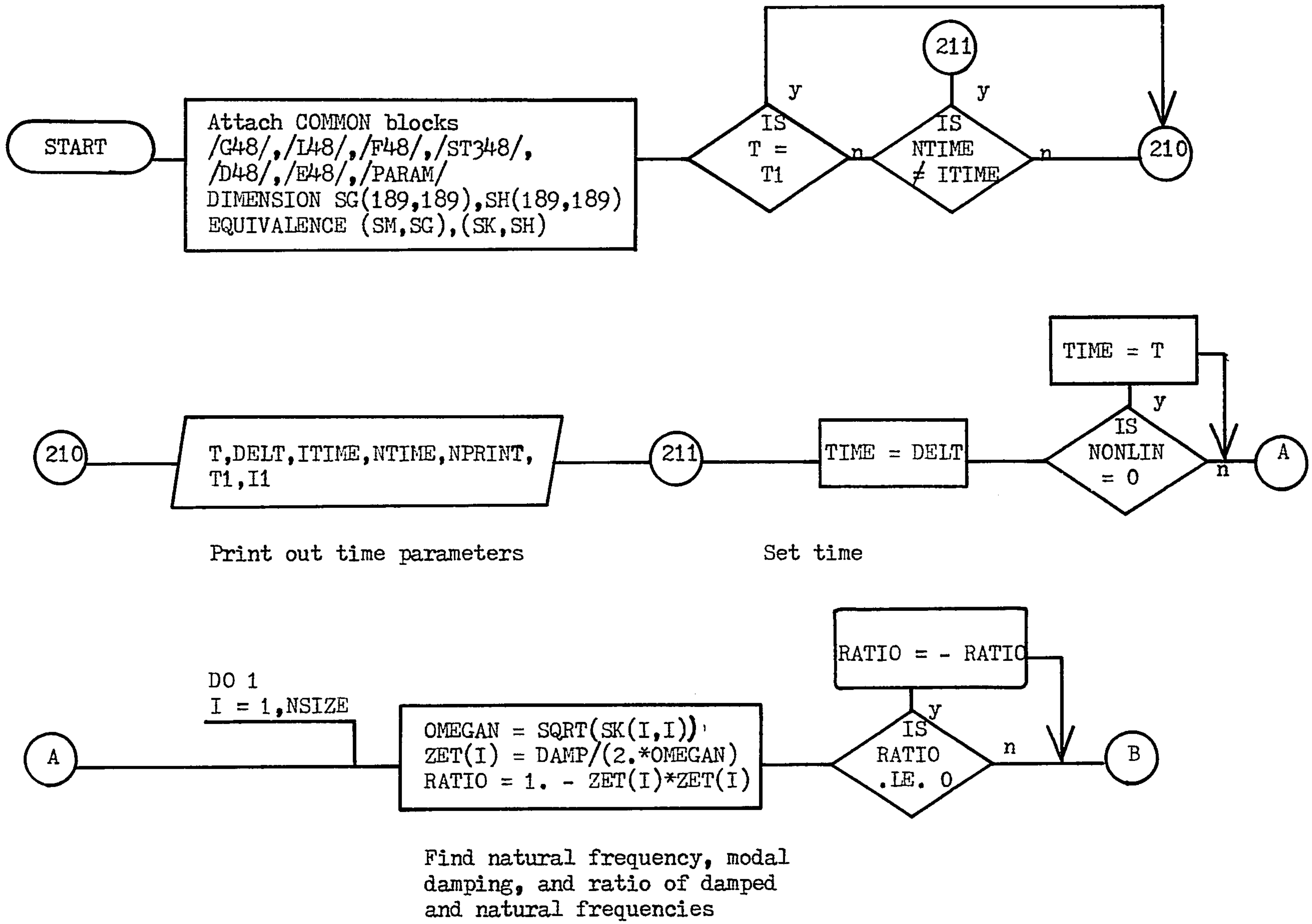


Fig. F-14a Flowchart of Subroutine DIS

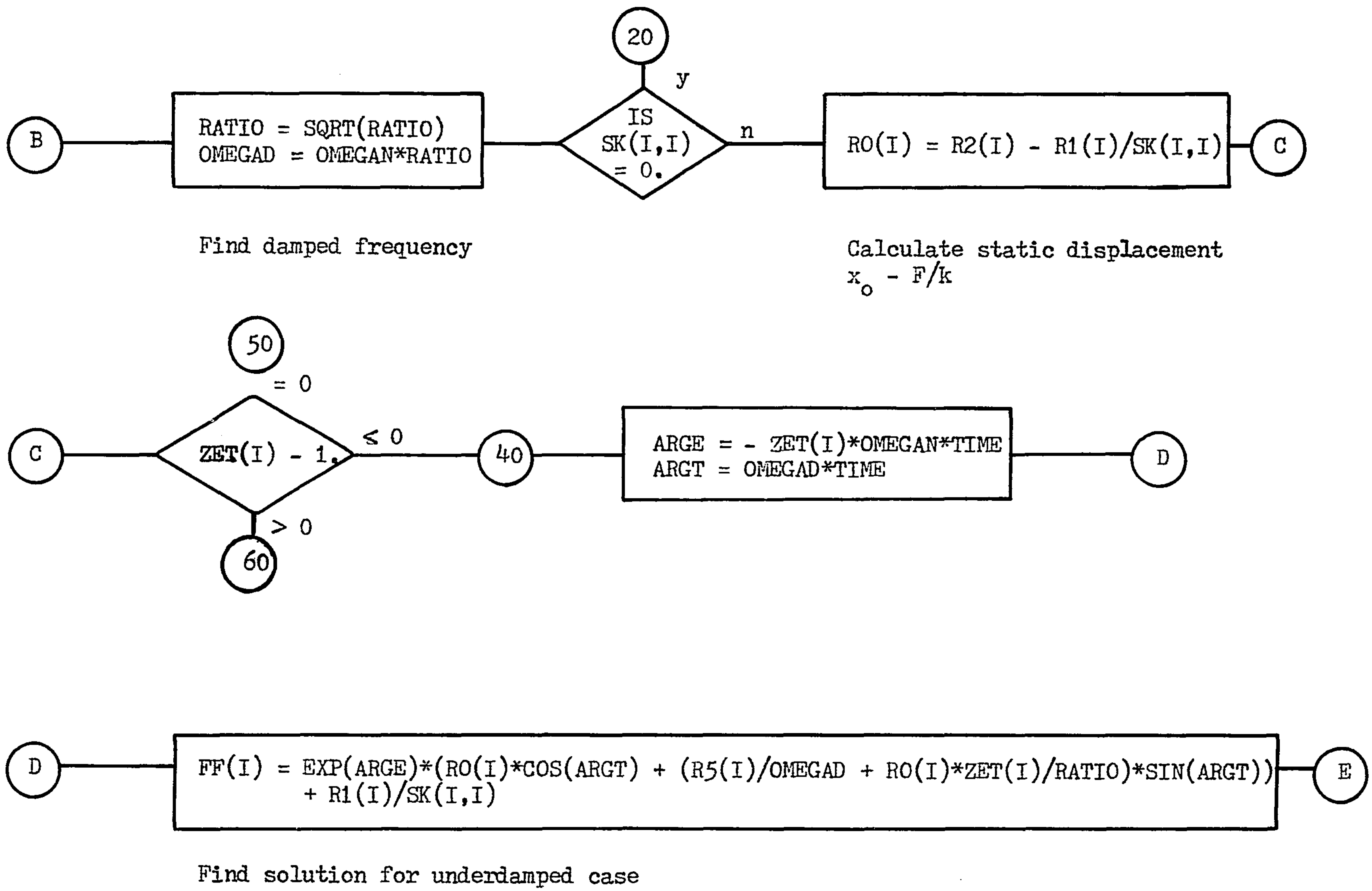


Fig. F-14b Flowchart of Subroutine DIS

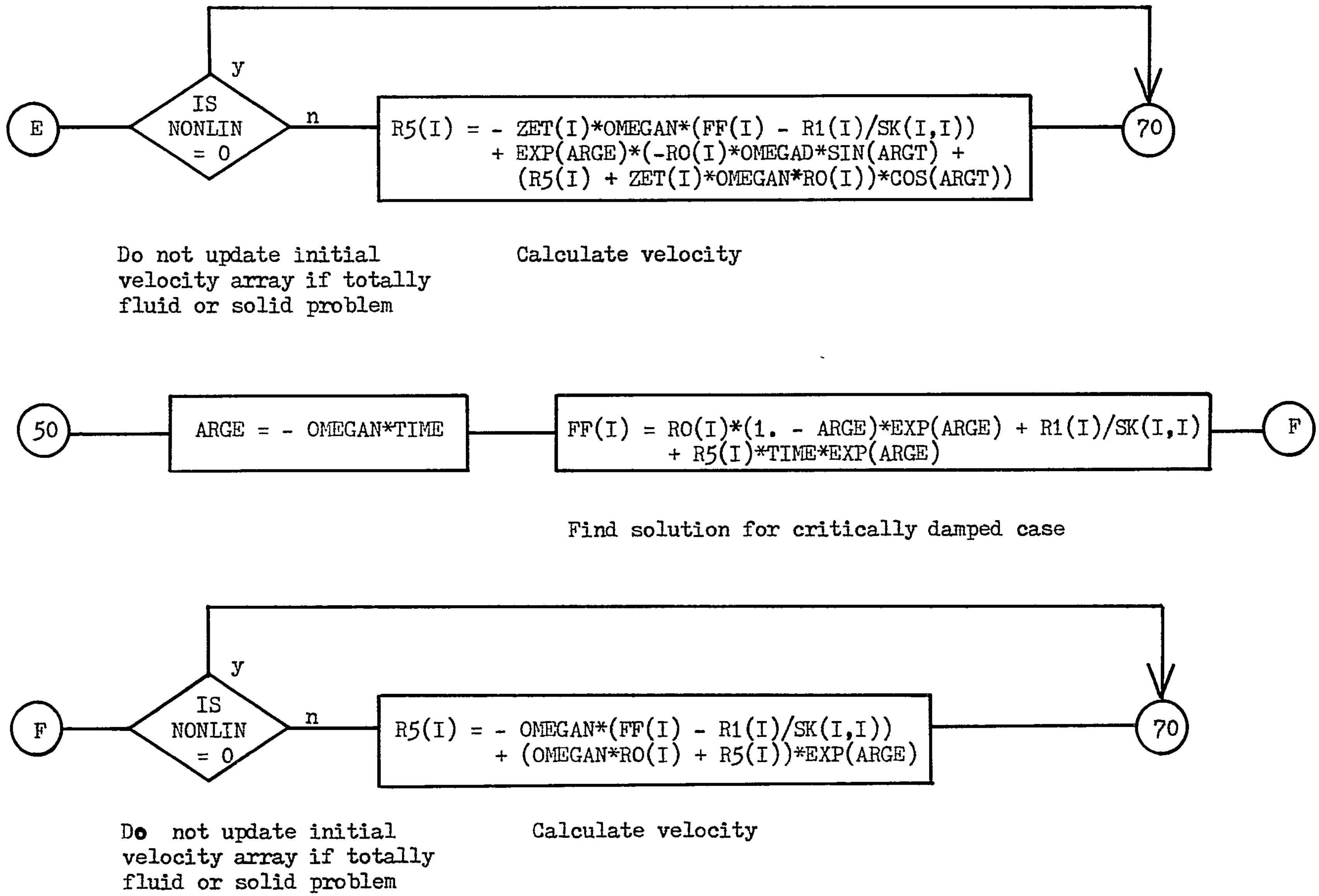


Fig. F-14c Flowchart of Subroutine DIS

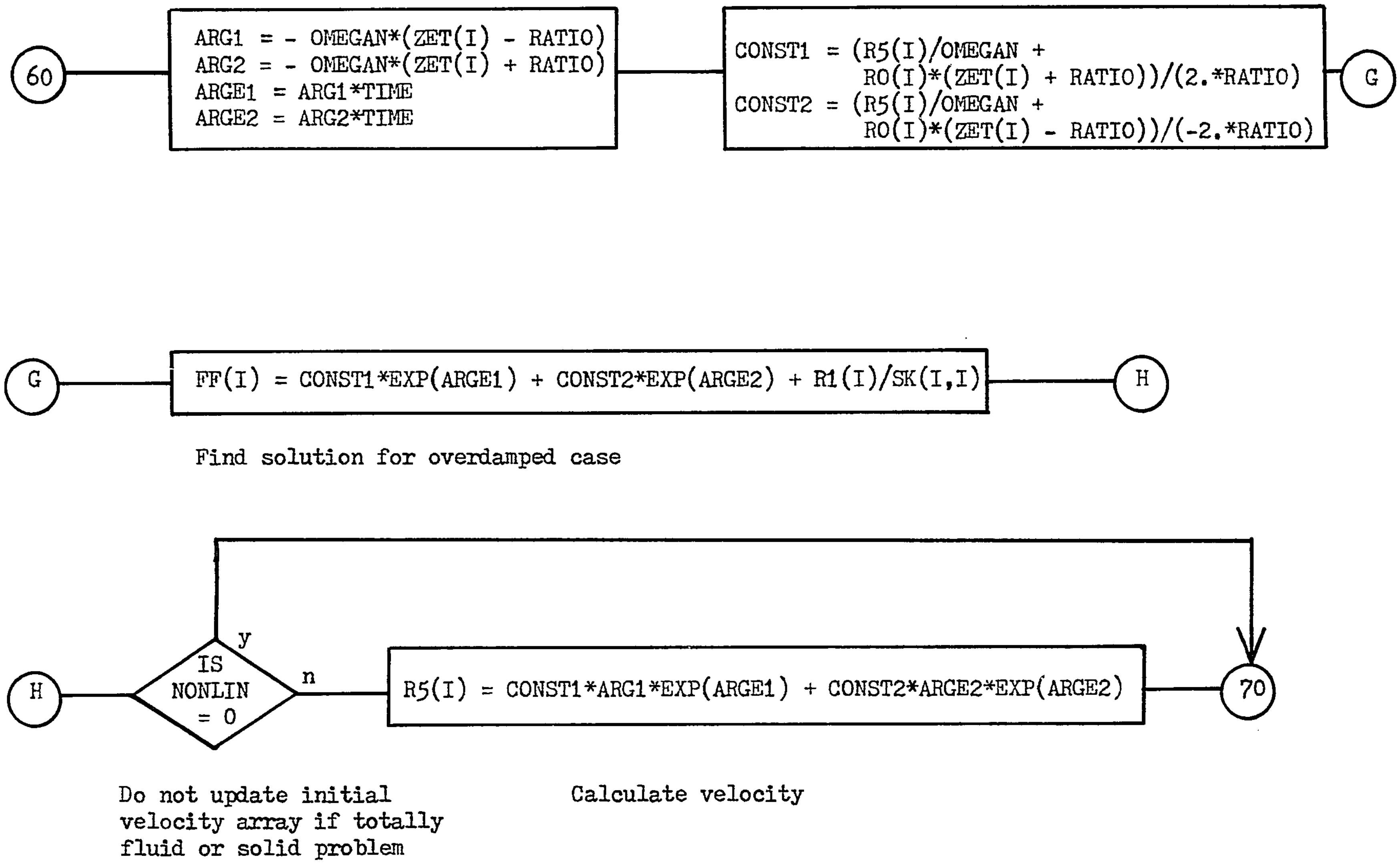


Fig. F-14d Flowchart of Subroutine DIS

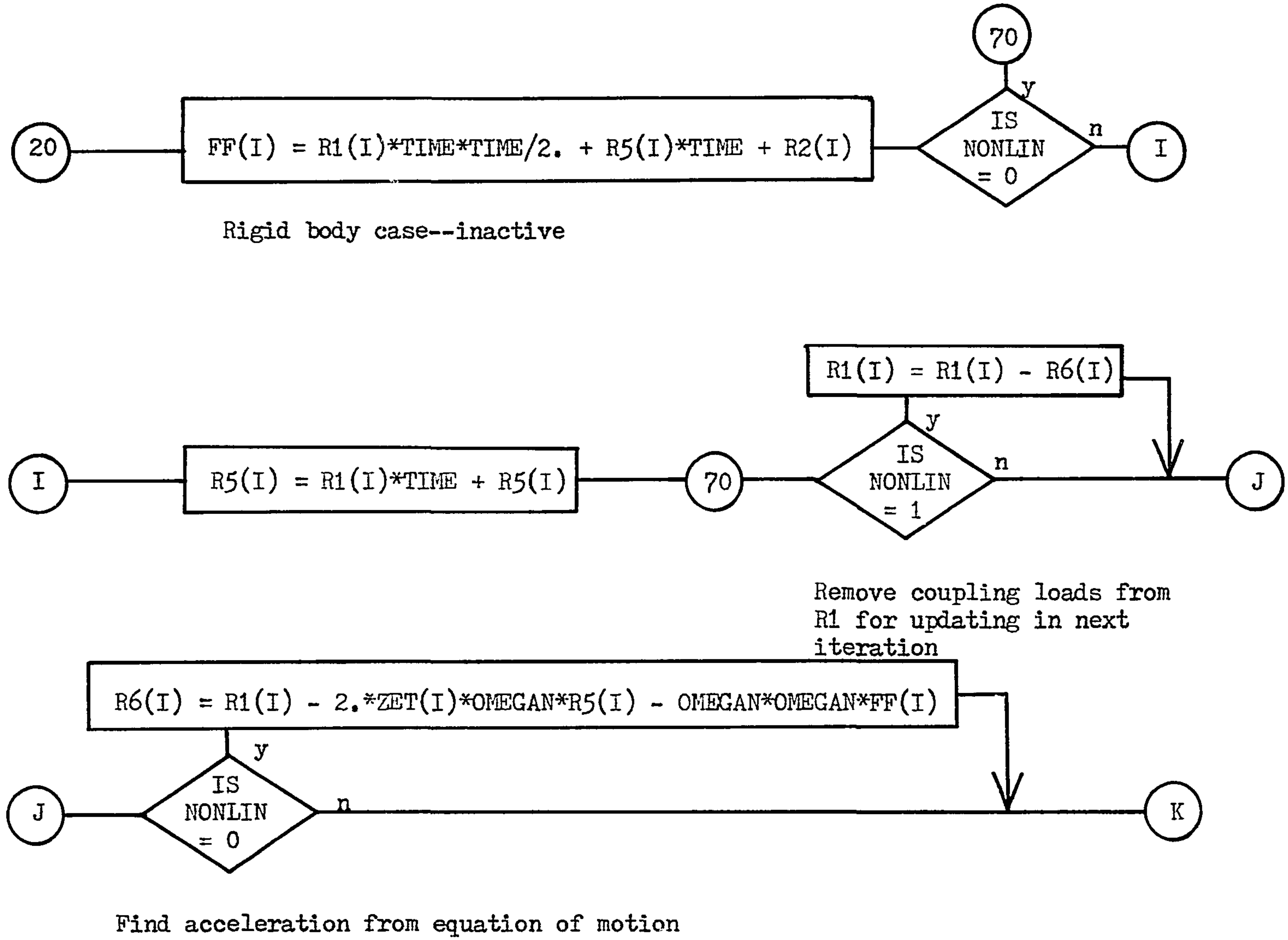


Fig. F-14e Flowchart of Subroutine DIS

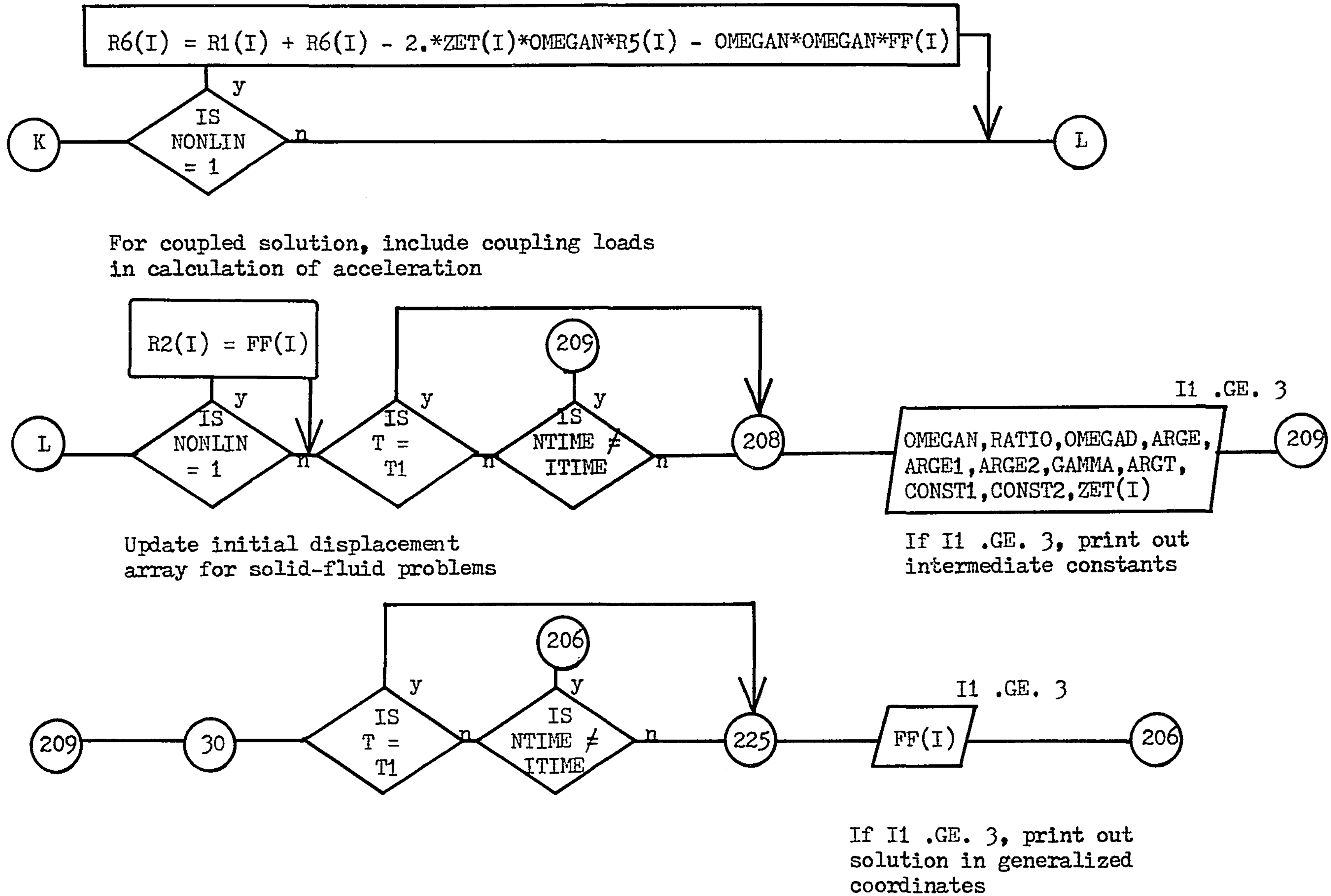


Fig. F-14f Flowchart of Subroutine DIS

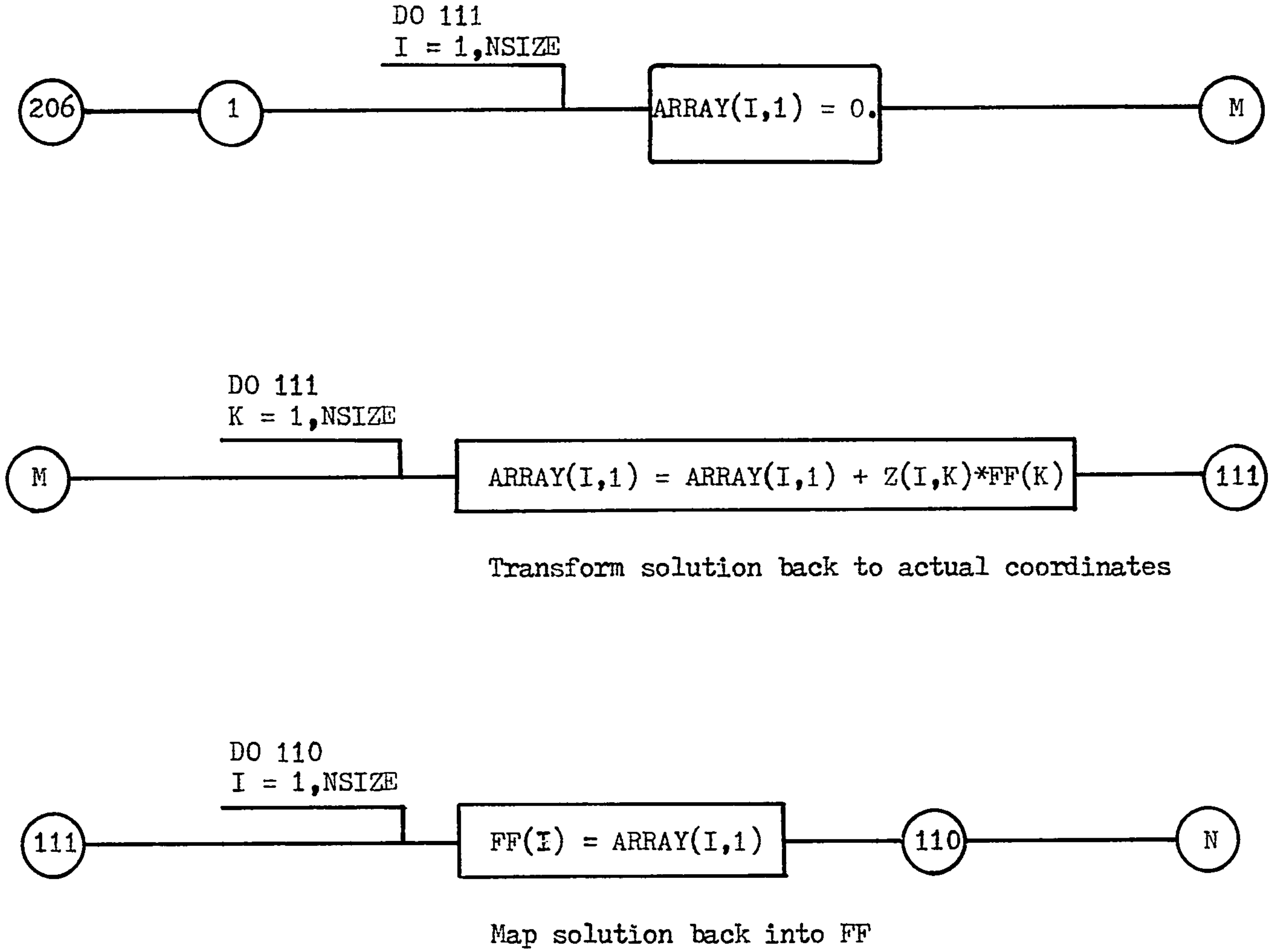


Fig. F-14g Flowchart of Subroutine DIS

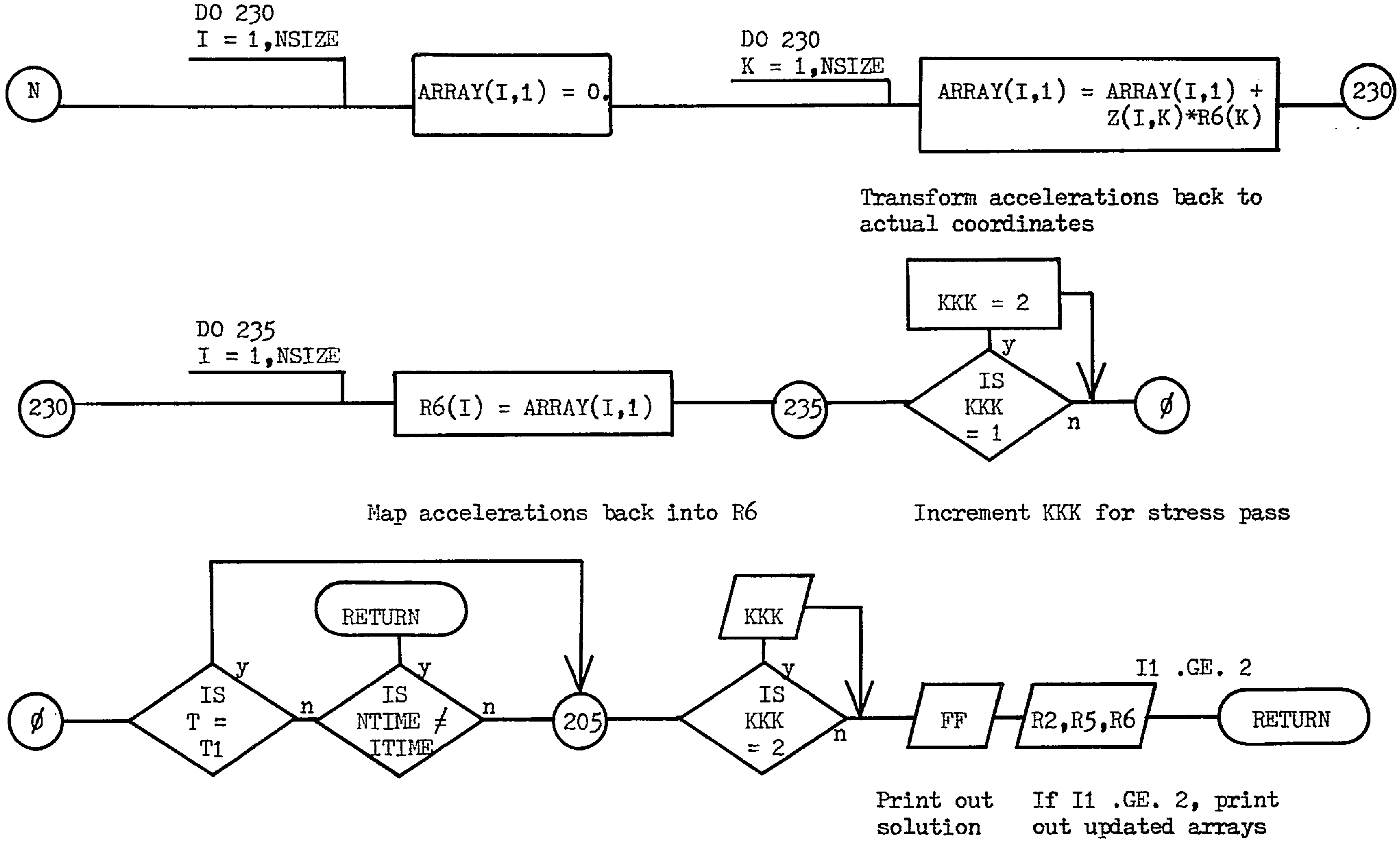


Fig. F-14h Flowchart of Subroutine DIS

proper locations in the solution array. Those expanded global arrays are then used in STIFT3 and STRESS to find the local arrays needed to calculate the solid-fluid interactive loads and the element stresses and velocities respectively. The second function is to add the nodal displacements to the coordinates of the solit nodes. This ultimately causes a modification in the solid mass and stiffness matrices and is used when the solid displacements are large. The calling sequence for this subroutine is

CALL MODMAK

The flowchart for MODMAK is given in Fig. F-15a through F-15c.

1.16 STRESS Subroutine

Subroutine STRESS controls the calling sequence of STIFT1, STIFT2, and STIFT3 to find the elemental stresses and velocities from the nodal displacements and pressures. At the end of STRESS, provision is made to enter the next iteration to calculate the nodal freedoms. The calling statement for this subroutine is

CALL STRESS

The flowchart for STRESS is given in Fig. F-16a through F-16b .

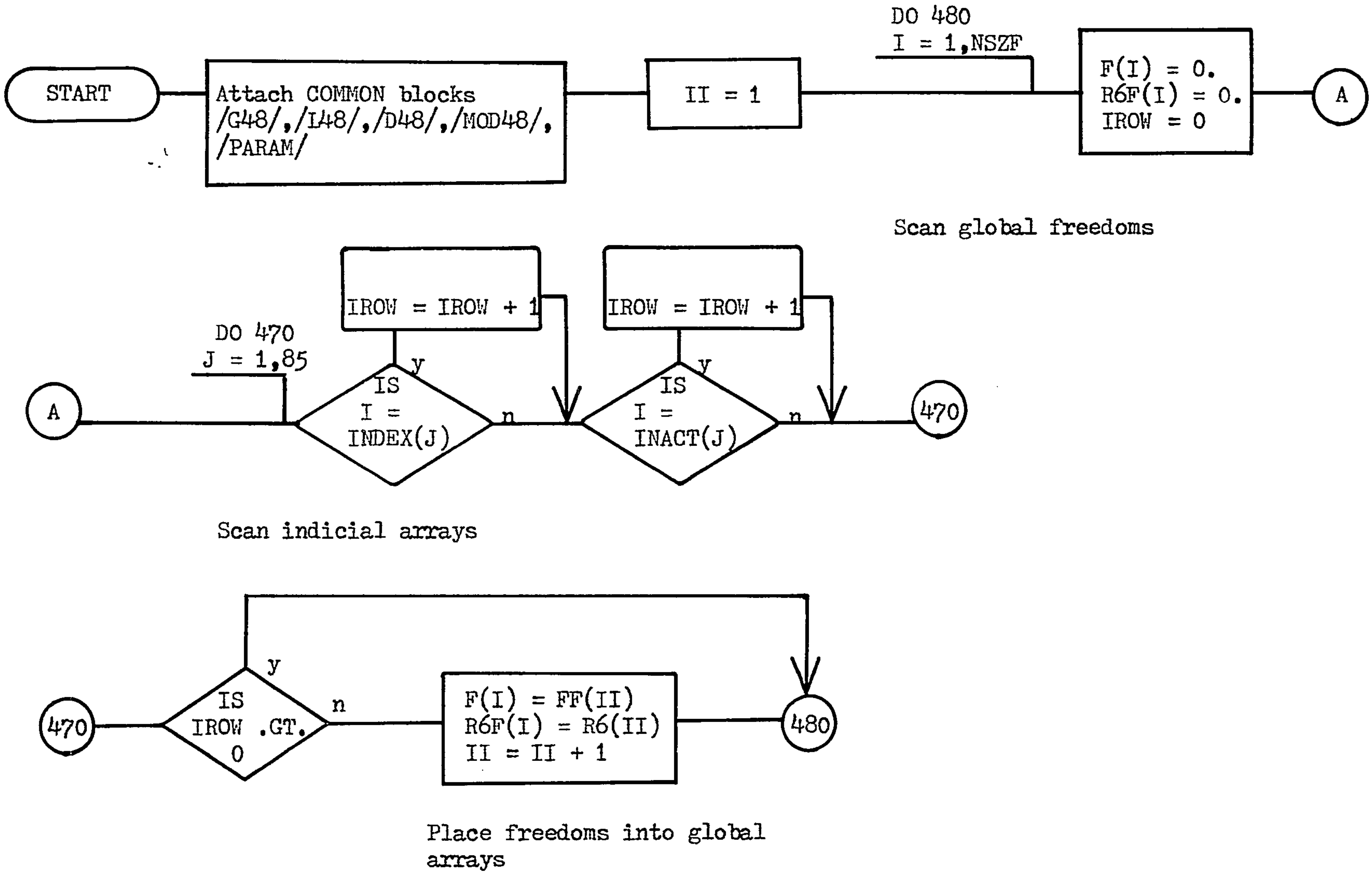


Fig. F-15a Flowchart of Subroutine MODMAK

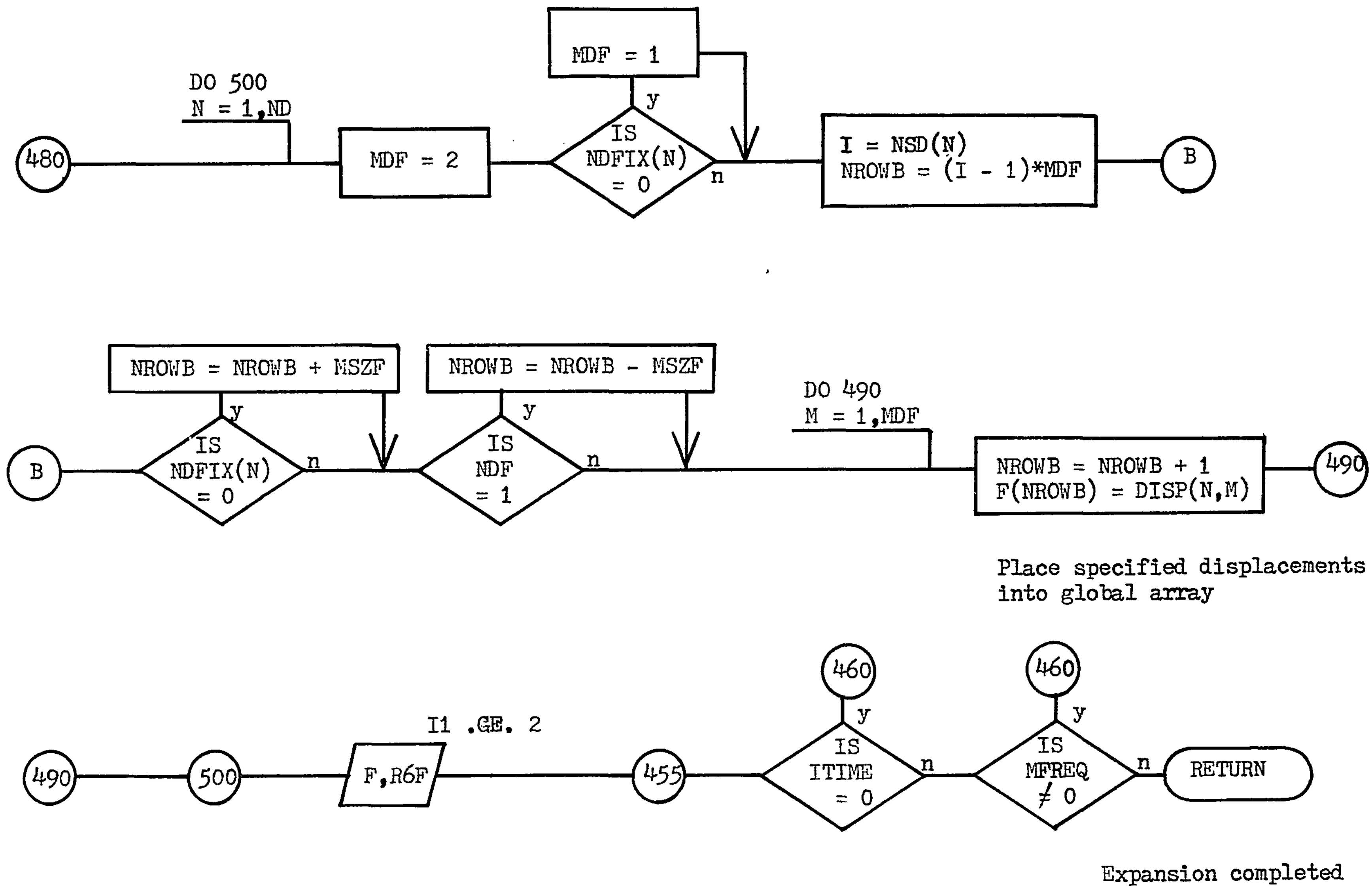


Fig. F-15b Flowchart of Subroutine MODMAK

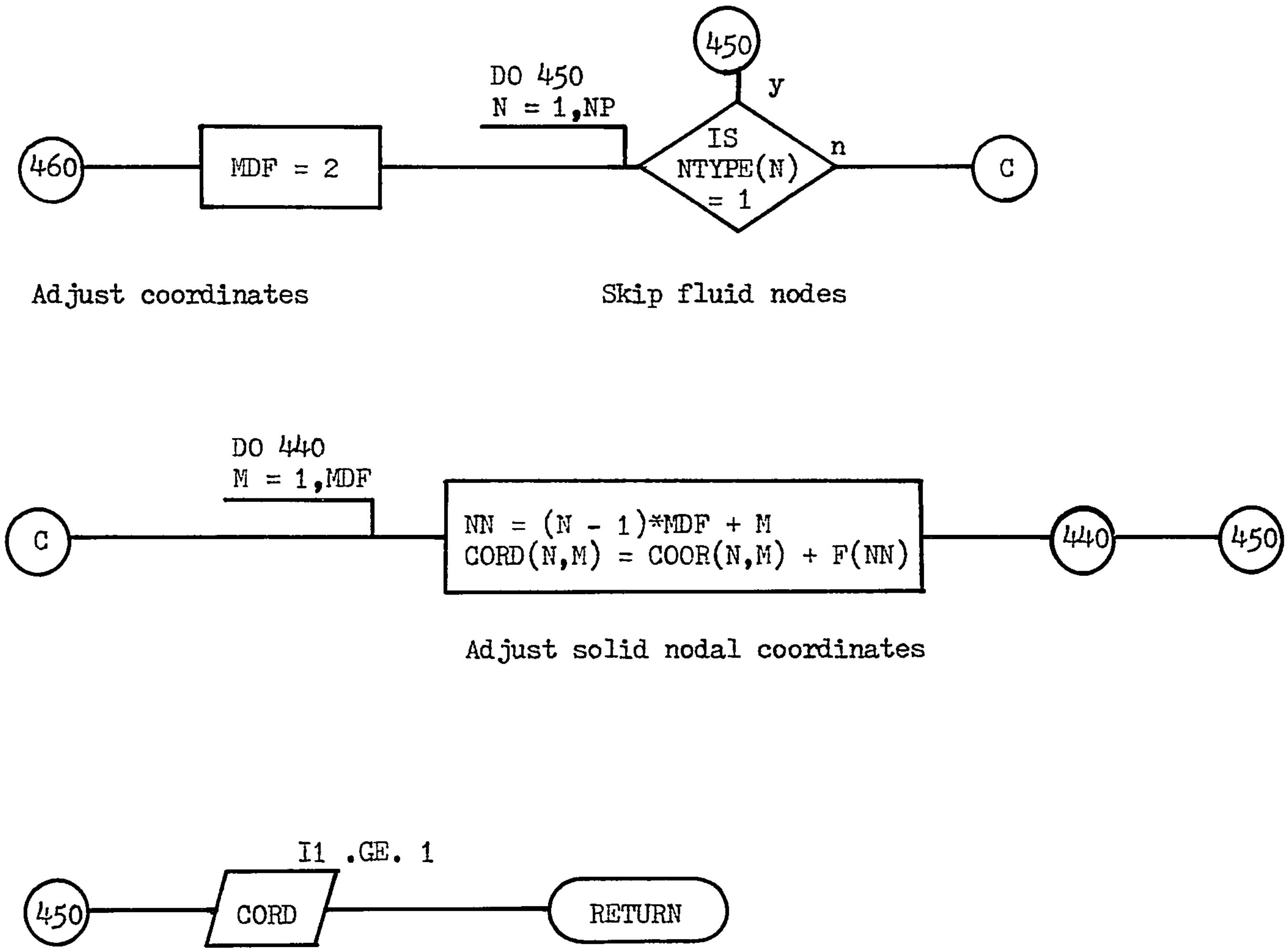


Fig. F-15c Flowchart of Subroutine MODMAK

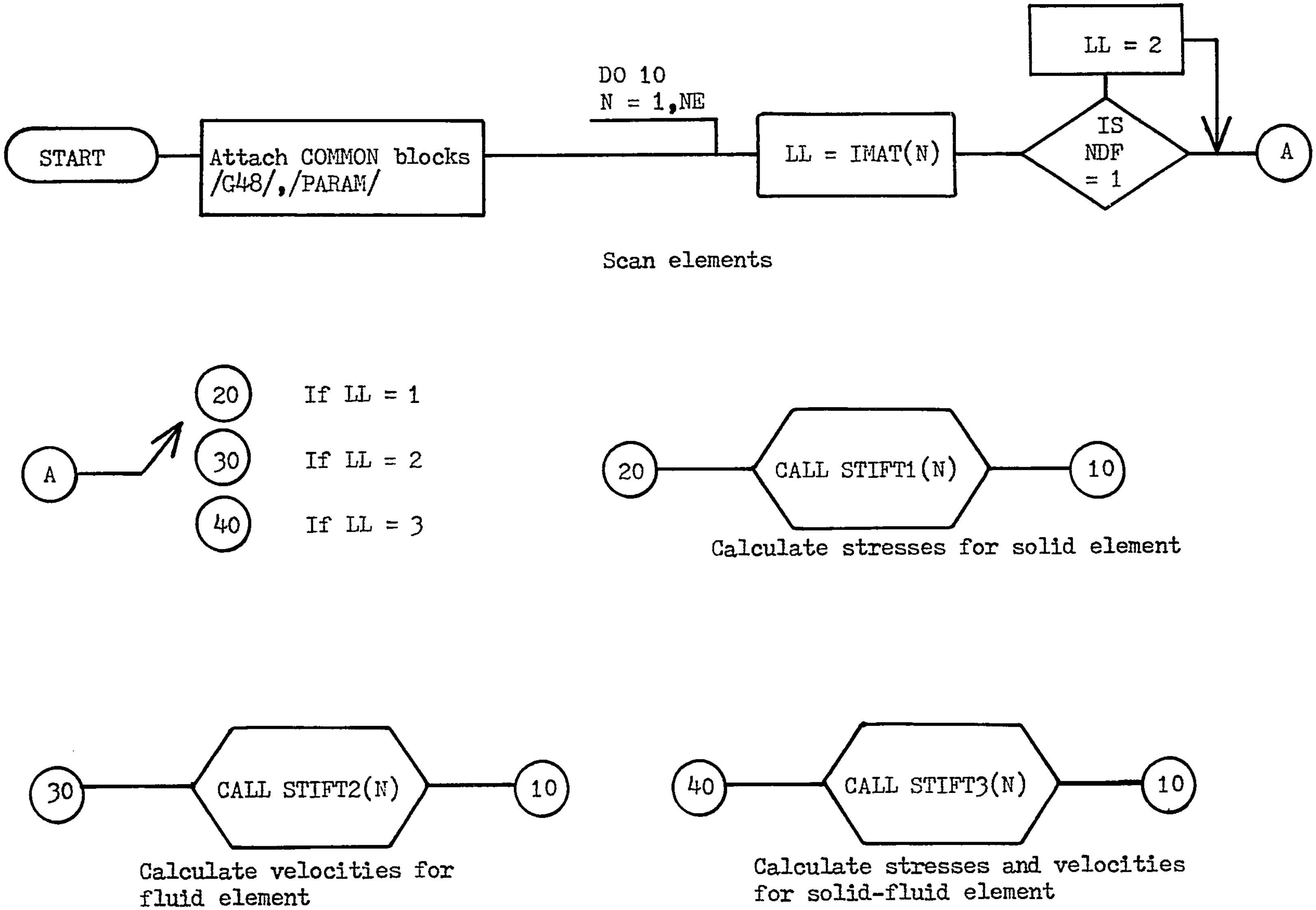


Fig. F-16a Flowchart of Subroutine STRESS

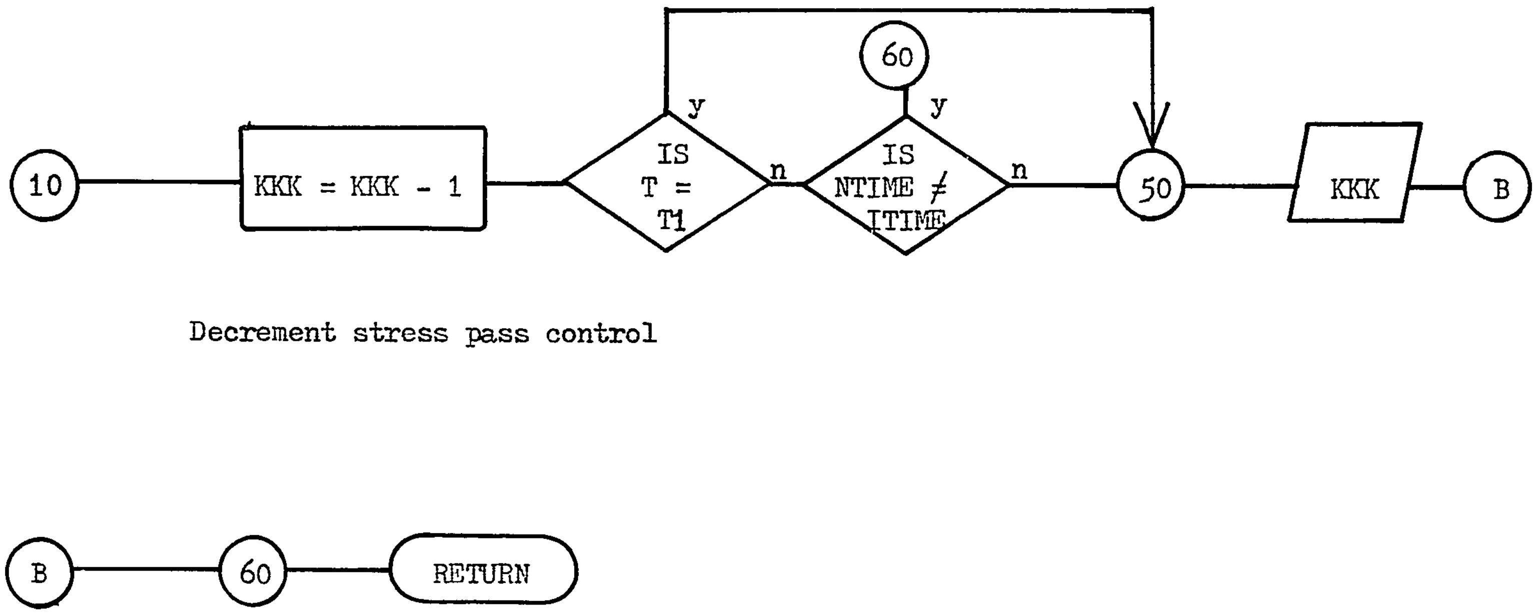


Fig. F-16b Flowchart of Subroutine STRESS

2 DESCRIPTION OF INPUT DATA

In this section, the input data for FLINTS is described in detail, followed by the printout of a sample data deck. The format for each card description is as follows:

Card number. Title of Card

Purpose of Card

Format

Columns	Variable	Comments

FLINTS Input Data

1. Title Card

   This card contains a 48-character descriptive title which is printed as a heading for the output data.

   Format (12A4)

Columns	Variable	Comments
1-48	TITLE	If no title is desired, insert a blank card.

2. System parameters card

   The data describing the system parameters such as the number of nodes, etc., are actually on two cards, 2a and 2b.

   Format (10I8) for both 2a and 2b.

Card 2a:

Columns	Variable	Comments
1-8	NP	Number of nodes; ≤ 63
9-16	NE	Number of elements; ≤ 88
17-24	NB	Number of nodes with applied loads; $1 \leq NB \leq 20$
25-32	--	Unused; leave blank.
33-40	NMAT	Number of different materials; ≤ 2
41-48	I1	Debug option $0 \leq I1 \leq 3$ = 0 yields solution only = 3 yields maximum debug printout.[1]
49-56	NPRINT	Frequency of printout; ≥ 1
57-64	ND	Number of nodes with specified displacements or pressures; $1 \leq ND \leq 20$

[1]It is suggested, in view of the voluminous amount of data resulting when I1 = 3, that the user limit the duration of his runs when using a high debug option ($I1 \geq 2$).

Columns	Variable	Comments
65-72	NDF	Number of nodal degrees of freedom of the system = 1 for fluid system = 2 for solid system = 3 for solid-fluid system[2]
73-80	KKK	Stress pass control option = 0 for pressures and displacements only = 1 for pressures, displacements, velocities, and stresses.
Card 2b:		
1-8	--	Unused--leave blank.
9-18	--	Unused--leave blank.
19-24	MFREQ	Frequency at which mass and stiffness matrices are updated for large solid displacements.

[2]Care must be taken that the product NP x NDF, which represents the total global number of degrees of freedom, does not exceed 189. The total number of active dynamic freedoms cannot exceed 90.

Columns	Variable	Comments
25-32	NONLIN	Solid-fluid indicator
		= 0 for fluid or solid problem
		= 1 for solid-fluid problem

3. Time Card

This card reads in the parameters associated with the time history history duration, as well as other system parameters.

Format (I10, 2F10.5, I10, 3F10.5, I10)

Columns	Variable	Comments
1-10	NIT	Number of iterations.
11-20	TBEG	Initial time of time history.
21-30	TEND	Final time of time history.
31-40	--	Unused--leave blank.
41-50	UO	Initial fluid velocity in axial (x) direction.
51-60	VO	Initial fluid velocity in transversal (y) direction.
61-70	ALPHA	Angle solid-fluid boundary makes with x-axis, measured counter-clockwise, in radians.

Columns	Variable	Comments
71-80	KMASS	Mass option = 0 for lumped mass analysis = 1 for distributed mass representation.

4. Material Properties Card

This card reads in the material properties for the number of materials specified in NMAT. One card is needed for each material.

Format (I10,3E12.5)

Columns	Variable	Comments
1-10	N	Material type number; ≤ 2
11-22	ORT(N,1)	= E Young's modules for solid = c Speed of sound for fluid
23-34	ORT(N,2)	= ν Poisson's ratio for solid Blank for fluid
35-46	ORT(N,3)	= ρ Density for solid and fluid

5. Damping Parameter Card

This card reads in the value of proportional damping based on mass which is to be used in the problem.

Format (6E12.5)

Columns	Variable	Comments
1-12	DAMP	Enter 0 if system is undamped

6. Nodal point data card

This card reads in the x and y coordinates of each nodal point. There must be 1 card for every node.

Format (I10, 2F10.3)

Columns	Variable	Comments
1-10	N	Node number
21-30	CORD(N,1)	x-coordinate of node
31-40	CORD(N,2)	y-coordinate of node

7. Nodal point freedom card

This card has the number of degrees of freedom which exist at each node. There must be 1 card for every node.

Format (2I5)

Columns	Variable	Comments
1-5	N	Node number.
6-10	NTYPE(N)	Number of degrees of freedom at node N.

8. Element connections card

This card contains the node number associated with each element and the element type number (see Card 4) referring to the material properties. Element connections should be numbered in a consistent counter-clockwise fashion. Solid-fluid elements are numbered listing the solid nodes first, and then the fluid nodes.

Format (8I5)

Columns	Variable	Comments
1-5	N	Element number.
6-10	NOP(N,1)	Nodes of fluid
11-15	NOP(N,2)	or solid finite
16-20	NOP(N,3)	element or solid nodes
		of solid-fluid finite element.
		element.
21-25	NOP(N,4)	Fluid nodes of
26-30	NOP(N,5)	solid-fluid finite
31-35	NOP(N,6)	element; blank for fluid
		and solid elements.
36-40	IMAT(N)	Material type number for
		finite element N. For totally
		fluid or solid problem, IMAT(N) = 1 .
		For interactive problem,
		= 1 for solid element
		= 2 for fluid element
		= 3 for solid-fluid element

9. Applied Load Type Card

This card contains the nodes at which applied loads act and the type of load which acts there. Even if no applied loads exist, the logic of the program requires that a dummy load of zero be applied at some arbitrary node. One card is needed for each applied load.

Format (2I5)

Columns	Variable	Comments
1-5	NBC	Node at which specified load acts.
6-10	NFIX	Code indication of direction of load
		= 01 load is in y-direction
		= 10 load is in x-direction
		= 11 load is in x and y directions
		= 0 dummy load

10. Specified Displacement Type Card

This card is quite similar to Card 9. It contains the nodes at which specified displacement or pressures act and the type of specified displacement which exists there. At least one specified displacement must be given in order to restrain the system and remove rigid body modes. If a specified displacement and a specified pressure exist at a node, that node must be indicated twice, once for the specified displacements and once for the pressures.

Format (2I5)

Columns	Variable	Comments
1-5	NSD	Node at which specified displacement or pressure acts.
6-10	NDFIX	Code indication of direction of specified displacement. = 0 pressure is specified = 01 displacement in y-direction is given = 10 displacement in x-direction is given = 11 displacement in x and y directions is given.

11. Load Card

This card reads in the load information for the current time step.

Format (2F10.4,I10)

Columns	Variable	Comments
1-10	PMAX	Magnitude of load.
11-20	---	Unused--leave blank
21-30	NLD	Load type code = 3 for constant loads Other values of NLD should be used in a future extension of the program to treat time-varying loads.

12. Specified Displacement Card

This card contains the specified displacements. One card is needed for each set of specified displacements and pressures.

Format (2F10.4)

Columns	Variable	Comments
1-10	DISP(N,1)	Specified pressure or specified displacement in x-direction.
11-20	DISP(N,2)	Specified displacement in y-direction.

```
*******************************************************************************
**************      THIS IS A PRINTOUT OF A FLINTS DATA DECK      ****************
*******************************************************************************


72 ELEMENT FLUID SOLID MODEL
        54       72        1        3        2        0        1       14        3        1
                                    1
          4        0.   .00001             0        0.        0.        0.        0
          1   .3E+08         .3E+00   .0007297
          2   .6E+05                9.35521E-05
           0.
          1       .6        0.
          2       .6        4.
          3       .6        8.
          4       .6       12.
          5       .6       12.5
          6       .6       13.
          7      2.7        0.
          8      2.7        4.
          9      2.7        8.
         10      2.7       12.
         11      2.7       12.5
         12      2.7       13.
         13      6.9        0.
         14      6.9        4.
         15      6.9        8.
         16      6.9       12.
         17      6.9       12.5
         18      6.9       13.
         19     11.1        0.
         20     11.1        4.
         21     11.1        8.
         22     11.1       12.
         23     11.1       12.5
         24     11.1       13.
         25     15.3        0.
         26     15.3        4.
         27     15.3        8.
         28     15.3       12.
         29     15.3       12.5
         30     15.3       13.
         31     19.5        0.
         32     19.5        4.
         33     19.5        8.
         34     19.5       12.
         35     19.5       12.5
         36     19.5       13.
         37     23.7        0.
         38     23.7        4.
         39     23.7        8.
         40     23.7       12.
         41     23.7       12.5
         42     23.7       13.
         43     27.9        0.
         44     27.9        4.
         45     27.9        8.
```

	45	27.9	12.
	47	27.9	12.5
	48	27.9	13.
	49	30.	0.
	50	30.	4.
	51	30.	8.
	52	30.	12.
	53	30.	12.5
	54	30.	13.
1	1		
2	1		
3	1		
4	3		
5	2		
6	2		
7	1		
8	1		
9	1		
10	3		
11	2		
12	2		
13	1		
14	1		
15	1		
16	3		
17	2		
18	2		
19	1		
20	1		
21	1		
22	3		
23	2		
24	2		
25	1		
26	1		
27	1		
28	3		
29	2		
30	2		
31	1		
32	1		
33	1		
34	3		
35	2		
36	2		
37	1		
38	1		
39	1		
40	3		
41	2		
42	2		
43	1		
44	1		
45	1		
46	3		
47	2		
44	2		
49	1		
50	1		

51	1						
52	3						
53	2						
54	2						
1	1	7	8				2
2	1	8	2				2
3	2	8	9				2
4	2	9	3				2
5	3	9	10				2
6	4	10	11	3	10	4	3
7	4	11	5				1
8	5	11	12				1
9	5	12	6				1
10	7	13	14				2
11	7	14	8				2
12	8	14	15				2
13	8	15	9				2
14	9	15	16				2
15	10	16	17	9	16	10	3
16	10	17	11				1
17	11	17	18				1
18	11	18	12				1
19	13	19	20				2
20	13	20	14				2
21	14	20	21				2
22	14	21	15				2
23	15	21	22				2
24	16	22	23	15	22	16	3
25	16	23	17				1
26	17	23	24				1
27	17	24	18				1
28	19	25	26				2
29	19	26	20				2
30	20	26	27				2
31	20	27	21				2
32	21	27	23				2
33	22	28	29	21	28	22	3
34	22	29	23				1
35	23	29	30				1
36	23	30	24				1
37	25	31	32				2
38	25	32	26				2
39	26	32	33				2
40	26	33	27				2
41	27	33	34				2
42	28	34	35	27	34	28	3
43	28	35	29				1
44	29	35	36				1
45	29	36	31				1
46	31	37	38				2
47	31	35	32				2
48	32	38	39				2
49	32	39	33				2
50	33	39	40				2
51	34	40	41	33	40	34	3
52	34	41	35				1
53	35	41	42				1
54	35	42	36				1
55	37	43	44				2

```
 56   37   44   38                  2
 57   38   44   45                  2
 58   38   45   39                  2
 59   39   45   46                  2
 60   40   45   47   39   46   40   3
 61   40   47   41                  1
 62   41   47   48                  1
 63   41   48   42                  1
 64   43   49   50                  2
 65   43   50   44                  2
 66   44   50   51                  2
 67   44   51   45                  2
 68   45   51   52                  2
 69   46   52   53   45   52   46   3
 70   45   53   47                  1
 71   47   53   54                  1
 72   47   54   48                  1
  1    0
  1    0
  2    0
  3    0
  4    0
  4   11
  5   11
  6   11
 49    0
 50    0
 51    0
 52    0
 52   11
 53   11
 54   11
       0.        0.        3
      10.
      10.
      10.
      10.
       0.        0.
       0.        0.
       0.        0.
       0.
       0.
       0.
       0.
       0.        0.
       0.        0.
       0.        0.

******************************************************************************
*************************     END OF SAMPLE DECK     *************************
******************************************************************************
```

3 DESCRIPTION OF OUTPUT DATA

The output data are classified as follows:

3.1 Output data not under debug option control

The output data which are not under the control of the debug option I1 (i.e., for I1 = 0) are as follows:

I1 = 0. The following is printed out when I1 $\geq$ 0 :

Subroutine	Output
GDATA	Geometric input data.
LOAD	Global parameters such as size of global matrices, etc.; Load input data; Local load array; Global load array; Specified displacements and pressures; Indicial arrays, sorted global load arrays.
FORMK	Size of condensed matrices; Reduced mass and stiffness matrices and load arrays.
STIFT1	Elemental stresses.
STIFT2	Elemental fluid velocities.
STIFT3	Elemental stresses and fluid velocities.
MODAL	Eigenvectors, eigenvalues; eigenvectors transformed back to global coordinates.

Subroutine	Output
DIS	Time parameters such as elapsed time, iteration number, etc.; stress pass control; solution in terms of global coordinates.
STRESS	Stress pass control.

All the above data are printed out with suitable headings.

3.2 Output data under the control of the debug option

This includes the aforementioned data and the following:

I1 ≥ 1. The following is printed out when I1 ≥ 1:

Subroutine	Output
FORMK	Global matrices and load array as adjusted for specified displacements.
STIFT1	Elemental strain-displacement matrix; elemental stress-strain matrix.
STIFT3	Elemental strain-displacement matrix; elemental stress-strain matrix.
MODMAK	Adjusted nodal coordinates

I1 ≥ 2. The following is printed out when I1 ≥ 2:

Subroutine	Output
FORMK	Unrestrained global mass and stiffness matrices and coupling load array; coupling array adjusted for specified displacements and inactive freedoms; Matrices used in EQUIVALENCE statement.

I1 $\geq$ 2 (continued)

Subroutine	Output
STIFT1	Local coordinates.
STIFT2	Local coordinates; fluid accelerations, proportional damping factor.
STIFT3	Local coordinates of fluid triangle; fluid accelerations, proportional damping factor; local coordinates of solid triangle.
MODAL	Generalized interactive and applied forces.
DIS	Updated initial displacement, velocity, and acceleration arrays.
MODMAK	Global displacement and acceleration arrays.

I1 $\geq$ 3. The following is printed out when I1 $\geq$ 3:

Subroutine	Output
FORMK	Fixed nodal coordinate array.
STIFT1	Element connections; elemental mass and stiffness matrices; area of finite element, etc.
STIFT2	Element connections; elemental inertia and fluidity matrices; area of finite element, etc.

I1 $\geq$ 3 (continued)

Subroutine	Output
STIFT3	Fluid element connections; area of fluid portion; elemental inertia and fluidity matrices; solid element connections; elemental mass and stiffness matrices; area of solid portion; coupling matrix; elemental coupling force on solid; elemental coupling force on fluid, etc.
MODAL	Modified matrices after calling REDUC1; L matrix; mass matrix formed by $[L]\,[L]^T$; matrix to convert initial displacements and velocities to generalized form; generalized initial displacements; generalized initial velocities.
DIS	Intermediate parameters such as natural frequency, damped frequency, etc.; solution in generalized coordinates.

All of the above data are printed out with suitable headings.

A working knowledge of the program is necessary to perform effective debugging runs.

3.3 Error Printouts

The output data which compose the error printout are as follows:

Subroutine	Output
STIFT1	Number of element with bad connections -- run is terminated.
STIFT2	Number of element with bad connections - run is terminated.
STIFT3	Number of element with bad connections -- run is terminated.
REDUC1	Error message if mass matrix is not positive definite.

The error messages for the bad connections indicate in which element the error lies. The error message for the matrix error is simply a warning and does not terminate the run.

3.4 Output Data Defining Problem Solution

a) Displacement Pass

All unrestrained nodal freedoms are printed as the problem solution. The program first prints out the heading "Finite Element Modal Superposition Solution." The elapsed time, time increment, iteration number, printout iteration number, frequency of printout, time at first iteration, and debug option are then printed. If the program is about to enter the stress pass, the stress pass control is then printed. The heading "Modal Solution For Unrestrained Nodes" is then printed, followed by the unrestrained displacements (u_{xi}, u_{yi}), and the unrestrained pressures p_i sequentially in column form.

b) Stress Pass

The elemental stresses and fluid velocities are printed after the solution. The heading for the velocities is "Elemental Velocities at Iteration No." followed by the iteration number. The heading for the stresses is "Elemental Stresses at Iteration No." followed by the iteration number. The elemental stresses and velocities are then printed out for every element in the model.

4 OPERATING PROCEDURE

It is recommended that the user become familiar with the mathematical formulation of the finite element model and its numerical solution, presented in Part 1, as well as the organization of the program described in Part 2, before attempting to solve a solid-fluid interaction problem using FLINTS.

In developing the finite element mesh, the user should number the nodes in an orderly fashion and take care to number the element connections in a consistently counterclockwise manner. Failure to do so would result in a bad connection error message printout and abnormal termination of the run. Solid-fluid elements are numbered starting with the solid nodes first and then the fluid nodes. Six nodes are therefore used to define the solid-fluid quadrilateral element. The input data may then be punched on cards according to the format presented in Part 2, Section 2, Description of Input Data.
The data cards are then used in conjunction with the FLINTS source deck and necessary job cards in the following manner

```
//jobname   JOB   (ho8177,1650),'name'
//*MAIN   F = R, T = 10, L = 50, G = E
//  EXEC   FGRFCLG, PARM.FORT = BCD,
         RGNG = 448K
//FORT.SYSIN    DD   *
         FLINTS source deck
/*
//GO.SYSIN   DD  *
         FLINTS data deck
/*
```

The above job cards are applicable to an IBM 370/360 system. No matter which computer is being used, the user should bear in mind that the runs made for the completion of this study required 448K bytes of storage and approximately 10 minutes to execute on the IBM 370/360. A maximum of 50,000 lines was found to be more than adequate.

The following procedure is recommended for effective use of the FLINTS program

1) Input data check run

   The user should make a run with the beginning time and final time of the run set equal to zero. The debug option should also be set equal to zero. The program will then print out all of the input data and all other problem data, such as eigenvectors and eigenvalues, up to the problem solution. This enables the user to check the input data and the intermediate results for correctness. If the input data is incorrect, this run should be repeated until all user errors have been rectified. If the intermediate results appear incorrect, the debug option may be increased and this run repeated to provide more debug printout to help the user determine the nature of the difficulty.

2) Test run

   The user may, after the data has been debugged, perform a dynamic run using a time step on the basis of Section 4 in Part 1.

3) Finite element mesh convergence run

   The convergence of the finite element mesh is determined by making a run having a finer grid (more elements) than the desired grid. Convergence is achieved when there is no appreciable difference between the runs.

4) Time step convergence run

The convergence of the time step is determined by making a run having a time step that is twice the desired time step. Convergence is achieved when there is no appreciable difference between the runs. If there is a discrepancy, the time step should be halved and this prodecure repeated until convergence is established.

5) Solution run

Once convergence of the finite element mesh and time step are assured, all the runs necessary to obtain the desired solution may be made.

5 FLINTS NOMENCLATURE

The variables used in FLINTS are listed, defined, and cross-referenced with the analysis notation, to aid the user in understanding the program.

5.1 Subscripted Variables

Program Notation	Analysis Notation	Subroutine	Description
A(3,6)	B	STIFT1,STIFT3	Strain-displacement matrix, in^{-1}.
AK(9,9)	D, K_e, H_e	STIFT1,STIFT2, STIFT3	In STIFT1 and STIFT3, the elasticity matrix, psi and later the stiffness matrix. In STIFT2, the fluidity matrix, in.
AN(3),AM(3)	a	STIFT2,STIFT3	Local area coordinates of fluid and solid triangles, respectively, in^2.
BACK(3,6)	-	STIFT1,STIFT3	Stress back substitution matrix, psi/in.
BN(3),BM(3)	b	STIFT1,STIFT2, STIFT3	Local y-coordinates of fluid and solid triangles, respectively, in
CN(3),CM(3)	c	STIFT1,STIFT2, STIFT3	Local x-coordinates of fluid and solid triangles, respectively, in.
COOR(63,2)	-	FORMK,MODMAK	Fixed array of nodal coordinates, in.
CORD(63,2)	-	GDATA	Array of nodal coordinates, in.
D(90)	ω_n^2	MODAL	Array of eigenvalues, $(radians/sec)^2$.
DISP(20,3)	-	LOAD	Matrix of specified displacements (u_x and u_y), in, and pressures (p), psi.

Program Notation	Analysis Notation	Subroutine	Description
DL(90)	-	REDUC1,REBAKA	Array of diagonal elements of L matrix.
F(189)	-	MODMAK,STIFT1, STIFT2,STIFT3	Array of expanded global freedoms.
FF(90)	x	DIS,STIFT1, STIFT2,STIFT3	Array of active freedoms.
IMAT(88)	-	GDATA,FORMK	Array of element material type numbers.
INACT(85)	-	LOAD	Indicial array of rows which contain inactive freedoms.
INDEX(85)	-	LOAD	Indicial array of rows at which specified displacements act.
LINDX(85)	-	LOAD	Indicial array of row numbers at which pressure freedoms act.
NACT(85)	-	LOAD	Indicial array of active row numbers in sequence.
NBC(20)	-	GDATA,LOAD	Array of nodes at which specified loads act.
NDFIX(20)	-	GDATA,LOAD	Array of code indicators of direction of specified displacements.
NFIX(20)	-	GDATA,LOAD	Array of code indicators of direction of applied loads.
NOP(88,6)	-	GDATA,STIFT1, STIFT2,STIFT3	Matrix of element connections, see Section 2.
NSD(20)	-	GDATA,LOAD	Array of nodes at which specified displacements act.
NTYPE(63)	-	GDATA,FORMK	Array of nodal degrees of freedom, see Section 2.
ORT(2,3)	-	GDATA,STIFT1, STIFT2,STIFT3	Matrix of material properties, see Section 2.

Program Notation	Analysis Notation	Subroutine	Description
R(9)	R_e'	LOAD,STIFT3	Local load array; elemental coupling load array.
R0(189)	q_o-F/k	DIS	Static displacement array.
R1(189)	F	FORMK,MODAL, DIS	Generalized force array.
R2(189)	q_o	MODAL,DIS	Array of initial displacements.
R4(189)	R	LOAD,FORMK	Array of applied loads.
R5(189)	$\dot{q}_o$	MODAL,DIS	Array of initial velocities.
R6(189)	$R', \ddot{x}$	FORMK,MODAL, DIS,MODMAK	Array of global coupling loads; later, array of accelerations.
R6F(189)	-	MODMAK,STIFT3	Expanded global acceleration array.
S(3,6)	S_e	STIFT3	Elemental solid-fluid coupling matrix, lbf-sec^2/in^2
SIGMA(3)	σ	STIFT1,STIFT2, STIFT3	Elemental stress or velocity array.
SK(189,189) SH(189,189)	$\overline{K}$	FORMK,MODAL, DIS	Global stiffness matrix. SK is equivalenced with SH.
SM(189,189) SG(189,189)	$\overline{M}$	FORMK,MODAL, DIS	Global mass matrix. SM is equivalenced with SG.
UI(88)	v_x	STIFT2,STIFT3	Array of past fluid velocities in x-direction
VI(88)	v_y	STIFT2,STIFT3	Array of past fluid velocities in y-direction.
XM(9,9)	M_e	STIFT1,STIFT2, STIFT3	In STIFT1 and STIFT3, elemental mass matrix, lbf-sec^2/in; in STIFT2, elemental inertia matrix, in-sec^2.
Z(90,90)	ϕ	MODAL,DIS	Modal matrix
ZET(189)	ζ	DIS	Array of modal damping ratios.

5.2 Nonsubscripted Variables

Program Notation	Analysis Notation	Subroutine	Description
ALPHA	α	STIFT3	Angle of solid-fluid boundary with longitudinal x axis, radians.
AREAN,AREAM	A_f,A_s	STIFT1,STIFT2, STIFT3	Area of fluid and solid triangles, in^2
BB	$\frac{\partial p}{\partial x}$	STIFT2,STIFT3	Velocity differential in x-direction, psi/in.
CC	$\frac{\partial p}{\partial y}$	STIFT2,STIFT3	Velocity differential in y-direction, psi/in.
DAMP	α	STIFT2,STIFT3, DIS	Damping factor based on mass, sec^{-1}.
DELT	Δt	FORMK	Time step, sec.
ITIME	-	GDATA	Iteration number.
I1	-	GDATA	Debug option.
KKK	-	GDATA	Stress pass control.
KMASS	-	GDATA	Mass option.
MFREQ	-	MAIN,MODMAK	Frequency of matrix updates for large solid displacements.
NB	-	GDATA,LOAD	Number of nodes with applied loads.
ND	-	GDATA,LOAD	Number of nodes with specified displacements.
NDF	-	GDATA,LOAD	Maximum number of nodal degrees of freedom.
NE	-	GDATA,FORMK, STRESS	Number of elements.
NIT	-	GDATA	Number of iterations.
NLD	-	GDATA,LOAD	Load type.

Program Notation	Analysis Notation	Subroutine	Description
NMAT	-	GDATA	Number of material types.
NONLIN	-	GDATA,FORMK,MODAL, DIS,MODMAK	Solid-fluid indicator.
NP	-	GDATA,LOAD	Number of nodes.
NPRINT	-	GDATA	Frequency of printout.
NSIZE	-	FORMK,MODAL, DIS,MODMAK	Size of reduced matrices
NSZF	-	LOAD,FORMK	Size of unconstrained global matrices.
NTIME	-	FORMK	Printout control parameter.
T	t	FORMK	Elapsed problem time, sec.
TBEG	-	GDATA	Time at beginning of time history.
TEND	-	GDATA	Time at end of time history.
T1	-	FORMK	Time at first iteration.
UO	-	LOAD	Initial fluid velocity in x-direction.
VO	-	LOAD	Initial fluid velocity in y-direction.

6 PROGRAM LISTING AND SAMPLE RUN

To aid the user in understanding the behavior of the program FLINTS, a listing of FLINTS together with the output of a small four-iteration run are presented in this section. The printout of the large global and condensed matrices has been partially removed to make examination of the sample run less tedious. For purposes of illustration, the debug option was turned off (I1 = 0) and the stress pass control was turned on (KKK = 1). The model being used to illustrate the behavior of the program is a 72 element solid-fluid model, similar to Fig. 4-12, with one inch walls. The system is undamped.

```
FORTRAN IV G1  RELEASE 2.0            MAIN            DATE = 77075        18/22/49            PAGE 0001
                  C      CONTROL MAIN PROGRAM
                  C      VARIABLE       DEFINITION
                  C      FOR MAIN PROGRAM
                  C      NCN            NO. OF NODES PER ELEMENT %MAX.<
                  C      MPASS          CONTROL TO ALLOW CALCULATION OF EIGENVALUES
                  C                     AND EIGENVECTORS ONCE ONLY
                  C      DELT           INTEGRATION TIME INTERVAL
 0001                    COMMON/G48/TITLE(12),ORI(2,3),ZEI(189),CORD(63,2),NOP(88,6)
                        1,IMAT(66),NBC(20),NFIX(20),NSD(20),NUFIX(20),NTYPE(63)
                        2,CODE(63,2)
 0002                    COMMON/L48/RO(189),R1(189),R(9),RR(3),R4(189),LINDX(85)
                        1,DISP(20,3),INDEX(65),INACT(65),R5(169),NACT(85)
                        2,R6(189),R6F(189)
 0003                    COMMON/E48/R2(189),SM(189,189),SK(189,189)
 0004                    COMMON/TEST1/A(3,6),AK(9,9),BACK(3,6),XM(9,9)
 0005                    COMMON/ST248/S(3,6),ARRAY(90,90),XME(6,6),DIL(6,6)
 0006                    COMMON/FUNCT/YF,Y1,AN(3),AM(3),BN(3),BM(3),CN(3),CM(3),AH,AREAN
                        1,AREAM,XI,XF,SLOPE,YD
 0007                    COMMON/E46/Z(90,90),D(90),DL(90)
 0008                    COMMON/D48/FF(90)
 0009                    COMMON/MOD48/F(189)
 0010                    COMMON/STR/SIGMA(3),U1(66),V1(66)
 0011                    COMMON/PARAM/I,ITIME,LFT,KKK,TEND,NCN,NDFT,NDT,MFREQ
                        1,NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,NIT,IBEG,U0,V0,ALPHA,KMASS
                        2,NELE,NSZF,MSZF,NDLE,NEZF,NEEQ
                        3,NTINC,DELT,NSIZE,II
                        4,BB,CC,NONLIN,DAMP
 0012                    DIMENSION SG(189,189),SH(189,189)
 0013                    EQUIVALENCE (SM,SG),(SK,SH)
                  C      ZERO OUT EVERYTHING IN COMMON
 0014                    CALL CLEAR(TITLE,1216)
 0015                    CALL CLEAR(RO,1546)
 0016                    CALL CLEAR(R2,71631)
 0017                    CALL CLEAR(A,198)
 0018                    CALL CLEAR(S,8190)
 0019                    CALL CLEAR(YF,27)
 0020                    CALL CLEAR(Z,8280)
 0021                    CALL CLEAR(FF,90)
 0022                    CALL CLEAR(F,189)
 0023                    CALL CLEAR(SIGMA,179)
                  C      SET NO. OF NODES PER ELEMENT %MAX.<
 0024                    NCN=3
                  C
                  C      READ INPUT GEOMETRY AND PROP.
                  C
 0025                    CALL GDATA
 0026                    MPASS=1
                  C
                  C      READ LOADS
                  C
 0027                3   CALL LOAD
 0028                    IF(MPASS.NE.1)GOTO1
                  C      THE FOLLOWING IS PERFORMED ONCE ONLY
```

```
FORTRAN IV G1  RELEASE 2.0          MAIN           DATE = 77075        18/22/49            PAGE 0002

                    C      1.  FORM GLOBAL MATRICES AND SET UP FOR MODAL ANALYSIS
                    C      2.  FIND EIGENVALUES AND EIGENVECTORS
      0029           300   CALL FORMK
      0030                 CALL MODAL
      0031                 MPASS=MPASS+1
                    C      FIND FREEDOM FUNCTION--INCREMENT TIME AND ITERATION NUMBER
      0032             1   CALL DIS
      0033                 T=T+DELT
      0034                 ITIME=ITIME+1
      0035                 NTIME=(ITIME/NPRINT)*NPRINT
      0036                 JTIME=0
      0037                 IF(MFREQ.NE.0)JTIME=(ITIME/MFREQ)*MFREQ
                    C
                    C      ENTER MODMAK FOR NONLINEAR ANALYSIS OR STRESS PASS
      0038                 NCALL=0
      0039                 IF(JTIME.EQ.ITIME)NCALL=1
      0040                 IF(NONLIN.EQ.1)NCALL=1
      0041                 IF(KKK.EQ.2)NCALL=1
      0042                 IF(NCALL.NE.0)CALL MODMAK
                    C
                    C      IF KKK = 2  ALSO FIND STRESSES AND VELOCITIES
      0043                 IF(KKK.NE.2)GOTO2
      0044                 CALL STRESS
                    C
                    C      ITERATIVE LOOP
      0045             2   IF(T.GE.TEND)STOP
                    C
                    C      REPEAT FOR NEXT TIME STEP
                    C      IF LOADS ARE A FUNCTION OF TIME, CALL LOAD
      0046                 IF(LFT.EQ.1)GOTO3
      0047                 IF(NDFT.NE.0)GOTO3
      0048                 IF(JTIME.EQ.ITIME.OR.NONLIN.EQ.1)GOTO300
                    C      IF NOT, THEN JUST GO TO DISP
      0049                 GOTO1
      0050                 END
```

FORTRAN IV G1 RELEASE 2.0 CLEAR DATE = 77075 18/22/49 PAGE 0001

```
      SUBROUTINE CLEAR(AMEMBR,LENGTH)
      DIMENSION AMEMBR(LENGTH)
      DO10I=1,LENGTH
   10 AMEMBR(I)=0.
      RETURN
      END
```

```
FORTRAN IV G1  RELEASE 2.0          GDATA          DATE = 77075          18/22/49          PAGE 0001

0001              SUBROUTINE GDATA
0002              COMMON/G48/TITLE(12),ORT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                 1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                 2,COCR(63,2)
0003              COMMON/PARAM/T,ITIME,LFT,KKK,TEND,NCN,NDFT,NDT,MFREQ
                 1,NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,NIT,TBEG,UO,VO,ALPHA,KMASS
                 2,NBCE,NSZF,MSZF,NDCE,NEZF,NEEQ
                 3,NTIME,DELT,NSIZE,I1
                 4,BB,CC,NONLIN,DAMP
            C
            C     READ AND PRINT TITLE AND GENERAL SYSTEM PROPERTIES
            C
0004              READ(5,7)(TITLE(I),I=1,12)
0005              WRITE(6,100)(TITLE(I),I=1,12)
0006              READ(5,1)NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,KKK
0007              WRITE(6,14)
0008              WRITE(6,1)NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,KKK
0009              READ(5,1)NDFI,NDT,MFREQ,NONLIN
0010              WRITE(6,12)
0011              WRITE(6,1)NDFI,NDT,MFREQ,NONLIN
0012              READ(5,10)NIT,TBEG,TEND,LFT,UO,VO,ALPHA,KMASS
0013              WRITE(6,11)
0014              WRITE(6,10)NIT,TBEG,TEND,LFT,UO,VO,ALPHA,KMASS
            C
            C     READ AND PRINT MATERIAL DATA
            C
0015              READ(5,8)(N,(ORT(N,I),I=1,3),L=1,NMAT)
0016              WRITE(6,108)
0017              WRITE(6,109)
0018              WRITE(6,8)(N,(ORT(N,I),I=1,3),N=1,NMAT)
            C
            C     READ AND PRINT DAMPING FACTOR
            C
0019              READ(5,19)DAMP
0020              WRITE(6,101)DAMP
            C
            C     READ NODAL POINT DATA
            C
0021              READ(5,2)(N,(CORD(N,M),M=1,2),L=1,NP)
0022              READ(5,4)(I,NTYPE(I),L=1,NP)
            C
            C     READ ELEMENT DATA
            C
0023              READ(5,3)(N,(NOP(N,M),M=1,6),IMAT(N),L=1,NE)
            C
            C     READ IN BOUNDARY DATA AND SPECIFIED DISPLACEMENTS
            C
0024              READ(5,4)(NBC(I),NFIX(I),I=1,NB)
0025              READ(5,4)(NSD(I),NDFIX(I),I=1,ND)
            C
            C     PRINT INPUT DATA
            C
0026              WRITE(6,102)
```

```
            WRITE(6,15)
            WRITE(6,2)(N,(CORD(N,M),M=1,2),N=1,NP)
            WRITE(6,13)
            WRITE(6,20)(I,NTYPE(I),I=1,NP)
            WRITE(6,103)
            WRITE(6,16)
            WRITE(6,3)(N,(NOP(N,M),M=1,6),IMAT(N),N=1,NE)
            WRITE(6,104)
            WRITE(6,17)
            WRITE(6,20)(NBC(I),NFIX(I),I=1,NB)
            WRITE(6,105)
            WRITE(6,17)
            WRITE(6,20)(NSD(I),NDFIX(I),I=1,ND)
      C
      C     INITIALIZE TIME AND ITERATION NUMBER
      C
            T=TBEG
            ITIME=0
            RETURN
          1 FORMAT(1018)
          2 FORMAT(I10,2F10.3)
          3 FORMAT(E15)
          4 FORMAT(2I5)
          7 FORMAT(12A4)
          8 FORMAT(I10,3E12.5)
         10 FORMAT(I10,2F10.5,I10,3F10.5,I10)
         11 FORMAT(1H0,4X,3HNIT,6X,4HTBEG,6X,4HTEND,7X,3HLFT,7X,2HUO,8X,2HVO,7
           1X,5HALPHA,5X,5HKMASS)
         12 FORMAT(1H0,3X,4HNDFT,6X,3HNDT,3X,5HMFREQ,5X,6HNONLIN)
         13 FORMAT(25H0NODAL DEGREES OF FREEDOM/1H ,12(4HNODE,1X,4HTYPE,1X))
         14 FORMAT(1H0,3X,2HNP,6X,2HNE,6X,2HNB,6X,3HNLD,4X,4HNMAT,5X,2HI1,5X,6
           1HNPRINT,3X,2HNG,6X,3HNDF,5X,3HKKK)
         15 FORMAT(1H0,3X,4HNODE,5X,6HX-CORD,4X,6HY-CORD)
         16 FORMAT(1H0,12X,5HNODES,7X,9HMATL. NO.)
         17 FORMAT(1H0,3X,12(4HNODE,1X,4HTYPE,1X))
         19 FORMAT(6E12.5)
         20 FORMAT(12(2I5))
        100 FORMAT(1H1,12A4)
        101 FORMAT(18HODAMPING FACTOR IS,E12.5,2X,5HSEC-1)
        102 FORMAT(20H0NODAL POINTS        )
        103 FORMAT(20H0ELEMENTS            )
        104 FORMAT(20H0BOUNDARY CONDITIONS)
        105 FORMAT(24H0SPECIFIED DISPLACEMENTS)
        108 FORMAT(1H0,19HMATERIAL PROPERTIES)
        109 FORMAT(11H0 MATL. NO.,5X,3HE,C,8X,4HNU,-,7X,7HDENSITY)
            END
```

```
FORTRAN IV G1  RELEASE 2.0            LOAD              DATE = 77075          18/22/49             PAGE 0001

0001                    SUBROUTINE LOAD
0002                    COMMON/G48/TITLE(12),ORT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                       1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                       2,CORR(63,2)
0003                    COMMON/L48/RO(189),R1(189),R(9),RR(3),R4(189),LINDX(85)
                       1,DISP(20,3),INDEX(85),INACT(85),R5(189),NACT(85)
                       2,R6(189),R6F(189)
0004                    COMMON/STR/SIGMA(3),UI(88),VI(88)
0005                    COMMON/PARAM/T,ITIME,LFT,KKK,TEND,NCN,NDFT,NDT,MFREQ
                       1,NP,NE,NB,NLD,NMAT,II,NPRINT,ND,NDF,NII,IBEG,UO,VO,ALPHA,KMASS
                       2,NBCE,NSZF,MSZF,NDCE,NEZF,NEEQ
                       3,NTIME,DELT,NSIZE,1)
                       4,BB,CC,NONLIN,DAMP
0006                    IF(ITIME.EQ.0)GOTO165
0007                    IF(LFT.EQ.0)GOTO160
0008              165   CONTINUE
               C        SET PARAMETERS
0009                    NBCE=NP*NDF
0010                    NSZF=NP*NDF
0011                    MSZF=NP*(NDF-1)
0012                    NDCE=ND*NDF
0013                    NEZF=NCN*NDF
0014                    NEEQ=NCN*(NDF-1)
0015                    IF(NDF.GT.1)GOTO169
0016                    MSZF=NSZF
0017                    NEEQ=NEZF
0018              169   CONTINUE
0019                    WRITE(6,103)NBCE,NSZF,MSZF,NDCE,NEZF,NEEQ
0020                    WRITE(6,100)(TITLE(I),I=1,12)
               C
               C        SET INITIAL VELOCITY ARRAYS UI AND VI TO UO AND VO
0021                    DO184I=1,NE
0022                    UI(I)=UO
0023                    VI(I)=VO
0024              184   CONTINUE
               C        PREPARE R AND R4 IN U V P ... FORM
0025                    DO177N=1,NB
0026                    NQ=NBC(N)
               C        NQ IS NODE AT WHICH LOAD ACTS
0027                    I=1
0028                    IFIX=NFIX(N)+1
0029                    GOTO(79,80),IFIX
0030                    GOTO81
0031               79   I=3
0032                    GOTO81
0033               80   I=2
0034               81   CONTINUE
0035              179   READ(5,20)PMAX,OMEGA,NLD
0036                    WRITE(6,21)
0037                    WRITE(6,20)PMAX,OMEGA,NLD
               C        READS IN MAGNITUDE OF LOAD, FREQUENCY, TYPE OF LOAD
0038                    IF(NLD.EQ.0)GOTO174
               C        LOGIC FOR RANDOM LOADS
```

```
FORTRAN IV G1  RELEASE 2.0          LOAD          DATE = 77075        18/22/49          PAGE 0002

0039                GOTO(172,173,174,175),NLD
              C     SINUSOIDAL LOAD
0040          172   ARG=OMEGA*T
0041                R(I)=PMAX*SIN(ARG)
0042                GOTO176
              C     RAMP LOAD
0043          173   R(I)=PMAX-(PMAX/OMEGA)*T
0044                GOTO176
              C     STEP FUNCTION
0045          174   R(I)=PMAX
0046                GOTO176
              C     DECAYING EXPONENTIAL LOAD
0047          175   ARG=-OMEGA*T
0048                R(I)=PMAX*(1.-EXP(ARG))
0049          176   CONTINUE
0050                I=I+1
0051                IFIX=IFIX-1
0052                IF(IFIX.EQ.1)GOTO179
0053          17    CONTINUE
0054                NN=N+1
0055                IF(NBC(N).EQ.NBC(NN))GOTO177
0056                NROWB=(NN-1)*NDF
0057                DO170II=1,NDF
0058                NROWB=NROWB+1
0059                R4(NROWB)=R(II)
0060          170   CONTINUE
0061                WRITE(6,77)NN,(R(K),K=1,NDF)
              C     SET REMAINING ELEMENTS EQUAL TO ZERO
0062                DO76J=1,NDF
0063                R(J)=0.
0064          76    CONTINUE
0065          177   CONTINUE
0066          178   CONTINUE
0067                WRITE(6,76)
0068                WRITE(6,113)(R4(I),I=1,NSZF)
              C     SORT LOAD ARRAY INTO U V ... P ... FORM
0069                IF(NDF.LT.5)GOTO195
0070                NSHIF=0
0071                NEND=NSZF-NDF
0072                DO181I=1,NEND
0073                KK=I/NDF
0074                II=KK*NDF
0075                IF(II.NE.I)GOTO181
              C     SKIPS SORTING
0076                II=II-NSHIF
              C     SAVE ELEMENT TO BE SORTED AND NOTE CORRECTED ROW NUMBER
0077                SAVE=R4(II)
0078                NN=NP*(NDF-1)+KK
0079                LINDX(KK)=NN
0080                NM=NN-1
0081                ISW=0
0082                DO190J=II,NM
0083                KSW=0
```

```
0084              JJ=J+1
0085              DO200K=1,KK
0086              IF(J.EQ.LINDX(K))KSW=KSW+1
            C     INDICATES DO NOT SORT
0087              IF(JJ.EQ.LINDX(K))ISW=ISW+1
            C     INDICATES SKIP CURRENT VALUE--STORE NEXT VALUE
0088          200 CONTINUE
0089              IF(KSW.NE.0)GOTO190
0090              IF(ISW.NE.0)JJ=JJ+KK-1
0091              R4(J)=R4(JJ)
0092          190 CONTINUE
0093              R4(NN)=SAVE
            C     PUTS SAVED VALUE INTO CORRECT LOCATION
0094              NSHIF=NSHIF+1
0095          181 CONTINUE
0096              LINDX(KK+1)=NSZF
            C     SORTING NOW COMPLETED
0097          195 CONTINUE
0098          160 CONTINUE
0099              IF(ITIME.EQ.0)GOTO155
0100              IF(NDFI.EQ.0)GOTO150
0101          155 CONTINUE
0102              DO182I=1,ND
0103              KK=1
0104              IF(NDFIX(I).EQ.1)KK=2
0105              MDF=1
0106              IF(NDFIX(I).EQ.0)GOTO82
            C     INDICATES P FIXED
0107              MOD=(NDFIX(I)/10)*10
0108              IF(MOD.NE.NDFIX(I))MDF=2
            C     INDICATES U OR V FIXED
            C
            C     READ, PRINT, AND STORE DISPLACEMENTS
            C
0109           82 READ(5,4)(DISP(I,K),K=KK,MDF)
0110          182 CONTINUE
            C
            C     DETERMINE AND STORE THE ROW NUMBERS IN THE GLOBAL MATRICES WHERE
            C     DISPLACEMENTS ARE SPECIFIED
            C
0111              KK=0
0112              DO50N=1,ND
0113              I=NSD(N)
            C     LOGIC FOR U, V ONLY
0114              MDF=2
0115              NROWB=(I-1)*MDF
0116              IF(NDFIX(N).EQ.1)NROWB=NROWB+1
0117              IF(NDFIX(N).NE.11)MDF=1
0118              IF(NDFIX(N).NE.0)GOTO52
            C     LOGIC FOR PRESSURE ONLY
0119              NROWB=MSZF+I-1
0120              IF(NDF.EQ.1)NROWB=NROWB-MSZF
0121           52 DO51M=1,MDF
```

```
0122                NROWB=NROWB+1
0123                KK=KK+1
0124                INDEX(KK)=NROWB
0125          51    CONTINUE
0126          50    CONTINUE
             C
             C      FIND INACTIVE ROW NUMBERS
0127                IF(NDF.LT.3)GOTO191
0128                NN=0
0129                DO185N=1,NP
0130                NT=NTYPE(N)
0131                GOTO(186,187,185),NT
0132         186    MDF=2
0133                NROWB=(N-1)*MDF
0134                GOTO188
0135         187    MDF=1
0136                NROWB=(N-1)*MDF+NSZF
0137         188    DO189L=1,MDF
0138                NN=NN+1
0139                NROWB=NROWB+1
0140                INACT(NN)=NROWB
0141         189    CONTINUE
0142         185    CONTINUE
0143         191    CONTINUE
             C
             C      FIND ACTIVE ROW NUMBERS FOR MODAL ANALYSIS IN SEQUENCE
0144                II=1
0145                DO145I=1,NSZF
0146                ISW=0
0147                DO140J=1,86
0148                IF(I.EQ.INACT(J))ISW=ISW+1
0149         140    CONTINUE
0150                IF(ISW.NE.0)GOTO145
0151                DO130J=1,85
0152                IF(I.EQ.INDEX(J))ISW=ISW+1
0153         130    CONTINUE
0154                IF(ISW.NE.0)GOTO145
0155                NACT(II)=I
0156                II=II+1
0157         145    CONTINUE
0158         150    CONTINUE
0159                WRITE(6,111)(INDEX(KK),KK=1,85)
0160                WRITE(6,115)(INACT(KK),KK=1,85)
0161                WRITE(6,120)(NACT(KK),KK=1,85)
0162                WRITE(6,118)
0163                DO53I=1,NU
0164          53    WRITE(6,110)(I,K,DISP(I,K),K=1,NDF)
0165                WRITE(6,112)(LINDX(KK),KK=1,85)
0166                WRITE(6,109)
0167                WRITE(6,113)(R4(I),I=1,NSZF)
0168                RETURN
0169           4    FORMAT(2E10.3)
0170          20    FORMAT(2F10.4,I10)
```

```
FORTRAN IV G1  RELEASE 2.0          LOAD          DATE = 77075          18/22/49          PAGE 0005

0171          21  FORMAT(1H0,3X,4HPMAX,6X,5HOMEGA,6X,3HNLD)
0172          77  FORMAT(1H0,3X,4HNODE,6X,4HLOAD/1H ,I5,2F10.2/2(1H ,5X,2F10.2/))
0173          78  FORMAT(21H0UNSORTED LOAD ARRAYS)
0174         100  FORMAT(1H1,12A4)
0175         103  FORMAT(18H0SYSTEM PARAMETERS/1H ,5X,4HNBCE,1X,4HNSZF,1X,4HMSZF,1X,
                 14HNDCE,1X,4HNEZE,1X,4HNFEQ/1H ,4X,6(I5))
0176         109  FORMAT(1H0,5HLOADS)
0177         110  FORMAT(1H ,5HDISP(,I4,1H,,I4,2H)=,3X,E10.3/)
0178         111  FORMAT(1H0,52HSPECIFIED DISPLACEMENTS OCCUR AT THE FOLLOWING ROWS
                 1/6(1H ,12(I6,1X)/))
0179         112  FORMAT(52H0PRESSURE FREEDOMS ARE LOCATED AT THE FOLLOWING ROWS/6(
                 1H ,12(I6,1X)/))
0180         113  FORMAT(6(1H ,10(E12.5,1X)/))
0181         115  FORMAT(33H0INACTIVE FREEDOMS ARE LOCATED AT/6(1H ,12(I6,1X)/))
0182         118  FORMAT(1H0,23HTHESE DISPLACEMENTS ARE)
0183         120  FORMAT(22H0MODAL ROW NUMBERS ARE/6(1H ,12(I6,1X)/))
0184              END
```

```
FORTRAN IV G1  RELEASE 2.0            FORMK            DATE = 77075          18/22/49                PAGE 0001

0001                  SUBROUTINE FORMK
                 C    FOR SUBROUTINE FORMK
                 C    IPASS          IN F/S CALCULATION, INDICATES WHICH ARRAYS
                 C                   ARE TO BE ASSEMBLED
                 C    MDF,LDF        DUMMY DEGREES OF FREEDOM WHICH VARY AS TO
                 C                   THE TYPE OF ELEMENT CONSIDERED
                 C    NTIME          PRINTOUT VARIABLE
0002                  COMMON/G48/TITLE(12),ORT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                     1,IMAT(66),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                     2,COOR(63,2)
0003                  COMMON/L48/RO(189),R1(189),R(9),RR(3),R4(189),LINDX(85)
                     1,DISP(20,3),INDEX(85),INACT(85),R5(189),NACT(85)
                     2,R6(189),R6F(189)
0004                  COMMON/F48/R2(189),SM(189,189),SK(189,189)
0005                  COMMON/TEST1/A(3,6),AK(9,9),BACK(3,6),XM(9,9)
0006                  COMMON/PARAM/I,ITIME,LET,KKK,IEND,NCN,NUET,ND1,MEREQ
                     1,NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,NIT,TBEG,UO,VO,ALPHA,KMASS
                     2,NBLE,NSZE,MSZE,NDCE,NEZE,NEEQ
                     3,NTIME,DELT,NSIZE,T1
                     4,BB,CC,NONLIN,DAMP
0007                  DIMENSION SG(189,189),SH(189,189)
0008                  EQUIVALENCE (SM,SG),(SK,SH)
                 C    LOGIC FOR NONLINEAR ANALYSIS
0009                  IF(ITIME.EQ.0)GOTO530
0010                  IF(NONLIN.NE.0)GOTO1020
0011          530     CONTINUE
                 C
                 C    FIND TIME INTERVAL; FIND FIRST TIME VALUE; SET PARAMETER FOR
                 C    PRINTOUT
                 C
0012                  DELT=(TEND-TBEG)/FLOAT(NIT)
0013                  T1=TBEG+DELT
0014                  NTIME=(ITIME/NPRINT)*NPRINT
                 C    PUT NODAL COORDINATES INTO FIXED ARRAY COOR
                 C    FIRST TIME ONLY
0015                  IF(ITIME.NE.0)GOTO125
0016                  DO120N=1,NP
0017                  DO115M=1,2
0018                  COOR(N,M)=CORD(N,M)
0019          115     CONTINUE
0020          120     CONTINUE
0021                  IF(I1.LT.3)GOTO125
0022                  WRITE(6,110)
0023                  DO105N=1,NP
0024          105     WRITE(6,993)N,(M,COOR(N,M),M=1,2)
0025          125     CONTINUE
                 C
                 C    ZERO OUT ACCELERATION VECTOR
                 C    R6 SO THAT IT MAY CONTAIN
                 C    THE NONLINEAR LOADS LATER
0026           20     CONTINUE
0027                  DO21JJ=1,NSZE
0028                  R6(JJ)=0.
```

```
FORTRAN IV G1  RELEASE 2.0           FORMK          DATE = 77075          18/22/49              PAGE 0002

0029            210  CONTINUE
                C
                C    SCAN ELEMENTS
                C    FIND AND ASSEMBLE ARRAYS
                C
0030                 DO404N=1,NE
0031                 LL=IMAT(N)
0032                 IF(NDF.EQ.1)LL=2
0033                 IF(ITIME.EQ.0)GOTO510
0034                 IF(NONLIN.EQ.0)GOTO510
0035                 IF(LL.NE.3)GOTO400
                C    SKIPS ALL BUT FLUID/SOLID NODES
0036            510  CONTINUE
0037                 GOTO(401,402,403),LL
                C    SKIP CALCULATION IF SFS ELEMENT
0038            401  CALL STIEF1(N)
                C    SOLID ELEMENT
0039                 MDF=2
0040                 LDF=2
0041                 M=N
0042                 MM=N
0043                 GOTO404
0044            402  CALL STIF12(N)
                C    FLUID ELEMENT
0045                 MDF=1
0046                 LDF=1
0047                 M=N
0048                 MM=N
0049                 GOTO404
0050            403  CALL STIFT4(N)
                C    FLUID/SOLID ELEMENT
0051            411  MDF=2
0052                 LDF=2
0053                 M=N
0054                 MM=N
0055            404  CONTINUE
                C
                C     ASSEMBLE ELEMENTAL MATRICES
                C
                C    SET ROW INDICES
0056                 I=0
0057                 NROW=NLN
0058                 IF(LL.GT.2)NROW=NEEQ
0059                 DO350JJ=1,NROW
0060                 IF(JJ.LE.NLN)GOTO351
0061                 MDF=1
0062            351  JK=JJ
0063            352  NROWB=(NOP(MM,JK)-1)*MDF
0064                 IF(MDF.EQ.1)NROWB=NROWB+NSZF
0065                 IF(NDF.EQ.1)NROWB=NROWB-NSZF
0066                 DO350J=1,MDF
0067                 NROWB=NROWB+1
0068                 I=I+1
```

```
                 C      ASSEMBLE NONLINEAR FORCES
0069                    R6(NROWB)=R6(NROWB)+R(I)
0070                    IF(ITIME.EQ.0)GOTO520
0071                    IF(NONLIN.EQ.0)GOTO520
0072                    GOTO350
0073             520    CONTINUE
                 C
                 C      SET COLUMN INDICES
0074                    L=0
0075                    NCOL=NCN
0076                    IF(LL.GT.2)NCOL=NEEQ
0077                    DO330KK=1,NCOL
0078                    IF(KK.LE.NCN)GOTO329
0079                    LDF=1
0080             329    KJ=KK
0081             328    NCOLB=(NOP(M,KJ)-1)*LDF
0082                    IF(LDF.EQ.1)NCOLB=NCOLB+NSZF
0083                    IF(NDF.EQ.1)NCOLB=NCOLB-NSZF
0084                    DO320K=1,LDF
0085                    NCOLB=NCOLB+1
0086                    L=L+1
                 C      ASSEMBLY
0087             405    SM(NROWB,NCOLB)=SM(NROWB,NCOLB)+XM(I,L)
0088                    SK(NROWB,NCOLB)=SK(NROWB,NCOLB)+AK(I,L)
0089             320    CONTINUE
0090             330    CONTINUE
0091                    IF(LL.LT.3)GOTO350
0092                    LDF=2
0093             350    CONTINUE
0094             400    CONTINUE
0095                    IF(I1.LT.2)GOTO135
0096                    WRITE(6,810)(R6(I),I=1,NSZF)
0097                    IF(ITIME.EQ.0)GOTO129
0098                    IF(NONLIN.EQ.0)GOTO129
0099                    GOTO142
0100             129    DO131MM=1,NSZF
0101                    IF(MM.EQ.1)WRITE(6,891)I1
0102             131    WRITE(6,993)MM,(NN,SM(MM,NN),NN=1,NSZF)
0103             133    DO134MM=1,NSZF
0104                    IF(MM.EQ.1)WRITE(6,892)I1
0105             134    WRITE(6,993)MM,(NN,SK(MM,NN),NN=1,NSZF)
0106             135    CONTINUE
0107                    IF(ITIME.EQ.0)GOTO136
0108                    IF(NONLIN.EQ.0)GOTO136
0109                    GOTO142
0110             136    CONTINUE
                 C      PROBLEM IS NOW READY FOR MODAL ANALYSIS
                 C      BUT FIRST INCLUDE SPECIFIED DISPLACEMENTS
                 C      NOW ADD FORCES DUE TO SPECIFIED DISPLACEMENTS $R1< TO
                 C      APPLIED LOADS $R4< AND PUT INTO R1
0111                    DO460I=1,NSZF
0112                    R1(I)=R1(I)+R4(I)
0113             460    CONTINUE
```

```
FORTRAN IV G1  RELEASE 2.0          FORMK          DATE = 77075          18/22/49          PAGE 0004

          DO500N=1,ND
          IFIX=NDFIX(N)
          MM=1
          IF(IFIX.EQ.1)MM=2
          MDF=2
          IF(IFIX.EQ.0)MDF=1
          NX=10**(MDF-1)
          I=NSD(N)
          NROWB=(I-1)*MDF
          IF(MDF.EQ.1)NROWB=NROWB+MSZF
          IF(MDF.EQ.1)NROWB=NROWB-MSZF
          MDF=1
          MOD=(IFIX/10)*10
          IF(MOD.NE.IFIX)MDF=2
          IF(IFIX.EQ.1)NROWB=NROWB+1
C
C         EXAMINE EACH DEGREE OF FREEDOM
          DO490M=MM,MDF
          NROWB=NROWB+1
          R1(NROWB)=DISP(N,M)
C
C         MODIFIES B ARRAY TO INCLUDE SPECIFIED DISPLACEMENTS
          DO53II=1,NSZF
          IROW=0
          DO55JJ=1,85
          IF(II.EQ.INDEX(JJ))IROW=IROW+1
     55   CONTINUE
          IF(IROW.GT.0)GOTO53
          R1(II)=R1(II)-SK(II,NROWB)*R1(NROWB)
     53   CONTINUE
          ICON=NDFIX(N)/NX
          IF(ICON)450,420,420
    420   SK(NROWB,NROWB)=1.
          SM(NROWB,NROWB)=1.
          NN=NROWB+1
          IF(NN.GT.NSZF)GOTO425
C
C         ZERO OUT ROWS AND COLUMNS FOR SPECIFIED DISPLACEMENTS
C         FIRST FOR ELEMENTS %I,J<--I,J GT NROWB
          DO430J=NN,NSZF
          SK(NROWB,J)=0.
          SM(NROWB,J)=0.
          SK(J,NROWB)=0.
          SM(J,NROWB)=0.
    430   CONTINUE
    425   CONTINUE
C
C         FOR ELEMENTS %I,J<--I,J .LE. NROWB
          NN=NROWB-1
          IF(NN.LT.1)GOTO445
          J=NN
    440   CONTINUE
          SK(NROWB,J)=0.
```

```
            SM(NROWB,J)=0.
            SK(J,NROWB)=0.
            SM(J,NROWB)=0.
            J=J-1
            IF(J.GE.1)GOTO440
      445   CONTINUE
            NOFIX(N)=NOFIX(N)-NX*ICON
      450   NX=NX/10
      490   CONTINUE
            NOFIX(N)=IFIX
      500   CONTINUE
C                             OUTPUT BLOCK
            IF(I1.LT.1)GOTO142
            DO711I=1,NSZF
            IF(I.EQ.1)WRITE(6,710)I1
      701   WRITE(6,993)I,(J,SK(I,J),J=1,NSZF)
            DO703I=1,NSZF
            IF(I.EQ.1)WRITE(6,712)
      703   WRITE(6,993)I,(J,SM(I,J),J=1,NSZF)
            DO711I=1,NSZF
            IF(I.EQ.1)WRITE(6,793)
      711   WRITE(6,700)R1(I)
      142   CONTINUE
C                             END OF OUTPUT BLOCK
C           NOW REDUCE MATRIX SO THAT UNAFFECTED PARTS ONLY ARE SENT INTO
C           MODAL
            I=1
            DO470II=1,NSZF
            J=1
            IROW=0
            DO471JJ=1,85
            IF(II.EQ.INDEX(JJ))IROW=IROW+1
            IF(II.EQ.INACT(JJ))IROW=IROW+1
      471   CONTINUE
            IF(IROW.GT.0)GOTO470
C           ADJUST COLUMN VECTORS
            R6(I)=R6(II)
            IF(ITIME.EQ.0)GOTO540
            IF(NONLIN.EQ.0)GOTO540
            GOTO550
      540   CONTINUE
            R1(I)=R1(II)
            R5(I)=R5(II)
            R2(I)=R2(II)
            DO472JJ=1,NSZF
            JROW=0
            DO473JK=1,85
            IF(JJ.EQ.INDEX(JK))JROW=JROW+1
            IF(JJ.EQ.INACT(JK))JROW=JROW+1
      473   CONTINUE
            IF(JROW.GT.0)GOTO472
            SK(I,J)=SK(II,JJ)
            SM(I,J)=SM(II,JJ)
```

```
FORTRAN IV G1  RELEASE 2.0            FORMK          DATE = 77075        18/22/49              PAGE 0006

 0206                 J=J+1
 0207            472  CONTINUE
 0208            550  CONTINUE
 0209                 I=I+1
 0210            470  CONTINUE
            C
            C                        OUTPUT BLOCK
 0211                 IF(I1.LT.2)GOTO560
 0212                 IF(ITIME.EQ.0)GOTO560
 0213                 IF(NONLIN.EQ.0)GOTO560
 0214                 WRITE(6,820)(R6(I),I=1,NSZF)
            C                        END OF OUTPUT BLOCK
            C
 0215            560  CONTINUE
            C         SCAN ARRAY TO DETERMINE SIZE OF MATRIX
 0216                 N=0
 0217                 DO477J=1,85
 0218                 IF(INDEX(J).NE.0)N=N+1
 0219                 IF(INACT(J).NE.0)N=N+1
 0220            477  CONTINUE
 0221                 NSIZE=NSZF-N
            C         ZERO OUT REMAINDER OF ARRAY
 0222                 NN=NSIZE+1
 0223                 DO357I=1,NSZF
 0224                 DO358J=NN,NSZF
 0225                 R6(J)=0.
 0226                 IF(ITIME.EQ.0)GOTO356
 0227                 IF(NONLIN.EQ.0)GOTO356
 0228                 GOTO358
 0229            356  CONTINUE
 0230                 SK(I,J)=0.
 0231                 SK(J,I)=0.
 0232                 SM(I,J)=0.
 0233                 SM(J,I)=0.
 0234                 R1(J)=0.
 0235                 R5(J)=0.
 0236                 R2(J)=0.
 0237            358  CONTINUE
 0238                 IF(ITIME.EQ.0)GOTO357
 0239                 IF(NONLIN.EQ.0)GOTO357
 0240                 RETURN
 0241            357  CONTINUE
            C         ARRAYS SH %STIFFNESS< AND SG %MASS< ARE NOW READY FOR
            C         MODAL ANALYSIS
            C                        OUTPUT BLOCK
 0242                 WRITE(6,717)NSIZE
 0243                 IF(I1.LT.0)RETURN
 0244                 NN=NSIZE
 0245                 IF(I1.GE.2)NN=NSZF
 0246                 DO476I=1,NN
 0247                 IF(I.EQ.1)WRITE(6,715)I1
 0248            476  WRITE(6,993)I,(J,SH(I,J),J=1,NN)
 0249                 DO478I=1,NN
```

```
            IF(I.EQ.1)WRITE(6,716)
      478   WRITE(6,993)I,(J,SG(I,J),J=1,NN)
            WRITE(6,705)
            WRITE(6,700)(R6(I),I=1,NN)
            WRITE(6,704)
            WRITE(6,700)(R1(I),I=1,NN)
      475   CONTINUE
C                       END OF OUTPUT BLOCK
C                       OUTPUT BLOCK
            IF(II.LT.2)GOTO480
            WRITE(6,715)II
            DO482I=1,NN
      482   WRITE(6,993)I,(J,SK(I,J),J=1,NN)
            WRITE(6,716)
            DO481I=1,NN
      481   WRITE(6,993)I,(J,SM(I,J),J=1,NN)
      480   CONTINUE
C                       END OF OUTPUT BLOCK
            RETURN
      110   FORMAT(32HCORIGINAL NODAL COORDINATE ARRAY)
      700   FORMAT(6(1H ,10(E12.5,1X)/))
      704   FORMAT(19HCREDUCED LOAD ARRAY)
      705   FORMAT(29HOREDUCED NONLINEAR LOAD ARRAY)
      710   FORMAT(12HODEBUG LEVEL,I4/54HOSTIFFNESS MATRIX MODIFIED FOR SPECIF
           1IED DISPLACEMENTS)
      712   FORMAT(49HOMASS MATRIX MODIFIED FOR SPECIFIED DISPLACEMENTS)
      715   FORMAT(12HODEBUG LEVEL,I4/46HOCONDENSED STIFFNESS MATRIX FOR MODAL
           1 ANALYSIS)
      716   FORMAT(41HOCONDENSED MASS MATRIX FOR MODAL ANALYSIS)
      717   FORMAT(36HOSIZE OF CONDENSED MATRICES IS,I4)
      793   FORMAT(3HOB#)
      810   FORMAT(31HOLOAD ARRAY FOR NONLINEAR LOADS/6(1H ,10(E12.5,1X)/))
      820   FORMAT(37HOREARRANGED LOADS--NONLINEAR ANALYSIS/6(1H ,10(E12.5,1X)
           1/))
      891   FORMAT(12HODEBUG LEVEL,I4/12HOMASS MATRIX)
      892   FORMAT(12HODEBUG LEVEL,I4/17HOSTIFFNESS MATRIX)
      893   FORMAT(12HODEBUG LEVEL,I4/15HOINERTIA MATRIX)
      894   FORMAT(12HODEBUG LEVEL,I4/16HOFLUIDITY MATRIX)
      895   FORMAT(12HODEBUG LEVEL,I4/16HOCOUPLING MATRIX)
      993   FORMAT(4H ROW,I4,5(I4,1H,,1X,E12.5,2X)/7(1H ,7X,5(I4,1H,,1X,E12.5,
           12X)/))
            END
```

```
0001            SUBROUTINE STIFT1(N)
          C     CALCULATES MASS AND STIFFNESS MATRICES FOR SOLID
          C     PLANE TRIANGULAR ELEMENT
          C     FOR SUBROUTINE STIFT1%N<
          C     I,J,K          ELEMENT CONNECTIONS--LATER USED AS LOOP COUNTERS
          C     BM(NCN),CM(NCN)  LOCAL COORDINATES OF SOLID TRIANGLE
          C     AREAM          AREA OF SOLID TRIANGULAR ELEMENT
0002            COMMON/G48/TITLE(12),URT(2,3),ZET(189),CORD(63,2),NOP(88,6)
               1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
               2,CORR(63,2)
0003            COMMON/TEST1/A(3,6),AK(9,9),BACK(3,6),XM(9,9)
0004            COMMON/FUNC1/YE,Y1,AN(3),AM(3),BN(3),BM(3),CN(3),CM(3),AH,AREAN
               1,AREAM,XI,XF,SLOPE,YO
0005            COMMON/D48/FF(90)
0006            COMMON/MOD48/F(189)
0007            COMMON/STR/SIGMA(3),UI(88),VI(88)
0008            COMMON/PARAM/T,ITIME,LFT,KKK,TEND,NCN,NUFT,NDT,MFREQ
               1,NP,NE,NB,NLU,NMAT,I1,NPRINT,ND,NUF,NIT,IBEG,UO,VO,ALPHA,KMASS
               2,NBCE,NSZF,MSZF,NUCE,NEZF,NEEQ
               3,NTIME,DELT,NSIZE,I1
               4,BB,CC,NONLIN,DAMP
          C
          C     SET UP FREEDOM ARRAY FF ONLY IN STRESS PASS
0009            IF(KKK.NE.2)GOTO230
0010            DO235I=1,90
0011      235   FF(I)=0.
0012            DO240LL=1,NCN
0013            MDF=2
0014            NN=NOP(N,LL)
0015            NROWB=(NN-1)*MDF
0016            IROW=(LL-1)*MDF
0017            DO245MM=1,MDF
0018            NROWB=NROWB+1
0019            IROW=IROW+1
0020            FF(IROW)=F(NROWB)
0021      245   CONTINUE
0022      240   CONTINUE
0023            MTIME=ITIME
0024      230   CONTINUE
          C
          C     DETERMINE ELEMENT CONNECTIONS
0025            I=NOP(N,1)
0026            J=NOP(N,2)
0027            K=NOP(N,3)
0028            L=IMAT(N)
0029            IF(I1.GE.3)WRITE(6,207)
0030            IF(I1.GE.3)WRITE(6,160)I1,N,I,J,K
          C     SET UP LOCAL COORDINATE SYSTEM
0031            CM(3)=CORD(J,1)-CORD(I,1)
0032            CM(2)=CORD(I,1)-CORD(K,1)
0033            CM(1)=CORD(K,1)-CORD(J,1)
0034            BM(3)=CORD(I,2)-CORD(J,2)
0035            BM(2)=CORD(K,2)-CORD(I,2)
```

```
0036          BM(1)=CORD(J,2)-CORD(K,2)
0037          IF(I1.GE.2)WRITE(6,140)I1,N,(BM(I),CM(I),I=1,NCN)
0038          AREAM=(CM(3)*BM(2)-CM(2)*BM(3))/2.
0039          IF(AREAM.LE.0.)GOTO220
        C     ERROR EXIT FOR BAD CONNECTIONS
        C     FORM STRAIN-DISPLACEMENT MATRIX
0040          A(1,1)=-BM(3)-BM(2)
0041          A(1,2)=0.
0042          A(1,3)=BM(2)
0043          A(1,4)=0.
0044          A(1,5)=BM(3)
0045          A(1,6)=0.
0046          A(2,1)=0.
0047          A(2,2)=-CM(2)-CM(3)
0048          A(2,3)=0.
0049          A(2,4)=CM(2)
0050          A(2,5)=0.
0051          A(2,6)=CM(3)
0052          A(3,1)=-CM(2)-CM(3)
0053          A(3,2)=-BM(3)-BM(2)
0054          A(3,3)=CM(2)
0055          A(3,4)=BM(2)
0056          A(3,5)=CM(3)
0057          A(3,6)=BM(3)
0058          IF(I1.GE.1)WRITE(6,223)(I,(A(I,J),J=1,6),I=1,3)
        C     FORM STRESS-STRAIN MATRIX
0059          CONST=ORT(L,1)/((1.-ORT(L,2)*ORT(L,2))*AREAM)
0060          AK(1,1)=CONST
0061          AK(1,2)=CONST*ORT(L,2)
0062          AK(1,3)=0.
0063          AK(2,1)=AK(1,2)
0064          AK(2,2)=AK(1,1)
0065          AK(2,3)=0.
0066          AK(3,1)=AK(1,3)
0067          AK(3,2)=AK(2,3)
0068          AK(3,3)=CONST*(1.-ORT(L,2))/2.
0069          IF(I1.GE.1)WRITE(6,227)(I,(AK(I,J),J=1,3),I=1,3)
        C     BACK IS BACKSUBSTITUTION MATRIX
0070          DO226I=1,NCN
0071          DO224J=1,6
0072          BACK(I,J)=0.
0073          DO233K=1,3
0074      233 BACK(I,J)=BACK(I,J)+(AK(I,K)/2.)*A(K,J)
0075      224 CONTINUE
0076      226 CONTINUE
        C     THIS IS ENOUGH FOR STRESS PASS
        C     FIND STRESSES IF DESIRED
0077          IF(KKK.NE.2)GOTO257
0078          DO255I=1,3
0079          SIGMA(I)=0.
0080          DO255K=1,6
0081      255 SIGMA(I)=SIGMA(I)+BACK(I,K)*FF(K)
0082          IF(I.EQ.I1)GOTO256
```

```
FORTRAN IV G1  RELEASE 2.0             STIFT1           DATE = 77075          18/22/49               PAGE 0003

                IF(NTIME.NE.ITIME)RETURN
          256   WRITE(6,260)MTIME,N,(SIGMA(I),I=1,3)
                RETURN
          257   CONTINUE
        C       AK NOW BECOMES STIFFNESS MATRIX
                DO212I=1,6
                DO213J=1,6
                AK(I,J)=0.
                DO214K=1,3
          214   AK(I,J)=AK(I,J)+(BACK(K,I)/2.)*A(K,J)
          213   CONTINUE
          212   CONTINUE
        C       STIFFNESS MATRIX IS NOW CALCULATED
        C       CALCULATE MASS MATRIX
                CONST=ORT(L,3)*AREAM/FLOAT(NCN)
        C       DISTRIBUTED MASS--LOGIC FOR DIAGONAL MATRIX
                DO200II=1,6
                DO201JJ=1,6
                XM(II,JJ)=0.
                IF(II.EQ.JJ)GOTO202
                IF(KMASS.EQ.0)GOTO201
        C       IF KMASS = 0 , LUMPED MASS REPRESENTATION
        C       IF KMASS.NE.0 , DISTRIBUTED MASS REPRESENTATION
                JI=II+JJ
                IJ=(JI/2)*2
                IF(IJ.NE.JI)GOTO201
                XM(II,JJ)=CONST/4.
                GOTO201
          202   XM(II,JJ)=CONST/2.
                IF(KMASS.EQ.0)XM(II,JJ)=CONST
          201   CONTINUE
          200   CONTINUE
                IF(I1.LT.3)GOTO203
                WRITE(6,170)
                WRITE(6,150)I1,N,(II,(AK(II,JJ),JJ=1,6),II=1,6)
                WRITE(6,190)
                WRITE(6,150)I1,N,(II,(XM(II,JJ),JJ=1,6),II=1,6)
          203   CONTINUE
                IF(I1.GE.3)WRITE(6,120)I1,N,CONST,AREAM
                RETURN
        C       ERROR EXIT FOR BAD CONNECTIONS
          220   WRITE(6,100)N
           50   CONTINUE
                STOP
          100   FORMAT(34H1ZERO OR NEGATIVE AREA ELEMENT NO.,I4/21H0EXECUTION TERM
               1INATED)
          120   FORMAT(12H0DEBUG LEVEL,I4/23H0PARAMETERS FOR EL. NO.,I4/1H ,5X,6HC
               1ONST=,E12.5,3X,5HAREA=,E10.4)
          140   FORMAT(12H0DEBUG LEVEL,I4/26H0LOCAL COORDINATES FOR EL.,I4,6X,1HB,
               18X,1HC/3(1H ,32X,F8.3,2X,F8.3/))
          150   FORMAT(12H0DEBUG LEVEL,I4/8H0EL. NO.,I4/6(4H ROW,I4,3X,6(E12.5,3X)
               1/))
          160   FORMAT(12H0DEBUG LEVEL,I4/24H0EL. CONNECTIONS FOR EL.,I4,3X,2HI=,I
```

```
                  I4,3X,2HJ#,I4,3X,2HK#,I4)
0125        170   FORMAT(27H0ELEMENTAL STIFFNESS MATRIX)
0126        190   FORMAT(22H0ELEMENTAL MASS MATRIX)
0127        207   FORMAT(17H0OUTPUT OF STIFT1)
0128        223   FORMAT(27H0STRAIN-DISPLACEMENT MATRIX/3(1H ,3X,3HROW,I4,3X,6(E12.5
                 1,3X)/))
0129        227   FORMAT(21H0STRESS-STRAIN MATRIX/3(1H ,3X,3HROW,I4,3X,3(E12.5,3X)/)
                 1)
0130        260   FORMAT(36H0ELEMENTAL STRESSES AT ITERATION NO.,I4/8H ELEMENT,I4,2X
                 1,7HSIGMAX=,E12.5,2X,7HSIGMAY=,E12.5,2X,8HSIGMAXY=,E12.5)
0131              END
```

```
FORTRAN IV G1  RELEASE 2.0          STIFT2            DATE = 77075        18/22/49          PAGE 0001

0001                    SUBROUTINE STIFT2(N)
                  C     CALCULATES INERTIAL AND FLUIDITY MATRICES FOR FLUID PLANE
                  C     TRIANGULAR ELEMENT
                  C     FOR SUBROUTINE STIFT2%N<
                  C     I,J,K          ELEMENT CONNECTIONS--LATER USED AS LOOP COUNTERS
                  C     BN%NCN<,CN%NCN< LOCAL COORDINATES OF FLUID TRIANGLE
                  C     AREAN          AREA OF TRIANGULAR FLUID ELEMENT
                  C     BB             PARTIAL DERIVATIVE OF P WITH RESPECT TO X
                  C     CC             PARTIAL DERIVATIVE OF P WITH RESPECT TO Y
0002                    COMMON/L48/TITLE(12),URT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                       1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                       2,CUCR(63,2)
0003                    COMMON/L46/RO(189),R1(189),R(9),RR(3),R4(189),LINDX(85)
                       1,DISP(20,3),INDEX(85),INACT(85),R5(189),NACT(85)
                       2,R6(189),R6F(189)
0004                    COMMON/TEST1/A(3,6),AK(9,9),BACK(3,6),XM(9,9)
0005                    COMMON/FUNCT/YF,YI,AN(3),AM(3),BN(3),BM(3),CN(3),CM(3),AM,AREAN
                       1,AREAM,XI,XF,SLOPE,YO
0006                    COMMON/L48/FF(90)
0007                    COMMON/E46/Z(90,90),D(90),DL(90)
0008                    COMMON/MOD46/F(189)
0009                    COMMON/STR/SIGMA(3),UI(88),VI(88)
0010                    COMMON/PARAM/T,ITIME,LFT,KKK,TEND,NCN,NDFT,NDT,MFREQ
                       1,NP,NE,NB,NLD,NMAT,II,NPRINT,ND,NDF,NIT,TBEG,UO,VO,ALPHA,KMASS
                       2,NBCE,NSZF,MSZF,NDCE,NEZF,NEEQ
                       3,NTIME,DELT,NSIZE,T1
                       4,BB,CC,NUNLIN,DAMP
                  C
                  C     SET UP FREEDOM ARRAY FF IN STRESS PASS
0011                    IF(KKK.NE.2)GOTO225
0012                    DO230I=1,90
0013              230   FF(I)=0.
0014                    DO235LL=1,NCN
0015                    MDF=1
0016                    NN=NOP(N,LL)
0017                    NROWB=(NN-1)*MDF+MSZF
0018                    IF(NDF.EQ.1)NROWB=NROWB-MSZF
0019                    IROW=(LL-1)*MDF
0020                    DO24CMM=1,MDF
0021                    NROWB=NROWB+1
0022                    IROW=IROW+1
0023                    FF(IROW)=F(NROWB)
0024              240   CONTINUE
0025              235   CONTINUE
0026                    MTIME=ITIME
0027              225   CONTINUE
                  C
                  C     DETERMINE ELEMENT CONNECTIONS
0028                    I=NOP(N,1)
0029                    J=NOP(N,2)
0030                    K=NOP(N,3)
0031                    L=IMAT(N)
0032                    IF(II.GE.3)WRITE(6,20)
```

```
FORTRAN IV G1  RELEASE 2.0            STIFT2          DATE = 77075         18/22/49             PAGE 0002

0033                   IF(II.GE.3)WRITE(6,160)II,N,I,J,K
                 C
                 C     SET UP LOCAL COORDINATE SYSTEM
0034                   CN(3)=CORD(J,1)-CORD(I,1)
0035                   CN(2)=CORD(I,1)-CORD(K,1)
0036                   CN(1)=CORD(K,1)-CORD(J,1)
0037                   BN(3)=CORD(I,2)-CORD(J,2)
0038                   BN(2)=CORD(K,2)-CORD(I,2)
0039                   BN(1)=CORD(J,2)-CORD(K,2)
0040                   IF(II.GE.2)WRITE(6,140)II,N,(BN(I),CN(I),I=1,NCN)
0041                   AREAN=(CN(3)*BN(2)-CN(2)*BN(3))/2.
                 C     ERROR EXIT FOR BAD CONNECTIONS
0042                   IF(AREAN.LE.0.)GOTO220
                 C     LOGIC FOR STRESS PASS
0043                   IF(KKK.EQ.2)GOTO206
0044                   DD=12.*ORI(L,1)*ORI(L,1)
                 C
                 C     FORM ELEMENT FLUIDITY AND INERTIA MATRICES
0045                   DO200II=1,3
0046                   DO201JJ=1,3
0047                   AK(II,JJ)=(BN(II)*BN(JJ)+CN(II)*CN(JJ))/(4.*AREAN)
                 C     DISTRIBUTED MASS MATRIX--LOGIC FOR DIAGONAL MATRIX
                 C     IF KMASS = 0 , LUMPED MASS REPRESENTATION
                 C     IF KMASS.NE.0 , DISTRIBUTED MASS REPRESENTATION
0048                   IF(II.EQ.JJ)GOTO202
0049                   XM(II,JJ)=0.
0050                   IF(KMASS.NE.0)XM(II,JJ)=AREAN/DD
0051                   GOTO201
0052             202   XM(II,JJ)=4.*AREAN/DD
0053                   IF(KMASS.NE.0)XM(II,JJ)=2.*AREAN/DD
0054             201   CONTINUE
0055             200   CONTINUE
0056                   IF(II.LT.3)GOTO203
0057                   WRITE(6,170)
0058                   WRITE(6,150)II,N,(II,(AK(II,JJ),JJ=1,3),II=1,3)
0059                   WRITE(6,190)
0060                   WRITE(6,150)II,N,(II,(XM(II,JJ),JJ=1,3),II=1,3)
0061             203   CONTINUE
                 C     SIGNIFIES END OF DISPLACEMENT PASS
0062                   IF(II.GE.3)WRITE(6,120)II,N,DD,AREAN
0063                   RETURN
0064             206   CONTINUE
                 C
                 C     FIND ELEMENT VELOCITIES
                 C
0065                   BB=0.
0066                   CC=0.
0067                   DO10IL=1,NCN
0068                   LL=IL
0069                   BB=BB+FF(LL)*BN(IL)
0070                   CC=CC+FF(LL)*CN(IL)
0071              10   CONTINUE
0072                   BB=BB*DELT/(2.*AREAN*ORT(L,3))+DAMP*DELT*UI(N)
```

```
FORTRAN IV G1  RELEASE 2.0            STIFT2            DATE = 77075          18/22/49             PAGE 0003

0073                 CC=CC+DELT/(2.*AREAN*ORT(L,3))+DAMP*DELT*VI(N)
0074                 SIGMA(1)=UI(N)-BB
0075                 SIGMA(2)=VI(N)-CC
               C     PLACE UPDATED VELOCITIES IN UI AND VI
               C     FOR USE IN NEXT TIME STEP
0076                 UI(N)=SIGMA(1)
0077                 VI(N)=SIGMA(2)
0078                 IF(I.EQ.I1)GOTO245
0079                 IF(NTIME.NE.ITIME)GOTO250
0080           245   IF(I1.GE.2)WRITE(6,101)BB,CC
0081                 IF(I1.GE.2)WRITE(6,115)DAMP
0082                 WRITE(6,110)NTIME,N,SIGMA(1),SIGMA(2)
0083           250   RETURN
               C     ERROR EXIT FOR BAD CONNECTIONS
0084           220   WRITE(6,100)N
0085            50   CONTINUE
0086                 STOP
0087            20   FORMAT(17H0OUTPUT OF STIFT2)
0088           100   FORMAT(34H1ZERO OR NEGATIVE AREA ELEMENT NO.,I4/21H0EXECUTION TERM
                    1INATED)
0089           101   FORMAT(20H0PARTIAL DERIVATIVES,5X,3HBB#,E12.5,3X,3HCC#,E12.5)
0090           110   FORMAT(38H0ELEMENTAL VELOCITIES AT ITERATION NO.,I4/8H ELEMENT,I4,
                    12X,2HU=,E12.5,2X,2HV=,E12.5)
0091           115   FORMAT(19H0ELEMENTAL DAMPING=,E12.5)
0092           120   FORMAT(12H0DEBUG LEVEL,I4/23H0PARAMETERS FOR EL. NO.,I4/1H ,5X,3HD
                    1D#,E12.5,3X,5HAREA#,F10.4)
0093           140   FORMAT(12H0DEBUG LEVEL,I4/26H0LOCAL COORDINATES FOR EL.,I4,6X,1HB,
                    18X,1HC/3(1H ,32X,F8.3,2X,F8.3/))
0094           150   FORMAT(12H0DEBUG LEVEL,I4/8H0EL. NO.,I4/3(4H ROW,I4,3X,3(E12.5,3X)
                    1/))
0095           160   FORMAT(12H0DEBUG LEVEL,I4/24H0EL. CONNECTIONS FOR EL.,I4,3X,2HI#,I
                    14,3X,2HJ#,I4,3X,2HK#,I4)
0096           170   FORMAT(26H0ELEMENTAL FLUIDITY MATRIX)
0097           190   FORMAT(25H0ELEMENTAL INERTIA MATRIX)
0098                 END
```

```
0001               SUBROUTINE STIFT3(N)
             C     FINDS MASS, INERTIA, STIFFNESS, FLUIDITY, AND COUPLING
             C     MATRICES FOR FLUID/SOLID ELEMENT
             C     FOR STIFT3%N<
             C     I,J,K          ELEMENT CONNECTIONS--LATER USED AS LOOP COUNTERS
             C     BN%NCN<,LN%NCN< LOCAL COORDINATES OF FLUID TRIANGLE
             C     BM%NCN<,CM%NCN< LOCAL COORDINATES OF SOLID TRIANGLE
             C     AN%NCN<,AM%NCN< LOCAL COORDINATES OF FLUID AND SOLID ELEMENTS
             C     S%3,6<         UNASSEMBLED COUPLING MATRIX
0002               COMMON/G48/TITLE(12),ORT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                  1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                  2,CORR(63,2)
0003               COMMON/L48/RO(189),RI(189),R(9),RR(3),R4(189),LINDX(85)
                  1,DISP(20,3),INDEX(85),INACT(85),R5(189),NACT(85)
                  2,R6(189),R6F(189)
0004               COMMON/TEST1/A(3,6),AK(9,9),BALK(3,6),XM(9,9)
0005               COMMON/ST348/S(3,6),ARRAY(90,90),XMF(6,6),DIL(6,6)
0006               COMMON/FUNC1/YF,YI,AN(3),AM(3),BN(3),BM(3),CN(3),CM(3),AH,AREAN
                  1,AREAM,XI,XF,SLOPE,YO
0007               COMMON/D48/FF(90)
0008               COMMON/E46/Z(90,90),D(90),DL(90)
0009               COMMON/MOL48/F(189)
0010               COMMON/STR/SIGMA(3),U1(88),V1(88)
0011               COMMON/PARAM/I,ITIME,LFT,KKK,TEND,NCN,NLF1,NDT,NFREQ
                  1,NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,NIT,TBEG,UO,VO,ALPHA,KMASS
                  2,NECE,NSZF,MSZF,NUCE,NEZF,NEEQ
                  3,NTIME,DELT,NSIZE,T1
                  4,BE,CC,NONLIN,DAMP
             C     FIND ELEMENTAL DISPLACEMENT
             C     AND ACCELERATION ARRAYS USED
             C     TO FIND NONLINEAR FORCES
0012               DO290I=1,9
0013         290   R(I)=0.
0014               DO280I=1,90
0015         280   FF(I)=0.
0016               DO330LL=1,NEEQ
0017               MDF=2
0018               NN=NOP(N,LL)
0019               IF(LL.GT.NCN)MDF=1
0020               NROWB=(NN-1)*MDF
0021               IF(LL.GT.NCN)NROWB=NROWB+MSZF
0022               IF(MDF.EQ.1)NROWB=NROWB-MSZF
0023               IROW=(LL-1)*MDF
0024               IF(LL.GT.NCN)IROW=IROW+NCN
0025               IF(MDF.EQ.1)IROW=IROW-NCN
0026               DO320MM=1,MDF
0027               NROWB=NROWB+1
0028               IROW=IROW+1
0029               FF(IROW)=F(NROWB)
0030               R(IROW)=R6F(NROWB)
0031         320   CONTINUE
0032         330   CONTINUE
0033               MTIME=ITIME
```

```
0034      300 CONTINUE
            C
            C     FIND MATRICES FOR FLUID ELEMENT
0035              L=2
            C
            C     FIND DENSITY TO BE USED
            C     IN CALCULATING COUPLING MATRIX S
            C
0036              RHOF=ORT(L,3)
            C
            C     DETERMINE ELEMENT CONNECTIONS
0037              I=NOP(N,4)
0038              J=NOP(N,5)
0039              K=NOP(N,6)
0040       17     CONTINUE
0041              IF(II.GE.3)WRITE(6,110)
0042              IF(II.GE.3)WRITE(6,160)II,N,I,J,K
            C
            C     SET UP LOCAL COORDINATE SYSTEM
0043              CN(3)=CORD(J,1)-CORD(I,1)
0044              CN(2)=CORD(I,1)-CORD(K,1)
0045              CN(1)=CORD(K,1)-CORD(J,1)
0046              BN(3)=CORD(I,2)-CORD(J,2)
0047              BN(2)=CORD(K,2)-CORD(I,2)
0048              BN(1)=CORD(J,2)-CORD(K,2)
            C     FIND LOCAL COORDINATES AN
0049              AN(3)=CORD(I,1)*CORD(J,2)-CORD(J,1)*CORD(I,2)
0050              AN(2)=CORD(K,1)*CORD(I,2)-CORD(I,1)*CORD(K,2)
0051              AN(1)=CORD(J,1)*CORD(K,2)-CORD(K,1)*CORD(J,2)
0052              IF(II.GE.2)WRITE(6,140)II,N,(BN(I),CN(I),AN(I),I=1,NCN)
0053              AREAN=(CN(3)*BN(2)-CN(2)*BN(3))/2.
            C     ERROR EXIT FOR BAD CONNECTIONS
0054              IF(AREAN.LE.0.)GOTO220
0055              IF(ITIME.EQ.0)GOTO146
            C     LOGIC FOR STRESS PASS
0056              IF(KAK.EQ.2)GOTO206
0057              GOTO2061
0058      146     CONTINUE
0059              DD=12.*ORT(L,1)*ORT(L,1)
            C
            C     FORM ELEMENT FLUIDITY AND INERTIA MATRICES
0060              DO20II=1,3
0061              IJ=II+NEEQ
0062              DO20JJ=1,3
0063              JI=JJ+NEEQ
0064              AK(IJ,JI)=(BN(II)*BN(JJ)+CN(II)*CN(JJ))/(4.*AREAN)
            C
            C     DISTRIBUTED MASS MATRIX--LOGIC FOR DIAGONAL MATRIX
            C     IF KMASS = 0 , LUMPED MASS REPRESENTATION
            C     IF KMASS.NE.0 , DISTRIBUTED MASS REPRESENTATION
0065              IF(II.EQ.JJ)GOTO202
0066              XM(IJ,JI)=C.
0067              IF(KMASS.NE.0)XM(IJ,JI)=AREAN/DD
```

```
FORTRAN IV G1  RELEASE 2.0            STIFT3          DATE = 77075        18/22/49            PAGE 0003

0068                GOTO201
0069        202     XM(IJ,JI)=4.*AREAN/DD
0070                IF(KMASS.NE.0)XM(IJ,JI)=2.*AREAN/DD
0071        201     CONTINUE
0072        200     CONTINUE
0073                NC=NFEQ+1
0074                IF(I1.LT.3)GOTO203
0075                WRITE(6,170)
0076                WRITE(6,150)I1,N,(II,(AK(II,JJ),JJ=NC,NEZF),II=NC,NEZF)
0077                WRITE(6,190)
0078                WRITE(6,150)I1,N,(II,(XM(II,JJ),JJ=NC,NEZF),II=NC,NEZF)
0079        203     CONTINUE
            C       DISPLACEMENT PASS ONLY
0080                IF(I1.GE.3)WRITE(6,120)I1,N,DD,AREAN
0081                GOTO2061
0082        206     CONTINUE
            C
            C       FIND ELEMENT VELOCITIES
            C
0083                BB=0.
0084                CC=0.
0085                DO10IL=1,NCN
0086                LL=NEEQ+IL
0087                BB=BB+FF(LL)*BN(IL)
0088                CC=CC+FF(LL)*CN(IL)
0089         10     CONTINUE
0090                BB=BB*DELT/(2.*AREAN*RHOF)+DAMP*DELT*UI(N)
0091                CC=CC*DELT/(2.*AREAN*RHOF)+DAMP*DELT*VI(N)
0092                SIGMA(1)=UI(N)-BB
0093                SIGMA(2)=VI(N)-CC
            C       PLACE UPDATED VELOCITIES IN UI AND VI FOR USE IN NEXT
            C       TIME STEP
0094                UI(N)=SIGMA(1)
0095                VI(N)=SIGMA(2)
0096                IF(T.EQ.T1)GOTO245
0097                IF(NTIME.NE.ITIME)GOTO2061
0098        245     IF(I1.GE.2)WRITE(6,101)BB,CC
0099                IF(I1.GE.2)WRITE(6,287)DAMP
0100                WRITE(6,1102)NTIME,N,SIGMA(1),SIGMA(2)
0101       2061     CONTINUE
            C
            C       FIND MATRICES FOR SOLID ELEMENT
0102                M=N
0103                L=1
            C       DETERMINE ELEMENT CONNECTIONS
0104                I=NOP(N,1)
0105                J=NOP(N,2)
0106                K=NOP(N,3)
0107                IF(I1.GE.3)WRITE(6,1101)
0108                IF(I1.GE.3)WRITE(6,160)I1,M,I,J,K
            C       SET UP LOCAL COORDINATE SYSTEM
0109                CM(3)=CORD(J,1)-CORD(I,1)
0110                CM(2)=CORD(I,1)-CORD(K,1)
```

```
      CM(1)=CORD(K,1)-CORD(J,1)
      BM(3)=CORD(I,2)-CORD(J,2)
      BM(2)=CORD(K,2)-CORD(I,2)
      BM(1)=CORD(J,2)-CORD(K,2)
C     FIND LOCAL COORDINATES AM
      AM(3)=CORD(I,1)*CORD(J,2)-CORD(J,1)*CORD(I,2)
      AM(2)=CORD(K,1)*CORD(I,2)-CORD(I,1)*CORD(K,2)
      AM(1)=CORD(J,1)*CORD(K,2)-CORD(K,1)*CORD(J,2)
      IF(I1.GE.2)WRITE(6,140)I1,M,(BM(I),CM(I),AM(I),I=1,NCN)
      AREAM=(CM(3)*BM(2)-CM(2)*BM(3))/2.
C     ERROR EXIT FOR BAD CONNECTIONS
      IF(AREAM.LE.0.)GOTO220
      IF(ITIME.EQ.0)GOTO147
      IF(KKK.EQ.2)GOTO147
      GOTO148
  147 CONTINUE
C     FORM STRAIN-DISPLACEMENT MATRIX
      A(1,1)=-BM(3)-BM(2)
      A(1,2)=0.
      A(1,3)=BM(2)
      A(1,4)=0.
      A(1,5)=BM(3)
      A(1,6)=0.
      A(2,1)=0.
      A(2,2)=-CM(2)-CM(3)
      A(2,3)=0.
      A(2,4)=CM(2)
      A(2,5)=0.
      A(2,6)=CM(3)
      A(3,1)=-CM(2)-CM(3)
      A(3,2)=-BM(3)-BM(2)
      A(3,3)=CM(2)
      A(3,4)=BM(2)
      A(3,5)=CM(3)
      A(3,6)=BM(3)
      IF(I1.GE.1)WRITE(6,223)(I,(A(I,J),J=1,6),I=1,3)
C     FORM STRESS-STRAIN MATRIX
      CONST=ORT(L,1)/((1.-ORT(L,2)*ORT(L,2))*AREAM)
      AK(1,1)=CONST
      AK(1,2)=CONST*ORT(L,2)
      AK(1,3)=0.
      AK(2,1)=AK(1,2)
      AK(2,2)=AK(1,1)
      AK(2,3)=0.
      AK(3,1)=AK(1,3)
      AK(3,2)=AK(2,3)
      AK(3,3)=CONST*(1.-ORT(L,2))/2.
      IF(I1.GE.1)WRITE(6,227)(I,(AK(I,J),J=1,3),I=1,3)
C     BACK IS BACKSUBSTITUTION MATRIX
      DO251I=1,NCN
      DO224J=1,6
      BACK(I,J)=0.
      DO223K=1,3
```

```
0159      2231 BACK(I,J)=BACK(I,J)+(AK(I,K)/2.)*A(K,J)
0160      2241 CONTINUE
0161      2251 CONTINUE
          C    SO MUCH FOR STRESS PASS
          C    FIND STRESSES IF DESIRED
0162           IF(KKK.NE.2)GOTO2501
0163           DO2551I=1,3
0164           SIGMA(I)=0.
0165           DO2551K=1,6
0166      2551 SIGMA(I)=SIGMA(I)+BACK(I,K)*EE(K)
0167           IF(I.EQ.TI)GOTO2651
0168           IF(NTIME.NE.ITIME)GOTO2661
0169      2651 WRITE(6,1103)MTIME,N,(SIGMA(I),I=1,3)
0170      2661  CONTINUE
0171           RETURN
0172      2501 CONTINUE
          C    AK NOW BECOMES STIFFNESS MATRIX
0173           DO2121I=1,6
0174           DO2131J=1,6
0175           AK(I,J)=0.
0176           DO2141K=1,3
0177      2141 AK(I,J)=AK(I,J)+(BACK(K,I)/2.)*A(K,J)
0178      2131 CONTINUE
0179      2121 CONTINUE
          C    STIFFNESS MATRIX IS NOW CALCULATED
          C
          C    CALCULATE MASS MATRIX
0180           CONST=UKT(L,3)*AREAM/FLOAT(NCN)
0181           DO2001II=1,6
0182           DO2011JJ=1,6
          C    DISTRIBUTED MASS--LOGIC FOR DIAGONAL MATRIX
0183           XM(II,JJ)=0.
0184           IF(II.EQ.JJ)GOTO2021
0185           IF(KMASS.EQ.0)GOTO2011
0186           JI=II+JJ
0187           IJ=(JI/2)*2
0188           IF(IJ.NE.JI)GOTO2011
0189           XM(II,JJ)=CONST/4.
0190           GOTO2011
0191      2021 XM(II,JJ)=CONST/2.
0192           IF(KMASS.EQ.0)XM(II,JJ)=CONST
0193      2011 CONTINUE
0194      2001 CONTINUE
0195           IF(II.LT.3)GOTO2031
0196           WRITE(6,1701)
0197           WRITE(6,1501)II,M,(II,(AK(II,JJ),JJ=1,6),II=1, )
0198           WRITE(6,1901)
0199           WRITE(6,1501)II,M,(II,(XM(II,JJ),JJ=1,6),II=1,6)
0200      2031 CONTINUE
0201           IF(II.GE.3)WRITE(6,1201)II,M,CONST,AREAM
0202       148 CONTINUE
          C
          C    ESTABLISH COUPLING MATRIX S
```

```
                C      DETERMINE INTERVAL OVER WHICH INTEGRATION
                C      TAKES PLACE $IN Y ONLY<
                C      FIND COMMON NODES
      0203             IPASS=0
      0204             DO35I=1,6
      0205             IF(IPASS.EQ.2)GOTO35
      0206             NN=NUP(N,1)
      0207             IF(NTYPE(NN).NE.3)GOTO35
      0208             IPASS=IPASS+1
      0209             IF(IPASS.EQ.2)GOTO40
      0210             NODIN=NUP(N,I)
      0211             GOTO35
      0212         40  NODEM=NOP(N,I)
      0213         45  YI=CORD(NODEN,2)
      0214             YF=CORD(NODEM,2)
      0215             XI=CORD(NODEN,1)
      0216             XF=CORD(NODEM,1)
      0217             IF(XI-XF)46,47,46
                C      VERTICAL WALL
                C      AH IS CONSTANT VALUE OF X TO BE USED TO FIND SF
      0218         47  AH=XI
      0219             SLOPE=0.
      0220         46  IF(YI-YF)35,49,35
                C      HORIZONTAL WALL
      0221         49  AH=YI
      0222         35  CONTINUE
      0223             IF(XI.NE.XF)SLOPE=(YF-YI)/(XF-XI)
      0224             Y0=YI-SLOPE*XI
                C      FUNCTION SF$I,J< IS APPLICABLE ONLY TO SPECIFIC PROBLEMS
      0225             DO50I=1,3
      0226             DO55J=1,6
      0227             JJ=(J/2)*2
      0228             IF(JJ.EQ.J)GOTO60
      0229             JN=(J/2)+1
      0230             S(I,J)=SF(I,JN)*SIN(ALPHA)*RHOF
      0231             GOTO55
      0232         60  S(I,J)=SF(I,JN)*COS(ALPHA)*RHOF
      0233         55  CONTINUE
      0234         50  CONTINUE
                C      S NOW ESTABLISHED
      0235             IF(I1.LT.3)GOTO502
      0236             WRITE(6,180)N
      0237             WRITE(6,1502)I1,M,(II,(S(II,JJ),JJ=1,6),II=1,3)
      0238        502  CONTINUE
                C
                C
                C      FLUID FORCE ON SOLID ST*P/RHOF AND PUT INTO ARRAY
      0239             DO114I=1,NEEQ
      0240             ARRAY(I,1)=0.
      0241             DO114K=1,NCN
      0242             KK=NEEQ+K
      0243             ARRAY(I,1)=ARRAY(I,1)+S(K,I)*FE(KK)/RHOF
      0244        114  CONTINUE
```

```
                C
                C                        OUTPUT BLOCK
0245                  IF(I1.LT.3)GOTO144
0246                  WRITE(6,901)(ARRAY(I,1),I=1,NEEQ)
0247             144  CONTINUE
                C                        END OF OUTPUT BLOCK
                C
                C
                C     FIND ACCELERATION FORCE ON FLUID
                C     -S*R AND PUT IN ARRAY
0248                  DO143I=1,NCN
0249                  II=I+NEEQ
0250                  ARRAY(II,1)=0.
0251                  DO143K=1,NEEQ
0252                  ARRAY(II,1)=ARRAY(II,1)-S(I,K)*R(K)
0253             143  CONTINUE
                C
                C     OUTPUT BLOCK
0254                  IF(I1.LT.3)GOTO153
0255                  NC=NEEQ+1
0256                  WRITE(6,902)(ARRAY(II,1),II=NC,NEZF)
0257             153  CONTINUE
                C                        END OF OUTPUT BLOCK
                C
                C
                C     MAP FORCES INTO R TO BE ASSEMBLED
                C     INTO GLOBAL FORCE ARRAY R6 IN FORMK
                C     NONLINEAR FORCE CALCULATION THEN COMPLETED
0258                  DO115I=1,NEZF
0259                  R(I)=ARRAY(I,1)
0260             115  CONTINUE
0261             350  CONTINUE
0262                  RETURN
0263               1  WRITE(6,11)N,M
0264                  STOP
                C     ERROR EXIT FOR BAD CONNECTIONS
0265             220  WRITE(6,100)N
0266                  STOP
0267            2201  WRITE(6,100)M
0268                  STOP
0269              11  FORMAT(31H0INCORRECT INTERFACE ON EL. NO.,I4,3X,3HAND,I4/21H EXECU
                     1TION TERMINATED)
0270             100  FORMAT(34H0NEGATIVE OR ZERO AREA ELEMENT NO.,I4/21H0EXECUTION TERM
                     1INATED)
0271             101  FORMAT(20H0PARTIAL DERIVATIVES,5X,3HBB#,E12.5,3X,3HCC#,E12.5)
0272             110  FORMAT(20H0STIFT3 FLUID OUTPUT)
0273             120  FORMAT(12H0DEBUG LEVEL,I4/23H0PARAMETERS FOR EL. NO.,I4/1H ,5X,3HD
                     1D#,E12.5,3X,5HAREA#,E10.4)
0274             140  FORMAT(12H0DEBUG LEVEL,I4/26H0LOCAL COORDINATES FOR EL.,I4,6X,1HB,
                     18X,1HC,6X,1HA/3(1H ,32X,3(E10.4,2X)/))
0275             150  FORMAT(12H0DEBUG LEVEL,I4/8H0EL. NO.,I4/3(4H ROW,I4,3X,3(E12.5,3X)
                     1/))
0276             160  FORMAT(12H0DEBUG LEVEL,I4/24H0EL. CONNECTIONS FOR EL.,I4,3X,2HI#,I
```

```
                    14,3X,2HJ#,14,3X,2HK#,I4)
0277          170   FORMAT(26HOELEMENTAL FLUIDITY MATRIX)
0278          180   FORMAT(30HCCOUPLING MATRIX FOR FLUID EL.,I4)
0279          190   FORMAT(25HOELEMENTAL INERTIA MATRIX)
0280          223   FORMAT(27HOSTRAIN-DISPLACEMENT MATRIX/3(1H ,3X,3HROW,I4,3X,6(E12.5
                   1,3X)/))
0281          227   FORMAT(21HOSTRESS-STRAIN MATRIX/3(1H ,3X,3HROW,I4,3X,3(E12.5,3X)/)
                   1)
0282          287   FORMAT(19HOELEMENTAL DAMPING=,E12.5)
0283          396   FORMAT(12HODEBUG LEVEL,I4/10HOM INVERSE)
0284          897   FORMAT(25HOMATRIX PRODUCT MINV*KT*K)
0285          898   FORMAT(20HOVOLUME CHANGE ARRAY/1H ,10(E12.5,1X))
0286          899   FORMAT(20HOMATRIX PRODUCT KT*K)
0287          900   FORMAT(34HODILATATION MATRIX FOR ELEMENT NO.,I4)
0288          901   FORMAT(25HONONLINEAR FORCE ON SOLID/(1H ,2X,E12.5)/)
0289          902   FORMAT(28HOACCELERATION FORCE ON FLUID/(1H ,2X,E12.5)/)
0290          903   FORMAT(23HOTOTAL NONLINEAR FORCES/(1H ,2X,E12.5)/)
0291          993   FORMAT(4H ROW,I4,5(I4,1H.,1X,E12.5,2X)/7(1H ,7X,5(I4,1H.,1X,E12.5,
                   12X)/))
0292          1101  FORMAT(20HOSTIFT3 SOLID OUTPUT)
0293          1102  FORMAT(38HOELEMENTAL VELOCITIES AT ITERATION NO.,I4/8H ELEMENT,I4
                   1,2X,2HU=,E12.5,2X,2HV=,E12.5)
0294          1103  FORMAT(36HOELEMENTAL STRESSES AT ITERATION NO.,I4/8H ELEMENT,I4,2X
                   1,7HSIGMAX=,E12.5,2X,7HSIGMAY=,E12.5,2X,8HSIGMAXY=,E12.5)
0295          1201  FORMAT(12HODEBUG LEVEL,I4/23HOPARAMETERS FOR EL. NO.,I4/1H ,5X,6HC
                   1ONST#,E12.5,3X,5HAREA#,F10.4)
0296          1501  FORMAT(12HODEBUG LEVEL,I4/8HOEL. NO.,I4/6(4H ROW,I4,3X,6(E12.5,3X)
                   1/))
0297          1502  FORMAT(12HODEBUG LEVEL,I4/8HOEL. NO.,I4/3(4H ROW,I4,3X,6(E12.5,3X)
                   1/))
0298          1701  FORMAT(27HOELEMENTAL STIFFNESS MATRIX)
0299          1702  FORMAT(36HOELEMENTAL FLUID MASS MATRIX INVERSE)
0300          1901  FORMAT(22HOELEMENTAL MASS MATRIX)
0301                END
```

```
FORTRAN IV G1  RELEASE 2.0            SF               DATE = 77075          18/22/49                  PAGE 0001

0001                  FUNCTION SF(I,J)
0002                  COMMON/FUNCT/YF,YI,AN(3),AM(3),BN(3),BM(3),CN(3),CM(3),AH,AREAN
                     1,AREAM,XI,XF,SLOPE,YO
                C     CALCULATES FLUID/SOLID COUPLING FUNCTION FOR VERTICAL
                C     SOLID BOUNDARY
                C     FIND DIFFERENCE FUNCTIONS
0003                  DIFF=ABS(YF-YI)
0004                  DIFSQ=ABS(YF*YF-YI*YI)/2.
0005                  DIFCU=ABS(YF*YF*YF-YI*YI*YI)/3.
0006                  DIFFX=ABS(XF-XI)
0007                  DIFSQX=ABS(XF*XF-XI*XI)/2.
0008                  DIFCUX=ABS(XF*XF*XF-XI*XI*XI)/3.
0009                  IF(DIFF.EQ.0.)GOTO10
0010                  IF(DIFFX.NE.0.)GOTO20
                C     FIND LINEAR TERM
0011                  ALIN=(AN(I)*AM(J)+(AM(J)*BN(I)+AN(I)*BM(J))*AH+BN(I)*BM(J)*AH*AH)*
                     1DIFF
                C     FIND SQUARED TERM
0012                  SQU=(AM(J)+CN(I)+AN(I)*CM(J)+(BN(I)*CM(J)+BM(J)*CN(I))*AH)*DIFSQ
                C     FIND CUBIC TERM
0013                  CUB=CN(I)*CM(J)*DIFCU
0014                  GOTO30
0015             10   CONTINUE
                C     FOR HORIZONTAL WALL
                C     FIND LINEAR TERM
0016                  ALIN=(AN(I)*AM(J)+(AN(I)*CM(J)+AM(J)*CN(I))*AH+CN(I)*CM(J)*AH*AH)*
                     1DIFFX
                C     FIND SQUARED TERM
0017                  SQU=(AN(I)*BM(J)+AM(J)*BN(I)+(BN(I)*CM(J)+BM(J)*CN(I))*AH)*DIFSQX
                C     FIND CUBIC TERM
0018                  CUB=BN(I)*BM(J)*DIFCUX
0019                  GOTO30
0020             20   CONTINUE
                C     SLOPED WALL
                C     FIND LINEAR TERM
0021                  ALIN=(AN(I)*AM(J)+(AN(I)*CM(J)+AM(J)*CN(I))*YO+CN(I)*CM(J)*YO*YO)*
                     1DIFFX
                C     FIND SQUARED TERM
0022                  SQU=(AN(I)*BM(J)+AM(J)*BN(I)+(AN(I)*CM(J)+AM(J)*CN(I))*SLOPE+(BN(I
                     1)*CM(J)+BM(J)*CN(I))*YO+CN(I)*CM(J)*2.*YO*SLOPE)*DIFSQX
                C     FIND CUBIC TERM
0023                  CUB=(BN(I)*BM(J)+(BN(I)*CM(J)+BM(J)*CN(I))*SLOPE+CN(I)*CM(J)*SLOPE
                     1*SLOPE)*DIFCUX
0024             30   CONTINUE
                C     FIND FUNCTION
0025                  SF=(ALIN+SQU+CUB)/(4.*AREAN*AREAM)
0026                  RETURN
0027                  END
```

```
FORTRAN IV G1  RELEASE 2.0           MODAL              DATE = 77075        18/22/49              PAGE 0001

0001                   SUBROUTINE MODAL
                 C     THIS SUBROUTINE FINDS THE EIGENVECTORS AND EIGENVALUES OF THE
                 C     PROBLEM AND SETS UP Z FOR USE IN DISP
0002                   COMMON/L48/R0(189),R1(189),R(9),RB(3),R4(189),LINDX(85)
                      1,DISP(20,3),INDEX(85),INACT(85),R5(189),NACT(85)
                      2,R6(189),R6E(189)
0003                   COMMON/F48/R2(189),SM(189,189),SK(189,189)
0004                   COMMON/ST3-6/S(3,6),ARRAY(90,90),XME(6,6),DIL(6,6)
0005                   COMMON/E48/Z(90,90),D(90),DL(90)
0006                   COMMON/PARAM/I,ITIME,LET,KKK,TEND,NCN,NDET,NDT,MEREQ
                      1,NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,NIT,TBEG,UD,VD,ALPHA,KMASS
                      2,NECE,NSZE,MSZE,NDCE,NEZE,NEEQ
                      3,NTIME,DELT,NSIZE,I1
                      4,BB,CC,NONLIN,DAMP
0007                   DIMENSION SG(189,189),SH(189,189)
0008                   EQUIVALENCE (SM,SG),(SK,SH)
0009                   IF(ITIME.EQ.0)GOTO150
0010                   IF(NONLIN.EQ.0)GOTO150
0011                   GOTO170
0012             150   CONTINUE
                 C     REDUCE         A X = LAMBDA B X       TO
                 C                    K Z = LAMBDA Z
0013                   CALL REDUC1(SH,SG,NSIZE)
                 C                    OUTPUT BLOCK
0014                   IF(I1.LT.3)GOTO155
0015                   WRITE(6,810)I1
0016                   DO160I=1,NSIZE
0017             160   WRITE(6,993)I,(J,SH(I,J),J=1,NSIZE)
0018                   WRITE(6,820)
0019                   DO165I=1,NSIZE
0020             165   WRITE(6,993)I,(J,SG(I,J),J=1,NSIZE)
0021             155   CONTINUE
                 C                    END OF OUTPUT BLOCK
                 C
                 C     FIND EIGENVECTORS AND EIGENVALUES.
                 C     RETURNS EIGENVECTORS IN Z; EIGENVALUES IN D
0022                   CALL JACOBI(NSIZE,SH,1,M,Z)
                 C
                 C     PUT EIGENVALUES INTO D
0023                   DO190I=1,NSIZE
0024                   D(I)=SH(I,I)
0025             190   CONTINUE
                 C                    OUTPUT BLOCK
0026                   IF(I1.LT.0)GOTO1
0027                   WRITE(6,215)
0028                   DO221I=1,NSIZE
0029                   WRITE(6,226)(Z(I,J),J=1,NSIZE)
0030             221   CONTINUE
0031                   WRITE(6,200)
0032                   WRITE(6,266)(D(I),I=1,NSIZE)
0033               1   CONTINUE
                 C                    END OF OUTPUT BLOCK
                 C
```

```
              C      TRANSFORM Z BACK TO X
 0034                CALL REBAKA(SG,NSIZE)
              C                     OUTPUT BLOCK
 0035                IF(I1.LT.0)GOTO205
 0036                WRITE(6,860)I1
 0037                DO210I=1,NSIZE
 0038          210   WRITE(6,993)I,(J,Z(I,J),J=1,NSIZE)
 0039          205   CONTINUE
              C                     END OF OUTPUT BLOCK
              C
              C      FIND GENERALIZED FORCE ZT*F (RHS)
 0040                DO275I=1,NSIZE
 0041                ARRAY(I,1)=0.
 0042                DO275K=1,NSIZE
 0043          275   ARRAY(I,1)=ARRAY(I,1)+Z(K,I)*R1(K)
 0044                DO270I=1,NSIZE
 0045          270   R1(I)=ARRAY(I,1)
              C                     OUTPUT BLOCK
 0046                IF(I1.LT.2)GOTO280
 0047                WRITE(6,825)
 0048                WRITE(6,996)(R1(I),I=1,NSIZE)
 0049          280   CONTINUE
              C                     END OF OUTPUT BLOCK
              C
              C      MULTIPLY NONLINEAR LOADS BY ZT
              C      AND ADD TO LINEAR LOADS
 0050          170   CONTINUE
 0051                DO230I=1,NSIZE
 0052                ARRAY(I,1)=0.
 0053                DO230K=1,NSIZE
 0054          230   ARRAY(I,1)=ARRAY(I,1)+Z(K,I)*R6(K)
 0055                DO235I=1,NSIZE
 0056                R6(I)=ARRAY(I,1)
 0057                R1(I)=R1(I)+R6(I)
 0058          235   CONTINUE
              C                     OUTPUT BLOCK
 0059                IF(I1.LT.2)GOTO295
 0060                WRITE(6,835)
 0061                WRITE(6,996)(R6(I),I=1,NSIZE)
 0062                WRITE(6,825)
 0063                WRITE(6,996)(R1(I),I=1,NSIZE)
 0064          295   CONTINUE
              C                     END OF OUTPUT BLOCK
 0065                IF(ITIME.EQ.0)GOTO180
 0066                IF(NONLIN.EQ.0)GOTO180
 0067                RETURN
 0068          180   CONTINUE
              C      ADJUST ARRAY OF INITIAL VALUES FROM X TO Q
              C      FORM L MATRIX IN SG
 0069                DO300I=1,NSIZE
 0070                DO305J=1,NSIZE
 0071          305   SG(I,J)=0.
 0072          300   SG(I,I)=DL(I)
```

```
C                OUTPUT BLOCK
          IF(I1.LT.3)GOTO0335
          WRITE(6,845)
          DO340I=1,NSIZE
      340 WRITE(6,993)I,(J,SG(I,J),J=1,NSIZE)
      335 CONTINUE
C                END OF OUTPUT BLOCK
C         FORM MASS MATRIX L*LT AND PUT IN SG
          DO310J=1,NSIZE
          DO310I=1,NSIZE
          ARRAY(I,J)=0.
          DO310K=1,NSIZE
      310 ARRAY(I,J)=ARRAY(I,J)+SG(I,K)*SG(J,K)
          DO315I=1,NSIZE
          DO315J=1,NSIZE
      315 SG(I,J)=ARRAY(I,J)
C                OUTPUT BLOCK
          IF(I1.LT.3)GOTO0345
          WRITE(6,855)
          DO350I=1,NSIZE
      350 WRITE(6,993)I,(J,SG(I,J),J=1,NSIZE)
      345 CONTINUE
C                END OF OUTPUT BLOCK
C         FORM MATRIX PRODUCT ZI*M AND PUT IN SG
          DO320J=1,NSIZE
          DO320I=1,NSIZE
          ARRAY(I,J)=0.
          DO320K=1,NSIZE
      320 ARRAY(I,J)=ARRAY(I,J)+Z(K,I)*SG(K,J)
          DO325I=1,NSIZE
          DO325J=1,NSIZE
      325 SG(I,J)=ARRAY(I,J)
C                OUTPUT BLOCK
          IF(I1.LT.3)GOTO0355
          WRITE(6,875)
          DO360I=1,NSIZE
      360 WRITE(6,993)I,(J,SG(I,J),J=1,NSIZE)
      355 CONTINUE
C                END OF OUTPUT BLOCK
C         FORM MATRIX PRODUCT ZI*M*R2 AND PUT INTO R2
C         CONVERSION THEN COMPLETED
          DO330I=1,NSIZE
          ARRAY(I,1)=0.
          DO330K=1,NSIZE
      330 ARRAY(I,1)=ARRAY(I,1)+SG(I,K)*R2(K)
          DO370I=1,NSIZE
      370 R2(I)=ARRAY(I,1)
C                OUTPUT BLOCK
          IF(I1.LT.3)GOTO0365
          WRITE(6,865)
          WRITE(6,996)(R2(I),I=1,NSIZE)
      365 CONTINUE
C                END OF OUTPUT BLOCK
```

```
              C      FORM MATRIX PRODUCT ZT*M*R5 AND PUT INTO R4
              C      CONVERSION FOR INITIAL VELOCITY THEN COMPLETED
0114                 DO375I=1,NSIZE
0115                 ARRAY(I,1)=0.
0116                 DO375K=1,NSIZE
0117           375   ARRAY(I,1)=ARRAY(I,1)+SG(I,K)*R5(K)
0118                 DO380I=1,NSIZE
0119           380   R5(I)=ARRAY(I,1)
              C                     OUTPUT BLOCK
0120                 IF(I1.LT.3)GOTO385
0121                 WRITE(6,882)
0122                 WRITE(6,996)(R5(I),I=1,NSIZE)
0123           385   CONTINUE
              C                     END OF OUTPUT BLOCK
              C      PROBLEM IS NOW READY FOR ANALYTICAL SOLUTION
0124                 RETURN
0125           200   FORMAT(12H0EIGENVALUES)
0126           206   FORMAT(1H ,5X,E12.5/)
0127           215   FORMAT(13H0EIGENVECTORS)
0128           226   FORMAT(1H ,5X,7(E12.5,3X))
0129           810   FORMAT(12H0DEBUG LEVEL,I4/17H0OUTPUT OF REDUC1/13H0LINV*A*LINVT)
0130           815   FORMAT(12H0DEBUG LEVEL,I4/29H0GENERALIZED STIFFNESS MATRIX)
0131           820   FORMAT(31H0MATRIX B AS MODIFIED BY REDUC1)
0132           825   FORMAT(24H0RHS IN GENERALIZED FORM)
0133           830   FORMAT(12H0DEBUG LEVEL,I4/34H0A MATRIX AFTER TRIDIAGONALIZATION)
0134           835   FORMAT(28H0GENERALIZED NONLINEAR LOADS)
0135           845   FORMAT(9H0L MATRIX)
0136           855   FORMAT(27H0MASS MATRIX FORMED BY L*LT)
0137           860   FORMAT(12H0DEBUG LEVEL,I4/35H0EIGENVECTORS TRANSFORMED BACK TO X)
0138           865   FORMAT(55H0FINAL INITIAL DISPLACEMENT ARRAY CONVERTED FROM X TO Q)
0139           870   FORMAT(12H0DEBUG LEVEL,I4/19H0MATRIX PRODUCT M*Z)
0140           875   FORMAT(20H0MATRIX PRODUCT ZT*M)
0141           880   FORMAT(12H0DEBUG LEVEL,I4/24H0GENERALIZED MASS MATRIX)
0142           882   FORMAT(51H0FINAL INITIAL VELOCITY ARRAY CONVERTED FROM X TO Q)
0143           890   FORMAT(12H0DEBUG LEVEL,I4/19H0MATRIX PRODUCT K*Z)
0144           993   FORMAT(4H ROW,I4,5(I4,1H.,1X,E12.5,2X)/7(1H ,7X,5(I4,1H.,1X,E12.5,
                    12X)/))
0145           996   FORMAT(6(1H ,10(E12.5,1X)/))
0146                 END
```

```
      SUBROUTINE REDUC1(A,B,N)
C     SUBROUTINE TO REDUCE
C                     A X = LAMBDA B X
C     TO              K Z = LAMBDA Z
C     IN WHICH        K = LINV*A*LINVT
C                     Z = LT*X
C                     B = L*LT
C     L IS LOWER TRIANGULAR MATRIX
C     FORMS K IN A
      COMMON/E48/Z(90,90),D(90),DL(90)
      DIMENSION A(189,189),B(189,189)
      DO1I=1,N
      II=I-1
      DO2J=1,N
      X=B(I,J)
      IF(II.LT.1)GOTO15
      K=II
    3 X=X-B(I,K)*B(J,K)
      K=K-1
      IF(K.GE.1)GOTO3
   15 CONTINUE
      IF(I.EQ.J)GOTO4
      B(J,I)=X/Y
      GOTO2
    4 IF(X.LE.0.)GOTO5
      Y=SQRT(X)
      DL(I)=SQRT(X)
      GOTO2
    5 WRITE(6,100)
    2 CONTINUE
    1 CONTINUE
      DO6I=1,N
      Y=DL(I)
      II=I-1
      DO7J=1,N
      X=A(I,J)
      IF(II.LT.1)GOTO16
      K=II
    8 X=X-B(I,K)*A(J,K)
      K=K-1
      IF(K.GE.1)GOTO8
   16 CONTINUE
      A(J,I)=X/Y
    7 CONTINUE
    6 CONTINUE
      DO9J=1,N
      JJ=J-1
      DO10I=J,N
      X=A(I,J)
      II=I-1
      IF(II.LT.J)GOTO17
      K=II
   11 X=X-A(K,J)*B(I,K)
```

```
0046          K=K-1
0047          IF(K.GE.1)GOTO11
0048     17   CONTINUE
0049          IF(JJ.LT.1)GOTO18
0050          K=JJ
0051     12   X=X-A(J,K)*B(I,K)
0052          K=K-1
0053          IF(K.GE.1)GOTO12
0054     18   CONTINUE
0055          A(I,J)=X/DL(I)
0056     10   CONTINUE
0057      9   CONTINUE
0058          DO13J=1,N
0059          DO14I=1,N
0060          IF(J.GE.I)GOTO14
0061          A(J,I)=A(I,J)
0062     14   CONTINUE
0063     13   CONTINUE
0064          RETURN
0065    100   FORMAT(27HGB IS NOT POSITIVE DEFINITE)
0066          END
```

```
FORTRAN IV G1  RELEASE 2.0          JACOBI          DATE = 77075        18/22/49          PAGE 0001

0001                   SUBROUTINE JACOBI (N,Q,JVEC,M,V)
                 C     SUBPROGRAM FOR DIAGONALIZATION OF MATRIX Q BY SUCCESSIVE ROTATIONS
                 C
                 C     THE ARGUMENTS USED IN THE SUBROUTINE JACOBI ARE DEFINED AS FOLLOWS
                 C
                 C     N       ORDER OF THE GIVEN REAL SYMMETRIC MATRIX, N .GE. 2
                 C     Q       THE MATRIX Q TO BE DIAGONALIZED.  THIS INPUT MATRIX IS LATER
                 C     DESTROYED
                 C     JVEC       A FIXED POINT INDEX
                 C             JVEC = 0 INDICATES EIGENVALUES ALONE ARE TO BE FOUND
                 C            JVEC .NE. 0 INDICATES EIGNEVALUES AND EIGENVECTORS ARE TO BE
                 C            FOUND
                 C     M       THE NUMBER OF ROTATIONS PERFORMED
                 C     V      STORAGE FOR EIGENVECTORS, REQUIRED EVEN IF JVEC = 0
                 C     KUO, SHAN S., NUMERICAL METHODS AND COMPUTERS, ADDISON-WESLEY, 1965
0002                   DIMENSION Q(189,189),V(90,90),X(90),IH(90)
                 C
                 C     NEXT 8 STATEMENTS FOR SETTING INITIAL VALUES OF MATRIX V
                 C
0003                   IF(JVEC)10,15,10
0004              10   DO14I=1,N
0005                   DO14J=1,N
0006                   IF(I-J)12,11,12
0007              11   V(I,J)=1.
0008                   GOTO14
0009              12   V(I,J)=0.
0010              14   CONTINUE
                 C
0011              15   M=0
                 C     NEXT 8 STATEMENTS SCAN FOR LARGEST OFF DIAG. ELEM. IN EACH ROW
                 C     X(I) CONTAINS LARGEST ELEMENT IN  ITH ROW
                 C     IH(I) HOLDS SECOND SUBSCRIPT DEFINING POSITION OF ELEMENT
                 C
0012              17   MI=N-1
0013                   DO30I=1,MI
0014                   X(I)=0.
0015                   MJ=I+1
0016                   DO30J=MJ,N
0017                   IF(X(I)-ABS(Q(I,J)))20,20,30
0018              20   X(I)=ABS(Q(I,J))
0019                   IH(I)=J
0020              30   CONTINUE
                 C
                 C     NEXT 7 STATEMENTS FIND FOR MAXIMUM OF X(I)S FOR PIVOT ELEMENT
                 C
0021              40   DO70I=1,MI
0022                   IF(I-1)60,60,45
0023              45   IF(XMAX-X(I))60,70,70
0024              60   XMAX=X(I)
0025                   IP=I
0026                   JP=IH(I)
0027              70   CONTINUE
                 C
```

```
      C     NEXT TWO STATEMENTS TEST FOR XMAX, IF LESS THAN 10**-8, GO TO 1000
      C
            EPSI=1.E-08
            IF(XMAX-EPSI)1000,1000,148
      C
        148 M=M+1
      C
      C     NEXT 11 STATEMENTS FOR COMPUTING TANG, SIN, COS, Q(I,I), Q(J,J)
      C
            IF(Q(IP,IP)-Q(JP,JP))150,151,151
        150 TANG=-2.*Q(IP,JP)/(ABS(Q(IP,IP)-Q(JP,JP))+SQRT((Q(IP,IP)-Q(JP,JP))
           1**2+4.*Q(IP,JP)**2))
            GOTO160
        151 TANG=2.*Q(IP,JP)/(ABS(Q(IP,IP)-Q(JP,JP))+SQRT((Q(IP,IP)-Q(JP,JP))*
           1*2+4.*Q(IP,JP)**2))
        160 COSN=1./SQRT(1.+TANG**2)
            SINE=TANG*COSN
            QII=Q(IP,IP)
            Q(IP,IP)=COSN**2*(QII+TANG*(2.*Q(IP,JP)+TANG*Q(JP,JP)))
            Q(JP,JP)=COSN**2*(Q(JP,JP)-TANG*(2.*Q(IP,JP)-TANG*QII))
      C
            Q(IP,JP)=0.
      C
      C     NEXT 4 STATEMENTS FOR PSUEDO RANK THE EIGENVALUES
      C
            IF(Q(IP,IP)-Q(JP,JP))152,153,153
        152 TEMP=Q(IP,IP)
            Q(IP,IP)=Q(JP,JP)
            Q(JP,JP)=TEMP
      C
      C     NEXT 6 STATEMENTS ADJUST SIN, COS FOR COMPUTATION OF Q(I,K), V(I,K)
      C
            IF(SINE)154,155,155
        154 TEMP=COSN
            GOTO170
        155 TEMP=-COSN
        170 COSN=ABS(SINE)
            SINE=TEMP
      C
      C     NEXT 10 STATEMENTS FOR INSPECTING THE IHS BETWEEN I+1 AND N-1 TO
      C     DETERMINE WHETHER A NEW MAXIMUM VALUE SHOULD BE COMPUTED SINCE
      C     THE PRESENT MAXIMUM IS IN THE I OR J ROW
      C
        153 DO350I=1,MI
            IF(I-IP)210,350,200
        200 IF(I-JP)210,350,210
        210 IF(IH(I)-IP)230,240,230
        230 IF(IH(I)-JP)350,240,350
        240 K=IH(I)
        250 TEMP=Q(I,K)
            Q(I,K)=0.
            MJ=I+1
            X(I)=0.
```

```
                C
                C     NEXT 5 STATEMENTS SEARCH IN DEPLETED ROW FOR NEW MAXIMUM
                C
0061                  DO320J=HJ,N
0062                  IF(X(I)-ABS(Q(I,J)))300,300,320
0063          300     X(I)=ABS(Q(I,J))
0064                  IH(I)=J
0065          320     CONTINUE
0066                  Q(I,K)=TEMP
0067          350     CONTINUE
                C
0068                  X(IP)=0.
0069                  X(JP)=0.
                C
                C     NEXT 30 STATEMENTS FOR CHANGING THE OTHER ELEMENTS OF Q
                C
0070                  DU53CI=1,N
                C
0071                  IF(I-IP)370,530,420
0072          370     TEMP=Q(I,IP)
0073                  Q(I,IP)=COSN*TEMP+SINE*Q(I,JP)
0074                  IF(X(I)-ABS(Q(I,IP)))380,390,390
0075          380     X(I)=ABS(Q(I,IP))
0076                  IH(I)=IP
0077          390     Q(I,JP)=-SINE*TEMP+COSN*Q(I,JP)
0078                  IF(X(I)-ABS(Q(I,JP)))400,530,530
0079          400     X(I)=ABS(Q(I,JP))
0080                  IH(I)=JP
0081                  GOTO530
                C
0082          420     IF(I-JP)430,530,480
0083          430     TEMP=Q(IP,I)
0084                  Q(IP,I)=COSN*TEMP+SINE*Q(I,JP)
0085                  IF(X(IP)-ABS(Q(IP,I)))440,450,450
0086          440     X(IP)=ABS(Q(IP,I))
0087                  IH(IP)=I
0088          450     Q(I,JP)=-SINE*TEMP+COSN*Q(I,JP)
0089                  IF(X(I)-ABS(Q(I,JP)))400,530,530
                C
0090          480     TEMP=Q(IP,I)
0091                  Q(IP,I)=COSN*TEMP+SINE*Q(JP,I)
0092                  IF(X(IP)-ABS(Q(IP,I)))490,500,500
0093          490     X(IP)=ABS(Q(IP,I))
0094                  IH(IP)=I
0095          500     Q(JP,I)=-SINE*TEMP+COSN*Q(JP,I)
0096                  IF(X(JP)-ABS(Q(JP,I)))510,530,530
0097          510     X(JP)=ABS(Q(JP,I))
0098                  IH(JP)=I
0099          530     CONTINUE
                C
                C     NEXT 6 STATEMENTS TEST FOR COMPUTATION OF EIGENVECTORS
                C
0100                  IF(JVEC)540,40,540
```

```
FORTRAN IV G1  RELEASE 2.0        JACOBI          DATE = 77075        18/22/49          PAGE 0004

      540  DO550I=1,N
           TEMP=V(I,IP)
           V(I,IP)=COSN*TEMP+SINE*V(I,JP)
      550  V(I,JP)=-SINE*TEMP+COSN*V(I,JP)
           GOTO40
     1000  RETURN
           END
```

```
FORTRAN IV G1  RELEASE 2.0          REBAKA          DATE = 77075          18/22/49          PAGE 0001

0001                    SUBROUTINE REBAKA(B,N)
                  C     PERFORMS THE OPERATION
                  C                    X = LINVT*Z
                  C     WHERE Z IS THE ARRAY OF EIGENVECTORS
0002                    COMMON/E46/Z(90,90),D(90),DL(90)
0003                    DIMENSION B(189,189)
0004                    DO1J=1,N
0005                    I=N
0006                2   X=Z(I,J)
0007                    II=I+1
0008                    IF(II.GT.N)GOTO4
0009                    DO3K=II,N
0010                    X=X-B(K,I)*Z(K,J)
0011                3   CONTINUE
0012                4   CONTINUE
0013                    Z(I,J)=X/DL(I)
0014                    I=I-1
0015                    IF(I.GE.1)GOTO2
0016                1   CONTINUE
0017                    RETURN
0018                    END
```

```
FORTRAN IV G1  RELEASE 2.0           DIS              DATE = 77075        18/22/49              PAGE 0001

 0001                 SUBROUTINE DIS
 0002                 COMMON/C48/TITLE(12),ORT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                     1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                     2,CODR(63,2)
 0003                 COMMON/L48/RO(189),R1(189),R(9),RR(3),R4(189),LINUX(85)
                     1,DISP(20,3),INDEX(85),INACT(85),R5(189),NACT(85)
                     2,R6(189),R6F(189)
 0004                 COMMON/E48/R2(189),SM(189,189),SK(189,189)
 0005                 COMMON/ST348/S(3,6),ARRAY(90,90),XMF(6,6),DIL(6,6)
 0006                 COMMON/D48/EE(90)
 0007                 COMMON/E48/Z(90,90),D(90),DL(90)
 0008                 COMMON/PARAM/T,ITIME,LFT,KKK,IEND,NCN,NUFT,NDT,MEREQ
                     1,NP,NE,NB,NLD,NMAT,II,NPRINT,ND,NDF,NIT,IBEG,UO,VO,ALPHA,KMASS
                     2,NBCE,NSZE,MSZE,NDCE,NEZE,NEEQ
                     3,NTIME,DELT,NSIZE,T1
                     4,BE,CC,NONLIN,DAMP
 0009                 DIMENSION SG(189,189),SH(189,189)
 0010                 EQUIVALENCE (SM,SG),(SK,SH)
 0011                 IF(T.EQ.T1)GOTO210
 0012                 IF(NTIME.NE.ITIME)GOTO211
 0013           210   CONTINUE
 0014                 WRITE(6,720)T,DELT,ITIME,NTIME,NPRINT,T1,II
 0015           211   CONTINUE
 0016                 TIME=DELT
 0017                 IF(NONLIN.EQ.0)TIME=T
                C
                C     DAMPING ARRAY ZET SHOULD BE DIVIDED BY THE DENSITY
                C     OF THE FLUID IN THE FLUID MODES ONLY
                C
 0018                 DO1I=1,NSIZE
                C     SET UP NATURAL AND DAMPED FREQUENCIES AND THIER RATIO
 0019                 OMEGAN=SQRT(SK(I,I))
                C
                C     SET UP DAMPING RATIO FOR EACH MODE BASED ON CONSTANT DAMP
 0020                 ZET(I)=DAMP/(2.*OMEGAN)
                C
 0021                 RATIO=1.-ZET(I)*ZET(I)
 0022                 IF(RATIO.LE.0.)RATIO=-RATIO
 0023                 RATIO=SQRT(RATIO)
 0024                 OMEGAD=OMEGAN*RATIO
 0025                 IF(SK(I,I).EQ.0.)GOTO20
                C     MODIFY INITIAL DISPLACEMENTS TO XO=F/(M*OMEGANSQ)
 0026                 RO(I)=R2(I)-R1(I)/SK(I,I)
 0027                 IF(ZET(I)-1.)40,50,60
                C     ZETA  .LT.  1.
                C     UNDERDAMPED CASE
 0028            40   CONTINUE
                C     SET UP ARGUMENTS
 0029                 ARGE=-ZET(I)*OMEGAN*TIME
 0030                 ARGT=OMEGAD*TIME
                C     FIND SOLUTION
 0031                 EE(I)=EXP(ARGE)*(RO(I)*COS(ARGT)+(R5(I)/OMEGAD+RO(I)*ZET(I)/RATIO)
                     1*SIN(ARGT))+R1(I)/SK(I,I)
```

```
FORTRAN IV G1  RELEASE 2.0          DIS             DATE = 77075          18/22/49              PAGE 0002

                  C      CALCULATE VELOCITY FOR USE IN NEXT TIME STEP
      0032               IF(NONLIN.EQ.0)GOTO70
      0033               R5(I)=-ZET(I)*OMEGAN*(FF(I)-R1(I)/SK(I,I))+EXP(ARGE)*(-RO(I)*OMEGA
                        1D*SIN(ARGT)+(R5(I)+ZET(I)*OMEGAN*RO(I))*COS(ARGT))
      0034               GOTO70
                  C      ZETA .EQ. 1.
                  C      CRITICALLY DAMPED CASE
      0035            50 CONTINUE
      0036               ARGE=-OMEGAN*TIME
                  C      FIND FUNCTION
      0037               FF(I)=RO(I)*(1.-ARGE)*EXP(ARGE)+R1(I)/SK(I,I)+R5(I)*TIME*EXP(ARGE)
                  C      CALCULATE VELOCITY FOR USE IN NEXT TIME STEP
      0038               IF(NONLIN.EQ.0)GOTO70
      0039               R5(I)=-OMEGAN*(FF(I)-R1(I)/SK(I,I))+(OMEGAN*RO(I)+R5(I))*EXP(ARGE)
      0040               GOTO70
                  C      ZETA .GT. 1.
                  C      OVERDAMPED CASE
      0041            60 CONTINUE
      0042               ARG1=-OMEGAN*(ZET(I)-RATIO)
      0043               ARG2=-OMEGAN*(ZET(I)+RATIO)
      0044               ARGE1=ARG1*TIME
      0045               ARGE2=ARG2*TIME
      0046               CONST1=(R5(I)/OMEGAN+RO(I)*(ZET(I)+RATIO))/(2.*RATIO)
      0047               CONST2=(R5(I)/OMEGAN+RO(I)*(ZET(I)-RATIO))/(-2.*RATIO)
                  C      FIND FUNCTION
      0048               FF(I)=CONST1*EXP(ARGE1)+CONST2*EXP(ARGE2)+R1(I)/SK(I,I)
                  C      CLACULATE VELOCITY FOR USE IN NEXT TIME STEP
      0049               IF(NONLIN.EQ.0)GOTO70
      0050               R5(I)=CONST1*ARG1*EXP(ARGE1)+CONST2*ARG2*EXP(ARGE2)
      0051               GOTO70
                  C      RIGID BODY CASE
      0052            20 CONTINUE
                  C      FIND FUNCTION
      0053               FF(I)=R1(I)*TIME*TIME/2.+R5(I)*TIME+R2(I)
                  C      FIND VELOCITY FOR USE IN NEXT TIME STEP
      0054               IF(NONLIN.EQ.0)GOTO70
      0055               R5(I)=R1(I)*TIME+R5(I)
      0056            70 CONTINUE
                  C
                  C      REMOVE NONLINEAR LOADS FROM
                  C      R1 SO THAT NEW LOADS MAY
                  C      BE ADDED IN LATER
                  C
      0057               IF(NONLIN.EQ.1)R1(I)=R1(I)-R6(I)
                  C
                  C      FIND ACCELERATION USING VELOCITY AND
                  C      DISPLACEMENT
      0058               IF(NONLIN.EQ.0)R6(I)=R1(I)-2.*ZET(I)*OMEGAN*R5(I)-OMEGAN*OMEGAN*FF
                        1(I)
      0059               IF(NONLIN.EQ.1)R6(I)=R1(I)+R6(I)-2.*ZET(I)*OMEGAN*R5(I)-OMEGAN*OME
                        1GAN*FF(I)
                  C      PUT SOLUTION INTO INITIAL DISPLACEMENT ARRAY FOR NEXT TIME STEP
      0060               IF(NONLIN.EQ.1)R2(I)=FF(I)
```

```
FORTRAN IV G1  RELEASE 2.0          DIS          DATE = 77098        16/24/57          PAGE 0003

                 C      IT IS NOT NECESSARY TO CONVERT VELOCITY ARRAY TO X
                 C      AS IT IS NOT NEEDED IN FINAL FORM
                 C                      OUTPUT BLOCK
 0061                   IF(T.EQ.T1)GOTO208
 0062                   IF(NTIME.NE.ITIME )GOTO209
 0063                   IF(I1.LT.3)GOTO30
 0064             208   CONTINUE
 0065                   WRITE(6,800)OMEGAN,RATIO,OMEGAD,ARGE,ARGE1,ARGE2,GAMMA,ARGT,CONST1
                       1,CONST2,ZET(I)
 0066             209   CONTINUE
 0067              30   CONTINUE
                 C                      END OF OUTPUT BLOCK
                 C                      OUTPUT BLOCK
 0068                   IF(T.EQ.T1)GOTO225
 0069                   IF(NTIME.NE.ITIME )GOTO206
 0070                   IF(I1.LT.3)GOTO1
 0071             225   CONTINUE
 0072                   WRITE(6,605)I,FF(I)
 0073             206   CONTINUE
 0074               1   CONTINUE
                 C      PROBLEM HAS NOW BEEN SOLVED ANALYTICALLY
                 C      FIND FREEDOMS FF=Z*FF USING ARRAY
 0075                   DO111I=1,NSIZE
 0076                   ARRAY(I,1)=0.
 0077                   DO111K=1,NSIZE
 0078             111'  ARRAY(I,1)=ARRAY(I,1)+Z(I,K)*FF(K)
 0079                   DO110I=1,NSIZE
 0080             110   FF(I)=ARRAY(I,1)
                 C
                 C      TRANSFORM ACCELERATIONS R6 = Z*R6 USING ARRAY
 0081                   DO230I=1,NSIZE
 0082                   ARRAY(I,1)=0.
 0083                   DO230K=1,NSIZE
 0084             230   ARRAY(I,1)=ARRAY(I,1)+Z(I,K)*R6(K)
 0085                   DO235I=1,NSIZE
 0086                   R6(I)=ARRAY(I,1)
 0087             235   CONTINUE
                 C
                 C      INCREMENT STRESS PASS CONTROL
 0088                   IF(KKK.EQ.1)KKK=2
 0089                   IF(T.EQ.T1)GOTO205
 0090                   IF(NTIME.NE.ITIME)RETURN
 0091             205   CONTINUE
 0092                   IF(KKK.GE.2)WRITE(6,840)KKK
 0093                   WRITE(6,810)
 0094                   NSAVE=C
 0095                   DO120I=1,NSIZE
 0096                   IF(NACT(I).GE.LINDX(I))GOTO125
 0097                   NODE=(NACT(I)+1)/2
 0098                   IF(NODE.NE.NSAVE)GOTO130
 0099                   WRITE(6,845)NODE,FF(I)
 0100                   GOTO120
 0101             130   WRITE(6,850)NODE,FF(I)
```

```
0102              NSAVE=NODE
0103              GOTO120
0104         125  NODE=NACT(I)-MSZF
0105              WRITE(6,855)NODE,FF(I)
0106         120  CONTINUE
0107              IF(I1.LT.2)RETURN
0108              WRITE(6,815)
0109              WRITE(6,820)(R2(I),I=1,NSIZE)
0110              WRITE(6,825)
0111              WRITE(6,820)(R5(I),I=1,NSIZE)
0112              WRITE(6,830)
0113              WRITE(6,820)(R6(I),I=1,NSIZE)
0114              RETURN
0115         720  FORMAT(44H0FINITE ELEMENT MODAL SUPERPOSITION SOLUTION/21H SOLUTIO
                 1N FOR TIME T=,E12.5/21H TIME INCREMENT DELT=,E12.5/21H ITERATION N
                 2UMBER IT=,I5/21H PRINTOUT VARIABLE N=,I5/13H OUTPUT EVERY,I5,2X,5H
                 3TIMES/27H TIME AT FIRST ITERATION IS,E12.5/12H DEBUG LEVEL,I4)
0116         800  FORMAT(35H0PARAMETERS FOR ANALYTICAL SOLUTION/8H OMEGAN=,E12.5/7H
                 1RATIO=,E12.5/8H OMEGAD=,E12.5/6H ARGE=,E12.5/7H ARGE1=,E12.5/7H AR
                 2GE2=,E12.5/7H GAMMA=,E12.5/6H ARGT=,E12.5/8H CONST1=,E12.5/8H CONS
                 3T2=,E12.5/6H ZETA=,E12.5)
0117         805  FORMAT(4HCROW,I5,3X,3HFF=,E12.5)
0118         810  FORMAT(38H0MODAL SOLUTION FOR UNRESTRAINED NODES)
0119         815  FORMAT(35H0UPDATED INITIAL DISPLACEMENT ARRAY)
0120         820  FORMAT(6(1H ,10(E12.5,1X)/))
0121         825  FORMAT(31H0UPDATED INITIAL VELOCITY ARRAY)
0122         830  FORMAT(27H0UPDATED ACCELERATION ARRAY)
0123         835  FORMAT(5E16.5)
0124         840  FORMAT(26H0ENTERING STRESS PASS,KKK=,I4)
0125         845  FORMAT(1H ,1X,2HV(,I5,3H )=,2X,E12.5)
0126         850  FORMAT(1H ,1X,2HU(,I5,3H )=,2X,E12.5)
0127         855  FORMAT(1H ,1X,2HP(,I5,3H )=,2X,E12.5)
0128              END
```

```
FORTRAN IV G1  RELEASE 2.0            MODMAK              DATE = 77075          18/22/49              PAGE 0001

 0001                SUBROUTINE MODMAK
 0002                COMMON/G48/TITLE(12),ORT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                    1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                    2,COOR(63,2)
 0003                COMMON/L46/RO(189),R1(189),R(9),RR(3),R4(189),LINDX(85)
                    1,DISP(20,3),INDEX(85),INACT(85),R5(189),NACT(85)
                    2,R6(189),R6F(189)
 0004                COMMON/D48/FF(90)
 0005                COMMON/MOD48/F(189)
 0006                COMMON/PARAM/I,ITIME,LET,KKK,TEND,NCN,NDFT,NUT,MEREQ
                    1,NP,NE,NB,NLD,NMAT,I1,NPRINT,ND,NDF,NIT,TBEG,UD,VD,ALPHA,KMASS
                    2,NBCE,NSZF,MSZF,NDCE,NEZF,NEEQ
                    3,NTIME,DELT,NSIZE,T1
                    4,BB,CC,NONLIN,DAMP
               C     THIS SUBROUTINE MODIFIES THE NODAL COORDINATES TO ACCOUNT FOR NODAL
               C     DISPLACEMENT
               C     ONLY THE SOLID NODES ARE EFFECTED
               C     FF IS THE ARRAY OF UNRESTRAINED FREEDOMS
               C     F IS TOTAL FREEDOM ARRAY INCLUDING FIXED NODES
               C     PLACE FREEDOM ARRAY IN PROPER ORDER
 0007                II=1
 0008                DO480I=1,NSZF
 0009                F(I)=0.
 0010                R6F(I)=0.
 0011                IROW=0
 0012                DO470J=1,85
 0013                IF(I.EQ.INDEX(J))IROW=IROW+1
 0014                IF(I.EQ.INACT(J))IROW=IROW+1
 0015           470  CONTINUE
 0016                IF(IROW.GT.0)GOTO480
 0017                F(I)=FF(II)
 0018                R6F(I)=R6(II)
 0019                II=II+1
 0020           480  CONTINUE
               C     NOW THE FREEDOM FUNCTIONS FF ARE IN THE PROPER PLACES IN
               C     THE GLOBAL ARRAY
               C     INSERT SPECIFIED DISPLACEMENTS
 0021                DO500N=1,ND
 0022                MDF=2
 0023                IF(NDFIX(N).EQ.0)MDF=1
 0024                I=NSD(N)
 0025                NROWB=(I-1)*MDF
 0026                IF(NDFIX(N).EQ.0)NROWB=NROWB+MSZF
 0027                IF(NDF.EQ.1)NROWB=NROWB-MSZF
 0028                DO490M=1,MDF
 0029                NROWB=NROWB+1
 0030                F(NROWB)=DISP(N,M)
 0031           490  CONTINUE
 0032           500  CONTINUE
 0033                IF(II.LT.2)GOTO455
 0034                IF(NTIME.NE.ITIME)GOTO455
 0035                WRITE(6,100)
 0036                WRITE(6,110)(F(I),I=1,NSZF)
```

```
FORTRAN IV G1  RELEASE 2.0          MODMAK          DATE = 77075          18/22/49          PAGE 0002

0037                WRITE(6,800)
0038                WRITE(6,110)(R6F(I),I=1,NSZE)
0039           455  CONTINUE
0040                IF(ITIME.EQ.0)GOTO460
0041                IF(MFREQ.NE.C)GOTO460
0042                RETURN
0043           460  CONTINUE
               C    MODIFY COORDINATES
0044                MDF=2
0045                DO450N=1,NP
0046                IF(NTYPE(N).EQ.1)GOTO450
               C    SKIPS FLUID NODES
0047                DO440M=1,MDF
0048                NN=(N-1)*MDF+M
0049                CORD(N,M)=COOR(N,M)+F(NN)
0050           440  CONTINUE
0051           450  CONTINUE
0052                IF(I1.LT.1)RETURN
0053                IF(NTIME.NE.ITIME)RETURN
0054                WRITE(6,120)
0055                DO430N=1,NP
0056           430  WRITE(6,993)N,(M,CORD(N,M),M=1,2)
0057                RETURN
0058           100  FORMAT(21HOGLOBAL FREEDOM ARRAY)
0059           110  FORMAT(6(1H ,10(E12.5,1X)/))
0060           120  FORMAT(21HOADJUSTED COORDINATES)
0061           800  FORMAT(26HOGLOBAL ACCELERATION ARRAY)
0062           993  FORMAT(4H ROW,I4,5(I4,1H.,1X,E12.5,2X)/7(1H ,7X,5(I4,1H.,1X,E12.5,
                   12X)/))
0063                END
```

```
FORTRAN IV G1  RELEASE 2.0          STRESS          DATE = 77075          18/22/49          PAGE 0001

0001                SUBROUTINE STRESS
              C
              C     THIS SUBROUTINE EVALUATES THE ELEMENTAL STRESS AND VELOCITIES
              C
0002                COMMON/G4B/TITLE(12),ORT(2,3),ZET(189),CORD(63,2),NOP(88,6)
                   1,IMAT(88),NBC(20),NFIX(20),NSD(20),NDFIX(20),NTYPE(63)
                   2,COLR(63,2)
0003                COMMON/PARAM/I,ITIME,LFI,KKK,TEND,NCN,NDFT,NDT,MEREQ
                   1,NP,NE,NB,NLU,NMAT,I1,NPRINT,ND,NDF,NIT,TBEG,UO,VO,ALPHA,KMASS
                   2,NBLE,NSZE,MSZE,NDCE,NEZE,NEEQ
                   3,NTIME,DELT,NSIZE,T1
                   4,BB,CC,NONLIN,DAMP
              C
              C     SCAN ELEMENTS
              C
0004                DO10N=1,NE
0005                LL=IMAT(N)
0006                IF(NDF.EQ.1)LL=2
0007                GOTO(20,30,40),LL
0008           20   CALL STIFT1(N)
0009                GOTO10
0010           30   CALL STIFT2(N)
0011                GOTO10
0012           40   CALL STIFT3(N)
0013           10   CONTINUE
0014                KKK=KKK+1
0015                IF(T.EQ.T1)GOTO50
0016                IF(ATIME.NE.ITIME)GOTO60
0017           50   WRITE(6,100)KKK
0018           60   CONTINUE
0019                RETURN
0020          100   FORMAT(24H0END OF STRESS PASS,KKK=,I4)
0021                END
```

72 ELEMENT FLUID SOLID MODEL

NP	NE	NB	NLD	NMAT	I1	NPRINT	ND	NDF	KKK
54	72	1	3	2	0	1	14	3	1

NDFI	NDT	MFREQ	NONLIN
0	0	0	1

NIT	TBEG	TEND	LFT	UO	VO	ALPHA	KMASS
4	0.0	0.00001	0	0.0	0.0	0.0	0

MATERIAL PROPERTIES

MATL. NO.	E.C	NU.-	DENSITY
1	0.30000E+08	0.30000E+00	0.72970E-03
2	0.60000E+05	0.0	0.93552E-04

DAMPING FACTOR IS 0.0 SEC-1

NODAL POINTS

NODE	X-CORD	Y-CORD
1	0.600	0.0
2	0.600	4.000
3	0.600	8.000
4	0.600	12.000
5	0.600	12.500
6	0.600	13.000
7	2.700	0.0
8	2.700	4.000
9	2.700	8.000
10	2.700	12.000
11	2.700	12.500
12	2.700	13.000
13	6.900	0.0
14	6.900	4.000
15	6.900	8.000
16	6.900	12.000
17	6.900	12.500
18	6.900	13.000
19	11.100	0.0
20	11.100	4.000
21	11.100	8.000
22	11.100	12.000
23	11.100	12.500
24	11.100	13.000
25	15.300	0.0
26	15.300	4.000
27	15.300	8.000
28	15.300	12.000
29	15.300	12.500
30	15.300	13.000
31	19.500	0.0
32	19.500	4.000
33	19.500	8.000
34	19.500	12.000
35	19.500	12.500
36	19.500	13.000
37	23.700	0.0
38	23.700	4.000
39	23.700	8.000
40	23.700	12.000

41	23.700	12.500
42	23.700	13.000
43	27.900	0.0
44	27.900	4.000
45	27.900	8.000
46	27.900	12.000
47	27.900	12.500
48	27.900	13.000
49	30.000	0.0
50	30.000	4.000
51	30.000	8.000
52	30.000	12.000
53	30.000	12.500
54	30.000	13.000

NODAL DEGREES OF FREEDOM

NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE
1	1	2	1	3	1	4	3	5	2	6	2	7	1	8	1	9	1	10	3	11	2	12	2
13	1	14	1	15	1	16	3	17	2	18	2	19	1	20	1	21	1	22	3	23	2	24	2
25	1	26	1	27	1	28	3	29	2	30	2	31	1	32	1	33	1	34	3	35	2	36	2
37	1	38	1	39	1	40	3	41	2	42	2	43	1	44	1	45	1	46	3	47	2	48	2
49	1	50	1	51	1	52	3	53	2	54	2												

ELEMENTS

		NODES			MATL. NO.		
1	1	7	8	0	0	0	2
2	1	8	2	0	0	0	2
3	2	8	9	0	0	0	2
4	2	9	3	0	0	0	2
5	3	9	10	0	0	0	2
6	4	10	11	3	10	4	3
7	4	11	5	0	0	0	1
8	5	11	12	0	0	0	1
9	5	12	6	0	0	0	1
10	7	13	14	0	0	0	2
11	7	14	8	0	0	0	2
12	8	14	15	0	0	0	2
13	8	15	9	0	0	0	2
14	9	15	16	0	0	0	2
15	10	16	17	9	16	10	3
16	10	17	11	0	0	0	1
17	11	17	18	0	0	0	1
18	11	18	12	0	0	0	1
19	13	19	20	0	0	0	2
20	13	20	14	0	0	0	2
21	14	20	21	0	0	0	2
22	14	21	15	0	0	0	2
23	15	21	22	0	0	0	2
24	16	22	23	15	22	16	3
25	16	23	17	0	0	0	1
26	17	23	24	0	0	0	1
27	17	24	18	0	0	0	1
28	19	25	26	0	0	0	2
29	19	26	20	0	0	0	2
30	20	26	27	0	0	0	2
31	20	27	21	0	0	C	2
32	21	27	28	0	0	0	2
33	22	28	29	21	28	22	3
34	22	29	23	0	0	0	1
35	23	29	30	0	0	0	1
36	23	30	24	C	C	C	1
37	25	31	32	0	C	0	2
38	25	32	26	0	C	0	2
39	26	32	33	0	0	0	2
40	26	33	27	0	0	0	2

41	27	33	34	0	0	0	2
42	28	34	35	27	34	28	3
43	28	35	29	0	0	0	1
44	29	35	36	0	0	0	1
45	29	36	30	0	0	0	1
46	31	37	38	0	0	0	2
47	31	38	32	0	0	0	2
48	32	38	39	0	0	0	2
49	32	39	33	0	0	0	2
50	33	39	40	0	0	0	2
51	34	40	41	33	40	34	3
52	34	41	35	0	0	0	1
53	35	41	42	0	0	0	1
54	35	42	36	0	0	0	1
55	37	43	44	0	0	0	2
56	37	44	38	0	0	0	2
57	38	44	45	0	0	0	2
58	38	45	39	0	0	0	2
59	39	45	46	0	0	0	2
60	40	46	47	39	46	40	3
61	40	47	41	0	0	0	1
62	41	47	48	0	0	0	1
63	41	48	42	0	0	0	1
64	43	49	50	0	0	0	2
65	43	50	44	0	0	0	2
66	44	50	51	0	0	0	2
67	44	51	45	0	0	0	2
68	45	51	52	0	0	0	2
69	46	52	53	45	52	46	3
70	46	53	47	0	0	0	1
71	47	53	54	0	0	0	1
72	47	54	48	0	0	0	1

BOUNDARY CONDITIONS

NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE
1	0																						

SPECIFIED DISPLACEMENTS

NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE	NODE	TYPE
1	0	2	0	3	0	4	0	4	11	5	11	6	11	49	0	50	0	51	0	52	0	52	11
53	11	54	11																				

SYSTEM PARAMETERS

NBCE	NSZF	MSZF	NDCE	NEZF	NEEQ
3	162	108	42	9	6

72 ELEMENT FLUID SOLID MODEL

PMAX	OMEGA	NLD
0.0	0.0	3

NODE	LOAD	
1	0.0	0.0
	0.0	

UNSORTED LOAD ARRAYS

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0								

SPECIFIED DISPLACEMENTS OCCUR AT THE FOLLOWING ROWS

109	110	111	112	7	8	9	10	11	12	157	158
159	160	103	104	105	106	107	108	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0											

INACTIVE FREEDOMS ARE LOCATED AT

1	2	3	4	5	6	113	114	13	14	15	16
17	18	119	120	25	26	27	28	29	30	125	126
37	38	39	40	41	42	131	132	49	50	51	52
53	54	137	138	61	62	63	64	65	66	143	144
73	74	75	76	77	78	149	150	85	86	87	88
89	90	155	156	97	98	99	100	101	102	161	162
0	0	0	0	0	0	0	0	0	0	0	0
0											

MODAL ROW NUMBERS ARE

19	20	21	22	23	24	31	32	33	34	35	36
43	44	45	46	47	48	55	56	57	58	59	60
67	68	69	70	71	72	79	80	81	82	83	84
91	92	93	94	95	96	115	116	117	118	121	122
123	124	127	128	129	130	133	134	135	136	139	140
141	142	145	146	147	148	151	152	153	154	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0											

```
THESE DISPLACEMENTS ARE
DISP(  1,  1)=   10.000
DISP(  1,  2)=    0.0
DISP(  1,  3)=    0.0
DISP(  2,  1)=   10.000
DISP(  2,  2)=    0.0
DISP(  2,  3)=    0.0
DISP(  3,  1)=   10.000
DISP(  3,  2)=    0.0
DISP(  3,  3)=    0.0
DISP(  4,  1)=   10.000
DISP(  4,  2)=    0.0
DISP(  4,  3)=    0.0
DISP(  5,  1)=    0.0
DISP(  5,  2)=    0.0
DISP(  5,  3)=    0.0
DISP(  6,  1)=    0.0
DISP(  6,  2)=    0.0
DISP(  6,  3)=    0.0
DISP(  7,  1)=    0.0
DISP(  7,  2)=    0.0
DISP(  7,  3)=    0.0
DISP(  8,  1)=    0.0
DISP(  8,  2)=    0.0
DISP(  8,  3)=    0.0
DISP(  9,  1)=    0.0
DISP(  9,  2)=    0.0
DISP(  9,  3)=    0.0
DISP( 10,  1)=    0.0
DISP( 10,  2)=    0.0
DISP( 10,  3)=    0.0
DISP( 11,  1)=    0.0
DISP( 11,  2)=    0.0
```

```
DISP( 11,  3)=       0.0
DISP( 12,  1)=       0.0
DISP( 12,  2)=       0.0
DISP( 12,  3)=       0.0
DISP( 13,  1)=       0.0
DISP( 13,  2)=       0.0
DISP( 13,  3)=       0.0
DISP( 14,  1)=       0.0
DISP( 14,  2)=       0.0
DISP( 14,  3)=       0.0

PRESSURE FREEDOMS ARE LOCATED AT THE FOLLOWING ROWS
   109   110   111   112   113   114   115   116   117   118   119   120
   121   122   123   124   125   126   127   128   129   130   131   132
   133   134   135   136   137   138   139   140   141   142   143   144
   145   146   147   148   149   150   151   152   153   154   155   156
   157   158   159   160   161   162     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0

     0     0     0     0     0     0     0     0     0     0     0     0
     0

LOADS
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0

  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0

  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
  0.0     0.0

SIZE OF CONDENSED MATRICES IS  70

DEBUG LEVEL   0

CONDENSED STIFFNESS MATRIX FOR MODAL ANALYSIS
ROW   1   1.  0.78579E+08    2. -0.10714E+06    3. -0.72692E+08    4.  0.10714E+08    5.  0.0
          6.  0.0            7. -0.19623E+07    8.  0.[illegible]9451E+07    9.  0.0   10. -0.10714E+08
         11.  0.0           12.  0.0           13.  0.0           14.  0.0           15.  0.0
         16.  0.0           17.  0.0           18.  0.0           19.  0.0           20.  0.0
         21.  0.0           22.  0.0           23.  0.0           24.  0.0           25.  0.0
         26.  0.0           27.  0.0           28.  0.0           29.  0.0           30.  0.0
         31.  0.0           32.  0.0           33.  0.0           34.  0.0           35.  0.0
         36.  0.0           37.  0.0           38.  0.0           39.  0.0           40.  0.0
```

	41.	0.0	42.	0.0	43.	0.0	44.	0.0	45.	0.0
	46.	0.0	47.	0.0	48.	0.0	49.	0.0	50.	0.0
	51.	0.0	52.	0.0	53.	0.0	54.	0.0	55.	0.0
	56.	0.0	57.	0.0	58.	0.0	59.	0.0	60.	0.0
	61.	0.0	62.	0.0	63.	C.0	64.	C.0	65.	0.0
	66.	0.0	67.	0.0	68.	0.0	69.	0.0	70.	0.0
ROW 2	1.	-0.10714E+08	2.	0.20975E+09	3.	0.10714E+08	4.	-0.20769E+09	5.	0.0
	6.	0.0	7.	0.57692E+07	8.	-0.68681E+06	9.	-0.10714E+08	10.	0.0
	11.	0.0	12.	0.0	13.	0.0	14.	C.0	15.	C.0
	16.	0.0	17.	0.0	18.	0.0	19.	0.0	20.	0.0
	21.	0.0	22.	0.0	23.	0.0	24.	0.0	25.	0.0
	26.	0.0	27.	0.0	28.	0.0	29.	0.0	30.	0.0
	31.	0.0	32.	0.0	33.	0.0	34.	0.0	35.	0.0
	36.	0.0	37.	0.0	38.	0.0	39.	C.0	40.	0.0
	41.	0.0	42.	0.0	43.	0.0	44.	0.0	45.	0.0
	46.	0.0	47.	0.0	48.	0.0	49.	0.0	50.	0.0
	51.	0.0	52.	0.0	53.	0.0	54.	0.0		
	56.	0.0	57.	0.0	58.	C.0	59.	0.0		
	61.	0.0	62.	0.0	63.	0.0				
	66.	C.0	67.	0.0	68.	[illegible]				

									10.	0.0
								[illegible]	15.	0.0
							19.	0.0	20.	0.0
						C.0	24.	0.0	25.	0.0
				[illegible]	28.	0.0	29.	0.0	30.	0.0
	31.	0.0	32.	0.0	33.	0.0	34.	0.0	35.	0.0
	36.	0.0	37.	0.0	38.	0.0	39.	0.0	40.	0.0
	41.	C.0	42.	0.0	43.	0.0	44.	0.0	45.	0.0
	46.	0.0	47.	0.0	48.	0.0	49.	0.0	50.	0.0
	51.	0.0	52.	0.0	53.	0.0	54.	0.0	55.	0.0
	56.	0.0	57.	0.0	58.	0.0	59.	0.0	60.	0.0
	61.	0.0	62.	0.0	63.	0.0	64.	0.0	65.	-0.95238E+00
	66.	0.0	67.	0.0	68.	-0.78750E+00	69.	0.44321E+01	70.	-0.78750E+00
ROW 70	1.	0.0	2.	0.0	3.	0.0	4.	0.0	5.	0.0
	6.	0.0	7.	0.0	8.	0.0	9.	0.0	10.	0.0
	11.	0.0	12.	0.0	13.	0.0	14.	0.0	15.	0.0
	16.	C.C	17.	C.C	18.	0.0	19.	0.0	20.	0.0
	21.	0.0	22.	0.0	23.	0.0	24.	0.0	25.	0.0
	26.	0.0	27.	0.0	28.	0.0	29.	0.0	30.	0.0
	31.	0.0	32.	0.0	33.	0.0	34.	0.0	35.	0.0
	36.	0.0	37.	0.0	38.	0.0	39.	0.0	40.	0.0
	41.	0.0	42.	C.0	43.	C.0	44.	0.0	45.	0.0
	46.	0.0	47.	0.0	48.	0.0	49.	0.0	50.	0.0
	51.	0.0	52.	C.0	53.	0.0	54.	0.0	55.	0.0
	56.	0.0	57.	0.0	58.	C.0	59.	0.0	60.	0.0
	61.	C.C	62.	C.0	63.	0.0	64.	0.0	65.	0.0
	66.	-0.47619E+00	67.	0.0	68.	0.0	69.	-C.78750E+00	7C.	0.22161E+01

CONDENSED MASS MATRIX FOR MODAL ANALYSIS

ROW 1	1.	0.63849E-03	2.	0.0	3.	0.0	4.	0.0	5.	0.0
	6.	0.0	7.	0.0	8.	0.0	9.	0.0	10.	0.0
	11.	C.0	12.	0.0	13.	0.0	14.	0.0	15.	0.0
	16.	0.0	17.	0.0	18.	C.0	19.	0.0	20.	C.0
	21.	0.0	22.	0.0	23.	0.0	24.	0.0	25.	0.0
	26.	0.0	27.	0.0	28.	0.0	29.	0.0	3C.	0.0
	31.	0.0	32.	0.0	33.	0.0	34.	0.0	35.	0.0
	36.	0.0	37.	0.0	38.	0.0	39.	0.0	40.	0.0

```
         41.  0.0         42.  0.0         43.  0.0         44.  0.0         45.  0.0
         46.  0.0         47.  0.0         48.  0.0         49.  0.0         50.  0.0
         51.  0.0         52.  0.0         53.  0.0         54.  0.0         55.  0.0
         56.  0.0         57.  0.0         58.  0.0         59.  0.0         60.  0.0
         61.  0.0         62.  0.0         63.  0.0         64.  0.0         65.  0.0
         66.  0.0         67.  0.0         68.  0.0         69.  0.0         70.  0.0

ROW   2   1.  0.0          2.  0.63849E-03   3.  0.0          4.  0.0          5.  0.0
          6.  0.0          7.  0.0          8.  0.0          9.  0.0         10.  0.0
         11.  0.0         12.  0.0         13.  0.0         14.  0.0         15.  0.0
         16.  0.0         17.  0.0         18.  0.0         19.  0.0         20.  0.0
         21.  0.0         22.  0.0         23.  0.0         24.  0.0         25.  0.0
         26.  0.0         27.  0.0         28.  0.0         29.  0.0         30.  0.0
         31.  0.0         32.  0.0         33.  0.0         34.  0.0         35.  0.0
         36.  0.0         37.  0.0         38.  0.0         39.  0.0         40.  0.0

         41.  0.0         42.  0.0         43.  0.0         44.  0.0         45.  0.0
         46.  0.0         47.  0.0         48.  0.0         49.  0.0         50.  0.0
         51.  0.0         52.  0.0         53.  0.0         54.  0.0         55.  0.0
         56.  0.0         57.  0.0         58.  0.0         59.  0.0
         61.  0.0         62.  0.0         63.  0.0
         66.  0.0         67.  0.0

                                                                          15.  0.0
                                                            19.  0.0         20.  0.0
                                           23.  0.0         24.  0.0         25.  0.0
                                           28.  0.0         29.  0.0         30.  0.0
         31.  0.0         32.  0.0         33.  0.0         34.  0.0         35.  0.0
         36.  0.0         37.  0.0         38.  0.0         39.  0.0         40.  0.0

         41.  0.0         42.  0.0         43.  0.0         44.  0.0         45.  0.0
         46.  0.0         47.  0.0         48.  0.0         49.  0.0         50.  0.0
         51.  0.0         52.  0.0         53.  0.0         54.  0.0         55.  0.0
         56.  0.0         57.  0.0         58.  0.0         59.  0.0         60.  0.0
         61.  0.0         62.  0.0         63.  0.0         64.  0.0         65.  0.0
         66.  0.0         67.  0.0         68.  0.0         69.  0.35000E-08   70.  0.0

ROW  70   1.  0.0          2.  0.0          3.  0.0          4.  0.0          5.  0.0
          6.  0.0          7.  0.0          8.  0.0          9.  0.0         10.  0.0
         11.  0.0         12.  0.0         13.  0.0         14.  0.0         15.  0.0
         16.  0.0         17.  0.0         18.  0.0         19.  0.0         20.  0.0
         21.  0.0         22.  0.0         23.  0.0         24.  0.0         25.  0.0
         26.  0.0         27.  0.0         28.  0.0         29.  0.0         30.  0.0
         31.  0.0         32.  0.0         33.  0.0         34.  0.0         35.  0.0
         36.  0.0         37.  0.0         38.  0.0         39.  0.0         40.  0.0

         41.  0.0         42.  0.0         43.  0.0         44.  0.0         45.  0.0
         46.  0.0         47.  0.0         48.  0.0         49.  0.0         50.  0.0
         51.  0.0         52.  0.0         53.  0.0         54.  0.0         55.  0.0
         56.  0.0         57.  0.0         58.  0.0         59.  0.0         60.  0.0
         61.  0.0         62.  0.0         63.  0.0         64.  0.0         65.  0.0
         66.  0.0         67.  0.0         68.  0.0         69.  0.0         70.  0.19444E-08

REDUCED NONLINEAR LOAD ARRAY
  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0

  0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
```

```
REDUCED LOAD ARRAY
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.95238E+01  0.19048E+02  0.19048E+02  0.95238E+01  0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0

0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0

EIGENVECTORS
     0.27965E-01  -0.24694E-01  0.92368E-03  0.46237E-02  -0.77967E-02  0.79055E-02  -0.48220E-02

0.30056E-01  0.33482E-01  0.20495E-01  0.21216E-01  -0.19543E-01  0.15199E-01  -0.83313E-02

0.40784E+00  -0.22977E-01  0.11745E-01  0.24704E-01  0.31624E-01  -0.28386E-01  0.16611E-01

0.53293E+00  -0.42827E+00  0.60667E-01  0.13425E+00  -0.16336E+00  -0.13882E+00  -0.78377E-01
0.31397E+00  -0.32830E+00  -0.14910E+00  -0.15498E+00  -0.43547E-01  0.12661E+00  -0.10746E-01

0.32659E-02  0.97705E-01  -0.14514E-01  0.18836E-01  -0.5[illegible]
0.0          0.0          0.0
0.0


                         [illegible]E+00  0.64028E-01  0.10744E+00  0.17591E+00  -0.67645E-02
[illegible]  0.18372E+00  -0.72747E-01  0.14992E+00  -0.41580E-01  -0.10958E+00  0.53577E-01
     0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0

0.13492E+00  0.13570E+00  0.16794E+00  -0.31786E+00  -0.30358E+00  -0.10940E+00  -0.14133E+00
0.84715E-01  0.31217E-01  0.34158E+00  0.37867E+00  -0.44264E-01  -0.18460E+00  -0.43393E-01

0.16833E-01  0.31820E+00  -0.35901E+00  -0.14852E+00  0.18345E+00  0.14348E+00  0.10454E+00
0.14148E+00  0.16103E+00  -0.11658E+00  0.11779E+00  -0.63467E-01  -0.82989E-01  0.43816E-01
EIGENVALUES
     0.74004E+12

     0.74002E+12

     0.73182E+12

     0.72905E+12

     0.72628E+12

     0.72428E+12

     0.72519E+12
```

0.36889E+12
0.36879E+12
0.36758E+12
0.36617E+12
0.36453E+12
0.36298E+12
0.36185E+12
0.26272E+12
0.26272E+12
0.25646E+12
0.25597E+12
0.25508E+12
0.25406E+12
0.25328E+12
0.13594E+12
0.13594E+12
0.13320E+12
0.13239E+12
0.13075E+12
0.12875E+12
0.12712E+12
0.11814E+11
0.11707E+11
0.69402E+10
0.65315E+10
0.39989E+10
0.39850E+10
0.38609E+10
0.27524E+10
0.18543E+10
0.16465E+10
0.73605E+09
0.47318E+09

```
0.21115E+09
0.26471E+08
0.18748E+10
0.18703E+10
0.16480E+10
0.16270E+10
0.15946E+10
0.14368E+10
0.13970E+10
0.12390E+10
0.12220E+10
0.11670E+10
0.11571E+10
0.10584E+10
0.10050E+10
0.95704E+09
0.94101E+09
0.94000E+09
0.92161E+09
0.83160E+09
0.76042E+09
0.71978E+09
0.71526E+09
0.55864E+09
0.53498E+09
0.38259E+09
0.33319E+09
0.26569E+09
0.15761E+09
0.40703E+08

DEBUG LEVEL    0

EIGENVECTORS TRANSFORMED BACK TO X
ROW   1   1.  0.11067E+01    2. -0.97727E+00    3.  0.36555E-01    4.  0.18298E+00    5. -0.30856E+00
          6.  0.31286E+00    7. -0.19083E+00    8. -0.11895E+01    9.  0.13251E+01   10.  0.81109E+00
```

11. 0.83964E+00	12. -0.77341E+00	13. 0.60151E+00	14. -0.32972E+00	15. -0.16141E+02
16. -0.90932E+00	17. 0.46481E+00	18. 0.97766E+00	19. 0.12515E+01	20. -0.11234E+01
21. 0.65739E+00	22. -0.21091E+02	23. -0.16949E+02	24. 0.24009E+01	25. 0.53131E+01
26. -0.64650E+01	27. -0.54940E+01	28. -0.31018E+01	29. 0.12426E+02	30. -0.12992E+02
31. -0.59008E+01	32. -0.61333E+01	33. -0.17234E+01	34. 0.50105E+01	35. -0.42527E+00
36. -0.12925E+00	37. 0.36667E+01	38. -0.57441E+00	39. 0.74543E+00	40. -0.20636E+01
41. -0.71344E+00	42. 0.41303E+00	43. 0.0	44. 0.0	45. 0.0
46. 0.0	47. 0.0	48. 0.0	49. 0.0	50. 0.0
51. 0.0	52. 0.0	53. 0.0	54. 0.0	55. 0.0
56. 0.0	57. 0.0	58. 0.0	59. 0.0	60. 0.0
61. 0.0	62. 0.0	63. 0.0	64. 0.0	65. 0.0
66. 0.0	67. 0.0	68. 0.0	69. 0.0	70. 0.0

ROW 2

1. -0.11820E+02	2. 0.10464E+02	3. 0.21409E+01	4. 0.26855E+01	5. -0.22985E+01
6. 0.15438E+01	7. -0.73585E+00	8. 0.14352E+02	9. -0.16056E+02	10. -0.10156E+02
11. -0.10621E+02	12. 0.97323E+01	13. -0.75094E+01	14. 0.41018E+01	15. -0.15767E+01
16. -0.88914E-01	17. -0.45331E+00	18. -0.93554E+00	19. -0.12879E+01	20. 0.12865E+01
21. -0.81062E+00	22. -0.16631E+01	23. -0.13358E+01	24. 0.25260E+00	25. 0.54189E+00
26. -0.67659E+00	27. -0.60374E+00	28. -0.35595E+00	29. -0.12892E+00	30. 0.13029E+00
31. -0.13789E-01	32. -0.11663E+00	33. -0.10639E+02	34. -0.44901E+01	35. 0.12885E+02
36. -0.77678E+01	37. 0.10074E+00	38. -0.66036E+01	39. 0.48000E+01	40. -0.17419E+00
41. -0.28457E+01	42. 0.11644E+01	43. 0.0	44. 0.0	45. 0.0
46. 0.0	47. 0.0	48. 0.0	49. 0.0	50. 0.0
51. 0.0	52. 0.0	53. 0.0	54. 0.0	
56. 0.0	57. 0.0	58. 0.0		
61. 0.0	62. 0.0			
66. 0.0				

				... 0.0
		... 0.44847E+04	44. -0.45439E+04	45. -0.25991E+04
	... 0.62477E+04	48. 0.28688E+04	49. 0.33728E+04	50. -0.68168E+03
51. -0.43734E+04	52. 0.23206E+04	53. 0.69727E+03	54. -0.22973E+04	55. 0.21906E+04
56. -0.23845E+04	57. -0.18121E+04	58. 0.55650E+04	59. -0.56272E+04	60. 0.10823E+04
61. 0.18160E+04	62. 0.29733E+04	63. -0.11434E+03	64. 0.12406E+04	65. 0.31054E+04
66. -0.12305E+04	67. 0.25342E+04	68. -0.70283E+03	69. -0.16523E+04	70. 0.99014E+03

ROW 70

1. 0.0	2. 0.0	3. 0.0	4. 0.0	5. 0.0
6. 0.0	7. 0.0	8. 0.0	9. 0.0	10. 0.0
11. 0.0	12. 0.0	13. 0.0	14. 0.0	15. 0.0
16. 0.0	17. 0.0	18. 0.0	19. 0.0	20. 0.0
21. 0.0	22. 0.0	23. 0.0	24. 0.0	25. 0.0
26. 0.0	27. 0.0	28. 0.0	29. 0.0	30. 0.0
31. 0.0	32. 0.0	33. 0.0	34. 0.0	35. 0.0
36. 0.0	37. 0.0	38. 0.0	39. 0.0	40. 0.0
41. 0.0	42. 0.0	43. -0.30598E+04	44. 0.30773E+04	45. 0.38085E+04
46. -0.72083E+04	47. -0.68845E+04	48. -0.24810E+04	49. -0.32050E+04	50. 0.19211E+04
51. 0.70794E+03	52. 0.77463E+04	53. 0.65874E+04	54. 0.10038E+04	55. -0.41863E+04
56. -0.98405E+03	57. -0.38174E+03	58. 0.72160E+04	59. -0.81415E+04	60. -0.33680E+04
61. 0.41602E+04	62. 0.32548E+04	63. 0.23707E+04	64. 0.32086E+04	65. 0.36518E+04
66. -0.26437E+04	67. 0.26712E+04	68. -0.14393E+04	69. -0.18820E+04	70. 0.99365E+03

FINITE ELEMENT MODAL SUPERPOSITION SOLUTION
SOLUTION FOR TIME T= 0.0
TIME INCREMENT DELT= 0.25000E-05

```
ITERATION NUMBER IT=     0
PRINTOUT VARIABLE N=     0
OUTPUT EVERY     1  TIMES
TIME AT FIRST ITERATION IS 0.25000E-05
DEBUG LEVEL    0

ENTERING STRESS PASS,KKK=   2

MODAL SOLUTION FOR UNRESTRAINED NODES
 U(   10 )=    0.0
 V(   10 )=    0.0
 U(   11 )=    0.0
 V(   11 )=    0.0
 U(   12 )=    0.0
 V(   12 )=    0.0
 U(   16 )=    0.0
 V(   16 )=    0.0
 U(   17 )=    0.0
 V(   17 )=    0.0
 U(   18 )=    0.0
 V(   18 )=    0.0
 U(   22 )=    0.0
 V(   22 )=    0.0
 U(   23 )=    0.0
 V(   23 )=    0.0
 U(   24 )=    0.0
 V(   24 )=    0.0
 U(   28 )=    0.0
 V(   28 )=    0.0
 U(   29 )=    0.0
 V(   29 )=    0.0
 U(   30 )=    0.0
 V(   30 )=    0.0
 U(   34 )=    0.0
 V(   34 )=    0.0
 U(   35 )=    0.0
 V(   35 )=    0.0
 U(   36 )=    0.0
 V(   36 )=    0.0
 U(   40 )=    0.0
 V(   40 )=    0.0
 U(   41 )=    0.0
 V(   41 )=    0.0
 U(   42 )=    0.0
 V(   42 )=    0.0
 U(   46 )=    0.0
 V(   46 )=    0.0
 U(   47 )=    0.0
 V(   47 )=    0.0
 U(   48 )=    0.0
 V(   48 )=    0.0
 P(    7 )=    0.15302E-01
 P(    8 )=    0.17001E-01
 P(    9 )=    0.17001E-01
 P(   10 )=    0.19124E-01
 P(   13 )=    0.31469E-05
 P(   14 )=    0.33821E-05
 P(   15 )=    0.32306E-05
 P(   16 )=    0.33216E-05
 P(   19 )=   -0.46333E-06
 P(   20 )=   -0.35367E-06
 P(   21 )=    0.13386E-06
 P(   22 )=    0.22305E-06
 P(   25 )=   -0.89174E-07
 P(   26 )=   -0.61304E-06
 P(   27 )=   -0.67125E-06
```

```
P(   28 )=  -0.11118E-05
P(   31 )=  -0.14366E-06
P(   32 )=  -0.22142E-06
P(   33 )=   0.32526E-06
P(   34 )=   0.55158E-06
P(   37 )=   0.66101E-06
P(   38 )=   0.54925E-06
P(   39 )=   0.52969E-06
P(   40 )=   0.45379E-06
P(   43 )=  -0.27661E-07
P(   44 )=  -0.19679E-06
P(   45 )=  -0.43550E-06
P(   46 )=  -0.65075E-06

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT    1  U= 0.12706E+00  V=-0.11348E-04

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT    2  U= 0.12704E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT    3  U= 0.12704E+00  V=-0.84145E-09

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT    4  U= 0.12704E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT    5  U= 0.12704E+00  V=-0.14182E-04

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT    6  U= 0.12701E+00  V= 0.0


            ...18E-06  V= 0.15613E-08

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT  68  U=-0.55418E-08  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   1
ELEMENT  69  U=-0.82809E-08  V= 0.14380E-08

ELEMENTAL STRESSES AT ITERATION NO.   1
ELEMENT  69  SIGMAX= 0.0          SIGMAY= 0.0          SIGMAXY= 0.0

ELEMENTAL STRESSES AT ITERATION NO.   1
ELEMENT  70  SIGMAX= 0.0          SIGMAY= 0.0          SIGMAXY= 0.0

ELEMENTAL STRESSES AT ITERATION NO.   1
ELEMENT  71  SIGMAX= 0.0          SIGMAY= 0.0          SIGMAXY= 0.0

ELEMENTAL STRESSES AT ITERATION NO.   1
ELEMENT  72  SIGMAX= 0.0          SIGMAY= 0.0          SIGMAXY= 0.0

END OF STRESS PASS,KKK=    1

FINITE ELEMENT MODAL SUPERPOSITION SOLUTION
SOLUTION FOR TIME T= 0.25000E-05
TIME INCREMENT DELT= 0.25000E-05
ITERATION NUMBER IT=     1
PRINTOUT VARIABLE N=     1
OUTPUT EVERY     1  TIMES
TIME AT FIRST ITERATION IS 0.25000E-05
DEBUG LEVEL    0
```

```
PARAMETERS FOR ANALYTICAL SOLUTION
OMEGAN= 0.86026E+06
RATIO= 0.10000E+01
OMEGAD= 0.86026E+06
ARGE= 0.0
ARGE1= 0.08328E-78
ARGE2=-0.34436E+30
GAMMA=-0.15450E+23
ARGT= 0.21506E+01
CONST1=-0.72407E+19
CONST2=-0.17891E-19
ZETA= 0.0

ROW    1    FF=-0.87421E-10

PARAMETERS FOR ANALYTICAL SOLUTION
OMEGAN= 0.86024E+06
RATIO= 0.10000E+01
OMEGAD= 0.86024E+06
ARGE= 0.0
ARGE1= 0.08328E-78
ARGE2=-0.34436E+30
GAMMA=-0.15450E+23
ARGT= 0.21506E+01

PARAMETERS FOR ANALYTICAL SOLUTION
OMEGAN= 0.12554E+05
RATIO= 0.10000E+01
OMEGAD= 0.12554E+05
ARGE= 0.0
ARGE1= 0.08328E-78
ARGE2=-0.34436E+30
GAMMA=-0.15450E+23
ARGT= 0.31386E-01
CONST1=-0.72407E+19
CONST2=-0.17891E-19
ZETA= 0.0

ROW   69    FF= 0.13174E-05

PARAMETERS FOR ANALYTICAL SOLUTION
OMEGAN= 0.63799E+04
RATIO= 0.10000E+01
OMEGAD= 0.63799E+04
ARGE= 0.0
ARGE1= 0.08328E-78
ARGE2=-0.34436E+30
GAMMA=-0.15450E+23
ARGT= 0.15950E-01
CONST1=-0.72407E+19
CONST2=-0.17891E-19
ZETA= 0.0

ROW   70    FF= 0.70687E-06

ENTERING STRESS PASS,KKK=    2

MODAL SOLUTION FOR UNRESTRAINED NODES
 U(   10 )=    0.13073E-09
 V(   10 )=    0.14655E-07
```

```
U(  11 )= -0.68986E-10
V(  11 )=  0.14119E-08
U(  12 )= -0.84046E-11
V(  12 )=  0.12235E-09
U(  16 )= -0.59833E-10
V(  16 )=  0.52010E-10
U(  17 )=  0.54192E-10
V(  17 )=  0.55959E-11
U(  18 )=  0.57104E-11
V(  18 )=  0.16086E-12
U(  22 )= -0.21972E-12
V(  22 )=  0.32029E-12
U(  23 )=  0.21132E-12
V(  23 )=  0.67266E-13
U(  24 )=  0.13549E-13
V(  24 )=  0.39854E-12
U(  28 )=  0.15047E-13
V(  28 )=  0.26525E-12
U(  29 )=  0.12086E-13
V(  29 )=  0.26973E-12
U(  30 )=  0.12592E-13
V(  30 )=  0.27207E-12
U(  34 )= -0.38259E-14
V(  34 )=  0.30109E-12
U(  35 )=  0.21818E-14
V(  35 )=  0.29469E-12
U(  36 )=  0.12420E-13
V(  36 )=  0.27404E-12
U(  40 )= -0.26225E-13
V(  40 )=  0.13477E-12
U(  41 )= -0.10059E-13
V(  41 )=  0.12124E-12
U(  42 )=  0.94629E-15
V(  42 )=  0.12711E-12
U(  46 )= -0.13313E-13
V(  46 )=  0.59217E-13
U(  47 )= -0.72154E-14
V(  47 )=  0.53660E-13
U(  48 )=  0.60976E-14
V(  48 )=  0.59189E-13
P(   7 )=  0.61143E-01
P(   8 )=  0.67914E-01
P(   9 )=  0.67920E-01
P(  10 )=  0.76379E-01
P(  13 )=  0.29401E-04
P(  14 )=  0.32327E-04
P(  15 )=  0.32431E-04
P(  16 )=  0.36141E-04
P(  19 )=  0.19325E-06
P(  20 )=  0.25844E-06
P(  21 )=  0.10065E-05
P(  22 )=  0.93482E-06
P(  25 )= -0.99582E-06
P(  26 )= -0.16394E-05
P(  27 )= -0.20734E-05
P(  28 )= -0.25586E-05
P(  31 )=  0.32037E-06
P(  32 )=  0.46683E-06
P(  33 )=  0.15770E-05
P(  34 )=  0.22296E-05
P(  37 )=  0.16815E-05
P(  38 )=  0.13870E-05
P(  39 )=  0.73316E-06
P(  40 )=  0.79744E-06
P(  43 )=  0.19365E-05
P(  44 )=  0.16172E-05
```

```
P(   45 )=   0.14144E-05
P(   46 )=   0.15907E-05

ELEMENTAL VELOCITIES AT ITERATION NO.   2
ELEMENT   1  U= 0.25353E+00  V=-0.56582E-04

ELEMENTAL VELOCITIES AT ITERATION NO.   2
ELEMENT   2  U= 0.25342E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   2
ELEMENT   3  U= 0.25342E+00  V=-0.43506E-07

ELEMENTAL VELOCITIES AT ITERATION NO.   2
ELEMENT   4  U= 0.25342E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   2
ELEMENT   5  U= 0.25342E+00  V=-0.70693E-04

ELEMENTAL VELOCITIES AT ITERATION NO.   2
ELEMENT   6  U= 0.25329E+00  V= 0.0

ELEMENTAL STRESSES AT ITERATION NO.   2
ELEMENT   6  SIGMAX=-0.25990E+00  SIGMAY=-0.87256E-..

... AT ITERATION NO.   2
...69  SIGMAX= 0.20900E-06  SIGMAY= 0.62699E-07  SIGMAXY=-0.32536E-06

ELEMENTAL STRESSES AT ITERATION NO.   2
ELEMENT  70  SIGMAX= 0.73132E-08  SIGMAY=-0.31922E-06  SIGMAXY=-0.15521E-06

ELEMENTAL STRESSES AT ITERATION NO.   2
ELEMENT  71  SIGMAX= 0.11327E-06  SIGMAY= 0.33981E-07  SIGMAXY=-0.29593E-06

ELEMENTAL STRESSES AT ITERATION NO.   2
ELEMENT  72  SIGMAX= 0.96870E-08  SIGMAY= 0.32265E-06  SIGMAXY=-0.17990E-07

END OF STRESS PASS,KKK=   1

FINITE ELEMENT MODAL SUPERPOSITION SOLUTION
SOLUTION FOR TIME T= 0.50000E-05
TIME INCREMENT DELT= 0.25000E-05
ITERATION NUMBER IT=    2
PRINTOUT VARIABLE N=    2
OUTPUT EVERY    1  TIMES
TIME AT FIRST ITERATION IS 0.25000E-05
DEBUG LEVEL   0

ENTERING STRESS PASS,KKK=   2

MODAL SOLUTION FOR UNRESTRAINED NODES
  U(   10 )=   0.13503E-06
  V(   10 )=   0.36130E-07
  U(   11 )=  -0.59348E-09
  V(   11 )=   0.14985E-07
  U(   12 )=  -0.35265E-09
  V(   12 )=   0.53872E-08
  U(   16 )=  -0.67031E-09
```

```
V(   16 )=    0.30410E-09
U(   17 )=    0.55531E-09
V(   17 )=    0.10428E-09
U(   18 )=    0.26504E-09
V(   18 )=    0.14071E-10
U(   22 )=   -0.46047E-11
V(   22 )=    0.30738E-11
U(   23 )=    0.38526E-11
V(   23 )=   -0.65750E-11
U(   24 )=    0.17941E-11
V(   24 )=    0.52154E-11
U(   28 )=    0.33126E-13
V(   28 )=    0.21627E-12
U(   29 )=    0.90036E-13
V(   29 )=    0.17206E-12
U(   30 )=   -0.49257E-14
V(   30 )=    0.24536E-12
U(   34 )=    0.11417E-13
V(   34 )=    0.28044E-12
U(   35 )=    0.80699E-14
V(   35 )=    0.23581E-12
U(   36 )=    0.13725E-13
V(   36 )=    0.20184E-12
U(   40 )=   -0.57495E-13
V(   40 )=    0.19296E-12
U(   41 )=   -0.42012E-13
V(   41 )=    0.16009E-12
U(   42 )=   -0.40329E-13
V(   42 )=    0.15143E-12
U(   46 )=   -0.23578E-13
V(   46 )=    0.85432E-13
U(   47 )=   -0.30416E-13
V(   47 )=    0.67044E-13
U(   48 )=   -0.30531E-14
V(   48 )=    0.64518E-13
P(    7 )=    0.13732E+00
P(    8 )=    0.15247E+00
P(    9 )=    0.15250E+00
P(   10 )=    0.17094E+00
P(   13 )=    0.13667E-03
P(   14 )=    0.15126E-03
P(   15 )=    0.15170E-03
P(   16 )=    0.61281E-04
P(   19 )=    0.10058E-05
P(   20 )=    0.10394E-05
P(   21 )=    0.27344E-05
P(   22 )=    0.12517E-05
P(   25 )=   -0.11101E-05
P(   26 )=   -0.23544E-05
P(   27 )=   -0.30138E-05
P(   28 )=   -0.42394E-05
P(   31 )=   -0.80839E-06
P(   32 )=   -0.76741E-06
P(   33 )=    0.13225E-05
P(   34 )=    0.21383E-05
P(   37 )=    0.13635E-05
P(   38 )=    0.65565E-06
P(   39 )=   -0.13039E-06
P(   40 )=   -0.11176E-06
P(   43 )=    0.24098E-05
P(   44 )=    0.19763E-05
P(   45 )=    0.12454E-05
P(   46 )=    0.12496E-05

ELEMENTAL VELOCITIES AT ITERATION NO.   3
ELEMENT    1  U= 0.37904E+00   V=-0.15779E-03
```

```
ELEMENTAL VELOCITIES AT ITERATION NO.   3
ELEMENT   2  U= 0.37874E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   3
ELEMENT   3  U= 0.37874E+00  V=-0.27560E-06

ELEMENTAL VELOCITIES AT ITERATION NO.   3
ELEMENT   4  U= 0.37874E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   3
ELEMENT   5  U= 0.37874E+00  V=-0.19388E-03

ELEMENTAL VELOCITIES AT ITERATION NO.   3
ELEMENT   6  U= 0.37837E+00  V= 0.0

ELEMENTAL STRESSES AT ITERATION NO.   3
ELEMENT   6  SIGMAX=-0.43662E+00  SIGMAY=-0.15197E+01  SIGMAXY= 0.16465E+00

ELEMENTAL STRESSES AT ITERATION NO.   3
ELEMENT   7  SIGMAX=-0.93168E-02  SIGMAY=-0.27950E-02  SIGM...

ELEMENTAL STRESSES AT ITERATION ..
ELEMENT   8  SIGM...


...  SIGMAX= 0.37015E-06  SIGMAY= 0.11104E-06  SIGMAXY=-0.46940E-06

ELEMENTAL STRESSES AT ITERATION NO.   3
ELEMENT  70  SIGMAX= 0.11377E-06  SIGMAY=-0.10692E-05  SIGMAXY=-0.52616E-06

ELEMENTAL STRESSES AT ITERATION NO.   3
ELEMENT  71  SIGMAX= 0.47749E-06  SIGMAY= 0.14325E-06  SIGMAXY=-0.36837E-06

ELEMENTAL STRESSES AT ITERATION NO.   3
ELEMENT  72  SIGMAX=-0.20309E-08  SIGMAY=-0.15215E-06  SIGMAXY= 0.27696E-06

END OF STRESS PASS,KKK=   1

FINITE ELEMENT-MODAL SUPERPOSITION SOLUTION
SOLUTION FOR TIME T= 0.75000E-05
TIME INCREMENT DELT= 0.25000E-05
ITERATION NUMBER IT=    3
PRINTOUT VARIABLE N=    3
OUTPUT EVERY    1  TIMES
TIME AT FIRST ITERATION IS 0.25000E-05
DEBUG LEVEL   0

ENTERING STRESS PASS,KKK=   2

MODAL SOLUTION FOR UNRESTRAINED NODES
 U(   10 )=    0.32707E-08
 V(   10 )=    0.54657E-07
 U(   11 )=   -0.95727E-09
 V(   11 )=    0.42033E-07
 U(   12 )=   -0.19181E-08
 V(   12 )=    0.32902E-07
 U(   16 )=   -0.19583E-08
 V(   16 )=    0.85611E-09
 U(   17 )=    0.13484E-08
 V(   17 )=    0.55077E-09
 U(   18 )=    0.17606E-08
 V(   18 )=    0.25532E-09
```

```
U(  22 )=  -0.26256E-10
V(  22 )=   0.94060E-11
U(  23 )=   0.19087E-10
V(  23 )=  -0.38042E-10
U(  24 )=   0.22428E-10
V(  24 )=   0.11607E-10
U(  28 )=  -0.14218E-12
V(  28 )=   0.12754E-11
U(  29 )=   0.60222E-12
V(  29 )=   0.80758E-12
U(  30 )=  -0.85166E-12
V(  30 )=   0.15318E-11
U(  34 )=  -0.12074E-13
V(  34 )=   0.11922E-11
U(  35 )=   0.72546E-14
V(  35 )=   0.11104E-11
U(  36 )=   0.26409E-13
V(  36 )=   0.11058E-11
U(  40 )=  -0.97625E-13
V(  40 )=   0.78493E-12
U(  41 )=  -0.29018E-13
V(  41 )=   0.75029E-12
U(  42 )=   0.90206E-14
V(  42 )=   0.70965E-12
U(  46 )=  -0.38927E-13
V(  46 )=   0.23460E-12
U(  47 )=  -0.32670E-13
V(  47 )=   0.23277E-12
U(  48 )=   0.59841E-13
V(  48 )=   0.20270E-12
P(   7 )=   0.24350E+00
P(   8 )=   0.27022E+00
P(   9 )=   0.27033E+00
P(  10 )=   0.30297E+00
P(  13 )=   0.42312E-03
P(  14 )=   0.46884E-03
P(  15 )=   0.46908E-03
P(  16 )=   0.36284E-03
P(  19 )=   0.13784E-05
P(  20 )=   0.12554E-05
P(  21 )=   0.32783E-05
P(  22 )=  -0.15832E-05
P(  25 )=  -0.23954E-05
P(  26 )=  -0.34161E-05
P(  27 )=  -0.34049E-05
P(  28 )=  -0.48429E-05
P(  31 )=  -0.66310E-06
P(  32 )=  -0.70035E-06
P(  33 )=   0.16131E-05
P(  34 )=   0.24326E-05
P(  37 )=   0.57742E-06
P(  38 )=  -0.30920E-06
P(  39 )=  -0.15721E-05
P(  40 )=  -0.15721E-05
P(  43 )=   0.29714E-05
P(  44 )=   0.24338E-05
P(  45 )=   0.14166E-05
P(  46 )=   0.15330E-05

ELEMENTAL VELOCITIES AT ITERATION NO.   4
ELEMENT   1  U= 0.50319E+00  V=-0.33628E-03

ELEMENTAL VELOCITIES AT ITERATION NO.   4
ELEMENT   2  U= 0.50255E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   4
```

```
ELEMENT   3  U= 0.50255E+00  V=-0.10178E-05

ELEMENTAL VELOCITIES AT ITERATION NO.   4
ELEMENT   4  U= 0.50255E+00  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   4
ELEMENT   5  U= 0.50255E+00  V=-0.41195E-03

ELEMENTAL VELOCITIES AT ITERATION NO.   4
ELEMENT   6  U= 0.50177E+00  V= 0.0

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT   6  SIGMAX=-0.20232E+00  SIGMAY=-0.83016E+00  SIGMAXY= 0.20385E+00

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT   7  SIGMAX=-0.15028E-01  SIGMAY=-0.45063E-02  SIGMAXY= 0.23095E+00

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT   8  SIGMAX=-0.19565E+00  SIGMAY=-0.60658E+00  SIGMAXY= 0.20678E+00

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT   9  SIGMAX=-0.30112E-01  SIGMAY=-0.90335E-02  SIGMAXY= 0.[illegible]

ELEMENTAL VELOCITIES AT ITERATION NO



ELEMENTAL VELOCITIES AT ITERATION NO.   4
ELEMENT  68  U= 0.46334E-07  V= 0.0

ELEMENTAL VELOCITIES AT ITERATION NO.   4
ELEMENT  69  U= 0.47370E-07  V=-0.54364E-09

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT  69  SIGMAX= 0.61110E-06  SIGMAY= 0.18333E-06  SIGMAXY=-0.12890E-05

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT  70  SIGMAX= 0.47664E-06  SIGMAY= 0.33091E-07  SIGMAXY=-0.11346E-05

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT  71  SIGMAX= 0.51287E-06  SIGMAY= 0.15386E-06  SIGMAXY=-0.12790E-05

ELEMENTAL STRESSES AT ITERATION NO.   4
ELEMENT  72  SIGMAX=-0.15345E-05  SIGMAY=-0.22647E-05  SIGMAXY= 0.10211E-05

END OF STRESS PASS,KKK=   1
```

VITA

The author was born on in . She began attending Newark College of Engineering in 1968, obtaining the BSME degree from that college in 1972. She continued at NCE for the master's degree in mechanical engineering, which was completed in 1974. She remained a full-time student studying for the doctorate until June 1976, when she accepted employment at Bell Telephone Laboratories in Holmdel, New Jersey. The development of this dissertation took place between September 1974 and June 1976 at New Jersey Institute of Technology, and it was completed at Bell Laboratories between June 1976 and April 1977.