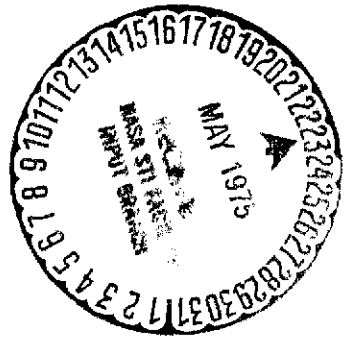
MCR-75-139
Contract NAS8-29946

Final
Report

April 1975

Finite Element Solution of Low Bond Number Sloshing



Prepared for:
Marshall Space Flight Center
Huntsville, Alabama 35823

MARTIN MARIETTA

MCR-75-139

FINITE ELEMENT SOLUTION OF LOW BOND NUMBER SLOSHING

Contract NAS8-29946

(Final Report)

Approved by:

George Morosow
Program Manager

April 1975

Authors: Richard L. Wohlen
A. Colton Park
David M. Warner

Prepared for: Marshall Space Flight Center
Huntsville, Alabama 35812

MARTIN MARIETTA CORPORATION
Denver Division
Denver, Colorado 80201

FOREWORD

This report, prepared by the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, under Contract NAS8-29946, presents the technical approach and the results of a study contract for the vibration characteristics of a liquid in a container of arbitrary axisymmetric shape with surface tension forces of the same order as acceleration forces (Bond Number ~ 1). The study was administered by the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama, under the direction of Mr. Frank Bugg, Systems Dynamics Laboratory.

CONTENTS

	Page
Foreword	ii
Contents	iii
Acknowledgements	vi
1. Introduction	1
2. Technical Approach	3
2.1 Free Surface Static Equilibrium Shape	3
2.1.1 Axisymmetric Meniscus	3
2.1.2 Meniscus Volume	7
2.1.3 State Equations	8
2.1.4 Container Definition	9
2.1.5 Contact Angle Definition	10
2.1.6 Implementation	11
2.2 Mass and Stiffness	14
2.2.1 Surface Tension Finite Element	16
2.2.2 Gravity Finite Element	20
2.2.3 Fluid Finite Element	21
2.3 Vibration Analysis	23
2.4 Finite Element Model	27
2.4.1 90° Model Boundary Conditions	28
2.4.2 Data Generation	30
3. Results	33
3.1 Free Surface Static Equilibrium Shape	33
3.2 Vibration Analysis	46
4. Conclusions	57
5. References	59
6. Computer Programs	61
6.1 Listings	64
6.1.1 Free Surface Static Equilibrium Shape	65
6.1.2 Vibration Analysis	87
6.2 Input Requirements	111
6.2.1 Free Surface Static Equilibrium Shape	111
6.2.2 Vibration Analysis	116

Pages

Figures

2-1	Force Balance on Axisymmetric Meniscus	4
2-2	Meniscus with Liquid Below	6
2-3	Meniscus with Liquid Above	7
2-4	Container Geometry	10
2-5	Contact Angle Geometry	11
2-6	Triangle Coordinate System	17
2-7	In-Plane Strain Due to Normal Deflection	18
2-8	Local Coordinate System for Tetrahedron Element	22
2-9	Tank/Fluid Geometry	28
2-10	90^{o} Model Boundary Conditions	29
3-1	Sample Tabular Output From Free Surface Equilibrium Shape Program, Survey Mode	34
3-2a	Ullage Volume vs Tank Axis Intercept	35
3-2b	Ullage Volume vs Tank Axis Intercept, Plot Symbols	36
3-3a	Contact Angle vs Tank Axis Intercept	37
3-3b	Contact Angle vs Tank Axis Intercept, Plot Symbols	38
3-4a	Ullage Volume vs Contact Angle, A=0.05	39
3-4b	Ullage Volume vs Contact Angle, A=0.04	40
3-4c	Ullage Volume vs Contact Angle, Plot Symbols	41
3-5	Free Surface Static Equilibrium Shape, Cylindrical Container	42
3-6	Free Surface Static Equilibrium Shape, Spherical Container	43
3-7	Free Surface Static Equilibrium Shape, Cosine Container	44

Page

3-8 Free Surface Static Equilibrium Shape, Tug-Like Container 45

3-9 Tug-Like Container, Grid 1 47

3-10 Tug-Like Container, Grid 2 48

3-11 First Slosh Mode, Surface Perspective View 53

3-12 Second Slosh Mode, Surface Perspective View 54

3-13 Third Slosh Mode, Surface Perspective View 55

6-1 Flow Chart, Low Bond Slosh Programs 62

6-2 Sample Problem Input Data - Free Surface Static Equilibrium Shape Program, Survey Mode 114

6-3 Sample Problem Input Data - Free Surface Static Equilibrium Shape Program, Search Mode 115

6-4 Sample Problem Input Data - Vibration Analysis Program 119

Tables

3-1 Number of DOF, Slosh Modes, Crunch Modes, and Circulation Modes 49

3-2 Effect of Simultaneous Acceleration and Surface Tension Variation on Frequency, Grid 1, 2 Sectors 50

3-3 Effect of Bulk Modulus Variation on Frequency, Grid 1, 2 Sectors 50

3-4 Effect of Bulk Modulus Variation on Frequency, Grid 2, 2 Sectors 51

3-5 Effect of Bulk Modulus Variation on Modal Displacement, Grid 2, 2 Sectors, Mode 39 (First Slosh Mode) 51

3-6 Slosh Frequency 52

ACKNOWLEDGEMENTS

The authors appreciate the assistance necessary for the successful completion of the study. Mr. Frank Bugg, of NASA, contributed many valuable comments and suggestions throughout the program. The very helpful comments and suggestions provided by Mr. Carl Bodley of the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, is very deeply appreciated. Thanks are also due to Mr. Wilcomb Benfield and Mr. Roger Philippus for their assistance. The authors are also appreciative of the helpful guidance provided by Mr. George Morosow, the Program Manager.

1. INTRODUCTION

Knowledge of the dynamics of liquid propellant in a low Bond number (defined to be mass density times acceleration of gravity and the square of a characteristic length divided by the surface tension) environment is critical to the design of certain spacecraft systems with respect to orbital propellant transfer and attitude control system. The proposed reusable Space Tug will be required to perform orbital and docking maneuvers with as much as 90% of the vehicle mass as liquid fuel. The effects of the liquid mass on the control system will be significant. In the absence of normal gravity, liquid free surfaces tend to distort drastically under the influence of relatively small disturbances. The propellant motions in low gravity may cause inefficient spacecraft operation or even mission failure, if the motion has not been accurately accounted for in the design of the vehicle.

Because the Bond number will be near or below unity for a large portion of the Space Tug flight environment, the equilibrium configuration of the liquid and natural frequencies and mode shapes will be primarily dependent on liquid surface tension and contact angle. A large body of experimental data have been produced on low Bond number sloshing using drop tower techniques and at 1-g using very small models, but there has been little successful analytical work applicable for tanks of general shape at Bond numbers of one. This study applied the finite element computer technique to the low Bond number slosh problem for tanks of general axisymmetric shape.

The study resulted in the development of digital computer programs for the determination of liquid free surface equilibrium shape, and lateral slosh natural vibration mode shapes and frequencies for a liquid in a container of arbitrary axisymmetric shape with surface tension forces the same order of magnitude as acceleration forces (Bond number ~ 1). For the vibration analysis, a finite volume element representation of the liquid was used.

The liquid free surface equilibrium shapes were computed for several tanks at various contact angles and ullage volumes. One of these configurations was selected for vibration analysis and lateral slosh mode shapes and natural frequencies were obtained.

This report provides documentation of the above results.

(This page blank)

2. TECHNICAL APPROACH

The problem was approached in two distinct steps:

a) establish the static equilibrium shape for a given axisymmetric container subject to contact angle and ullage volume constraints;

b) given the static equilibrium shape, model the system as an assemblage of finite elements and obtain natural frequencies and vibration modes.

2.1 Equilibrium Configuration for a Liquid Bounded by Free Surface and Axisymmetric Container

Of paramount importance to the solution of the low Bond number slosh problem is definition of the equilibrium free surface configuration. We seek a solution, for a given container geometry, such that two conditions are satisfied. The conditions are:

a) ullage (or liquid) volume consistent with user specifications, and

b) liquid/container contact angle consistent with user specifications.

In the following sections we develop a formulation for the equilibrium surface shape consistent with the above two specifications. The analytical developments presented herein assume a geometrically axisymmetric container subjected to an axisymmetric acceleration field.

2.1.1 Axisymmetric Meniscus Shape - Under the assumptions listed above, it is evident that the meniscus configuration will also be axisymmetric. Definition of the meniscus shape can be accomplished through examination of a force balance. Consider the annular ring cut from the meniscus as shown in Figure 2-1.

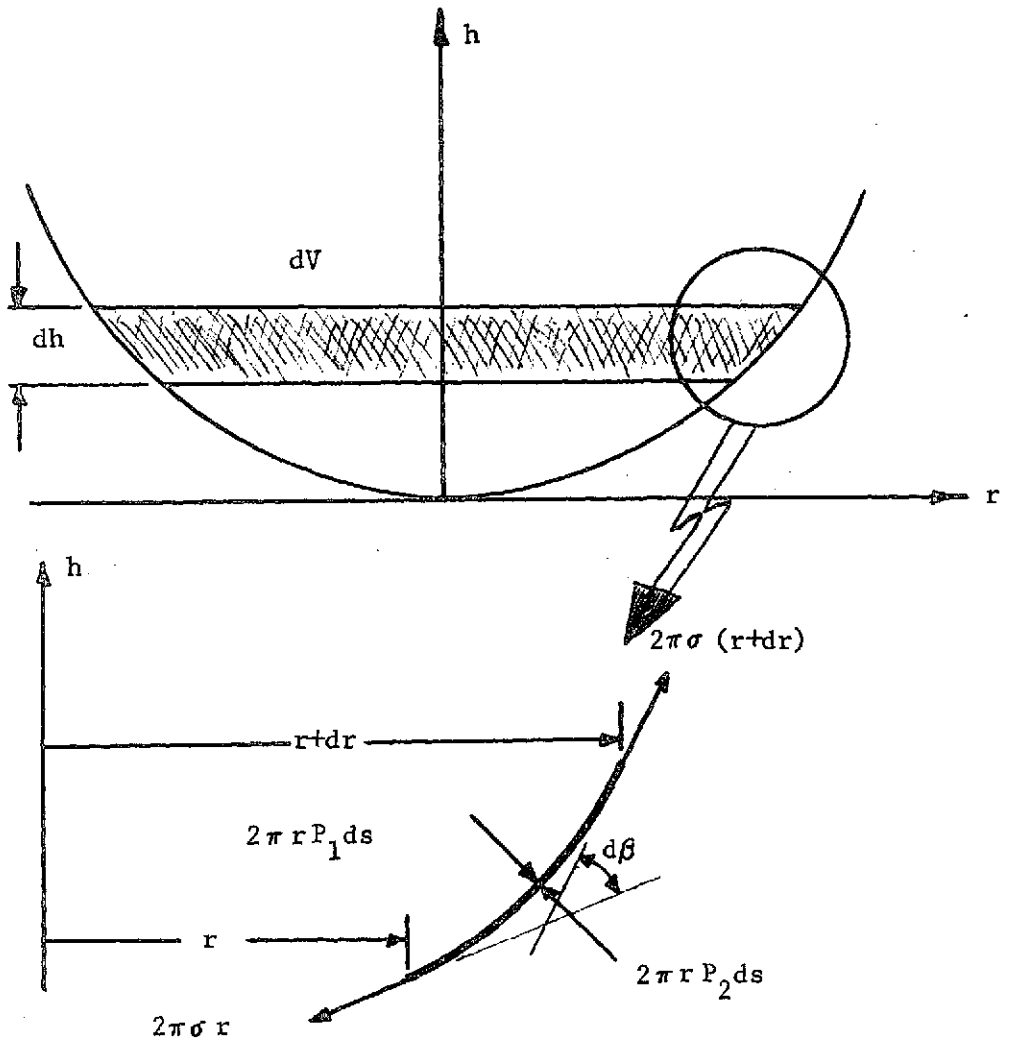


Figure 2-1. Force Balance on Axisymmetric Meniscus

A force balance in the vertical direction yields

$$2\pi\sigma(r+dr)\left[\sin(\beta+d\beta)\right] - 2\pi\sigma r \sin\beta = (P_1-P_2)\, 2\pi r \cos\left[\frac{\beta+\beta+d\beta}{2}\right] ds \qquad (1)$$

where σ = surface tension

P_1 = pressure above meniscus

P_2 = pressure below meniscus

r = radius

h = height

s = arc length

and it follows that

$$(r+dr)(\sin\beta \cos d\beta + \cos\beta \sin d\beta) - r \sin\beta$$

$$= \left(\frac{P_1-P_2}{\sigma}\right) r \left(\cos\beta \cos\frac{d\beta}{2} - \sin\beta \sin\frac{d\beta}{2}\right) ds \quad . \qquad (2)$$

For $d\beta$ small we have

$$\sin d\beta \simeq d\beta$$

$$\cos d\beta \simeq 1$$

and Equation (2) becomes

$$(r+dr)(\sin\beta + \cos\beta\, d\beta) - r \sin\beta$$

$$= \left(\frac{P_1-P_2}{\sigma}\right) r \left(\cos\beta - \sin\beta \frac{d\beta}{2}\right) ds \qquad (3)$$

or

$$r \cos\beta \frac{d\beta}{ds} + \sin\beta \frac{dr}{ds} = \left(\frac{P_1-P_2}{\sigma}\right) r \cos\beta \qquad (4)$$

where second order terms have been neglected.

Introducing

$$\sin\beta = \frac{dh}{ds} = h'$$

$$\cos\beta = \frac{dr}{ds} = r'$$

$$\cos\beta \frac{d\beta}{ds} = \frac{d^2h}{ds^2} = h''$$

yields

$$r\beta' \cos\beta + r' \sin\beta = \left(\frac{P_1-P_2}{\sigma}\right) r \cos\beta \qquad (5)$$

or

$$rh'' + r' h' = \left(\frac{P_1-P_2}{\sigma}\right) r\, r' \qquad (6)$$

and finally, we have the differential equation for the axisymmetric meniscus* as

$$\frac{d}{ds}(rh') = \left(\frac{P_1 - P_2}{\sigma}\right) rr' \quad . \tag{7}$$

A second differential equation arising from the geometry is

$$(r')^2 + (h')^2 = 1 \quad . \tag{8}$$

a) Liquid Below Meniscus

For a free meniscus with liquid below (Figure 2-2) we have

$P_1 = P_G$ = ullage gas pressure

$P_2 = P_L = P_{L_o} - \rho gh$ = liquid pressure

where ρ = liquid density

g = acceleration

P_{L_o} = liquid pressure at origin

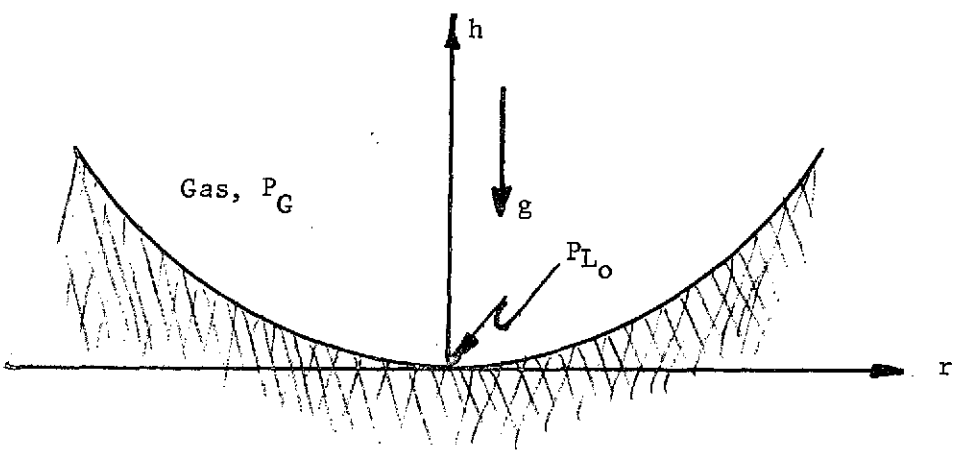


Figure 2-2. Meniscus With Liquid Below

* This is exactly the result presented in NASA SP-106, H. N. Abramson, editor

and the governing differential equations are

$$rh' + r'h' = rr'\left(\frac{P_G - P_{L_o} + \rho gh}{\sigma}\right)$$
$$(r')^2 + (h')^2 = 1 \qquad (9)$$

b) Liquid Above Meniscus

For a free meniscus with liquid above (Figure 2-3) we have

$P_1 = P_L = P_{L_o} - \rho gh$ = liquid pressure

$P_L = P_G$ = ullage gas pressure

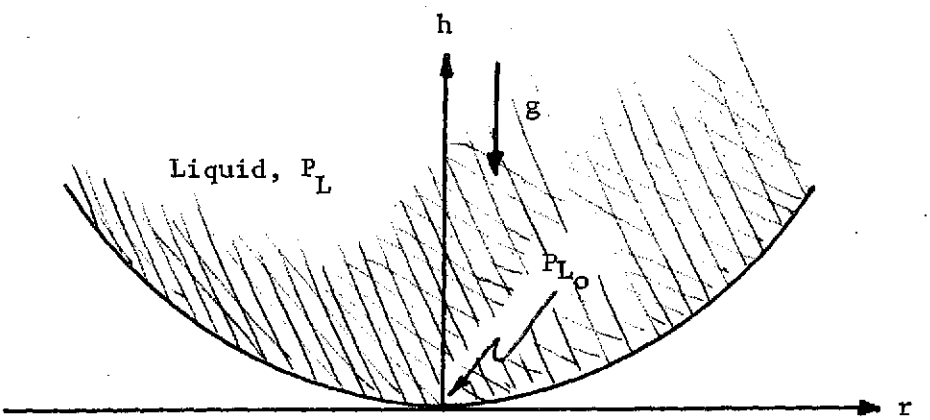


Figure 2-3. Meniscus With Liquid Above

and the governing differential equations are

$$rh'' + r'h' = -rr'\left(\frac{P_G - P_{L_o} + \rho gh}{\sigma}\right)$$
$$(r')^2 + (h')^2 = 1 \qquad (10)$$

2.1.2 Volume Contained Within Meniscus - With reference to Figure 2-1, it is seen that the volume of liquid (or ullage gas) contained within the boundary of the meniscus can be represented as

$$dV = \pi r^2 dh \qquad (11)$$

or $V' = \pi r^2 h'$ (11)

and when $h' > 0$ then $V' > 0$

and when $h' < 0$ then $V' < 0$

so that for

a. liquid below meniscus $V' > 0$ is ullage volume

$V' < 0$ is liquid volume

b. liquid above meniscus $V' > 0$ is liquid volume

$V' < 0$ is ullage volume

2.1.3 The State Equations - Definition of the applicable set of state equations follows from Equations (7), (8) and (11). We have

$$rh'' + r'h' = \left(\frac{P_1 - P_2}{\sigma}\right) rr'$$

$$r'r'' + h'h'' = 0 \qquad (12)$$

$$V' = \pi r^2 h'$$

with $P_1 = P_G$

$P_2 = P_{L_o} - \rho gh$ for liquid below meniscus,

and $P_1 = P_{L_o} - \rho gh$

$P_2 = P_G$ for liquid above meniscus.

We now define a set of state variables as

$$Y = \begin{Bmatrix} r \\ h \\ r' \\ h' \\ V/\pi \end{Bmatrix} \quad ; \quad Y' = \begin{Bmatrix} r' \\ h' \\ r'' \\ h'' \\ V'/\pi \end{Bmatrix} \qquad (13)$$

and the governing state equations become

$$rh'' + r'h' = (A + Bh)\, rr'$$

$$r'r'' + h'h'' = 0 \qquad (14)$$

$$V'/\pi = r^2 h'$$

with $A = (P_G - P_{L_o})/\sigma$

$B = \rho g/\sigma$ for liquid below meniscus,

and $A = (P_{L_o} - P_G)/\sigma$

$B = -\rho g/\sigma$ for liquid above meniscus.

The system governing state equations follow as

$$Y_1' = r' = Y_3$$

$$Y_2' = h' = Y_4$$

$$Y_3' = r'' = -Y_4\,(A + BY_2 - Y_4/Y_1)$$

$$Y_4' = h'' = Y_3\,(A + BY_2 - Y_4/Y_1)$$

$$Y_5' = Y_1^2\, Y_4$$

subject to the initial values

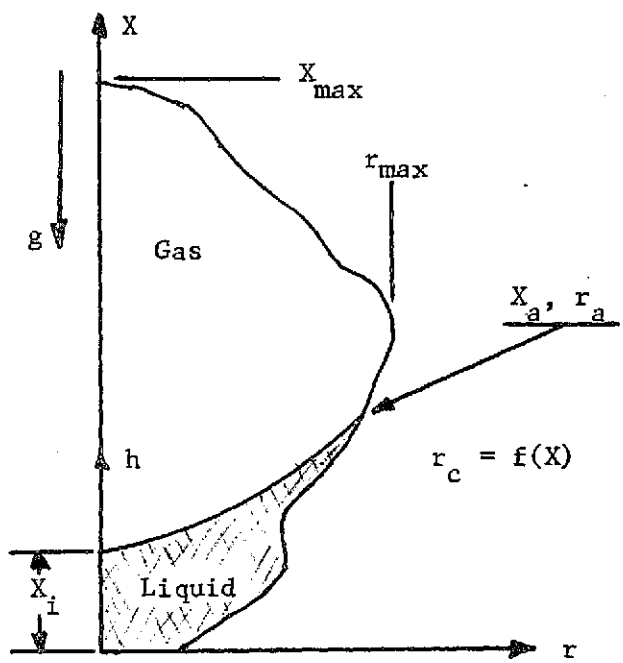
$$Y_1(0) = 0 \qquad Y_4(0) = 0$$

$$Y_2(0) = 0 \qquad Y_5(0) = 0. \qquad (16)$$

$$Y_3(0) = 1$$

The equations can be integrated using a numerical procedure to yield a trajectory which defines (for specified values of the coefficients A and B) continuous values of h, r and the contained volume V. The apparent singularity at the origin occuring in the 3rd and 4th of Equation (15) can be resolved through application of L'Hospital's Rule.

2.1.4 Definition of the Container - The development to this point has considered only the axisymmetric meniscus shape without regard to the nature of the container. For purposes of this investigation, it will be convenient to describe the container geometry as shown in Figure 2-4.



Bond No. = $B_o = \frac{\rho g X_{max}^2}{\sigma}$

Figure 2-4. Container Geometry

2.1.5 Definition of the Contact Angle - Consider the fluid free surface intersection with the container. Using the usual definition of contact angle (i.e., the angle between the container and the fluid surface measured through the fluid) we can establish a measure of the contact angle through examination of Figure 2-5. We define*

$$\tan \gamma = \left. \frac{dr_c}{dX_c} \right|_{X=X_a} = r_c'$$

$$\tan \alpha = r_f'/X_f' \Big|_{X=X_a}$$

where $X_f' = \frac{dX_f}{ds} = \frac{d}{ds}(X_i + h) = h'$

---

* The subscripts f and c refer to fluid and container, respectively; the subscript a denotes the fluid/container intersection point.

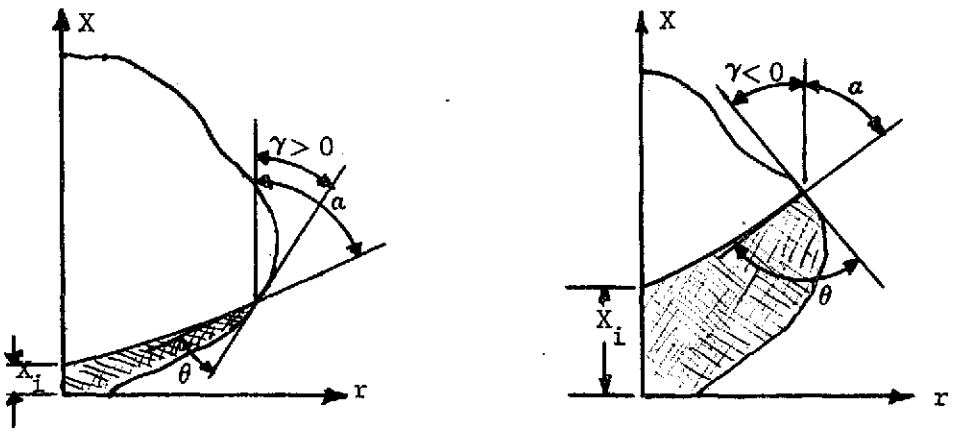


Figure 2-5. Contact Angle Geometry

and, with reference to the figure, it is evident that the contact angle is

$$\theta = \alpha - \gamma \tag{17}$$

2.1.6 Implementation - The digital computer program to define the static equilibrium shape runs in one of two distinct and independent modes of operation. The mode selection is controlled by the Boolean input control variable SEARCH. If SEARCH is input as .TRUE., the search mode is selected; if SEARCH is input as .FALSE., the survey mode is selected. The same program is used in both cases, however, the input requirements and the outputs obtained are different enough to justify separate discussion of each mode.

2.1.6.1 Survey Mode - While developing and checking out what is now the search mode, it was found to be quite difficult, in many cases, to supply the program with initial values of the iteration parameters, A_o and X_i, sufficiently close to the desired values to guarantee convergence of the solution algorithm. (A complete discussion of A_o and X_i will be found in Section 2.1.6.2.) This difficulty is especially severe prior to the user obtaining any computational experience with a given container. In other words, the user thinks of the problem in terms of contact angle (θ) and ullage volume (V_u) but the solution algorithm developed from the differential equations derived in Section 2.1.3 requires A_o and X_i to effect a solution. Therefore, the survey mode was developed and incorporated into the program to assist the user in generating a mapping, or a transformation from θ, V_u coordinates to

A_o, X_i coordinates. It is strongly recommended that the user take advantage of this feature before attempting to arrive at a specific solution via the search mode.

The survey mode operates by cycling through a user specified range of the two iteration parameters producing a table of θ and V_u vs X_i for each A_o. On option, the table may be plotted, three plot frames (θ vs X_i, V_u vs X_i, and θ vs V_u) for each A_o. This procedure may need to be repeated several times with different ranges on A_o and X_i to obtain the desired degree of correspondence but is inexpensive compared to running the search mode blind with trial and error initial values. Given this survey data and a desired θ and V_u, the user can interpolate and/or cross-plot to obtain good initial values of the iteration parameters for input to the search mode.

Specific input requirements for the survey mode are given in Section 6.2.1.1. A sample of the tabular printed output is shown in Figure 3-1, and samples of the optional plotted output are displayed in Figures 3-2 through 3-5.

2.1.6.2 Search Mode - The solution algorithm as encoded in the search mode of the digital computer program consists of three phases: input and initialization, iteration, and output of solution.

The input and initialization phase reads input data in NAMELIST format (as fully described in Section 6.2.1), performs several checks on the consistency of the input data, computes required constants, and initializes certain variables.

The iteration phase performs the computations required to arrive at the coordinates of the free surface static equilibrium shape. The four nested loops which comprise the iteration phase logic are, from inner to outer, the integration loop, the X loop, the A loop, and the main loop. The main loop is entered with current values of A_o, ΔA_o, X_i, ΔX_i, and the direction of search in the A coordinates (as denoted by the variable SGN) and control immediately falls through to the A-loop. In the search mode, the A-loop is repeated at most twice for each entry from the main loop. The A-loop increments the current A_o by ΔA_o in the proper direction ($A=A_o+SGN*\Delta A_o$), initializes the state vector and other data associated with integration of the state equations (Section 2.1.3) and enters the integration loop.

The integration loop uses the Runge-Kutta-Gill numerical integration algorithm to integrate the state equations, saving the r,X coordinates of the free surface trajectory obtained. The integration loop exits on any one of the following conditions:

a) the trajectory passes through the maximum radius of the container, $r > R_{max}$;

b) the trajectory passes through the axis of symmetry of the container, $r < 0$;

c) the trajectory passes through itself, $dr/ds > 0$ and $dX/ds < 0$, or $dr/ds < 0$ and $dX/ds > 0$, where s is the arc length coordinate.

These conditions are monitored by the Boolean function GOBACK and the loop is exited with the curfent trajectory in the array SOLN.

Upon exit from the integration loop, initialization is performed for the X-loop and the X-loop is entered. The X-loop is repeated the number of times specified by the input variable NX for each entry from the A-loop. It's function is to search each X_i in the range $X+\frac{1}{2}(NX)*\Delta X \leq X_i \leq X-\frac{1}{2}(NX)*\Delta X_i$ for an approximate value of the error function, ψ, where $\psi = \left[(V_{u_{current}} - V_{u_{desired}})/100.\right]^2 + \left[(\theta_{current} - \theta_{desired})/180.\right]^2$, less than the current value. If a smaller error is found, information identifying the current solution is saved and the loop is repeated until completion. Upon completion of the X-loop, one of two conditions exist: either a better solution was found, or a better solution was not found. If a better solution was not found, control passes to the top of the A-loop and the process is repeated for $A=A_o-SGN*\Delta A_o$. If no better solution is found on the second pass, the presumption is that ΔA_o and ΔX_i are too large; they are halved and control passes to the top of the main loop for a new search about the same A_o and X_i as before but with smaller values of ΔA_o and ΔX_i.

If a better solution is found upon completion of the X-loop, an accurate value of is computed for the current solution, and, if ψ is greater than the input tolerance, EPSC, A_o is set equal to A and control passes to the main loop for another search with no change in direction. This procedure is repeated until either convergence is established ($\psi <$ EPSC), in which case final output is generated, or until ΔA_o becomes less than a prescribed tolerance (currently 10^{-5} times the initial input ΔA_o) in which case the message HALVING LOOP,EXECUTION TERMINATED is printed and control passes to the input phase for the next case, if any.

Final output consists of tabulated values of the free surface equilibrium shape coordinates R and X as well as values of R', X', and incremental ullage volume V*.

2.1.6.3 Container Definition - In order to provide the maximum flexibility in specifying a container shape, it was decided to impose on the user the burden of writing a FORTRAN subroutine called CAN, with entry points RCAN and RCANP. RCAN is passed a value of X as an argument and computes the corresponding value of r. RCANP is passed a value of X as an argument and computes the corresponding value of dr/dX. The version of CAN presented herein (Section 6.1.1) computes these values for the container selected for the vibration analysis demonstration problem and can be used as a model for other containers.

2.2 Mass and Stiffness Matrices

Mass and stiffness matrices of the complete structure (only fluid is used in this study because the tank wall is assumed rigid) are calculated using a finite-element approach. In this approach, a continuous structure is assumed to be composed of simple, small structural elements such as tetrahedrons, pentahedrons, and hexahedrons for the volumetric fluid elements and triangles and quadrilaterals for the surface elements (both gravitational and surface tension). The derivation to obtain the finite element mass and stiffness matrices is based on kinetic energy and strain energy principles, respectively.

The kinetic energy for a complete structure may be expressed as

$$T = \frac{1}{2}\int\int\int \rho(X, Y, Z)\, \dot{\delta}^2\, (X, Y, Z, t)\, dX\, dY\, dZ \qquad (18)$$

where T = kinetic energy

ρ = mass density

$\dot{\delta}$ = time rate change of deflection

t = time

X, Y, Z = global coordinates

The difficulty in integrating equation (18) is expressing the deflection $\delta\,(X, Y, Z, t)$ as a continuous function over the complete structure. In the finite-element approach, however, this apparent difficulty is circumvented by idealizing the structure to be comprised of many small structural elements for which $\delta\,(X, Y, Z, t)$ can be expressed as a continuous function within the element boundaries. Thus, the expression (18) is valid for each of the finite-elements of the structure. Then the kinetic energy of the structure is the summation of the kinetic energies of each of the finite elements, that is,

$$T = \sum T_i \qquad (19)$$

where i refers to one particular finite element "i".

The common junction of finite elements is denoted as panel points, nodes or joints. Joints will be used here. The deflection $\delta\,(X, Y, Z, t)$ is easily expressed as a simple function of the joint deflections. These element joint deflections are then generalized coordinates or degrees of freedom of the complete structure.

The approach is to derive the mass matrix for finite-element, "i", in a convenient local coordinate system and then transform it to the global coordinate system. The technique is outlined here:

$$T_i = \tfrac{1}{2} \left\{ \dot{h}_L(t) \right\}_i^T \left[m_L \right]_i \left\{ \dot{h}_L(t) \right\}_i \tag{20}$$

where $\left[m_L \right]_i$ = the mass matrix in the local coordinate system for the ith element. This mass matrix is obtained by integration using an assumed displacement function. The discussion is deferred till later.

$\left\{ \dot{h}_L(t) \right\}_i$ = the time rate of change of the joint deflections of finite-element, "i". This is in the local system.

The deflections in the local coordinate systems are related to deflections in the global coordinate directions by a transformation matrix, $\left[\gamma \right]_i$ of direction cosines. Thus,

$$\left\{ h_L(t) \right\}_i = \left[\gamma \right]_i \left\{ h_G(t) \right\}_i \tag{21}$$

where $\left\{ h_G(t) \right\}_i$ = the joint deflections of finite element, "i", in the global coordinate system.

Using equation (21) in equation (20)

$$T_i = \tfrac{1}{2} \left\{ \dot{h}_G(t) \right\}_i^T \left[m_G \right]_i \left\{ \dot{h}_G(t) \right\}_i \tag{22}$$

where $\left[m_G \right]_i = \left[\gamma \right]_i^T \left[m_L \right]_i \left[\gamma \right]_i$ (23)

is the mass matrix with respect to the global coordinate system for the ith finite-element. Further, all the elemental mass matrices are finally assembled to give the mass matrix of the total structure, as shown in equation (19).

The development of the finite-element stiffness matrices is similar to that of the mass matrices. The strain energy for the structure may be expressed as the summation of the strain energies of each finite elements. That is,

$$U = \sum U_i \tag{24}$$

As was done for the finite-element mass matrix, the stiffness matrix for finite-element, "i", is derived in a convenient local coordinate system. Thus,

$$U_i = \tfrac{1}{2} \left\{h_L(t)\right\}_i^T \left[K_L\right]_i \left\{h_L(t)\right\}_i \tag{25}$$

where $\left[K_L\right]_i$ = the stiffness matrix with respect to local coordinate directions for finite element, "i". This stiffness matrix is obtained by integration using an assumed displacement function. This will be discussed later.

$\left\{h_L(t)\right\}_i$ = the joint deflections of finite element, i, measured in local coordinate system.

The same transformation matrix, $[\gamma]_i$, which was used in equation (22) is used here to relate the deflections in local coordinates to deflections in global coordinates. Substitute then to give

$$U_i = \tfrac{1}{2} \left\{h_G(t)\right\}_i^T \left[K_G\right]_i \left\{h_G(t)\right\}_i \tag{26}$$

where $$\left[K_G\right]_i = [\gamma]_i^T \left[K_L\right]_i [\gamma]_i \tag{27}$$

is the stiffness matrix with respect to the global coordinate system for the ith element.

Euler angle rotations at some joints (where the body coordinate is needed to be different than that of the global coordinates) are input in the program to allow the joint degree of freedom at these joints to be different than that of global X, Y, Z directions.

2.2.1 Surface Tension Finite Element - The basic surface tension element is a triangle; quadrilateral elements are formed by taking the average of the four overlapping triangles created by the diagonals.

For each triangle element, a local Cartesian coordinate system is defined such that vertex 1 is at the origin, vertex 2 is on the positive x-axis, vertex 3 is in the positive y direction as shown below.

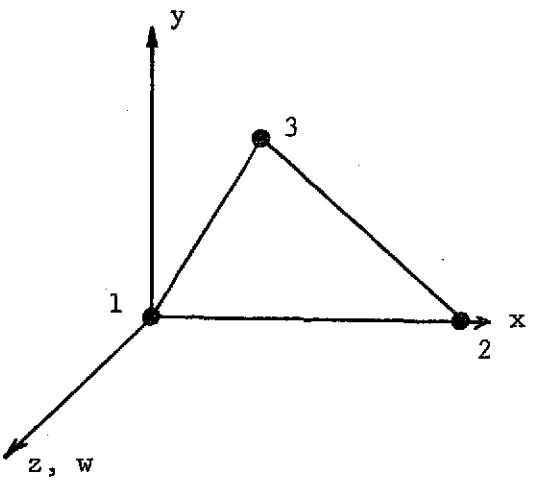


Figure 2-6. Triangle Coordinate System

The displacement field, in the normal direction (z), is chosen to be compatible with the triangle gravity element and the tetrahedron fluid element and will be linear of the form

$$w = a + b\,x + c\,y \tag{28}$$

The coefficients (a, b, c) are eliminated in terms of the 3 vertex displacements (w_1, w_2, w_3).

The mass matrix for the surface tension element is zero because there is no contribution of this element to the kinetic energy of the total system.

The stiffness matrix for the surface tension element is obtained as follows. Surface energy is associated with the liquid/vapor interface and the work done by the external liquid molecules to extend the surface. To simplify calculations, a hypothetical tension that acts in all directions parallel to the surface is substituted for the surface energy. This hypothetical tension is generally termed "surface tension". Surface tension has the same dimensions as surface energy per unit surface area, and it must have the same numerical magnitude. The concept of liquid surfaces behaving like a stretched membrane must not be misconstrued because surface energy is the fundamental liquid property and surface tension is merely a mathematical equivalent. The derivation given here is similar to that given in Reference (2) under Flexure of Plates with Simultaneous In-Plane Forces.

The effect of the normal deflection, w, is to introduce additional in-plane strains. Considering the figure (2-7) below, if points A and B move vertically, then the original length Δx becomes

$$\left\{ (\Delta x)^2 + \left(\frac{\partial w}{\partial x}\, \Delta x \right)^2 \right\}^{\frac{1}{2}} \tag{29}$$

$$= \Delta x \left\{ 1 + \tfrac{1}{2} \left(\frac{\partial w}{\partial x} \right)^2 + \ldots \right\} \tag{30}$$

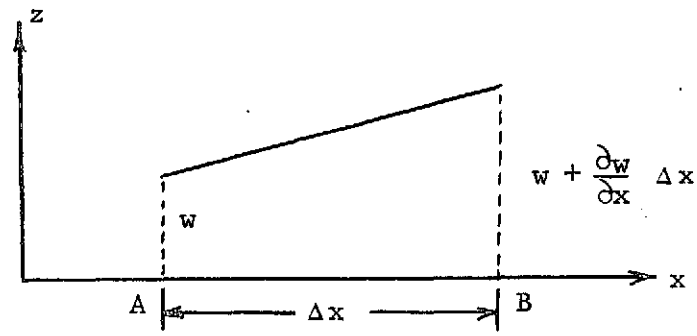


Figure 2-7. In-Plane Strain Due to Normal Deflection

The extension of the plate is then (neglecting higher order terms)

$$u = \tfrac{1}{2} \Delta x \left(\frac{\partial w}{\partial x} \right)^2 \tag{31}$$

and the in-plane strain ($u/\Delta x$) is

$$e_x = \tfrac{1}{2} \left(\frac{\partial w}{\partial x} \right)^2 \tag{32}$$

Similarly, in the y direction,

$$e_y = \tfrac{1}{2} \left(\frac{\partial w}{\partial y} \right)^2 \tag{33}$$

The in-plane strain energy thus becomes

$$U_{st} = \tfrac{1}{2} \int\int \sigma \left\{ \left(\frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial y} \right)^2 \right\} dxdy \tag{34}$$

Where σ is the coefficient of surface tension. Noting that the slopes are related to vertex displacements, we may write

$$\begin{Bmatrix} \frac{\partial w}{\partial x} \\ \frac{\partial w}{\partial y} \end{Bmatrix} = [G]\{\delta\} \tag{35}$$

where

$$\{\delta\}^T = \begin{bmatrix} w_1, & w_2, & w_3 \end{bmatrix} \tag{36}$$

and $[G]$ is a differentiation matrix.

The potential energy for the element becomes

$$U_{st} = \tfrac{1}{2} \{\delta\}^T \begin{bmatrix} K_{st} \end{bmatrix} \{\delta\} \tag{37}$$

where $[K_{st}]$ is the stiffness matrix for the surface tension element and is defined as

$$\begin{bmatrix} K_{st} \end{bmatrix} = \sigma \iint [G]^T [G] \, dxdy \tag{38}$$

Using the linear displacement function of Equation (28)

$$w = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{39}$$

To evaluate a, b, c

$$\underbrace{\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}}_{\{\delta\}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & x_2 & 0 \\ 1 & x_3 & y_3 \end{bmatrix}}_{[A]} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{40}$$

From which

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = [A]^{-1} \{\delta\} \tag{41}$$

where

$$[A]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1/x_2 & 1/x_2 & 0 \\ (x_3-x_2)/x_2y_3 & -x_3/x_2y_3 & 1/y_3 \end{bmatrix} \tag{42}$$

$$w = [1 \quad x \quad y] \quad [A]^{-1} \{\delta\} \tag{43}$$

Now

$$\begin{bmatrix} \frac{\partial w}{\partial x} \\ \frac{\partial w}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [A]^{-1} \{\delta\} \tag{44}$$

By comparing the above equation with Equation (35), we see that

$$[G] = \frac{1}{2A} \begin{bmatrix} -y_3 & y_3 & 0 \\ x_3-x_2 & -x_3 & x_2 \end{bmatrix} \tag{45}$$

where $A = \frac{1}{2} x_2 y_3$ (the area).

Then from Equation (38)

$$\begin{aligned} [K_{st}] &= \sigma A \quad [G]^T \quad [G] \\ &= \frac{\sigma}{4A} \left[\begin{array}{c|c|c} y_3^2 + (x_3-x_2)^2 & -y_3^2 - x_3 (x_3-x_2) & x_2 (x_3-x_2) \\ -y_3^2 - x_3 (x_3-x_2) & y_3^2 + x_3^2 & -x_2 x_3 \\ x_2 (x_3-x_2) & -x_2 x_3 & x_2^2 \end{array}\right] \end{aligned} \tag{46}$$

2.2.2 Gravity Finite Elements - The basic gravity element is a triangle; quadrilateral elements are formed by taking the average of the four overlapping triangles created by the diagonals. This element uses a linear displacement field ($\overline{w}$) that is boundary conformable.

The mass matrix for the gravity element is zero because there is no contribution of this element to the kinetic energy of the total system.

The stiffness matrix for the gravity element is obtained by expressing the gravitational potential energy in terms of the vertex displacements as

$$U_g = \tfrac{1}{2} \rho g \int_{\text{area}} (\bar{w} \cdot \bar{n}) \, (\bar{w} \cdot \bar{e}) \, ds \qquad (47)$$

where U_g = gravitational potential energy

$\bar{n}$ = unit outer normal

$\bar{e}$ = a unit vector parallel with the gravity vector $\bar{g}$, but of opposite sense, i.e., $\bar{e} = -\bar{g}/g$

A noteworthy observation can be made with reference to the gravitational potential energy expressed in Equation (47). We note that since we have a boundary conformable element, the surface integrals such as Equation (47) will all cancel each other throughout the interior of the fluid in a container, since $\bar{n}$ on common element boundaries is equal and opposite. Thus, the gravitational potential energy will depend only on displacement coordinates at the boundary of the entire volume of fluid (the free surface and the wetted container wall).

Notice also that for a rigid tank, $\bar{w} \cdot \bar{n}$ is non-zero only at the free surface where $\bar{e} = \bar{n}$; thus,

$$U_g = \tfrac{1}{2} \rho g \int_{\substack{\text{Free} \\ \text{Surface}}} (\bar{w} \cdot \bar{n})^2 \, ds, \qquad (48)$$

a much more familiar expression than that of Equation (47).

2.2.3 Fluid Finite Element - The basic fluid element is a tetrahedron; pentahedron elements and hexahedron elements are synthesized, simply by placing six and ten overlapping tetrahedrons together, respectively and averaging the result. The averaging is carried out to eliminate the bias, if any.

For each tetrahedron element, a local Cartesian coordinate system is defined so that vertex 1 is the origin, the x-axis includes vertex 2, vertex 3 lies in the x-y plane and vertex 4 always has a positive z-coordinate (Figure 2-8).

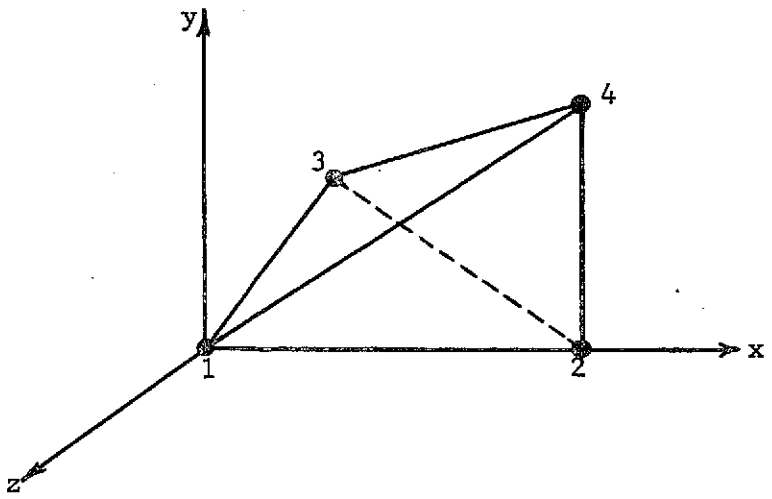


Figure 2-8. Local Coordinate System for Tetrahedron Element

This element uses a linear displacement field (constant strain) that is boundary conformable. The displacement field throughout the element is expressed in terms of coordinate locations and appears as

$$\bar{w}\ (x,\ y,\ z,\ t) = \bar{a}_o + \bar{a}_1\ x + \bar{a}_2\ y + \bar{a}_3\ z \tag{49}$$

The coefficients $\bar{a}_k(t)$, k = 0, 1, 2, 3 are eliminated in terms of the 12 vertex displacements.

The mass matrix for the fluid elements is obtained by expressing the kinetic energy as

$$T = \tfrac{1}{2} \int_{Vol} \dot{\bar{w}} \cdot \dot{\bar{w}}\ \rho dv \tag{50}$$

where T = kinetic energy

$\dot{\bar{w}} = \dot{\bar{a}}_o + \dot{\bar{a}}_1\ x + \dot{\bar{a}}_2\ y + \dot{\bar{a}}_3\ z$

ρ = mass density

This gives rise to a (12x12) mass matrix.

The stiffness matrix for the fluid element is obtained by expressing the volumetric dilatation strain energy in terms of vertex displacement coordinates as

$$U_D = \tfrac{1}{2} \int_{Vol} K\theta^2\ dv \tag{51}$$

where U_D = volumetric dilatation energy

K = fluid bulk modulus

θ = volumetric strain

2.3 Vibration Analysis

Two methods are included in this study for the solution of the eigenvalue/vector (i.e., frequencies/mode shapes) problem. The first method uses a Jacobi technique and is documented in Reference 3 in the description of Subroutine MODE1. This subroutine is used to calculate frequencies/mode shapes for small and intermediate size problems (approximately 120 DOF on a computer with 65000 core). No description of the method will be given here because the use of this subroutine is straightforward.

For larger size problems, a second method of calculating the frequencies/mode shapes is included. This is the iterative Rayleigh-Ritz method which is described in References (1) and (4). In this method, a large problem is reduced to a smaller problem in a particular frequency range. Mode shapes are initially assumed and then by the iterative technique are improved until they converge to the normal vibration modes of the structure. Because the iterative Rayleigh-Ritz method is not as straightforward as the first method described above for smaller size problems, the iterative Rayleigh-Ritz method is briefly described here.

For a discrete coordinate model of a structure having n degrees of freedom, the equations of motion can be written as

$$[M] \{\ddot{h}\} + [K] \{h\} = 0 \tag{52}$$

where $\{h\} = \{h(t)\}$ vector of discrete coordinate displacements,

$[M]$ = mass matrix

$[K]$ = stiffness matrix

If a solution of the type $\{h\} = \{h\} e^{iwt}$, implying a simple harmonic motion is assumed, equation (52) can be written as

$$\left([K] - \omega^2 [M] \right) \{h\} = \{0\} \tag{53}$$

Equation (53) is recognized as a matrix eigenvalue problem of order n, whose eigenvectors $[\Phi]$ are the mode shapes and whose eigenvalues $[\omega^2]$ are the frequencies. A complete sequence of trial vectors

$$\{V\}_1 , \{V\}_2 , \{V\}_3 , \text{----------} \{V\}_n \tag{54}$$

which are linearly independent, is assumed. The displacement $\{h\}$ is then expressed as a linear sum of the first "m" trial vector, that is,

$$\underset{(nx1)}{\{h\}} = \underset{(nxm)}{[V]} \quad \underset{(mx1)}{\{q\}} \tag{55}$$

Substitution of Equation (55) into (53) and multiplying by $[V]^T$ gives

$$\left([K^*] - \omega^2 [M^*]\right) \{q\} = \{0\} \tag{56}$$

where

$$[K^*] = [V]^T [K] [V] \tag{57}$$

and

$$[M^*] = [V]^T [M] [V] \tag{58}$$

Equation (56) is a matrix eigenvalue problem of reduced order "m" whose eigenvectors are $[\Phi^*]$ and eigenvalues are $[\omega^2]$. The solution of equation (56) has the form

$$\{q\} = [\Phi^*] \{q^*\} \tag{59}$$

where $\{q^*\}$ is the normalized coordinate vector. The eigenvalues, $[\omega^2]$, approximate the first "m" eigenvalues of the original structure. The associated eigenvectors $[\Phi]$ of the original structure are obtained by substitution of Equation (59) into (55), yielding

$$\underset{(nx1)}{\{h\}} = \underset{(nxm)}{[V]} \quad \underset{(mxm)}{[\Phi^*]} \quad \underset{(mx1)}{\{q^*\}}$$

or

$$\{h\} = [\Phi]\{q^*\} \tag{60}$$

where

$$[\Phi] = [V][y^*] \tag{61}$$

The accuracy of the mode shapes $[\Phi]$ and frequencies $[\omega^2]$ obtained depends entirely upon the trial vector $[V]$. If $[V]$ contains the true modal patterns, then the eigensolution for $[\Phi]$ and $[\omega^2]$ are exact. However, in general, that is not the case. Exact results can be obtained for the first "m" modes of the structure if the trial vectors $[V]$ do not have any contribution from modes higher than "m". Thus, an improved set of trial vectors can be calculated by suppressing the contribution of higher modes in approximate mode shapes. The procedure for suppressing the contribution of the higher modes is well known; in fact, it is the basis of the Power or Stodola-Vianello' matrix iteration method of modal analysis. Here, however, the method is applied to all modes simultaneously and is given as,

$$[K][V] = [M][\Phi] \tag{62}$$

The solution is carried out for $[V]$, which is then used to repeat equations (56) through (62). The cycle can be repeated until all the mode shapes $[\Phi]$ and frequencies ω^2 have converged to within a prescribed tolerance. Convergence is assured because the technique is equivalent to a power iteration applied simultaneously to all modes. Thus, the convergence theorems associated with the power method are directly applicable. The role of the eigensolution (equation (56)) is to prevent all modes from converging on the lowest mode.

Associated with the iterative Rayleigh-Ritz technique are parameters that affect the convergence and, hence, computer time which will be briefly discussed here. They are:

a. the initial mode shapes assumed to start the iteration process,
b. the number of modes used,
c. the repression of higher modes, and
d. shifting.

a. Initially Assumed Mode Shapes - The choice of initial mode shapes plays a very important role in the success of the technique. Inherent with the initial mode shape selection are two basic problems: (1) modes may be missed, and (2) the triple product $[M^*] = [V]^T [M] [V]$ may be ill-conditioned if the columns of $[V]$ are not sufficiently independent. It does not appear that there is a way to guarantee that the above two conditions will be met with any selection of $[V]$, however, the chance of them occurring can be minimized with some judicious selection of the vectors. If the elements of the vector or of matrix $[V]$ are randomly generated, it has been found that the chances of the above two conditions being violated is very remote.

b. Number of Modes Used - An increase in the number of modes used will, in general, decrease the number of iterations required for convergence. However, if more modes are used, the computer time for each iteration will increase because of the increase in sizes of the matrices used. Determination of the optimum number of modes to use requires an empirical assessment.

c. Repression of Higher Modes - As pointed out earlier, exact results can be obtained for the first "m" modes of the structure if the trial vectors in $[V]$ do not contain any contribution from modes higher than "m". Generalizing, it can be said that an improved set of trial vectors can be calculated by suppressing the contribution from the higher modes in the approximate mode shapes at each step. This is achieved as follows.

$$[V]_j = [K]^{-1} [M] [V]_{j-1} \tag{63}$$

The subscript j denotes the iteration number. If this iteration is repeated sufficient number of times, modes corresponding to the lowest frequency will be reached. If this iteration is repeated too many times, the mode will repeat itself in one or more columns of $[V]$ and will render $[V]^T [M] [V]$ to be ill-conditioned. The use here is not to converge to a mode but just to repress the higher modes and, hence, just a one time application is advisable.

d. Shifting - Shifting is an useful technique to speed the convergence of modes whose eigenvalues are close to the shift value. An additional benefit of shifting process is the conversion of the stiffness matrix (in case of a free-free structure) from singular to a non-singular matrix. The method is as follows.

To introduce the shift value, λ_s, the quantity $\lambda_s [M]$ is added and subtracted in Equation (53) to give

$$\left([K] - \lambda_s [M] - \omega^2 [M] + \lambda_s [M] \right) \{h\} = \{0\} \quad . \tag{64}$$

Define

$$[\hat{K}] = [K] - \lambda_s [M] \tag{65}$$

and

$$\Omega^2 = \omega^2 - \lambda_s \tag{66}$$

to give

$$\left([\hat{K}] - \Omega^2 [M] \right) \{h\} = \{0\} \tag{67}$$

This is now the eigen-problem to be solved rather than (53). Note that $[\hat{K}]$ is non-singular even if $[K]$ was not.

The eigenvalues of the original system are easily obtained as

$$\omega^2 = \Omega^2 + \lambda_s \tag{68}$$

The convergence will be to the lowest absolute value of Ω^2. Thus, shifting by a value, λ_s, the eigenvalues, ω^2, around this shift point are converged to first.

Some general remarks on shifting follows.

1. Analysis of a Free-Structure - Because a free structure has a singular stiffness matrix, the solution of the simultaneous equations in the iteration loop is not possible. However, the shift technique alleviates the problem.

2. Specific Frequency Range - When a shift value is used, the modes with eigenvalues closest to the shift value will converge first, which enables one to obtain the modes in the desired frequency range only.

3. Large number of modes - By repeated use of different shift values, any number of modes can be obtained.

4. The following observations are made based on the results of Reference (4).

   a. If the lowest eigenvalues in the range $\omega_1^2, \omega_2^2, \ldots, \omega_i^2$ are needed, a shift value of zero should be used for a restrained structure and one for a free-free structure.

   b. If the modes are needed in an intermediate range, a shift midway between the lowest and the highest expected eigenvalues should be used.

2.4 Finite Element Model

After the static free surface shape has been established, the total fluid is modeled as an assemblage of finite elements for the vibration analyses. Instead of analyzing the complete tank and fluid, it is convenient to reduce the number of degrees of freedom (and, thus, reduce the computer time) by using a 90° sector. This 90° model requires four

sets of boundary conditions to completely represent the total 360° model. This technique will be discussed in Section 2.4.1. To reduce the amount of input data describing joint coordinate locations, degrees of freedom, and Euler angles along with the finite element joint numbers, a subroutine to generate this data was developed and is described in Section 2.4.2.

The vibration analysis is for an axi-symmetric tank with rigid walls as shown in Figure 2-9. Euler angle rotations are used to give body coordinate systems normal to the tank walls and, thus, allow for fluid slippage tangentially and zero penetration normally. For consistency, the "v" DOF is normal to the tank wall.

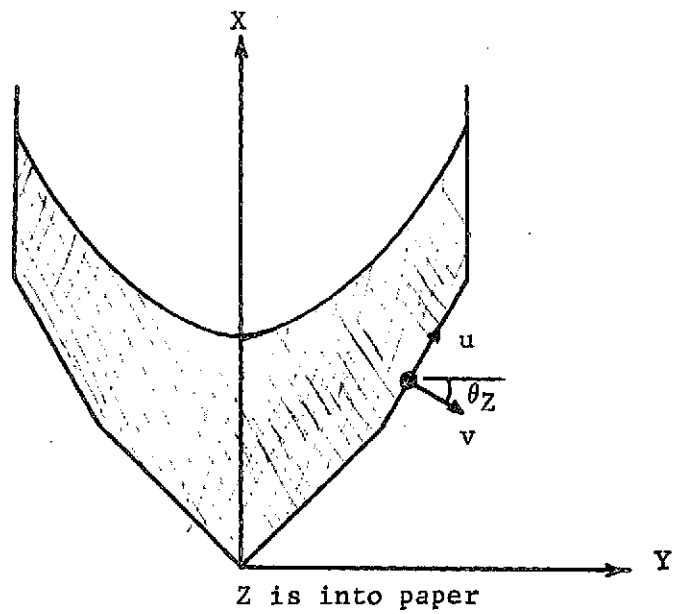
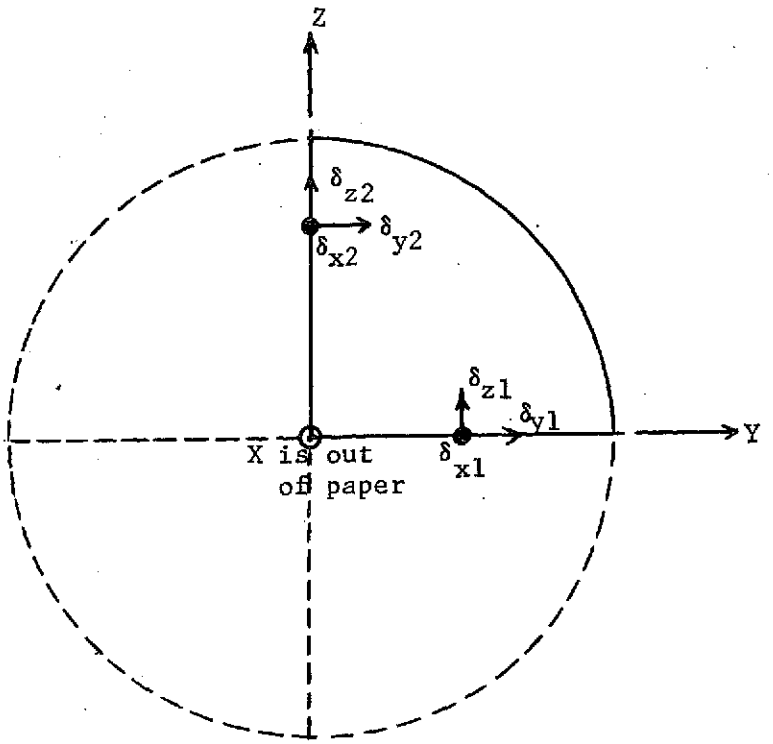


Figure 2-9. Tank/Fluid Geometry

2.4.1 90° Model Boundary Conditions - Using the four sets of boundary conditions given in Figure 2-10, the complete 360° model is represented. Because this study is only concerned with the lateral "Y" slosh modes, the symmetric/anti-symmetric boundary condition was the only condition used for the data generator.



Type of Motion		Boundary 1 (XY plane) Constraints		Boundary 2 (XZ plane) Constraints		Center Constraints	
Boundary 1 (XY plane)	Boundary 2 (XZ plane)	Allow	Fix	Allow	Fix	Allow	Fix
Symmetric	Symmetric	δ_x, δ_y	δ_z	δ_x, δ_z	δ_y	δ_x	δ_y, δ_z
Symmetric	Anti-Symmetric	δ_x, δ_y	δ_z	δ_y	δ_x, δ_z	δ_y	δ_x, δ_z
Anti-Symmetric	Symmetric	δ_z	δ_x, δ_y	δ_x, δ_z	δ_y	δ_z	δ_x, δ_y
Anti-Symmetric	Anti-Symmetric	δ_z	δ_x, δ_y	δ_y	δ_x, δ_z		δ_x, δ_y, δ_z

Figure 2-10. 90° Model Boundary Conditions

2.4.2 Data Generation - A computer subroutine (LBDGEN) was developed to generate joint coordinate locations, degrees of freedom, and Euler angles along with the finite element joint numbers. Using a drawing of the container wall and fluid surface in the XY plane, the analyst sketches the desired grid and obtains the joint X, Y coordinates and Euler angles (at the container wall). With this information, the number of sectors desired in 90^{o}, and other information detailed in Section 6.2.2, the data is generated to calculate the finite element mass and stiffness matrices for the fluid compressibility, gravity, and surface tension.

To avoid any ambiguity of the mode number of the first slosh mode, an algorithm to calculate the first slosh mode number was obtained as follows. The total number of degrees of freedom is given as

$$NDOF = M_o + M_s + M_c \tag{69}$$

where M_o = number of circulation (zero frequency) modes,

M_s = number of slosh modes which is the number of surface degrees of freedom,

M_c = number of crunch (high frequency) modes which is the number of fluid elements.

Because NDOF, M_s and M_c are easily calculated from the finite element geometry, the number of circulation modes is obtained as

$$M_o = NDOF - M_s - M_c \tag{70}$$

The mode number of the first slosh mode is then $M_o + 1$.

In terms of the grid sketched in the XY plane; NDOF, M_s and M_c are given as

$$NDOF = NGPAX + (2*NGPCW+3*NGPIS)*NSECT$$

$$M_s = 1 + (3*NGPFS+2)*NSECT$$

$$M_c = NGPEL*NSECT$$

where NGPAX = number of grid points on X-axis,

NGPCW = number of grid points on container wall (except at X-axis)

NGPIS = number of grid points on interior and surface (except at X-axis and container wall)

NSECT = number of sectors in 90° model

NGPFS = number of grid points on fluid surface (except X-axis and container wall)

NGPEL = number of grid point elements, i.e., number of elements on XY plane

Usage of these algorithms is demonstrated in Section 3.2.

3. RESULTS

3.1 Free Surface Static Equilibrium Shape

The digital computer program for generating the free surface static equilibrium shape has been verified by running several containers in both the survey mode and the search mode. The survey mode results consist of tabular printout and several plot frames. The tabular printout, a sample of which is shown in Figure 3-1, consists of values of ullage volume and contact angle vs tank axis intercept (X) for each value of A specified in the input data (Section 6.2.1). The asterisks indicate that no solution exists for this value of A and X. The plot output graphically displays this data in a manner intended to allow the user to visually bracket the desired values of ullage volume and contact angle and thus provide good initial values as input to the search mode. Ullage volume percent vs tank axis intercept plots are shown in Figure 3-2a; Figure 3-2b gives the correspondence between the numbers associated with each curve and the value of A. Contact angle vs tank axis intercept plots are shown in Figure 3-3a; Figure 3-3b gives the correspondence between the numbers associated with each curve and the value of A.

The range on tank axis intercept is XUP- ΔX to XLO+ ΔX where ΔX is given by (XUP-XLO)/NX and XUP, XLO and NX are input by the user. Figures 3-4a and 3-4b show the cross plot of ullage volume percentage vs contact angle for two consecutive values of A. The range of A is from ACOFO+DACOF to ACOFO+NA*DACOF where ACOFO, DACOF, and NA are input by the user. Figure 3-4c gives the correspondence between the numbers associated with each curve and the value of tank axis intercept. All of the above figures (3-1 through 3-4) apply to the tug like tank shown in Figure 3-8.

As an example of how to use the survey mode plotted output, consider determining initial conditions to the search mode to find the equilibrium shape for an ullage volume of 50% and a contact angle of 45°. For this particular tank, the cross plots, Figures 3-4a and 3-4b, are the most useful. The idea is to bracket the desired point and the solutions for A=0.05 (Figure 3-4a) lie to the left of the desired point and the solutions for A=0.04 (Figure 3-4b) lie to the right. Therefore, reasonable input values might be ACOFO=0.04 and DACOF=0.005. To determine the range on tank axis intercept note that the desired point lies between points numbered 17 and 19 on the plots. Referring to Figure 3-4c, point 17 corresponds to X=84.2 and point 19 corresponds to X=76.9; therefore, reasonable input values might be XUP=84, XLO=77, and NX=20.

The search mode has been run with a number of containers, Bond numbers, contact angles and ullage volumes. Results are plotted for a cylindrical tank (Figure 3-5), a spherical tank (Figure 3-6), a cosine tank (Figure 3-7) and the Tug-like tank (Figure 3-8) selected for vibration analysis. These results are presented to demonstrate the wide range of axisymmetric tanks to which the program may be applied.

RUN HY

SAMPLE SURVEY MODE RUN FOR
TUG-LIKE TANK

SURVEY SUMMARY

X COORD	A = .100 VUPCT	PHITD	A = .090 VUPCT	PHITD	A = .080 VUPCT	PHITD	A = .070 VUPCT	PHITD	A = .060 VUPCT	PHITD
142.837	.2	137.7	.2	139.2	.2	140.7	.2	142.3	.2	144.0
139.175	.7	119.3	.7	121.5	.8	123.7	.9	126.0	.9	128.2
135.512	1.4	104.7	1.6	107.4	1.7	110.1	1.9	112.8	2.1	115.5
131.850	2.4	91.8	2.7	95.0	2.9	98.2	3.2	101.3	3.6	104.3
128.187	3.6	80.0	4.0	83.7	4.4	87.2	4.9	90.7	5.5	94.0
124.525	4.9	68.8	5.5	73.1	6.2	77.1	6.9	80.7	7.7	84.0
120.862	6.3	58.1	7.2	63.1	8.1	67.4	9.1	71.1	10.2	74.5
117.200	7.7	47.5	8.9	53.2	10.2	58.2	11.5	62.0	12.9	65.0
113.537	9.9	36.3	10.6	43.8	12.3	49.3	14.1	53.4	15.9	55.6
109.875	9.8	23.0	12.1	34.1	14.5	40.8	16.8	44.9	19.1	46.4
106.212	******	******	13.3	22.8	16.6	32.6	19.7	37.0	22.5	36.9
102.550	******	******	******	******	18.2	24.1	22.5	29.8	26.0	27.2
98.887	******	******	******	******	19.3	11.3	25.1	23.3	29.6	16.7
95.225	******	******	******	******	******	******	27.9	17.3	33.3	13.2
91.562	******	******	******	******	******	******	28.7	3.7	36.9	13.2
87.900	******	******	******	******	******	******	******	******	40.5	13.2
84.237	******	******	******	******	******	******	******	******	44.2	13.2
80.575	******	******	******	******	******	******	******	******	47.8	13.2
76.912	******	******	******	******	******	******	******	******	51.4	13.2
73.250	******	******	******	******	******	******	******	******	55.1	13.2
69.587	******	******	******	******	******	******	******	******	58.7	13.2
65.925	******	******	******	******	******	******	******	******	62.3	13.2
62.262	******	******	******	******	******	******	******	******	66.0	13.2
58.600	******	******	******	******	******	******	******	******	69.6	13.2
54.937	******	******	******	******	******	******	******	******	73.2	13.2
51.275	******	******	******	******	******	******	******	******	76.9	13.2
47.612	******	******	******	******	******	******	******	******	80.5	13.2
43.950	******	******	******	******	******	******	******	******	84.1	13.2
40.287	******	******	******	******	******	******	******	******	87.8	7.7
36.625	******	******	******	******	******	******	******	******	91.3	12.0
32.962	******	******	******	******	******	******	******	******	94.4	5.0
29.300	******	******	******	******	******	******	******	******	96.4	19.7
25.637	******	******	******	******	******	******	97.5	18.2	97.7	27.6
21.975	******	******	98.3	5.7	98.5	19.6	98.5	27.5	98.6	33.8
19.312	99.1	16.4	99.1	23.9	99.2	29.7	99.2	34.7	99.2	39.0
18.650	99.6	29.6	99.6	33.7	99.6	37.5	99.6	40.7	99.6	43.8
10.987	99.8	38.9	99.8	41.4	99.8	44.0	99.8	46.0	99.8	48.3
7.325	99.9	46.8	99.9	48.1	99.9	49.5	100.0	50.8	100.0	52.1
3.662	100.0	53.7	100.0	54.3	100.0	55.0	100.0	55.6	100.0	56.2

Figure 3-1. SAMPLE TABULAR OUTPUT FROM FREE SURFACE EQUILIBRIUM SHAPE PROGRAM, SURVEY MODE

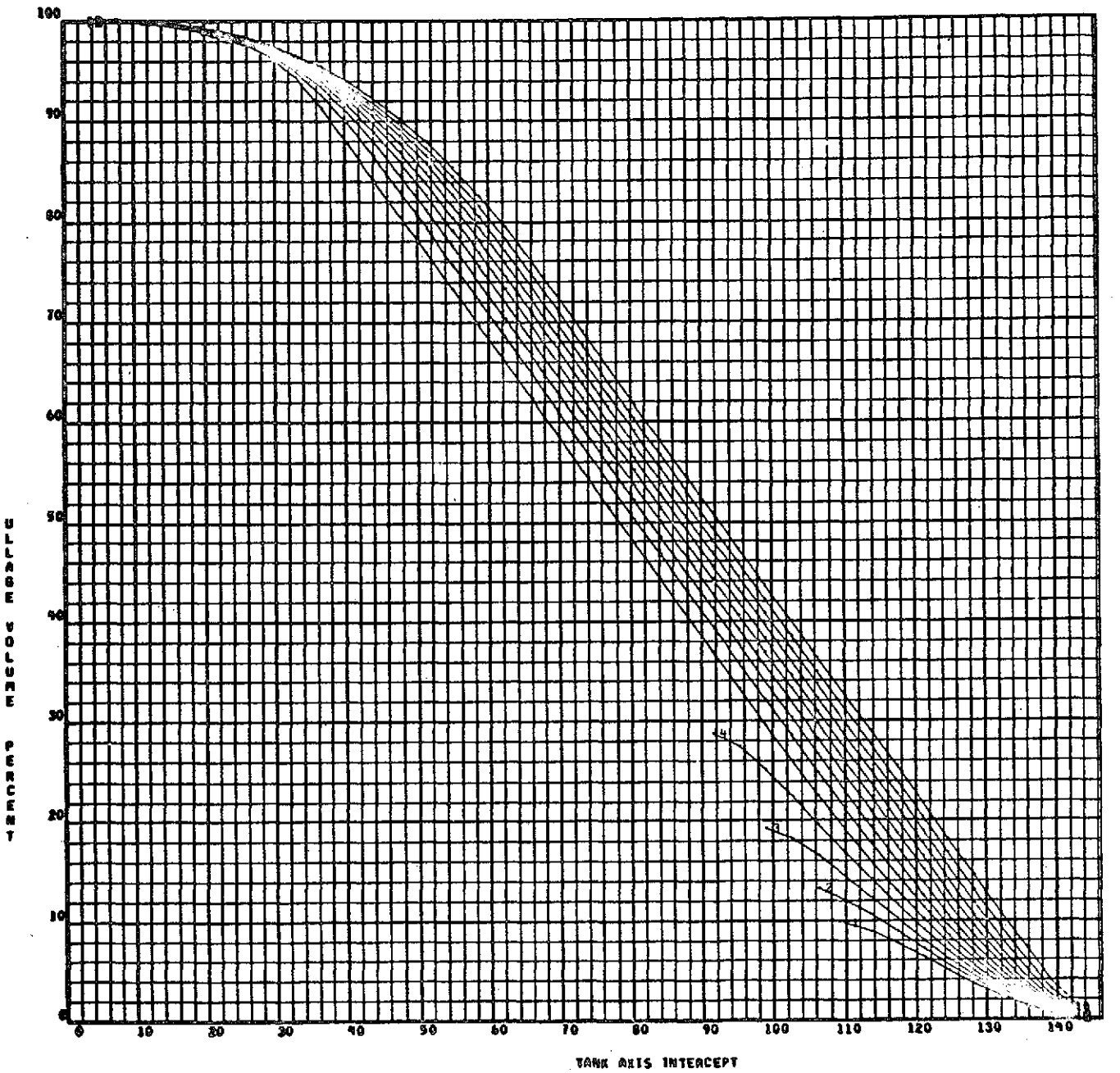


Figure 3-2a. ULLAGE VOLUME VS TANK AXIS INTERCEPT

PLOT SYMBOLS FOR
ULLAGE VOLUME PERCENT VS TANK AXIS INTERCEPT

SYMBOL	&COF
1	.10000000+00
2	.90000000-01
3	.79999999-01
4	.69999999-01
5	.60000000-01
6	.50000000-01
7	.40000000-01
8	.30000000-01
9	.20000000-01
10	.99999999-02
11	.00000000
12	-.99999999-02
13	-.20000000-01
14	-.30000000-01

Figure 3-2b. ULLAGE VOLUME VS TANK AXIS INTERCEPT, PLOT SYMBOLS

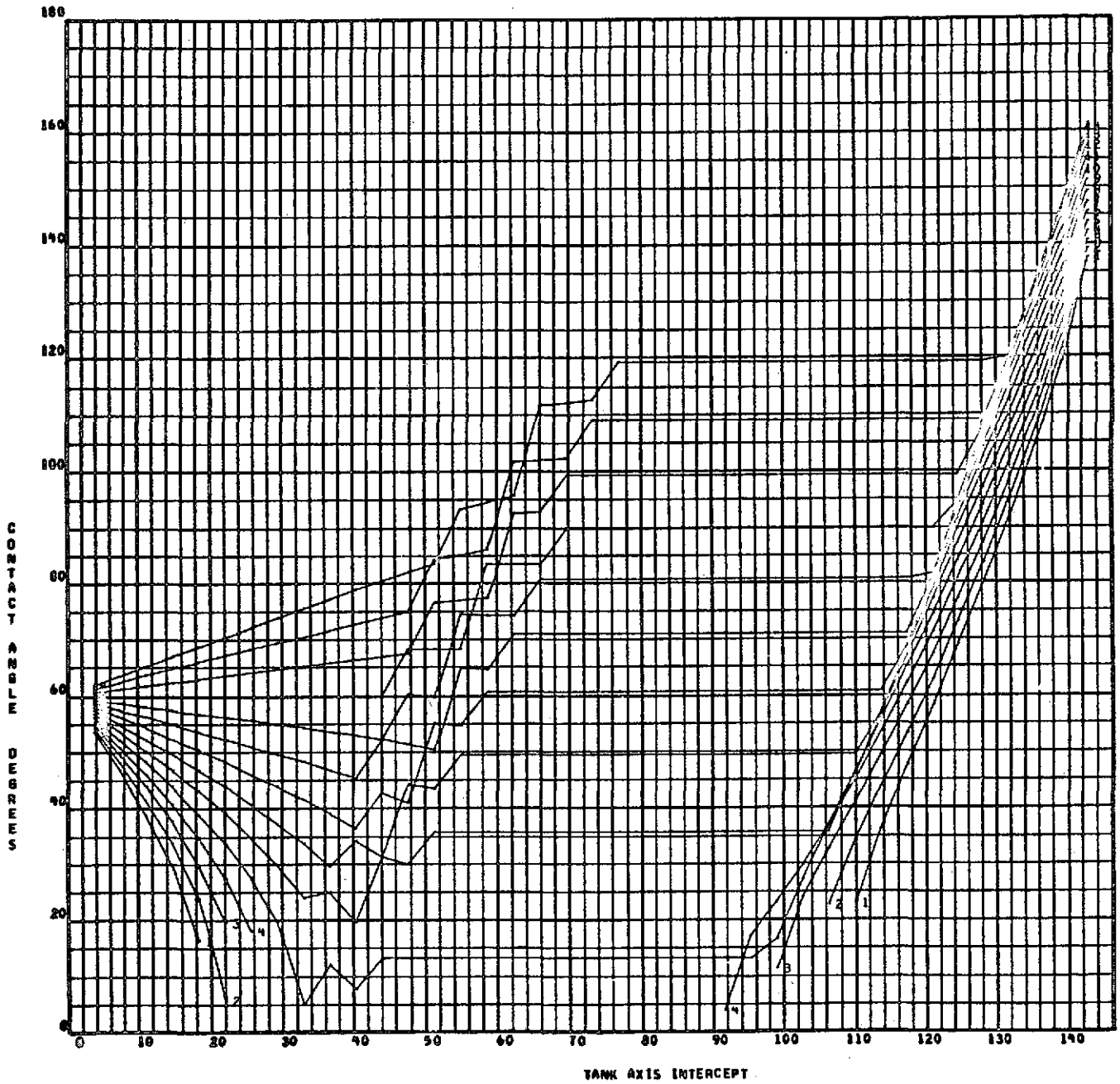


Figure 3-3a. CONTACT ANGLE VS TANK AXIS INTERCEPT

PLOT SYMBOLS FOR
CONTACT ANGLE DEGREES VS TANK AXIS INTERCEPT

SYMBOL	ACOF
1	.10000000+00
2	.90000000-01
3	.79999999-01
4	.69999999-01
5	.60000000-01
6	.50000000-01
7	.40000000-01
8	.30000000-01
9	.20000000-01
10	.99999999-02
11	.00000000
12	-.99999999-02
13	-.20000000-01
14	-.30000000-01

Figure 3-3b. CONTACT ANGLE VS TANK AXIS INTERCEPT, PLOT SYMBOLS

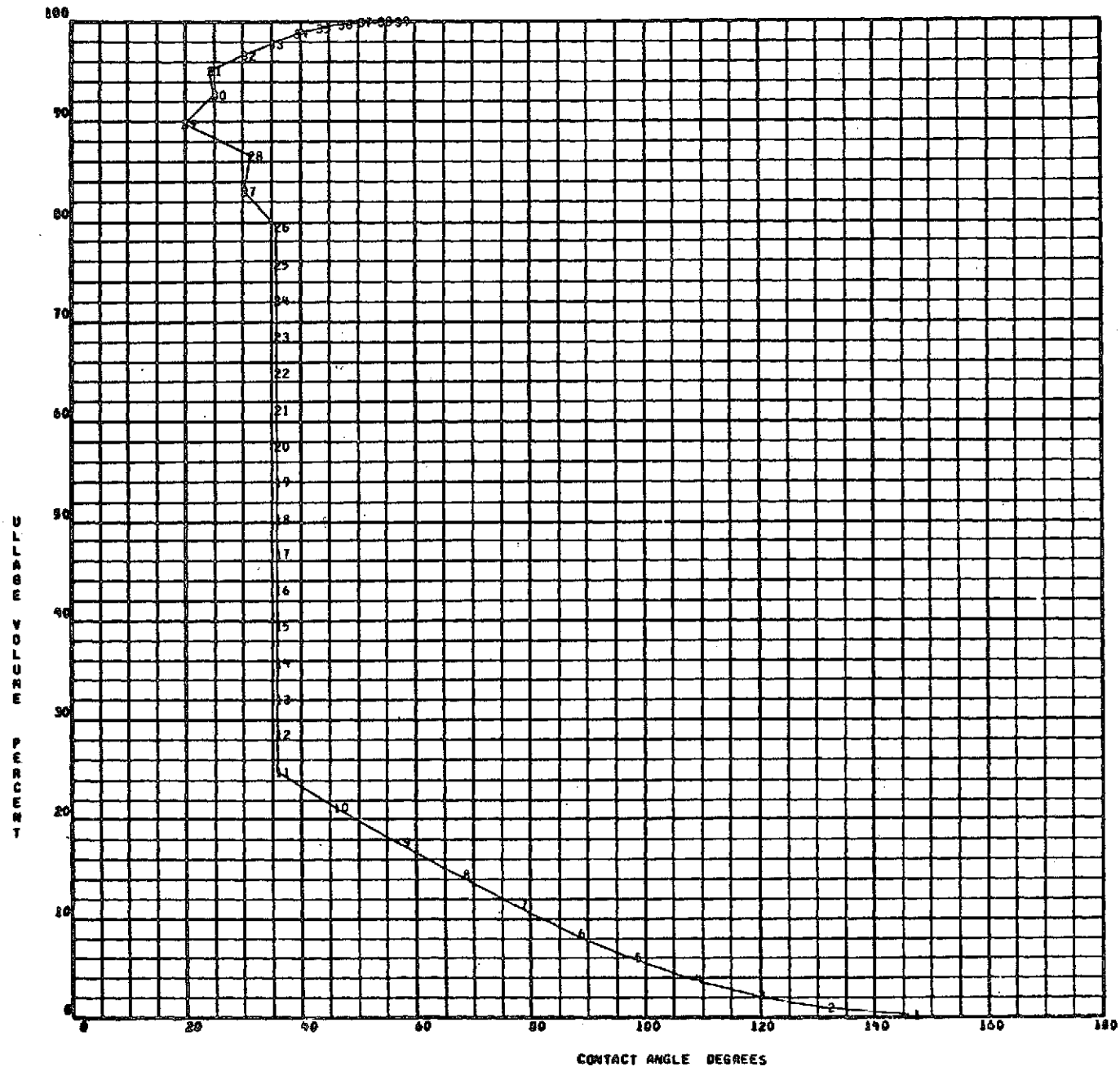


Figure 3-4a. ULLAGE VOLUME VS CONTACT ANGLE,
A = 0.05

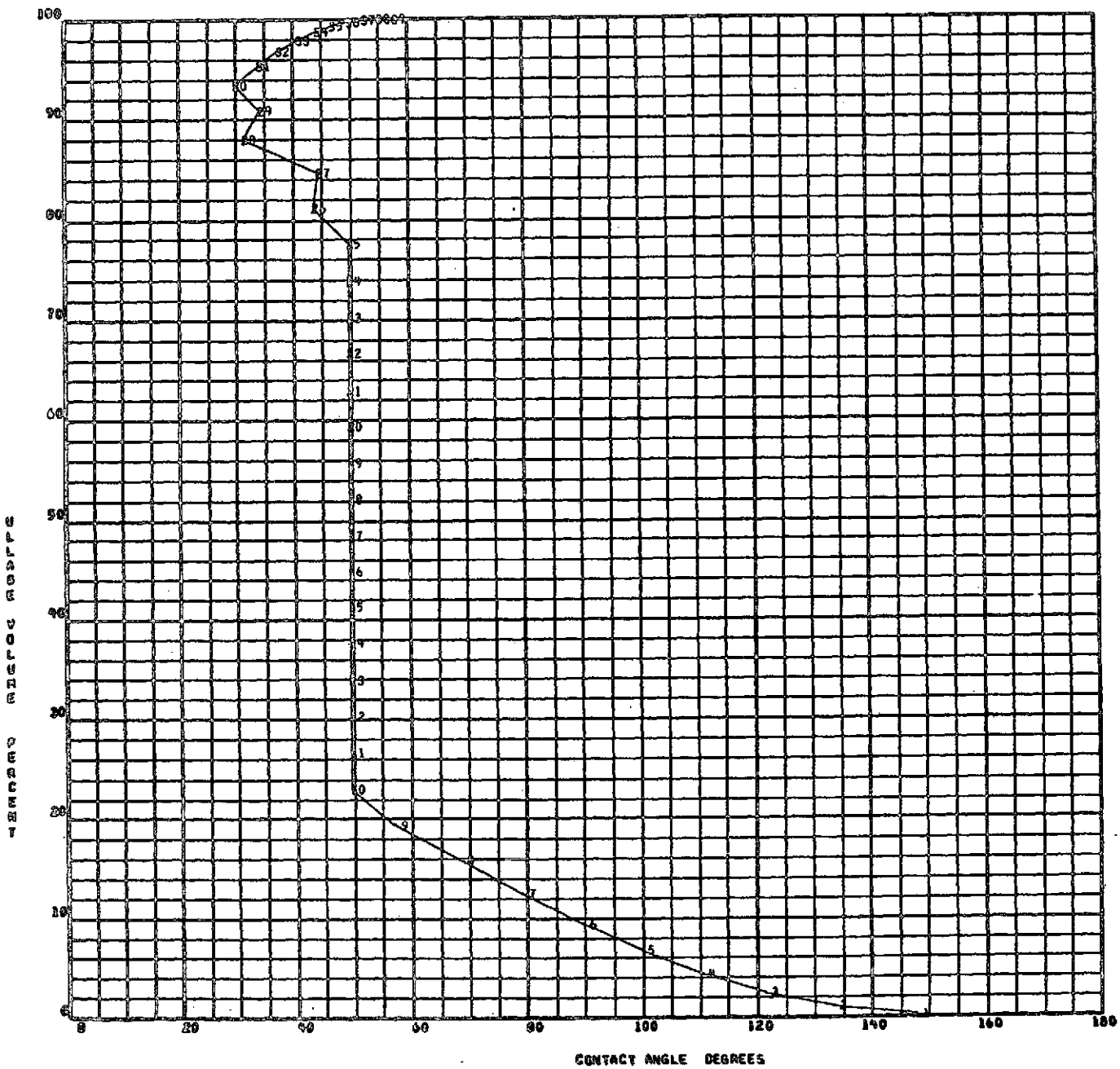


Figure 3-4b. ULLAGE VOLUME VS CONTACT ANGLE,
A = 0.04

PLOT SYMBOLS FOR
ULLAGE VOLUME PERCENT VS CONTACT ANGLE DEGREES

SYMBOL	INTERCEPT
1	.14283750+03
2	.13917500+03
3	.13551250+03
4	.13185000+03
5	.12818750+03
6	.12452500+03
7	.12086250+03
8	.11720000+03
9	.11353750+03
10	.10987500+03
11	.10621250+03
12	.10255000+03
13	.98987500+02
14	.95324999+02
15	.91662500+02
16	.87900000+02
17	.84237493+02
18	.80575000+02
19	.76912493+02
20	.73250000+02
21	.69587500+02
22	.65924999+02
23	.62262500+02
24	.58600000+02
25	.54937500+02
26	.51275000+02
27	.47612500+02
28	.43950000+02
29	.40287500+02
30	.36625000+02
31	.32962500+02
32	.29300000+02
33	.25637500+02
34	.21975000+02
35	.18312500+02
36	.14650000+02
37	.10987500+02
38	.73250000+01
39	.36625000+01

Figure 3-4c. ULLAGE VOLUME VS CONTACT ANGLE, PLOT SYMBOLS

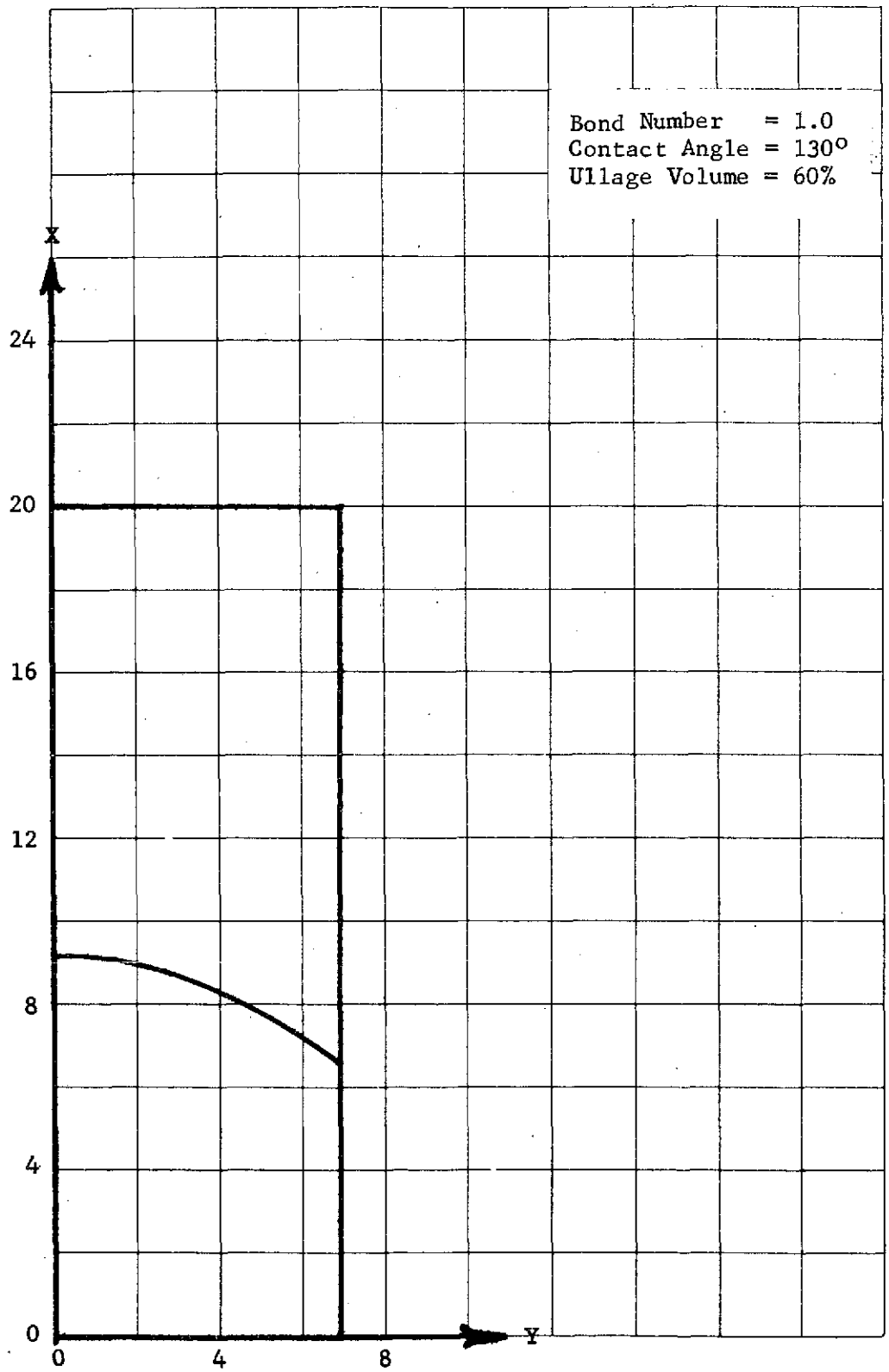


Figure 3-5. FREE SURFACE STATIC EQUILIBRIUM SHAPE, CYLINDRICAL CONTAINER

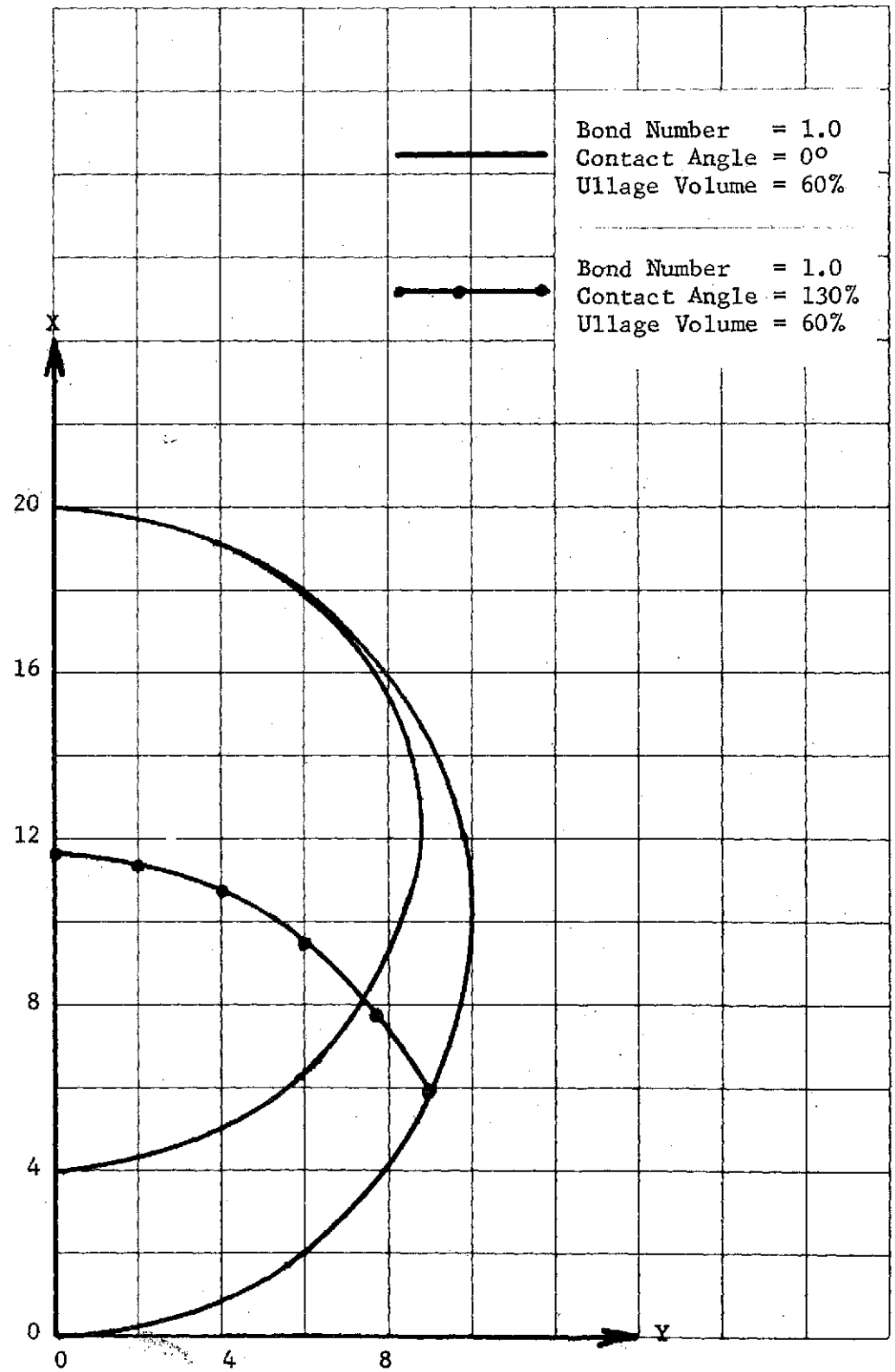


Figure 3-6. FREE SURFACE STATIC EQUILIBRIUM SHAPE, SPHERICAL CONTAINER

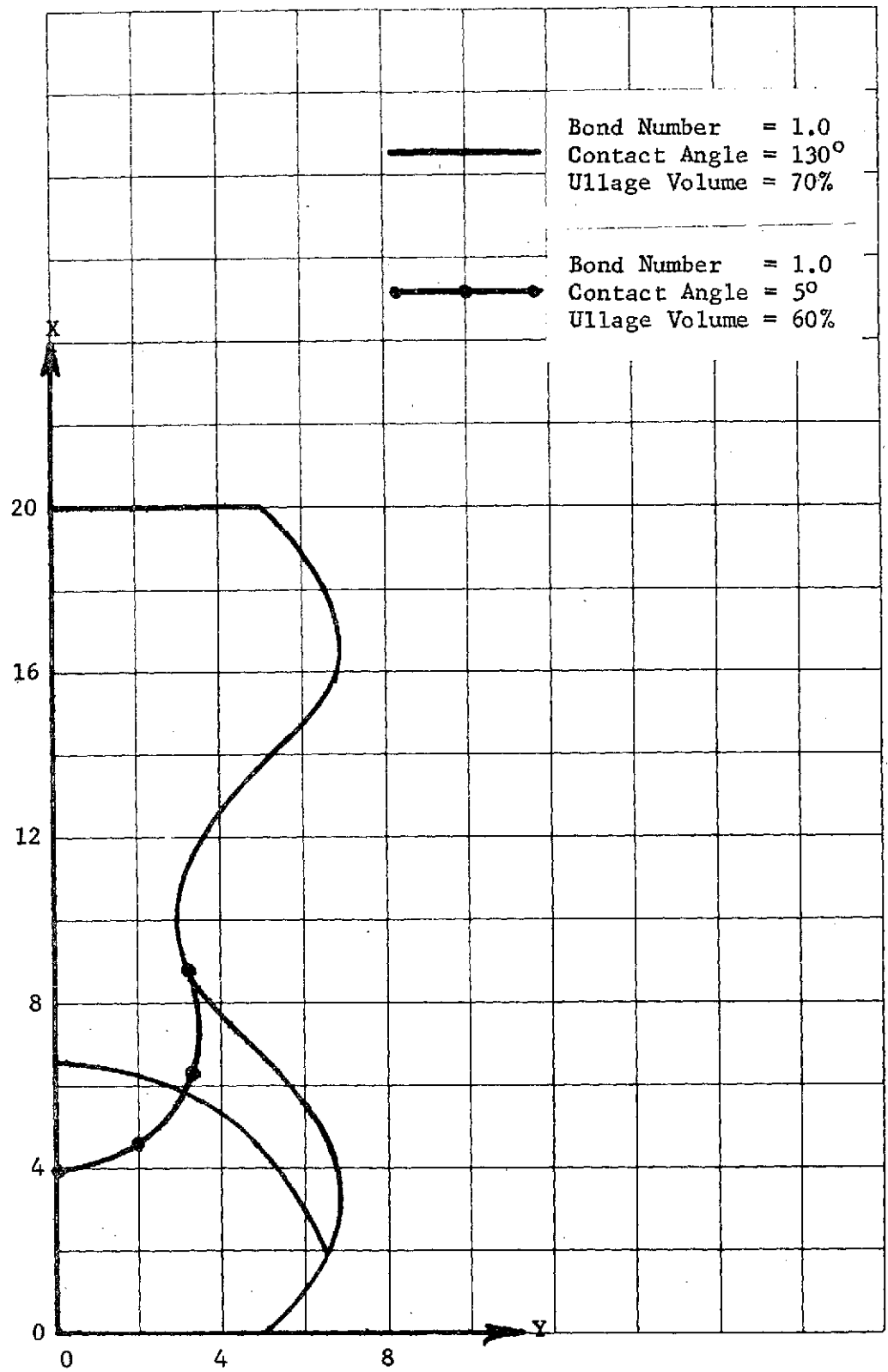


Figure 3-7. FREE SURFACE STATIC EQUILIBRIUM SHAPE, COSINE CONTAINER

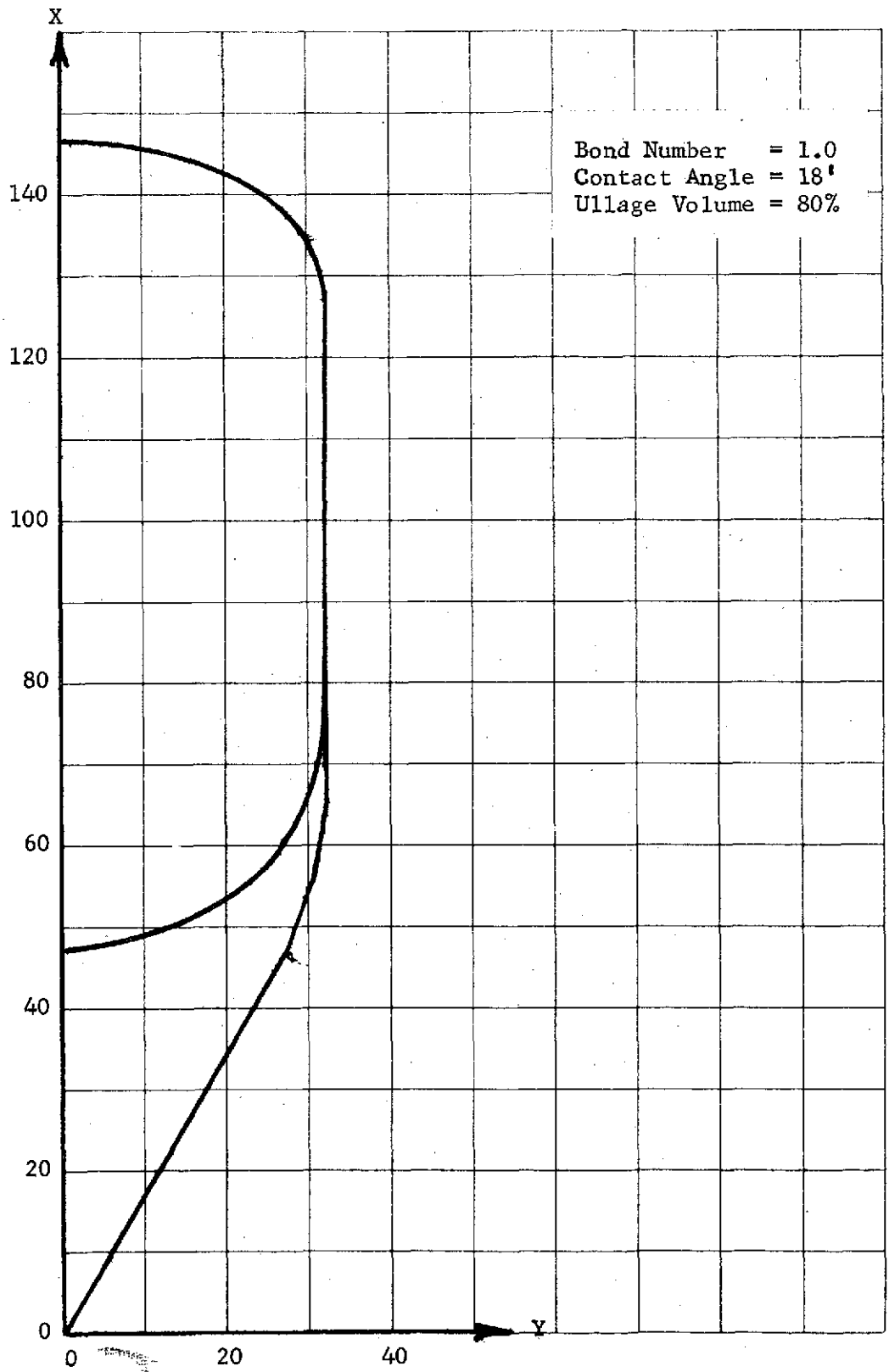


Figure 3-8. FREE SURFACE STATIC EQUILIBRIUM SHAPE, TUG-LIKE CONTAINER

3.2 Vibration Analysis

The Tug-like container of Figure 3-8, with a fluid of nitrogen tetroxide (N_2O_4), was used for the vibration analysis.

From Table A-1 of Reference (5), the density (ρ) and surface tension (σ) of this fluid are obtained as ρ = 1.454 dyne sec^2/cm^4 (1.36×10^{-4} lb-sec^2/in^4) and σ = 27.4 dyne/cm (1.56×10^{-4} lb/in). From this same reference, the contact angle is 0^o-2^o for the liquid, its vapor and titanium. From Martin test work, the bulk modulus is estimated at 90,300 N/cm^2 (131,000 psi).

The acceleration (g) is calculated from the equation for Bond number which is

$$B_o = \frac{\rho g r_{max}^2}{\sigma}$$

Using the tank radius (r_{max}) as 81.28 cm (32 in) and a Bond number of 1 gives g = .00285 cm/sec^2 (.00112 in/sec^2) which is 2.9×10^{-6} of the gravitational acceleration at the Earth's surface.

Note that there is an inconsistency in the characteristic length used here (r_{max}) and that used in the static equilibrium section 2.1.4 (X_{max}). Because X_{max} = 372.11 cm (146.5 in), the acceleration used in the vibration analysis should have been reduced by $(372.11/81.28)^2$ = 20.96. Because the terms of the gravitational stiffness matrix are already a factor of 10^{-2} less than the terms of the surface tension stiffness matrix, further reduction in the gravitational stiffness matrix terms should not effect the vibration analysis results that were obtained.

Four different size models were used in the vibration analysis, A course grid with one sector size and a fine grid with three sector sizes. The grids are shown in Figures 3-9 and 3-10. The algorithms developed in Section 2.4.2 are used here to calculate the total number of degrees of freedom (NDOF), the number of slosh modes (M_s), the number of "crunch" modes (M_c), and the number of circulation modes (M_o) in terms of the grid point geometry. These results are given in Table 3-1. The mode number of the first slosh mode is simply $M_o + 1$.

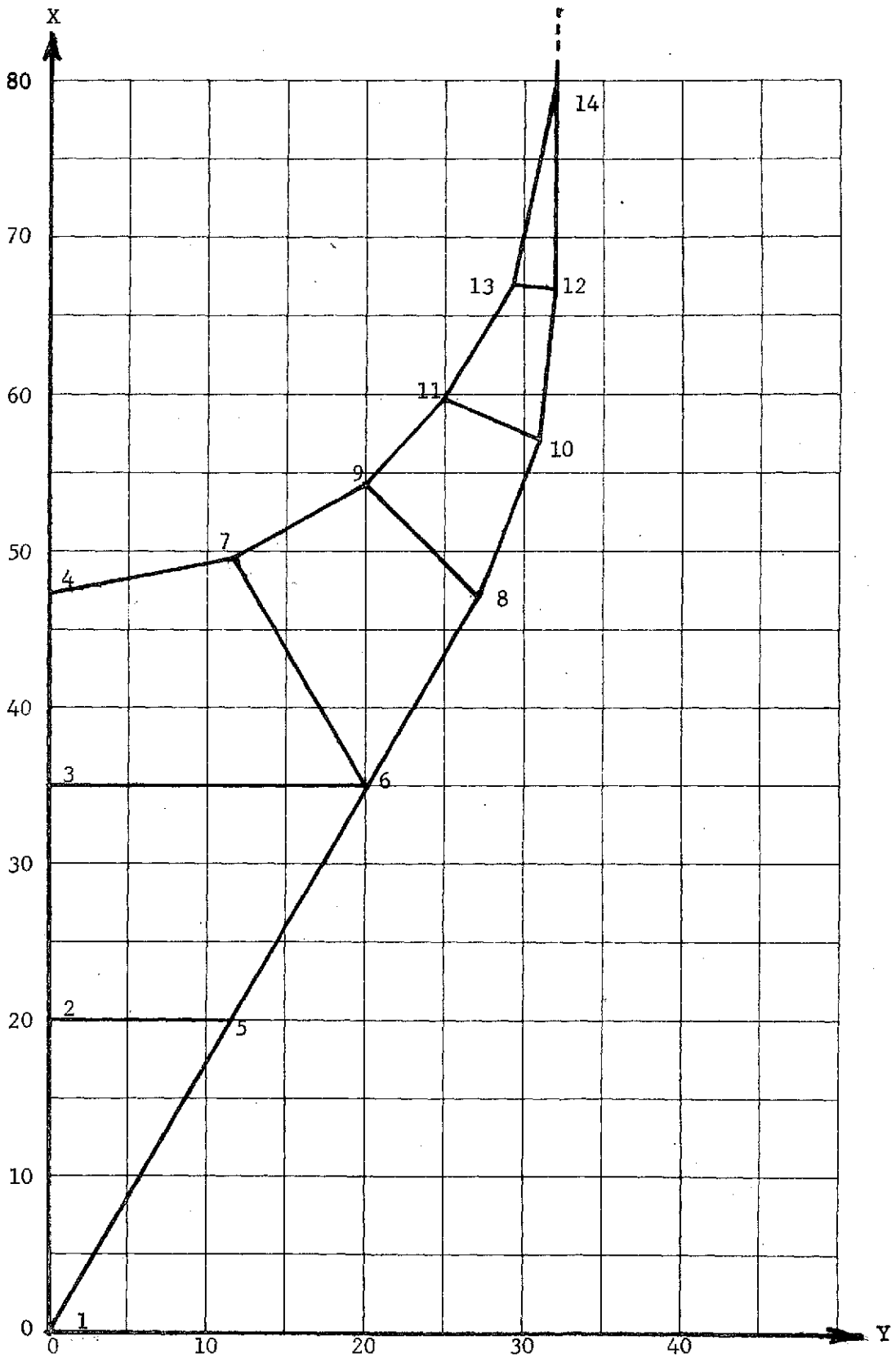


Figure 3-9. TUG-LIKE CONTAINER, GRID 1

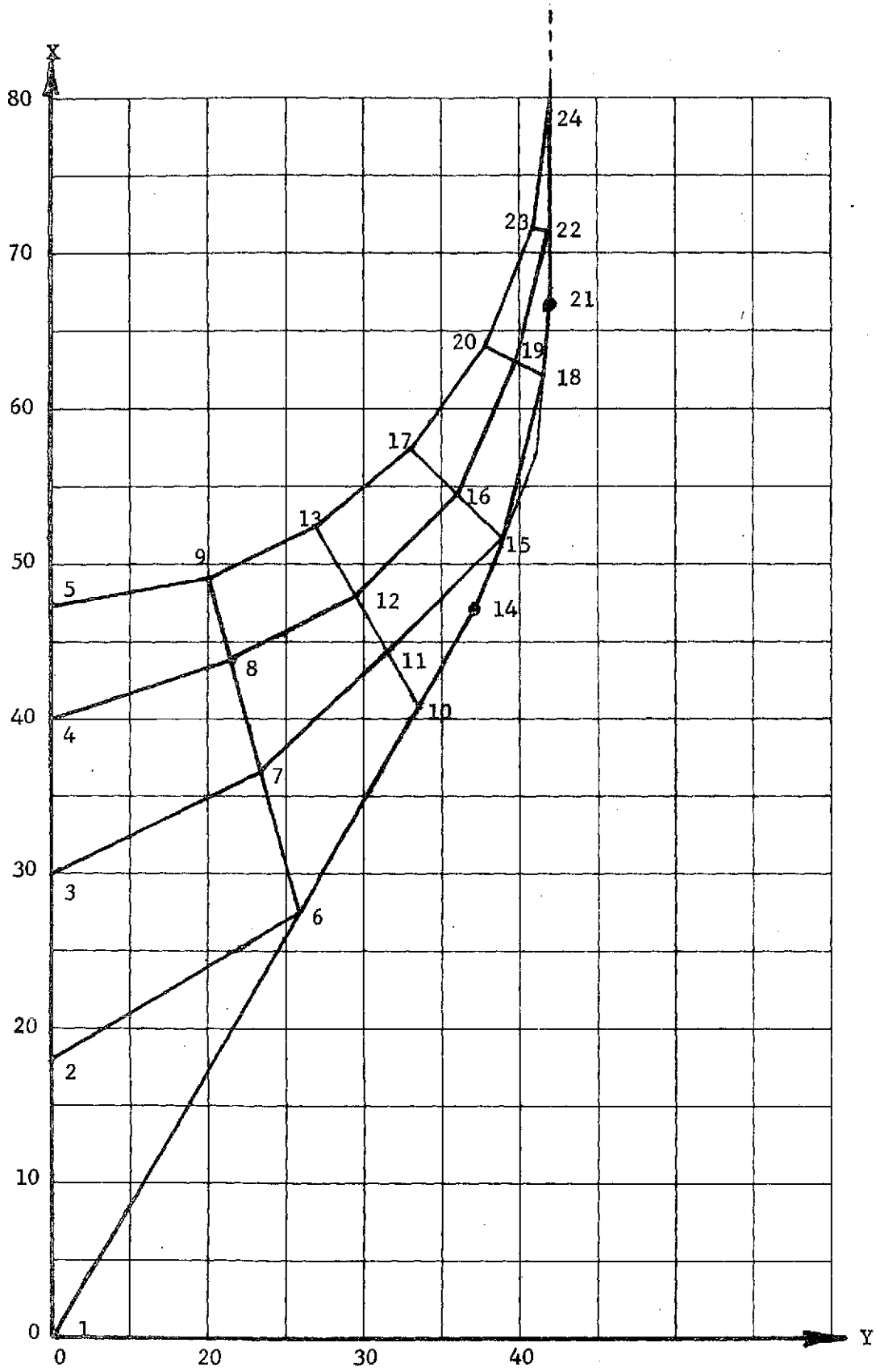


Figure 3-10. TUG-LIKE CONTAINER, GRID 2

Table 3-1

NUMBER OF DOF, SLOSH MODES, CRUNCH MODES, AND CIRCULATION MODES

	Grid 1	Grid 2		
	2 Sectors	2 Sectors	3 Sectors	4 Sectors
NGPAX(1)	4	5	5	5
NGPCW(1)	6	8	8	8
NGPIS(1)	4	11	11	11
NGPFS(1)	4	5	5	5
NGPEL(1)	7	15	15	15
NSECT(1)	2	2	3	4
NDOF	52	103	152	201
M_S	29	35	52	69
M_C	14	30	45	60
M_O	9	38	55	72

(1) Symbols are defined in Section 2.4.2

Using nominal values of bulk modulus, acceleration, fluid density and surface tension, a stiffness matrix composed of

$$\left[K_{\text{fluid}}\ (\text{BKM})\right] + \left[K_{\text{gravity}}\ (g, \rho)\right] + \left[K_{\text{SURFTN}}\ (\sigma)\right]$$

with terms on the order of 10^4, 10^{-7}, 10^{-5} respectively with Grid 1 and 2 sectors in the 90° model. Based on frequency, there was no problem separating the high frequency "crunch" modes. However, it was difficult (based only on frequency) to identify which frequencies were slosh modes and which frequencies were circulation modes (should be zero). Factors of 10^2, 10^4, 10^6 were applied simultaneously to acceleration and surface tension. A summary of results (Table 3-2) shows the ω^2 of the slosh modes to vary directly with the factor used and the ω^2 of the circulation and crunch modes to be uneffected. Scale factors were also applied to bulk modulus. A summary of results (Table 3-3) shows the ω^2 of the slosh modes to be uneffected but the ω^2 of the circulation modes decreased almost directly with the factor used and the ω^2 of the crunch modes decreased directly with the factor used.

Table 3-2

EFFECT OF SIMULTANEOUS ACCELERATION AND SURFACE TENSION VARIATION ON FREQUENCY (ω^2)
(E-7 = 10^{-7}) GRID 1, 2 SECTORS

Mode Type	Mode No.	Nom g,σ	Nomx10^2	Nomx10^4	Nomx10^6
Circulation	9	2.578E-7	1.931E-7	6.335E-7	3.638E-7
Slosh	10	1.403E-6	1.289E-4	1.289E-2	1.289E-0
	11	2.499E-6	2.389E-4	2.390E-2	2.390E-0
	.				
	.				
	.				
	37	9.765E-2	9.765E-0	9.765E+2	9.763E+4
	38	1.194E-1	1.194E+1	1.194E+3	1.193E+5
Crunch	39	1.453E+7	1.453E+7	1.453E+7	1.454E+7

Table 3-3

EFFECT OF BULK MODULUS VARIATION ON FREQUENCY (ω^2)

(E-7 = 10^{-7}) GRID 1, 2 SECTORS

Mode Type	Mode No.	Nom BKM	Nomx10^{-2}	Nomx10^{-4}	Nomx10^{-6}	Nomx10^{-8}	Nomx10^{-10}
Circulation	9	2.578E-7	3.299E-9	1.118E-10	2.633E-13	2.590E-15	5.307E-17
Slosh	10	1.403E-6	1.293E-6	1.289E-6	1.289E-6	1.289E-6	1.285E-6
	11	2.499E-6	2.392E-6	2.390E-6	2.390E-6	2.390E-6	2.370E-6
	.						
	.						
	.						
	37	9.765E-2	9.765E-2	9.765E-2	9.763E-2	9.565E-2	1.023E-2
	38	1.194E-1	1.194E-1	1.194E-1	1.193E-1	1.148E-1	1.297E-2
Crunch	39	1.453E+7	1.453E+5	1.453E+3	1.454E+1	1.565E-1	1.312E-2

Scale factors were also applied to bulk modulus for Grid 2. A summary of results (Table 3-4) shows the same variation of ω^2 with scale factor as was obtained with Grid 1.

Table 3-4

EFFECT OF BULK MODULUS VARIATION ON FREQUENCY (ω^2)

(E-7 = 10^{-7}) GRID 2, 2 SECTORS

Mode Type	Mode No.	Nom BKM	Nomx10^{-4}	Nomx10^{-8}	Nomx10^{-10}
Circulation	38	2.980E-6	3.603E-10	3.403E-14	4.034E-16
Slosh	39	5.545E-6	2.072E-6	2.072E-6	2.060E-6
	40	6.055E-6	4.829E-6	4.829E-6	4.816E-6
	.				
	.				
	.				
	72	2.840E-1	2.840E-1	2.185E-1	1.329E-1
	73	4.309E-1	4.309E-1	2.324E-1	1.356E-1
Crunch	74	1.082E+7	1.082E+3	2.794E-1	1.545E-2

In addition to frequency, the effect on modal displacement with variation of bulk modulus was also noted for Grid 2 with the results summarized in Table 3-5.

Table 3-5

EFFECT OF BULK MODULUS VARIATION ON MODAL DISPLACEMENT. GRID 2, 2 SECTORS MODE 39 (FIRST SLOSH MODE)

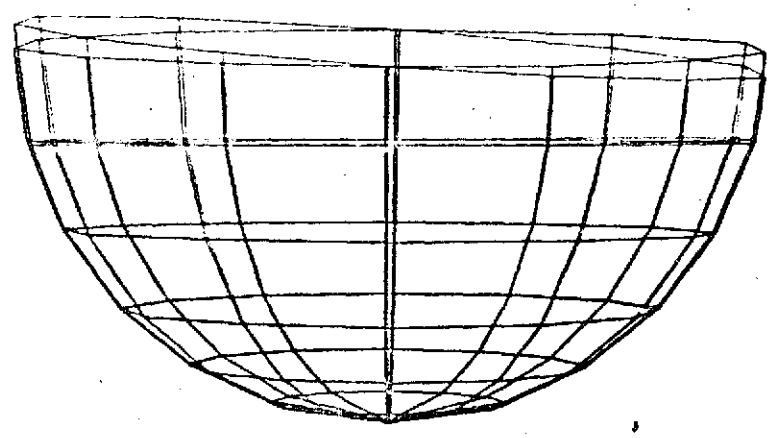
Grid Point Number	Nom BKM	Nomx10^{-4}	Nomx10^{-8}	Nomx10^{-10}
20 (δX)	-1.193	-.2069	-.2069	-.2075
24 (δX)	5.803	-2.814	-2.814	-2.788

Based on the results of the above four tables, it was decided to use a scale factor of 10^{-8} on bulk modulus to assure valid frequencies and modal displacements for the slosh modes. With this scale factor, Table 3-6 shows the variation in the first three slosh modes with the various grids used. The plotted mode shapes for the first 3 slosh modes are given in Figures 3-11 through 3-13 for Grid 2, 3 sectors. The undeformed and deformed surface joints are shown in a perspective view. Inclusion of the internal and wall joints was tried but the large amount of plotted data made the viewing too difficult and was thus abandoned in favor of just showing the surface. Plots of only the joints in the XY plane was also tried but did not give as satisfactory a plot as the surface perspective plots. Capability exists in the computer program for any of these plots at the user option, however.

Table 3-6

SLOSH FREQUENCY (HZ)
NOM BKM*10^{-8}

Slosh Mode	Grid 1	Grid 2		
	2 Sectors	2 Sectors	3 Sectors	4 Sectors
1	.0001807	.0002291	.0001841	.0001600
2	.0002460	.0003497	.0002404	.0001605
3	.0004080	.0006205	.0003224	.0002299

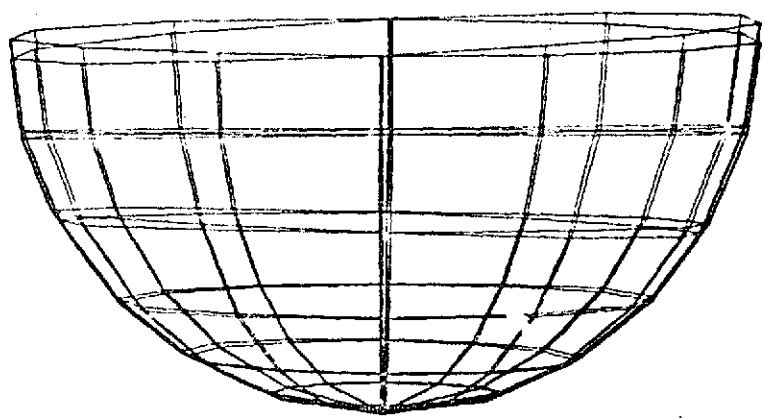


MODE 4, F = .000184 HZ. IUS OX TANK. BN=1. ULL VOL = 80. GRID 2, 3 SECTORS

CENTER OF EYES LOCATION	VIEW POINT LOCATION	ROLL ANGLE = 180.0DEG
X = 7.00000000E+01	X = 5.00000000E+01	CONE ANGLE = 30.0DEG
Y = 0.	Y = 0.	EYE TO EYE = 5.0 IN
Z =-2.00000000E+02	Z = 0.	
RUN NO. = G2/3-S	DATE = 24MR75	

Figure 3-11. FIRST SLOSH MODE, SURFACE PERSPECTIVE VIEW

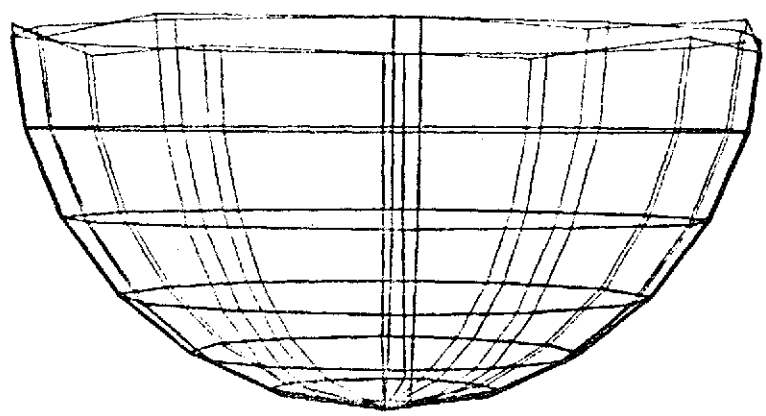
LEFT EYE VIEW



MODE 5, F = .000240 HZ. IUS OX TANK. BN=1. ULL VOL = 80. GRID 2, 3 SECTORS

CENTER OF EYES LOCATION	VIEW POINT LOCATION	ROLL ANGLE = 180.0DEG
X = 7.00000000E+01	X = 5.00000000E+01	CONE ANGLE = 30.0DEG
Y = 0.	Y = 0.	EYE TO EYE = 5.0 IN
Z =-2.00000000E+02	Z = 0.	
RUN NO. = G2/3-S	DATE = 24MR75	

Figure 3-12. SECOND SLOSH MODE, SURFACE PERSPECTIVE VIEW



LEFT EYE VIEW

MODE 6, F = .000322 HZ. IUS OX TANK, BN=1. ULL VOL = 80. GRID 2, 3 SECTORS

CENTER OF EYES LOCATION	VIEW POINT LOCATION	ROLL ANGLE = 180.0DEG
X = 7.00000000E+01	X = 5.00000000E+01	CONE ANGLE = 30.0DEG
Y = 0.	Y = 0.	EYE TO EYE = 5.0 IN
Z =-2.00000000E+02	Z = 0.	
RUN NO. = G2/3-S	DATE = 24MR75	

Figure 3-13. THIRD SLOSH MODE, SURFACE PERSPECTIVE VIEW

4. CONCLUSIONS

It is felt that a significant contribution was made by this study to the state of the art in finite element fluid analysis at low Bond number. In this study methods and computer programs for definition of the free surface static equilibrium shape at low Bond number, calculation of the stiffness matrix due to surface tension, generation of joint coordinate locations, degree of freedom values, Euler angles, and element joint numbers, and calculation of the vibration mode shapes/frequencies and plotting of these mode shapes were derived and coded. As with probably all new investigative analytical studies, review of the work performed reveals that a "blind-alley" was investigated and that there are several items that should be studied further.

To determine the free surface static equilibrium shape, an energy minimization technique was originally attempted. In this method, the displacement state was sought for static equilibrium corresponding to the minimum potential caused by gravitational potential energy, surface tension potential energy and the virtual work done by ullage pressure acting through virtual displacements, all subject to the constraints of constant volume and contact angle at the container boundaries. This is the method that was outlined in the proposal for this study, Reference (7). Unfortunately, no results were obtained by this original method because the attempts at solution continually diverged. Thus, this approach had to be abandoned in favor of the force balance method described in Section 2.1.

One item that should be investigated further is the separation of the slosh modes from the circulation modes and "crunch" modes. One possible approach, as described in Reference (8), is to describe all the joint coordinates in terms of the surface and ignorable coordinates by means of constraint equations. With this relationship, the original mass matrix is reduced to only the surface and ignorable coordinates. This reduced system is further reduced to only the surface coordinates by expressing the surface and ignorable coordinates in terms of only the surface coordinates. By this technique, only the modal properties of the surface slosh coordinates are calculated.

The data generator subroutine, used to calculate joint X, Y, Z locations, degree of freedom values, Euler angles, and finite element joint numbers was coded for the lateral slosh boundary conditions, that is, symmetric/anti-symmetric boundaries as defined in Figure 2-10. Expansion of the data generator to include the symmetric/symmetric and anti-symmetric/anti-symmetric boundary conditions should be done.

Investigation into the non axi-symmetric acceleration field should be performed in the definition of the free surface static equilibrium shape. This will allow complete generality for the low Bond number

problem. A non axi-symmetric acceleration field will require a new data generator because a 360^{o} model definition would be required. The gravity stiffness matrix currently allows a non axi-symmetric acceleration.

5. REFERENCES

1. J. R. Admire: "Modal Analysis of Structures by an Iterative Rayleigh-Ritz Technique". NASA TMX-64528, Marshall Space Flight Center, 1970.

2. O. C. Zienkiewicz and Y. K. Cheung: "The Finite Element Method in Structural and Continuum Mechanics". McGraw-Hill, New York, 1967.

3. R. L. Wohlen: "Synthesis of Dynamic Systems Using FORMA - Fortran Matrix Analysis". Contract NAS8-25922. MCR-71-75. Martin Marietta Corporation, Denver, Colorado, May 1971.

4. R. L. Wohlen, W. A. Benfield, R. A. Philippus: "Evaluation of Modal Analysis Techniques". Contract NAS8-26750. MCR-73-310, Martin Marietta Corporation, Denver, Colorado, December 1973.

5. H. Norman Abramson: "The Dynamic Behavior of Liquids in Moving Containers", NASA SP-106, 1966.

6. Univac 1100 Series FORTRAN V Programmer Reference Manual, UP-4060, Revision 2, Sperry Rand Corporation, 1972.

7. Technical Proposal, "Finite Element Solution of Low Bond Number Sloshing," P74-48317, Martin Marietta Corporation, January 1974.

8. D. A. Hunt, "Discrete Element Idealization of an Incompressible Liquid for Vibration Analysis". AIAA Journal, Volume 8, No. 6, June 1970.

6. COMPUTER PROGRAMS

The finite element solution for liquid sloshing at low Bond number is accomplished in two main steps. The first step is the free surface static equilibrium shape definition and the second step is the vibration analysis. Computer programs have been coded for these steps and are listed in Section 6.1. Input data to the programs are explained in Section 6.2 using a sample problem listing.

A schematic flow chart of the analysis steps is given in Figure 6-1, and a brief summary of the important subroutine functions are presented in the following pages.

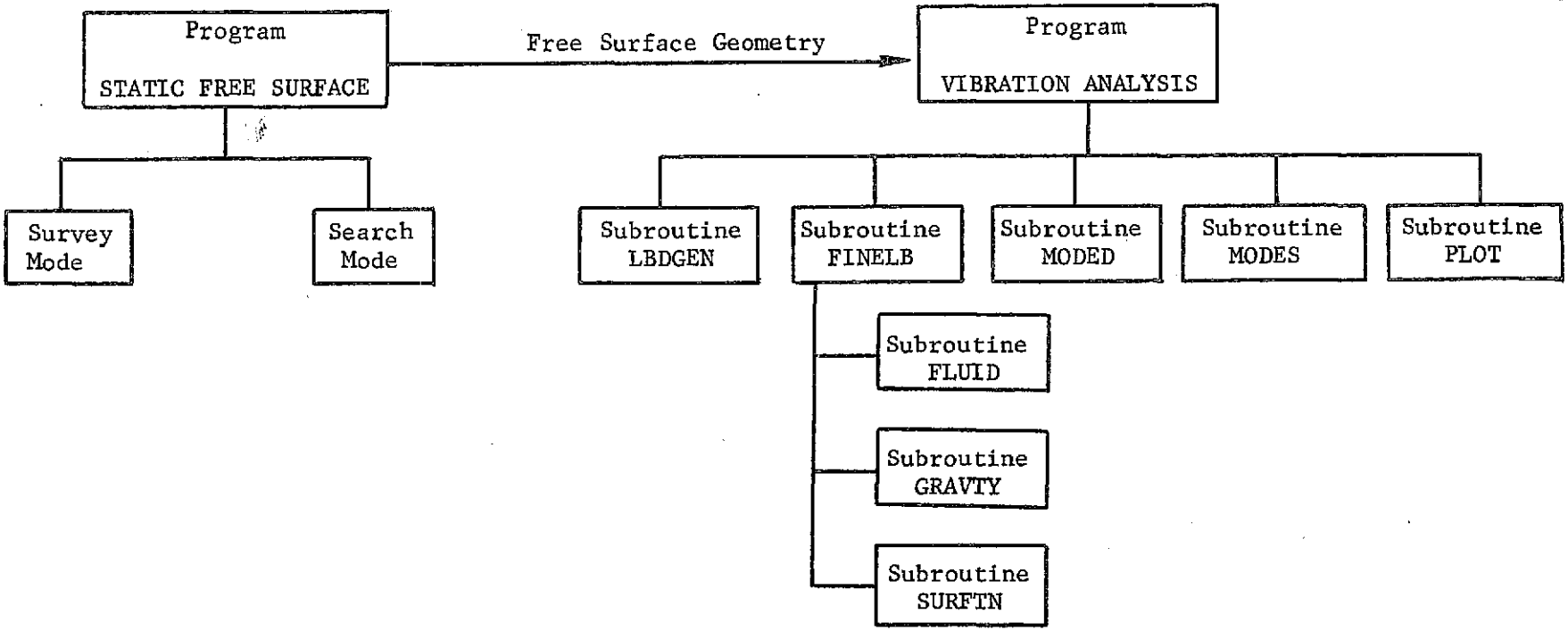


Figure 6-1. FLOW CHART, LOW BOND SLOSH PROGRAMS

Summary of Programs:

STATIC FREE SURFACE	obtains the free surface static equilibrium shape
VIBRATION ANALYSIS	obtains the vibration characteristics of the system (frequencies and mode shapes)

Summary of Subroutines (used in the VIBRATION ANALYSIS Program):

LBDGEN	automatic generation of joint X, Y, Z values, DOF numbers, Euler angles, and element joint numbers for FINELB
FINELB	generates mass and stiffness matrices
FLUID	generates mass and stiffness for fluid only
GRAVTY	generates gravity contribution to stiffness matrix
SURFTN	generates surface tension contribution to stiffness matrix
MODED	obtains frequencies and mode shapes (small size)
MODES	obtains frequencies and mode shapes (large size)
PLOT	plots the mode shapes

6.1 Computer Program Listings

Listings of the free surface static equilibrium shape computer program and associated computer subroutines is given in Section 6.1.1. Listing of the Vibration Analysis Computer program and associated computer subroutines is given in Section 6.1.2.

```
FMBUGGRIN207*TPF$.LOBOND
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
      C       PROGRAM TO DETERMINE THE STATIC EQUILIBRIUM SHAPE OF THE
      C       FREE SURFACE OF A FLUID WITH A LOW BOND NUMBER IN A
      C       GIVEN AXISYMMETRIC CONTAINER
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
              LOGICAL           GO BACK,DUMMY,ACCEPT,PRINT,SEARCH
              REAL              SURVPL,PHITO,PHICO,PHIFO,VUPCTS,DASH
              EQUIVALENCE       (SOLN(500)),SURVPL(1))
              DIMENSION         SURVPL(51,51,2)
              COMMON            SOLN(1000,5,2)
              COMMON            /CONSTS/ PI,ANUM
              COMMON            /PARAMS/ ACOF,BCOF,ZMAX,RMAX
              COMMON            /QPRKTA/ QRK(5),PRK(4)
              COMMON            /TIMESS/ DELTAT,T
              COMMON            /VECTOR/ Y(5),YDT(5)
              NAMELIST          /INDATA/ ACOFO,BONDNO,DACOF,DELTAT,EPSC,NA,
             1                           IPRNT,NX,PHIO,PRINT,RMAX,SEARCH,TVOL,ULPCT,
             2                           XLO,XMAX,XUP
      C
              DATA              NIT/5/,NOT/5/,NEQ/5/,IPRNT/10/
              DATA              DELTAT /0.1D000/, EPSC /1.0D-04/, EPST /1.0D+08/
              DATA              DASH/1H-/
              DATA              PRINT/.TRUE./
              DATA              ACOFO/0.0D00/,DACOF/0.1D00/,BONDNO/1.0D00/,NX/10/
              DATA              PHIO/90.0D00/,ULPCT/50.0D00/,XLO/-1.0D00/
              DATA              RMAX/-1.0D00/
              DATA              XMAX/-1.0D00/,XUP/-1.0D00/
              DATA              SEARCH/.TRUE./,NA/10/
      C
      C
      C       INITIALIZATION AND INPUT
      C
      C
      C      INITIALIZE CONSTANTS
              ANUM = DBLE(ATAN2(1.0,1.0)) / 45.0)
              PI = 180.0D00 * ANUM
              PRK(1) = 0.5D00
              PRK(2) = 1.0D00 - DSQRT(0.5D00)
              PRK(3) = 1.0D00 + DSQRT(0.5D00)
              PRK(4) = 0.5D00
      C      READ INPUT DATA
       1000 CONTINUE
              CALL START
              READ(NIT,INDATA)
              NZ = NX
              ZUP = XUP
              ZLO = XLO
              ZMAX = XMAX
      C      INITIALIZE ITERATION PARAMETERS
              IF((NA.GT.50 .OR. NZ.GT.50) .AND. .NOT. SEARCH) GO TO 9020
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      NUMA = 2
      IF(.NOT. SEARCH) NUMA = NA
      IF(RMAX.LT.0.0000 .OR. ZMAX.LT.0.0000) GO TO 9010
      IF(ZUP .LT. 0.0000) ZUP = ZMAX
      IF(ZLO .LT. 0.0000) ZLO = 0.0000
      DACOFO = DACOF
      ACOF = HONONO / (ZMAX*ZMAX)
      YVOL = VOLL(0.0000,0.0000,ZMAX)
      IF(MOD(NZ,2) .NE. 0) NZ = NZ + 1
      DZI = (ZUP-ZLO)/FLOAT(NZ)
      Z = (ZUP+ZLO)/2.0000
      PSIT = PSI
      ACCEPT = .FALSE.
      SGN = 1.0000
C     WRITE DATA
      CALL PAGEHD
      WRITE(NOT,INDATA)
C
C
C     MAIN LOOP - ITERATE WITH NEW ACOFO AND DACOF
C
C
 2000   CONTINUE
        CALL PAGEHD
        NLOOP = NZ - 1
C
C
C      A LOOP - GET NEW TRAJECTORY
C
C
          DO 3000 KA = 1,NUMA
            IA = KA
            IF(.NOT. SEARCH) IA = 1
            IF(IA .EQ. 2) SGN = -SGN
            ACOF = ACOFO + SGN*DACOF
            IF(.NOT. SEARCH) SURVPL(1,KA+1,1) = ACOF
            IF(PRINT) WRITE(NOT,2009) (DASH, I=1,50), ACOF, (DASH, I=1,50)
            Y(1) = 0.0000
            Y(2) = 0.0000
            Y(3) = 1.0000
            Y(4) = 0.0000
            Y(5) = 0.0000
            NT = 0
            T = 0.0000
            DO 3100 II = 1,NEQ
            ORK(II) = 0.0000
 3100       CONTINUE
            CALL SAVE(NT+1,NEQ,IA)
            CALL YDOT
            IF(PRINT) CALL PRINTS(NT+1,NEQ,IA)
C
C           INTEGRATION LOOP, INTEGRATE D.E.Q. FROM 0
C           UNTIL GOBACK = .FALSE.
C
 3200        CALL RUNKTA(NEQ,NT)
             CALL SAVE(NT+1,NEQ,IA)
```

```
112                     IF(PRINT .AND. MOD(NT,IPRNT).EQ.0)
113          1                        CALL PRINTS(NT+1,NEQ,IA)
114                    IF(GO BACK(DUMMY)) GO TO 3200
115          C
116          C              END OF INTEGRATION LOOP
117          C
118                     IF(PRINT) CALL PRINTS(NT+1,NEQ,IA)
119          C
120          C
121          C               LOOP ON ZI AT FIXED ACOF TO FIND SMALLER ERROR
122          C               IF ONE EXISTS
123          C               LOOP ON ZI AT FIXED ACOF TO FIND SMALLER ERROR
124          C               IF ONE EXISTS
125          C
126          C
127                      ZI = Z + DBLE(FLOAT(NZ/2))*DZI
128                      IF(PRINT) WRITE(NOT,2002)
129                      DO 4000 IZ = 1,NLOOP
130                        ZI = ZI - DZI
131                        IF(.NOT. SEARCH .AND. KA.EQ.1) SURVPL(IZ+1,KA,1) = ZI
132                        IF(ZI.GT.ZUP .OR. ZI.LT.ZLO) GO TO 4300
133                        J = 0
134           4200         CONTINUE
135                          J = J + 1
136                          IF(J .GT. NT+1) GO TO 4300
137                        IF(RCAN(SOLN(J,2,IA))+ZI) .GT. SOLN(J,1,IA)) GO TO 4200
138          C              HAVE BRACKETED CAN. DETERMINE PRELIMINARY VALUES OF
139          C              ULLAGE VOLUME, CONTACT ANGLE, AND ERROR
140                        VUPCT=VULL(SOLN(J,5,IA),SOLN(J,2,IA),ZI,ZMAX)*100./TVOL
141                        PHIF=DBLE(ATAN2(SNGL(SOLN(J,3,IA)),SNGL(SOLN(J,4,IA))))
142                        PHIC  = DATAN(RCANP(SOLN(J,2,IA))+ZI))
143                        PHIT  = PHIF - PHIC
144                        IF(PHIF .LT. 0.0000 .AND. PHIC .GT. 0.0000)
145                 *         PHIT = PHIT + 2.0*PI
146                        PHITD = PHIT/ANUM
147                        PHICD = PHIC/ANUM
148                        PHIFD = PHIF/ANUM
149                        PSIN = ((VUPCT-ULPCT)/100.0000)**2 +
150                 1                  ((PHITD-PHID)/180.0000)**2
151                        VUPCTS = SNGL(VUPCT)
152                        IF(PRINT) WRITE(NOT,2007) ZI,PSIN,VUPCTS,PHITD,
153                 1                                PHIFD,PHICD
154                        IF(SEARCH) GO TO 4250
155                        SURVPL(IZ+1,KA+1,1) = VUPCT
156                        SURVPL(IZ+1,KA+1,2) = PHITD
157                        GO TO 4000
158           4250         CONTINUE
159                        IF(PSIN .GT. PSIT) GO TO 4000
160          C             SAVE VALUES INDICATING CURRENT MINIMUM ERROR
161                        ZNEW = ZI
162                        PSIT = PSIN
163                        JSAVE = J
164                        ISAVE = IA
165                        ASAVE = ACOF
166                        SSGN  = SGN
167                        GO TO 4000
```

```
      4300            CONTINUE
C                     NO SOLUTION FOR THIS (ACOF,ZI)
                      IF(PRINT) WRITE(NOT,2001) ZI
                      IF(SEARCH) GO TO 4000
                      SURVPL(IZ+1,KA+1,1) = -1000.0
                      SURVPL(IZ+1,KA+1,2) = -1000.0
      4000           CONTINUE
C
C
C               END OF Z LOOP
C
C
                  IF(SEARCH) GO TO 3300
                  ACOFD = ACOF
                  IF(PRINT) CALL PAGEHD
                  GO TO 3000
      3300        CONTINUE
                  IF(PSII.LT.PSI .AND. ACCEPT) GO TO 2200
                  IF(IA .EQ. 2) WRITE(NOT,2011) ACOFD,DACOF
                  IF(IA.EQ.1 .AND. ACCEPT) GO TO 2100
                  ACCEPT = .TRUE.
      3000       CONTINUE
                 IF(.NOT. SEARCH) GO TO 1500
C
C
C          END OF A LOOP
C
C
      2100    CONTINUE
              IF(DABS(DACOF) .LT. DACOFD*1.D-05) GO TO 9000
              DACOF = DACOF/2.0000
              DZI   = DZI/2.0000
              Z = ZNEW
              SGN = -SGN
              ACCEPT = .FALSE.
              GO TO 2000
C       GET MORE ACCURATE VALUES OF ERROR,ULLAGE
C       VOLUME, AND CONTACT ANGLE AND TEST FOR
C       CONVERGENCE
      2200    CONTINUE
              CALL STATE(ISAVE,JSAVE,ZNEW,RS,ZS,RPS,ZPS,VS)
              VU = VULL(VS,ZS,ZMAX)*100.0000/TVOL
              PHIF = DBLE(ATAN2(SNGL(RPS),SNGL(ZPS)))
              PHIC = DATAN(RCANP(ZS))
              PHIT = PHIF - PHIC
              PHITD = PHIT / ANUM
              IF(PHIF.LT.0.0000 .AND. PHIC.GT.0.0000) PHIT = PHIT + 2.0000*PI
              PSI = ((VU-ULPCT)/100.0000)**2 +
     1              ((PHITD-PHID)/180.0000)**2
              PSII = PSI
              ACOFD = ASAVE
              SGN = SSGN
              WRITE(NOT,2005) ASAVE,PSI,ZNEW,VU,PHITD
              IF(PSI .GT. EPSC) GO TO 2000
C
C
```

```
C       END OF MAIN LOOP
C
C
C
C     CONVERGENCE ESTABLISHED, TABULATE (R,Z) COORDINATES
C     OF THE FREE SURFACE SHAPE AND GO READ DATA FOR
C     NEXT CASE
C
C
      CALL PAGEHD
      WRITE(NOT,2006) (DASH,I=1,50),BONDNO,PHID,ULPCT,(DASH,I=1,50)
      T=0.0000
      DO 1100 I = 1,JSAVE
      SOLN(I,2,ISAVE) = ZNEW + SOLN(I,2,ISAVE)
      CALL PRINTS(I,NEQ,ISAVE)
      T = DELTAT+DBLE(FLOAT(II))
 1100 CONTINUE
      GO TO 1000
C     SURVEY COMPLETED, TABULATE RESULTS
 1500 CONTINUE
      I1 = 2
 1510 CONTINUE
      I2 = I1+4
      IF(I2 .GT. NA) I2 = NA
      CALL PAGEHD
      WRITE(NOT,2004) (DASH,I=1,100),(SURVPL(1,I),I=I1,I2)
      WRITE(NOT,2003)
      DO 1520 I = 2,NZ
      WRITE(NOT,2008)SURVPL(I,1),(SURVPL(I,J,1),SURVPL(I,J,2),J=I1,I2)
 1520 CONTINUE
      I1 = I2+1
      IF(I1 .GT. NA) GO TO 1540
      GO TO 1510
 1540 CONTINUE
C  PLOT RESULTS
      CALL LOHOPL (SURVPL,NA-1,NZ-1,SNGL(ZMAX),5I)
      GO TO 1000
C
C     ERROR EXITS
C
 9000 CONTINUE
      WRITE(NOT,2010)
      STOP
 9010 CONTINUE
      WRITE(NOT,2020)
      STOP
 9020 CONTINUE
      WRITE(NOT,2030)
      STOP
C
C     FORMAT STATEMENTS
C
 2002 FORMAT(1H1///40X,20HRESULTS FOR X SWEEP                    ///
     1       6X,11HX-INTERCEPT      ,12X,5HERROR                 ,
     2       7X,10HULLAGE PCT       ,9X,13HCONTACT ANGLE         ,
     3       6X,11HFLUID ANGLE      ,9X,9HCAN ANGLE              ///
```

```
     &        )
2001  FORMAT(5X,D12.6,31X,23H***** NO SOLUTION *****    )
2002  FORMAT(///,40X,5HA)                                 ,/
     1         58X,15HSURVEY SUMMARY                      ,/
     2        40X,5HA)                                    ,///
     3         17X,1HX,10X,5(4HA = ,F6.3,9X)               )
2003 FORMAT(15X,5HCOORD,5(5X,6H VUPCT,2X,6H PHITD)        ,/)
2005  FORMAT(///2X,33HMINIMUM ERROR SOLUTION FOR ACOF =,D12.6,3H IS/
     1         22H     TRUE ERROR          =          ,D12.6,/
     1         22H     TRUE X-INTERCEPT    =          ,D12.6,/
     2         22H     TRUE ULLAGE PCT     =          ,D12.6,/
     3         22H     TRUE CONTACT ANG    =          ,D12.6
     4        )
2006  FORMAT (1H1,///,10X,5HA)                                          ,/
     1          10X,35HFREE SURFACE EQUILIBRIUM SHAPE FOR               ,/
     2          25X,15HBOND NUMBER    =      ,D12.6                     ,/
     3          25X,15HCONTACT ANGLE  =      ,D12.6                     ,/
     4          25X,15HULLAGE VOLUME  =      ,D12.6                     ,/
     5          10X,5HA)                                                ,///
     6           7X,3HARC                                               ,/
     7           4X,6HLENGTH,8X,1HR,12X,1HX,8X,7HR PRIME,6X,
     8              7HX PRIME,6X,5HVSTAR                                ,//
     9           )
2007  FORMAT(2(5X,D12.6),4(5X,F12.3))
2008  FORMAT(10X,F10.3,5(5X,F6.1,2X,F6.1))
2009  FORMAT(///20X,5HA),/
     1         20X,34H     SOLUTION TRAJECTORY FOR ACOF =    ,D12.6/
     2         20X,5HA),///
     3        )
2010  FORMAT(36H HALVING LOOP, EXECUTION TERMINATED          )
2011  FORMAT(///2X,35HNO MINIMUM ERROR SOLUTION FOR ACOF =,D12.6,
     1         19H PLUS OR MINUS ,D12.5)
2020  FORMAT(52H RMAX AND/OR XMAX NOT DEFINED, EXECUTION TERMINATED     )
2030  FORMAT(54H NA OR NX GT 50 FOR SEARCH=FALSE, EXECUTION TERMINATED)
C
      END
```

$PRT CAN

```
FMSUGGRIN207*TPFS.CAN
            DOUBLE PRECISION FUNCTION CAN(Z)
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
      C     EVALUATE, FOR A GIVEN Z, R OF THE CONTAINER OR THE DERIVATIVE
      C     OF R WITH RESPECT TO Z OF THE CONTAINER
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
            COMMON          /CONSTS/ PI ,ANUM
            COMMON          /PARAMS/  ACOF, BCOF,ZMAX,RMAX
      C
            ENTRY RCAN(Z)
            IZ = 1
            GO TO 100
            ENTRY RCANP(Z)
            IZ= 2
      C
      C   CODE COMMON TO THE EVALUATION OF BOTH R(Z)
      C
        100 CONTINUE
            IF(Z .GT. 47.3000) GO TO 110
            JZ = 1
            X = 27.3000/47.3000
            GO TO 199
        110 IF(Z .GT. 55.9000) GO TO 120
            JZ = 2
            X = 3.9000/8.6000
            GO TO 199
        120 IF(Z .GT. 65.5000) GO TO 130
            JZ = 3
            X = 4.1000/9.5000
            GO TO 199
        130 IF(Z .GT. 123.2000) GO TO 140
            JZ = 4
            X = 32.2000
            GO TO 199
        140 IF(Z. GT. 146.5000) GO TO 400
            JZ = 5
            X = (32.2000**2) - (32.2000/23.3000)**2*(Z-123.2000)**2
            IF(X .GE. 0.0000) X = DSQRT(X)
            IF(X .LT. 0.0000) X = 0.0000
        199 CONTINUE
            GO TO (200,300) , IZ
      C
      C   CODE FOR R(Z)
      C
        200 CONTINUE
            GO TO (210,220,230,240,250) , JZ
        210 CAN = X*Z
            RETURN
        220 CAN = 27.3000 + X*(Z-47.3000)
```

```
56                RETURN
57          230   CAN = 31.1000 + X*(Z-56.3000)
58                RETURN
59          240   CAN = X
60                RETURN
61          250   CAN = X
62                RETURN
53          C
64          C     CODE FOR R PRIME (Z)
55          C
66           300  CONTINUE
67                GO TO(310,320,330,340,350) , JZ
68           310  CAN = X
69                RETURN
70           320  CAN = X
71                RETURN
72           330  CAN = X
73                RETURN
74           340  CAN = 0.0000
75                RETURN
76           350  CAN = (470.5877000 - 3.8197060000*Z)/(2.0000*X +1.0D-10)
77                RETURN
78          C
79          C     ERROR EXITS
80          C
81           400  CONTINUE
82                CAN = 1.00+10
83                RETURN
84          C
85          C
86                END
```

```
$PRT  SOMACK
```

```
FMBUGGHIN2D7*TPFS.GOBACK
               LOGICAL FUNCTION GO BACK(GOOD)
               IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C
          C       ROUTINE TO DETERMINE WHEN TO EXIT INTEGRATION LOOP
          C
          C       ROUTINE TO DETERMINE WHEN TO EXIT INTEGRATION LOOP
          C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C
               LOGICAL         GOOD
               COMMON          /VECTOR/ Y(5),YDT(5)
               COMMON          /TIMESS/ DELTAT,T
               COMMON          /PARAMS/ ACOF,BCOF,ZMAX,RMAX
          C
               GO BACK = .TRUE.
               GOOD = .TRUE.
               IF(Y(1) .GT. RMAX       ) GO BACK = .FALSE.
               IF(Y(1) .LT. 0.0000) GO BACK = .FALSE.
               IF(Y(1) .LT. 0.0000) GOOD  = .FALSE.
               IF(Y(2) .GT. 0.0000 .AND. Y(4) .LT. 0.0000) GO BACK = .FALSE.
               IF(Y(2) .GT. 0.0000 .AND. Y(4) .LT. 0.0000) GOOD = .FALSE.
               IF(Y(2) .LT. 0.0000 .AND. Y(4) .GT. 0.0000) GO BACK = .FALSE.
               IF(Y(2) .LT. 0.0000 .AND. Y(4) .GT. 0.0000) GOOD = .FALSE.
          C
               RETURN
               END


@PRT  LOHOPL
```

```
FMBUGGRIN207*TPFS.LOBOPL
      1              SUBROUTINE LOBOPL (DATA,NA,NZ,ZMAX,KDATA)
      2        C
      3        C     DATA  = INPUT DATA ARRAY
      4        C     NA    = NO OF ACOF VALUES IN DATA
      5        C     NZ    = NO OF Z SWEEP VALUES IN DATA
      6        C     ZMAX  = MAX AXIAL TANK COORDINATE
      7        C     KDATA = ROW DIMENSION OF DATA ARRAY IN CALLING PROGRAM
      8        C           NOTE---MAIN PROGRAM MUST CALL IDENT(1) TO INITIALIZE SC4020
      9        C                  AND CALL ENDJOB TO TERMINATE SC4020
     10        C
     11              DIMENSION DATA(KDATA,KDATA),TY(12),TX(12),ISY(50),Y(50),X(50)
     12        C
     13              DATA NOT  /  6  /
     14              DATA ISY  /
     15             *2H  ,2H 1 ,2H 2 ,2H 3 ,2H 4 ,2H 5 ,2H 6 ,2H 7 ,2H 8 ,2H 9 ,2H10 ,2H11 ,2H12 ,2H13 ,
     16             *2H14 ,2H15 ,2H16 ,2H17 ,2H18 ,2H19 ,2H20 ,2H21 ,2H22 ,2H23 ,2H24 ,2H25 ,2H26 ,
     17             *2H27 ,2H28 ,2H29 ,2H30 ,2H31 ,2H32 ,2H33 ,2H34 ,2H35 ,2H36 ,2H37 ,2H38 ,2H39 ,
     18             *2H40 ,2H41 ,2H42 ,2H43 ,2H44 ,2H45 ,2H46 ,2H47 ,2H48 ,2H49 ,2H50  /
     19        C
     20              DO 2 I=1,12
     21              TX(I) = 6H
     22            2 TY(I) = 6H
     23        C
     24              TX(5) = 6HTANK A
     25              TX(6) = 6HXIS IN
     26              TX(7) = 6HTFRCFP
     27              TX(8) = 6HT
     28        C
     29              NR = NZ + 1
     30              NC = NA + 1
     31        C
     32        C  ULLAGE VOLUME PLOT  L = 1
     33        C  CONTACT ANGLE PLOT  L = 2
     34        C
     35              DO 100 L=1,2
     36              GO TO (101,102), L
     37          101 YMAX = 100.
     38              TY(5) = 6HULLAGE
     39              TY(6) = 6H VOLUM
     40              TY(7) = 6HE  PER
     41              TY(8) = 6HCENT
     42              GO TO 103
     43          102 YMAX = 190.
     44              TY(5) = 6HCONTAC
     45              TY(6) = 6HT ANGL
     46              TY(7) = 6HE  DEG
     47              TY(8) = 6HREES
     48          103 CONTINUE
     49        C
     50              IER = 0
     51              WRITE(NOT,1000) (TY(I),I=5,8), (TX(I),I=5,8)
     52         1000 FORMAT(1H0,10X,16HPLOT SYMBOLS FOR,/, 5X,4A6,4H AS ,A6,///,
     53            ,            10X, 6HSYMBOL, 15X, 4HACOF,//)
     54        C
     55              DO 120 K= 2,NC
```

```
      AC = DATA(I,KA,1)
      IS = ISY(KA-1)
      WRITE(NOT,1004) IS, AC
 1001 FORMAT(12X,A2,10X,E17.8)
C
      KNT = 0
      DO 80 KZ = 2,NR
      XX = DATA(KZ,I,1)
      IF(L .EQ. 1) YY = DATA(KZ,KA,1)
      IF(L .EQ. 2) YY = DATA(KZ,KA,2)
      IF(YY .GE. 0.0 .AND. YY .LE. YMAX) GO TO 91
      IF(KNT .EQ. 0) GO TO 80
C
   75 IF(IFR .EQ. 0) CALL QUIK3L(-1.0,.ZMAX,0.,.YMAX,35,TX,TY,-KNT,X,Y)
      IF(IFR .EQ. 1) CALL QUIK3L( 0.0,.ZMAX,0.,.YMAX,35,TX,TY,-KNT,X,Y)
C
      CALL XSCLV1(X(  1),IXRAS,IXERR)
      CALL YSCLV1(Y(  1),IYRAS,IYERR)
      CALL PRINTV(2,IS,IXRAS,IYRAS)
      CALL XSCLV1(X(KNT),IXRAS,IXERR)
      CALL YSCLV1(Y(KNT),IYRAS,IYERR)
      CALL PRINTV(2,IS,IXRAS,IYRAS)
C
      IFR = 1
      KNT = 0
      GO TO 80
   91 KNT = KNT + 1
      X(KNT) = XX
      Y(KNT) = YY
      IF(KZ .EQ. NR) GO TO 75
   80 CONTINUE
  120 CONTINUE
  100 CONTINUE
C
C  CROSS PLOTS   ULLAGE VOLUME VS CONTACT ANGLE -- ONE FRAME
C
      TY(5) = 6HULLAGE
      TY(6) = 6H VOLUM
      TY(7) = 6HE  PER
      TY(8) = 6HCENT
      TX(5) = 6HCONTAC
      TX(6) = 6HT ANGL
      TX(7) = 6HE  DEG
      TX(8) = 6HREES
C
      WRITE(NOT,1002) (TY(I),I=5,8), (TX(I),I=5,8)
 1002 FORMAT(1H1, 10X, 16HPLOT SYMBOLS FOR,/, 5X,4A6, 4H VS , 4A6,///
     *            10X, 6HSYMBOL, 15X, 9HINTERCEPT,//)
C
      DO 130 KZ=2,NR
      IS = ISY(KZ-1)
      ZI = DATA(KZ,1,1)
  130 WRITE(NOT,1001) IS,ZI
C
      DO 140 KA=2,NC
      IFR = 0
```

```
      KNT = 0
      DO 150 KZ=2,NR
      XX = DATA(KZ,KA,2)
      YY = DATA(KZ,KA,1)
      IF(YY .GE. 0.0  .AND.  YY .LE. 100.0) GO TO 151
      IF(KNT .EQ. 0) GO TO 150
C
  175 IF(IFR .EQ. 0) CALL GUIK3L(-1.0,.180.,0.,100.,35,TX,TY,-KNT,X,Y)
      IF(IFR .EQ. 1)  CALL GUIK3L( 0.0,.190.,0.,100.,35,TX,TY,-KNT,X,Y)
C
      DO 160 KL=1,KNT
      IS = ISY(ISL + KL - 1)
      CALL XSCLV1(X(KL),IXRAS,IXERR)
      CALL YSCLV1(Y(KL),IYRAS,IYERR)
  160 CALL PRINTV(2,IS,IXRAS,IYRAS)
C
      IFR = 1
      KNT = 0
      GO TO 150
  151 KNT = KNT + 1
      IF(KNT .EQ. 1) ISL = KZ - 1
      X(KNT) = XX
      Y(KNT) = YY
      IF(KZ .EQ. NR) GO TO 175
  150 CONTINUE
  190 CONTINUE
C
      RETURN
      END
```

```
@PRT  PAGEHD
```

```
FMBUGGRIN207*IPFS.PAGEHD
      1              SUBROUTINE PAGEHD                                        |
      2              COMMON/LSTART/ IRUNNO, IDATE, NPAGE, UNAME(3), TITLE1(12), TITLE2(12)
      3              DATA  NIT,NOT/5,6/                                       |
      4        C
      5        C  BRINGS UP NEW PAGE AND PUTS HEADING AT TOP.                 |
      6        C
      7        C     INTERNAL VARIABLES.  (TRANSFERRED THRU COMMON).          |
      8        C  IRUNNO = RUN NUMBER. (A6 FORMAT)
      9        C  IDATE  = DATE. (A5 FORMAT)                                  |
     10        C  NPAGE  = PAGE NUMBER.
     11        C  UNAME  = USERS NAME. (3A6 FORMAT)                           |
     12        C  TITLE1 = FIRST  TITLE. (12A6 FORMAT)
     13        C  TITLE2 = SECOND TITLE. (12A6 FORMAT)                        |
     14        C
     15         2001 FORMAT (9H1RUN NO. ,A6,42X,5X ,6X,42X,9HPAGE NO. ,I5 /
     16             ,55X,6HRUN BY,1X,3A6/,10X,12A6/,10X,12A6)
     17        C                                                              |
     18              NPAGE = NPAGE + 1
     19              WRITE(NOT,2001) IRUNNO,NPAGE,UNAME,TITLE1,TITLE2         |
     20              RETURN
     21              END


#PRT  PRINTS
```

```
FMBUGG*IN207*TPFS.PRINTS
      SUBROUTINE PRINTS(N,NEQ,I)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     OUTPUT ROUTINE - PRINT STATE FOR EACH DELTAT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON         /TIMESS/ DELTAT,T
      COMMON         SOLN(1000,5,2)
C
      DATA      NOT  /6/
C
      TS = SNGL(T)
      WRITE(NOT,1000) TS,(SOLN(N,J,I),J = 1,NEQ)
 1000 FORMAT(F10.3,5D13.5)
C
      RETURN
      END

@PRT   RCAN2
```

```
FMBUGGHIN2D7*TPFS.RCAN2
            DOUBLE PRECISION FUNCTION RCAN2(Z)
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C           ROUTINE TO EVALUATE THE SQUARE OF R OF THE CONTAINER TO BE
C           USED IN COMPUTING THE ULLAGE VOLUME
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
            RCAN2 = (RCAN(Z))**2
C
            RETURN
            END

@PRT   RUNKTA
```

```
FMBUGGRIN207*TPFS.RUNKTA
              SUBROUTINE RUNKTA(NEQTN,NT)
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       RUNGE-KUTTA-GILL NUMERICAL INTEGRATION ALGORITHM
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON        /QPRKTA/ QRK(5),PRK(4)
      COMMON        /TIMESS/ DELTAT,T
      COMMON        /VECTOR/ Y(5),YDT(5)
C
      DO 120 J = 1,4
        JIL = J
        DO 110 I = 1,NEQTN
          Z = YDT(I)*DELTAT
          GO TO (103,101,101,105), JIL
  101       R = PRK(JIL)*(Z - QRK(I))
            GO TO 107
  103       R = PRK(JIL)*Z - QRK(I)
            GO TO 107
  105       R = (Z - 2.0D00*QRK(I)) / 6.0D00
  107     Y(I) = Y(I) + R
  110   QRK(I) = QRK(I) + 3.0D00*R - PRK(JIL)*Z
        IF(JIL.EQ.1 .OR. JIL.EQ.3) T = T + DELTAT/2.0D00
  120 CALL YDOT
C
      NT = NT + 1
      T = DBLE(FLOAT(NT))*DELTAT
C
      RETURN
      END

@PRT  SAVE
```

```
FMBUGGBIN207*TPF$.SAVE
      SUBROUTINE SAVE(NROW,NEQ,IA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     ROUTINE TO SAVE INTEGRATED STATE SPACE SOLUTION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON        SOLN(1000,5,2)
      COMMON   /VECTOR/ Y(5),YDT(5)
C
      DO 10 I=1,NEQ
   10 SOLN(NROW,I,IA) = Y(I)
C
      RETURN
      END

@PRT  START
```

```
FMBUGGRIN207*TPF$.START
      1              SUBROUTINE START
      2              DIMENSION MONTHN(12),MONTHL(12),ADAY(22)
      3              COMMON /LSTART/ IRUNNO,IDATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
      4              DATA NIT,NOT/5,6/
      5              DATA MONTHN/2H01,2H02,2H03,2H04,2H05,2H06,
      6             *            2H07,2H08,2H09,2H10,2H11,2H12/,
      7             *     MONTHL/2HJA,2HFE,2HMR,2HAP,2HMY,2HJN,
      8             *            2HJL,2HAU,2HSE,2HOC,2HNO,2HDE/
      9              DATA IIST / 0 /
     10        C
     11         1001 FORMAT (A6, 9X, 3A6)
     12         1002 FORMAT (12A6)
     13         2003 FORMAT (35H1END OF INPUT DATA HAS BEEN REACHED.)
     14        C
     15              IF (IIST.EQ.0) CALL IDENT (3,ADAY)
     16              IIST = 1
     17              READ (NIT,1001) IRUNNO,UNAME
     18              IF (IRUNNO .NE. 4HSTOP) GO TO 10
     19              CALL ENDJOB
     20              WRITE (NOT,2003)
     21              STOP
     22        C
     23           10 READ (NIT,1002) TITLE1
     24              READ (NIT,1002) TITLE2
     25              NPAGE = 0
     26              RETURN
     27              END

$PRT   STATE
```

```
FMBUGGAIN207*TPFS.STATE
                SUBROUTINE STATE(I,J,ZI,RS,ZS,RPS,ZPS,VS)
                IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
      C         HAVING BRACKETED CAN, FIND A GOOD STATE
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
                COMMON          SOLN(1000,5,2)
      C
                CALC(K,Z) = SOLN(J-1,K,I)+(FACTOR*(SOLN(J,K,I)-SOLN(J-1,K,I)))+Z
      C
                RCIN = RCAN(SOLN(J-1,2,I)+ZI)
                RCOUT = RCAN(SOLN(J,2,I)+ZI)
                F1 = RCIN - SOLN(J-1,1,I)
                F2 = SOLN(J,1,I) - RCOUT
                FACTOR = F1 / (F1+F2)
                RS = CALC(1,0.0000)
                ZS = CALC(2,ZI)
                RS = CALC(1,0.0000)
                ZS = CALC(2,ZI)
                RPS = CALC(3,0.0000)
                ZPS = CALC(4,0.0000)
                VS = CALC(5,0.0000)
                RSS = DSQRT(RPS**2 + ZPS**2)
                RPS = RPS / RSS
                ZPS = ZPS / RSS
                RETURN
                END

@PRT  SIMPS
```

```
FMBUGGRIN207*TPFS.SIMPS
            DOUBLE PRECISION FUNCTION SIMPS(A,B,F)
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
      C     FUNCTION TO NUMERICALLY INTEGRATE THE FUNCTION F
      C     FROM A TO B USING SIMPSONS RULE
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
            DATA            INT/40/
      C
      C     A-LOWER LIMIT OF INTEGRATION
      C     B-UPPER LIMIT OF INTEGRATION
      C     F-INTEGRAND FUNCTION (DECLARED EXTERNAL IN CALLING PGM)
      C     IPAR-PARAMETER PASSED TO INTEGRAND FUNCTION
      C
      C     INITIALIZE PARAMETERS
      C
            TWOH=(B-A)/DBLE(FLOAT(INT))
            H = TWOH/2.0D00
            SUMEND=0.0D00
            SUMMID = 0.0D00
      C
      C     TWOH-INTERVAL
      C     H-HALF INTERVAL
      C     SUMMEND-SUM OF F(X SUB I), FOR EVEN I
      C     SUMMID-SUM OF F(X SUB I), FOR ODD I.
      C
      C     EVALUATE SUMEND AND SUMMID.
      C
            DO 1  K = 1, INT
              X=A+DBLE(FLOAT(K-1))*TWOH
              SUMEND   =   SUMEND   +   F(X)
          1 SUMMID = SUMMID + F(X+H)
      C
      C     RETURN ESTIMATED VALUE OF THE INTEGRAND
      C
            SIMPS = (2.0D00*SUMEND+4.0D00*SUMMID-F(A)+F(B))*H/3.0D00
      C
            RETURN
            END


@PRT  VULL
```

```
FMBUGGBIN207*TPFS.VULL
              DOUBLE PRECISION FUNCTION VULL(VSTAR,ZA,ZMAX)
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
      C       COMPUTE ULLAGE VOLUME FOR A GIVEN SOLUTION
      C
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
              EXTERNAL         RCAN2
              COMMON          /CONSTS/ PI, ANUM
      C
              VULL = PI*(VSTAR + SIMPS(ZA,ZMAX,RCAN2))
      C
              RETURN
              END

@PRT  VDOT
```

```
FMBUGGBIN2D7*TPFS.YDOT
      SUBROUTINE YDOT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     ROUTINE TO COMPUTE YDOT AS A FUNCTION OF Y
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON        /VECTOR/ Y(5),YDT(5)
      COMMON        /PARAMS/ ACOF,BCOF,ZMAX,RMAX
C
      RSS = DSQRT(Y(3)*Y(3) + Y(4)*Y(4))
      Y(3) = Y(3) / RSS
      Y(4) = Y(4) / RSS
      CCOF = 0.0000
      IF(Y(1) .NE. 0.0000) CCOF = Y(4) / Y(1)
      TRM = ACOF + BCOF*Y(2) - CCOF
C
      YDT(1) = Y(3)
      YDT(2) = Y(4)
      YDT(3) = -Y(4)*TRM
      YDT(4) = Y(3)*TRM
      YDT(5) = Y(1)*Y(1)*Y(4)
C
      RETURN
      END

@FIN
```

```
PHILIPBIN207*F2.MAIN
      1            COMPILER (XM=1),(EQUIV=CRM)
      2      C
      3      C  MAIN PROGRAM TO CALCULATE LOW BOND NUMBER TANK SLOSH MODES.
      4      C  DEVELOPED BY RL WOHLEN. FEBRUARY 1975.
      5            COMMON / DOUBLE / IWORDS(36000)
      6            COMMON / RAP2 / IWRD2
      7            COMMON / RAP5 / IWRD5
      8      C
      9      C  INPUT DATA READ IN THIS PROGRAM.
     10      C  10 CALL START
     11      C     IFINIT,TAPEID                                    FORMAT (2A6)
     12      C     CALL DGEN1  (SEE SUBROUTINE FOR INPUT)
     13      C     MOPT                                             FORMAT (A6)
     14      C     IF (MOPT .EQ. 6HMODED ) CALL MODED  (NO INPUT REQUIRED)
     15      C     IF (MOPT .EQ. 6HMODES ) CALL MODES  (SEE SUBRT FOR INPUT)
     16      C     IOPT                                             FORMAT (A6)
     17      C     IF (IOPT .EQ. 6HPLOT  ) CALL OPLT2S (SEE SUBRT FOR INPUT)
     18      C     GO TO 10
     19      C
     20            DATA NIT,NOT/5,6/
     21      C
     22      C  DEFINE READ,WRITE TAPES FOR FINEL.
     23            DATA NUTEL,NUTXYZ /
     24           *       29,    18 /
     25            DATA NUTLT,NUTST,NUTMX,NUTKX,NUTBX /
     26           *        3,     1,     2,    26,     7 /
     27      C  DEFINE BUFFER IN,OUT TAPES FOR MASS, STIF, MODES, FREQ MATRICES.
     28            DATA NUTM,NUTK,NUTP,NUTF/
     29           *       21,    22,    23,    25/
     30      C  DEFINE READ,WRITE UTILITY TAPES.
     31            DATA NUTR1,NUTR2,NUTR3 /
     32           *        8,     9,    10 /
     33      C  DEFINE BUFFER IN,OUT UTILITY TAPES.
     34            DATA NUTB1,NUTB2,NUTB3,NUTB4,NUTB5,NUTB6,NUTB7 /
     35           *       11,    12,    13,    14,    15,    16,    17 /
     36      C  DEFINE FORMA LIBRARY TAPES.
     37            DATA NRSVT1          / 28    /
     38      C
     39       1001 FORMAT (12A6)
     40      C
     41            REWIND NRSVT1
     42         10 CALL START
     43            READ (NIT,1001) IFINIT,TAPEID
     44            IF (IFINIT .EQ. 6HINITIL) CALL INTAPE (NRSVT1,TAPEID)
     45      C
     46            REWIND NUTEL
     47            REWIND NUTXYZ
     48            CALL LBDGEN (NUTEL,NUTXYZ)
     49            REWIND NUTEL
     50            CALL LBFINE (NUTEL,NUTXYZ,NUTM,NUTK,NUTLT,NUTST,NUTMX,NUTKX,NUTBX,
     51           *             NUTB1,NUTB2,NUTB3)
     52            READ (NIT,1001) MOPT
     53            IF (MOPT .EQ. 6HMODED ) GO TO 50
     54            IF (MOPT .EQ. 6HMODES ) GO TO 70
     55                                                             NERROR=1
```

```
      GO TO 999
   50 CALL MODED  (NUTM,NUTK,NUTF,NUTF,NRSVTI,NUTBI,NUTRI)
      GO TO 110
   70 CALL MODES  (NUTM,NUTK,NUTP,NUTF,NRSVTI,
     *              NUTB1,NUTB2,NUTB3,NUTB4,NUTB5,NUTB6,NUTB7)
  110 READ (NIT,1001) IOPT
      IF (IOPT .NE. 6HPLOT  ) GO TO 10
      CALL OPLT2C (NUTEL,NUTXYZ,NUTP,NUTF,NUTR1,NUTR2,NUTR3,
     *             NUTB1,NUTB2,NUTB3,NUTB4,NUTB5,NUTB6,NUTB7,NRSVTI)
      GO TO 110
C
  999 CALL ZZBOMB (6HLOWBND,NERROR)
      END
```

```
)PRT  F1.FINELB
```

```
PHILIPB1N207*F1.FINELB
      1              COMPILER (XR=1),(EQUIV=CMN)
      2              SUBROUTINE FINELB (XYZ,JDOF,EUL,NUTEL,NJ,
      3             *                   NUTM,NUTK,                    V,LV,KV,
      4             *                   KRX,KRJ,KRE,NUTMX,NUTKX,NUT1,NUT2,NUT3)
      5              DIMENSION XYZ(KRX,3), JDOF(KRJ,3), EUL(KRE,3), V(1), LV(1)
      6              DIMENSION M1(24,24), M2(24,24), M3(24,24)
      7              DATA KR/24/, IBLANK/6H      /, II/1/
      8              DATA NIT,NOT/5,6/
      9        C
     10        C  SUBROUTINE TO CALCULATE (ON OPTION) FINITE ELEMENT...
     11        C     ASSEMBLED MASS MATRIX (ON NUTM),
     12        C     ASSEMBLED STIFFNESS MATRIX (ON NUTK),
     13        C  IVEC GIVES ELEMENT DOF INTO GLOBAL DOF. EXAMPLES...
     14        C     IVEC(6)=834 PLACES ELEMENT DOF 6 INTO GLOBAL DOF 834.
     15        C     IVEC(3)=0   OMITS ELEMENT DOF 3 FROM GLOBAL DOF. THIS CONSTRAINS
     16        C                 ELEMENT DOF 3 TO ZERO MOTION.
     17        C  DATA ARRANGEMENT ON NUTM, NUTK FOR THE ASSEMBLED MATRICES IS IN
     18        C  SPARSE (Y) FORMA SUBROUTINE FORMAT.
     19        C  DATA ARRANGEMENT ON NUTLT, NUTKX, NUTST, NUTB FOR EACH FINITE
     20        C  ELEMENT (WRITTEN IN SUBROUTINE FLUID, ETC) IS (R=K)
     21        C     WRITE (NUTR) NAME,NEL,NR,NC,NAMEL,(IBLANK,I=1,5),
     22        C                  ((W(I,J),I=1,NR),J=1,NC),(IVEC(I),I=1,NC)
     23        C     NAMEL = FLUID,ETC.
     24        C  LAST RECORD (TO DENOTE TERMINATION) IS,
     25        C     WRITE (NUTR) IBLANK,(II,I=1,30)
     26        C  THE FOLLOWING UTILITY TAPES USE BASIC FORTRAN READ, WRITE. DO NOT
     27        C  USE THESE TAPES IN SPARSE (Y) FORMA SUBROUTINES WHICH USE FORMA
     28        C  SUBROUTINES YIN, YOUT (BECAUSE THEY USE BUFFER IN, BUFFER OUT).
     29        C     NUTMX, NUTKX.
     30        C  THE FOLLOWING UTILITY TAPES USE FORMA YIN, YOUT.
     31        C     NUTM, NUTK, NUT1, NUT2, NUT3.
     32        C  CALLS FORMA SUBROUTINES FLUID ,GRAVTY,PAGEHD,SURFTN,YRVA02,YZERO ,
     33        C                             ZZBOMB.
     34        C  DEVELOPED BY WA BENFIELD, CS BODLEY, RL WOHLEN. JANUARY 1973.
     35        C  LAST REVISION BY RL WOHLEN. FEBRUARY 1975.
     36        C
     37        C  INPUT DATA READ IN THIS SUBROUTINE FROM NUTEL. IF NUTEL = 5, DATA IS
     38        C  READ FROM CARDS.
     39        C  50 NAMEL                                          FORMAT (A6)
     40        C     IF (NAMEL .EQ. 6HRETURN) RETURN
     41        C     IF (NAMEL .EQ. 6HFLUID ) CALL FLUID  (SEE SUBRT FOR INPUT)
     42        C     IF (NAMEL .EQ. 6HGRAVTY) CALL GRAVTY (SEE SUBRT FOR INPUT)
     43        C     IF (NAMEL .EQ. 6HSURFTN) CALL SURFTN (SEE SUBRT FOR INPUT)
     44        C     GO TO 50
     45        C
     46        C  DEFINITION OF INPUT VARIABLES.
     47        C  NAMEL  = FLUID, ETC AS SHOWN ABOVE. GIVES SUBROUTINE CALLED.
     48        C
     49        C  EXPLANATION OF INPUT FORMATS, NUMBER INDICATES CARD COLUMNS USED.
     50        C     A = ANY KEYPUNCH SYMBOL.
     51        C     X = CARD COLUMNS SKIPPED.
     52        C
     53        C     SUBROUTINE ARGUMENTS (ALL INPUT)
     54        C  XYZ     = MATRIX OF JOINT GLOBAL X,Y,Z LOCATIONS. ROWS CORRESPOND
     55        C            TO JOINT NUMBERS. COLUMNS 1,2,3 CORRESPOND TO THE JOINT
```

```
      C                X,Y,Z LOCATIONS RESPECTIVELY. SIZE(NJ,3).
      C                MAY BE EQUIVALENCED TO V(1) IN CALLING PROGRAM.
      C  JDOF    =     MATRIX OF JOINT GLOBAL DEGREES OF FREEDOM. ROWS CORRESPOND
      C                TO JOINT NUMBERS. COLUMNS 1,2,3 CORRESPOND TO THE JOINT
      C                TRANSLATION DOFS AND COLUMNS 4,5,6 CORRESPOND TO THE JOINT
      C                ROTATION DOFS. SIZE(NJ,6).
      C                MAY BE EQUIVALENCED TO LV(1) IN CALLING PROGRAM.
      C  EUL     =     MATRIX OF JOINT EULER ANGLES (DEGREES). ROWS CORRESPOND
      C                TO JOINT NUMBERS. COLUMNS 1,2,3 CORRESPOND TO THE
      C                GLOBAL X,Y,Z PERMUTATION. SIZE(NJ,3). MAY BE
      C                EQUIVALENCED TO V(KRX*(XYZ COL DIM)+1) IN CALLING PROGRAM.
      C  NUTEL   =     LOGICAL NUMBER OF TAPE CONTAINING ELEMENT INPUT DATA FOR
      C                THIS SUBROUTINE AND SUBROUTINES AXIAL, ETC GIVEN BY NAMEL.
      C                IF NUTEL = 5, DATA WILL BE READ FROM CARDS.
      C  NJ      =     NUMBER OF JOINTS OR ROWS IN MATRICES (XYZ), (JDOF), (EUL).
      C  NUTM    =     LOGICAL NUMBER OF UTILITY TAPE ON WHICH ASSEMBLED
      C                MASS MATRIX IS OUTPUT IN SPARSE NOTATION.
      C                NUTM MAY BE ZERO IF MASS MATRIX IS NOT FORMED.
      C                USES FORMA YIN, YOUT.
      C  NUTK    =     LOGICAL NUMBER OF UTILITY TAPE ON WHICH ASSEMBLED
      C                STIFFNESS MATRIX IS OUTPUT IN SPARSE NOTATION.
      C                NUTK MAY BE ZERO IF STIFFNESS MATRIX IS NOT FORMED.
      C                USES FORMA YIN, YOUT.
      C  V       =     VECTOR WORK SPACE.
      C  LV      =     VECTOR WORK SPACE.
      C  KV      =     DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
      C  KRX     =     ROW DIMENSION OF XYZ IN CALLING PROGRAM.
      C  KRJ     =     ROW DIMENSION OF JDOF IN CALLING PROGRAM.
      C  KRE     =     ROW DIMENSION OF EUL IN CALLING PROGRAM.
      C  NUTMX   =     LOGICAL NUMBER OF UTILITY TAPE ON WHICH ELEMENT
      C                MASS MATRICES AND IVECS ARE STORED.
      C                NUTMX MAY BE ZERO IF MASS MATRIX IS NOT FORMED.
      C                USES FORTRAN READ, WRITE.
      C  NUTKX   =     LOGICAL NUMBER OF UTILITY TAPE ON WHICH ELEMENT
      C                STIFFNESS MATRICES (SAME AS GLOBAL LOADS TRANSFORMATION
      C                MATRICES) AND IVECS ARE STORED.
      C                NUTKX MAY BE ZERO IF STIFFNESS MATRIX IS NOT FORMED.
      C                USES FORTRAN READ, WRITE.
      C  NUT1    =     LOGICAL NUMBER OF UTILITY TAPE. USES FORMA YIN, YOUT.
      C  NUT2    =     LOGICAL NUMBER OF UTILITY TAPE. USES FORMA YIN, YOUT.
      C  NUT3    =     LOGICAL NUMBER OF UTILITY TAPE. USES FORMA YIN, YOUT.
      C
 1001 FORMAT (A6)
 2001 FORMAT (//41X 35HJOINT DATA USED IN SUBROUTINE FINEL)
 2002 FORMAT (//35X 47HJOINT DATA USED IN SUBROUTINE FINEL (CONTINUED))
 2003 FORMAT ( /18X 18HDEGREES OF FREEDOM
     *           18X 28HGLOBAL CARTESIAN COORDINATES
     *           12X 22HEULER ANGLES (DEGREES)
     *          /14X 11HTRANSLATION 8X 8HROTATION
     *          / 2X5HJOINT 6X1HU 5X1HV 5X1HW 5X1HP 5X1HQ 5X1HR
     *           11X1HX 11X1HY 11X1HZ 14X1HX 12X1HY 12X1HZ /)
 2004 FORMAT (1X I5, 3X 6I6, 3X 3F12.4, 4X 3F11.4)
      C
        IF (NUTMX .GT. 0) REWIND NUTMX
        IF (NUTKX .GT. 0) REWIND NUTKX
        NUTLT = 0
```

```
      NUTST = 0
C
C  DETERMINE SIZE OF FINAL MASS-STIFFNESS MATRIX FROM THE MAXIMUM DOF
C  NUMBER IN JDOF.
      NDOF = JDOF(1,1)
      DO 35 I=1,NJ
      DO 35 J=1,6
      IF (JDOF(I,J) .GT. NDOF) NDOF=JDOF(I,J)
   35 CONTINUE
C
C  PRINT JOINT DOF, XYZ COORDINATES, EULER ANGLES.
      CALL PAGEHD
      WRITE (NOT,2001)
      WRITE (NOT,2003)
      NLINE = 0
      DO 40 IJ=1,NJ
      NLINE = NLINE+1
      IF (NLINE .LE. 42) GO TO 40
      CALL PAGEHD
      WRITE (NOT,2002)
      WRITE (NOT,2003)
      NLINE = 1
   40 WRITE (NOT,2004) IJ, (JDOF(IJ,J), J=1,6), (XYZ(IJ,J), J=1,3),
     *                 (EUL(IJ,J), J=1,3)
C
C  READ FINITE ELEMENT TYPE.
   50 READ (NUTEL,1001) NAMEL
      IF (NAMEL .EQ. 6HRETURN) GO TO 500
      IF (NAMEL .EQ. 6HFLUID ) GO TO 151
      IF (NAMEL .EQ. 6HGRAVTY) GO TO 171
      IF (NAMEL .EQ. 6HSURFTN) GO TO 190
                                                        NERROR=1
                                                          GO TO 999
C  FLUID ELEMENT.
  151 CALL FLUID  (XYZ,JDOF,EUL,NUTEL,NJ,
     *             NUTMX,NUTKX,      NUTLT,NUTST,
     *             W1,W2,W3,KRX,KRJ,KRE,KW)
      GO TO 50
C  GRAVITY ELEMENT.
  171 CALL GRAVTY (XYZ,JDOF,EUL,NUTEL,NJ,
     *                 NUTKX,
     *             W1,W2,W3,KRX,KRJ,KRE,KW)
      GO TO 50
C  SURFACE TENSION ELEMENT.
  190 CALL SURFTN (XYZ,JDOF,EUL,NUTEL,NJ,
     *                 NUTKX,
     *             W1,W2,W3,KRX,KRJ,KRE,KW)
      GO TO 50
C
C  TERMINATE FINITE ELEMENT DATA ON STORAGE DISKS.
  500 IF (NUTMX .GT. 0) WRITE (NUTMX) IBLANK,(II,I=1,30)
      IF (NUTKX .GT. 0) WRITE (NUTKX) IBLANK,(II,I=1,30)
C
C  SUM FINITE ELEMENT MATRICES.
      IF (NUTM.GT.0) CALL YZERO (NUTM,NDOF,NDOF)
      IF (NUTK.GT.0) CALL YZERO (NUTK,NDOF,NDOF)
```

```
      IF (NUTMX .GT. 0) CALL YRVAD2 (NUTMX,NUTM,NDOF,W1,KW,V,LV,KV,
     *                               NUT1,NUT2,NUT3)
      IF (NUTKX .GT. 0) CALL YRVAD2 (NUTKX,NUTK,NDOF,W1,KW,V,LV,KV,
     *                               NUT1,NUT2,NUT3)
      RETURN
C
  999 CALL ZZBOMB (6HFINELB,NERROR)
      END
```

$PRT FI,K2ST1

```
PHILIPBIN207*FI.K2STI
      COMPILER (XM=1),(EQUIV=CMN)
      SUBROUTINE  K2STI   (X2,X3,Y3,ST,Z,KZ)
      DIMENSION Z(KZ,1)
C
C  SUBROUTINE TO CALCULATE FINITE ELEMENT...
C     STIFFNESS MATRIX,
C  FOR A SURFACE TENSION TRIANGLE ELEMENT WITH UNRESTRAINED BOUNDARIES.
C  LINEAR DISPLACEMENT FIELD IS USED.
C  STIFFNESS MATRIX IS IN LOCAL COORDINATE SYSTEM.
C  THE LOCAL COORDINATE SYSTEM ASSUMES THE PLATE TO LIE IN AN X-Y PLANE
C  WITH JOINT 1 AT THE X-Y ORIGIN, JOINT 2 LIES ALONG THE POSITIVE
C  X AXIS, AND JOINT 3 IS IN THE POSITIVE Y DIRECTION.
C  LOCAL COORDINATE ORDER IS
C     DZ1,D22,DZ3.
C  WHERE DZ IS TRANSLATION (OUT OF PAPER).
C  DEVELOPED BY  RL WOHLEN.  FEBRUARY 1975.
C
C     SUBROUTINE ARGUMENTS
C  X2       = INPUT  LOCAL X COORDINATE LOCATION OF JOINT 2.
C  X3       = INPUT  LOCAL X COORDINATE LOCATION OF JOINT 3.
C  Y3       = INPUT  LOCAL Y COORDINATE LOCATION OF JOINT 3.
C  ST       = INPUT  SURFACE TENSION (FORCE/LENGTH).
C  Z        = OUTPUT STIFFNESS MATRIX. SIZE(3,3).
C  KZ       = INPUT  ROW DIMENSION OF Z  IN CALLING PROGRAM. MIN=3.
C
      A = X2*Y3/2.
      CONST = ST/(4.*A)
      X3MX2 = X3-X2
      Z(1,1) = CONST * (Y3**2 + X3MX2**2)
      Z(1,2) =-CONST * (Y3**2 + X3*X3MX2)
      Z(1,3) = CONST * X2* X3MX2
      Z(2,2) = CONST * (Y3**2 + X3**2)
      Z(2,3) =-CONST * X2 * X3
      Z(3,3) = CONST * X2**2
C
C  SYMMETRIZE LOWER HALF.
      DO 10 J=1,3
      DO 10 I=J,3
   10 Z(I,J) = Z(J,I)
C
      RETURN
      END


$PRT  FI.LBDGEN
```

```
PHILIPBIN207*FI.LBDGEN
      1               COMPILER IXM=1',EQUIV=CHK)
      2               SUBROUTINE LBDGEN (NUTEL,NUTXYZ)
      3               COMMON / DOUBLE / XYTZ(50,3), ICW(50), IFS(50), JRN(50),
      4              *                  XYZ(900,3), JDOF(900,6), EUL(900,3),
      5              *                  IWORDS(24900)
      6               DATA KGP,K3, KJ,K4/
      7              *      50, 3,900, 4/
      8               DATA EPS/1.E-10/, DTR/.0174532925/, IZ/0/, I3/3/, I6/6/
      9               DATA   NAMEF , NAMEG , NAMEST, NAMEK , IBLANK, IRTN /
     10              *     6HFLUID ,6HGRAVTY,6HSURFTN,6HK1    ,6H      ,6HRETURN/
     11               DATA NIT,NOT/5,6/
     12          C
     13           1001 FORMAT (10X,I5)
     14           1002 FORMAT (10X,2E17.0)
     15           1005 FORMAT (A6)
     16           1010 FORMAT (5X,4I5)
     17           1021 FORMAT (5(A6,4X))
     18           1022 FORMAT (3(5X,E10.3))
     19           2010 FORMAT (9I5)
     20           2022 FORMAT (3(5X,1PE10.4))
     21          C
     22          C  DATA GENERATOR FOR LOW BOND CONTRACT.
     23          C  AXI-SYMMETRIC FLUID CONTAINER. RIGID WALL.
     24          C  90 DEGREE MODEL. USER SUPPLIED GRID.
     25          C  GENERATES (1) SIZES AND JDOF,XYZ,EUL MATRICES ON NUTXYZ FOR ARGUMENT
     26          C                  INPUT TO SUBROUTINE FINEL.
     27          C                (2) MASS TYPE, STIF TYPE, DENSITY, BULK MODULUS, GRAVITY,
     28          C                    FLUID ELEMENT JOINT NUMBERS ON NUTEL TO BE READ IN
     29          C                    SUBROUTINES FLUID, GRAVTY.
     30          C  SYMMETRIC, ANTI-SYMMETRIC CASE, THAT IS,
     31          C  U=DX=SOMETHING, V=DY=SOMETHING, W=DZ=0 ON XY PLANE.
     32          C  U=DX=0, V=DZ=0, W=-DY=SOMETHING ON XZ PLANE.
     33          C  X=CONTAINER AXIS OF SYMMETRY (+UP), Y=+RIGHT, Z=+INTO PAPER.
     34          C  EULER ANGLES ONLY USED ON BOUNDARIES WHERE CONSTRAINTS ARE APPLIED,
     35          C  THAT IS, AT XY PLANE, XZ PLANE, CONTAINER WALL.
     36          C  EULER ANGLE THETA X = 0 DEG ON XY PLANE, 90 DEG ON XZ PLANE,
     37          C  90/SECTION BETWEEN XY AND XZ PLANES AT CONTAINER WALL.
     38          C  EULER ANGLE THETA Y = 0 EVERYWHERE.
     39          C  EULER ANGLE THETA Z SUCH THAT V IS NORMAL OUT AND U IS TANGENT AT
     40          C  CONTAINER WALL.
     41          C  FIRST GRID POINTS ARE ON X-AXIS, 1 AT CONTAINER BOTTOM.
     42          C  DEFINITION. GRID POINTS = JOINTS ON XY PLANE.
     43          C  DEVELOPED BY RL WOHLEN, JANUARY 1975.
     44          C  LAST REVISION BY RL WOHLEN, FEBRUARY 1975.
     45          C
     46               READ (NIT,1001) NSECT
     47               CALL READ   (XYTZ, NGP,I3, KGP,K3)
     48               READ (NIT,1002) ZAXIS
     49          C  IF CONTAINER IS A CYLINDER, READ GRID POINT OF CORNER.
     50          C  IF CONTAINER IS NOT A CYLINDER, INPUT IGPCYL=0.
     51               READ (NIT,1001) IGPCYL
     52          C  READ CONTAINER WALL GRID POINT NUMBERS.
     53               CALL READIM (ICW, I1,NCW, 1,KGP)
     54          C  READ FLUID SURFACE GRID POINT NUMBERS FROM AXIS OUT.
     55               CALL READIM (IFS, I1,NFS, 1,KGP)
```

```
      READ (NIT,1005) NAMEM
      READ  (NIT,1022) RO,BRM,ST
      READ  (NIT,1022) GX
      GY = 0.0
      GZ = 0.0
C  READ ELEMENT GRID POINT NUMBERS AT STATEMENT 210.
C
C  CALCULATE XYZ, JDOF, EUL MATRICES.
      YAXIS = XYTZ(1,2)
      SECT = NSECT
      TXINC = 90./SECT
C  ZERO OUT MATRICES.
      DO 25 J=1,3
      DO 20 I=1,KJ
      XYZ(I,J) = 0.0
   20 EUL(I,J) = 0.0
      DO 25 J=1,6
      DO 25 I=1,KJ
   25 JDOF(I,J) = 0.0
C  CALCULATE NUMBER OF JOINTS ON X-AXIS.
      NPAX = 0
      DO 50 IGP=1,NGP
      IF (ABS(XYTZ(IGP,2)-YAXIS) .LT. EPS) NPAX=NPAX+1
   50 CONTINUE
C
C  X-AXIS JOINTS. TZ=90 AT JOINT 1, TZ=0 AT OTHER JOINTS.
      DO 102 IGP=1,NPAX
      XYZ(IGP,1) = XYTZ(IGP,1)
      XYZ(IGP,2) = XYTZ(IGP,2)
  102 XYZ(IGP,3) = ZAXIS
      EUL(1,3) = XYTZ(1,3)
      JDOF(1,1) = 1
      DO 105 IGP=2,NPAX
  105 JDOF(IGP,2) = IGP
C
C  JOINTS OTHER THAN X-AXIS.
      NPAXP1 = NPAX+1
      NPLANE = NSECT+1
      J = NPAX
      IDOF = NPAX
      DO 129 IGP=NPAXP1,NGP
C  SEE IF POINT IS ON CONTAINER WALL.
      IFCW = 0
      DO 122 IC=1,NCW
      IF (IGP .EQ. ICW(IC)) GO TO 123
  122 CONTINUE
      GO TO 124
  123 IFCW = 1
  124 DO 129 IPLANE=1,NPLANE
      TX = FLOAT(IPLANE-1)*TXINC
      CTX = COS(TX*DTR)
      STX = SIN(TX*DTR)
      YLOCAL = XYTZ(IGP,2)-YAXIS
      J = J+1
      XYZ(J,1) = XYTZ(IGP,1)
      XYZ(J,2) = YAXIS + YLOCAL*CTX
```

C-2

```
112             XYZ(J,3) = ZAXIS + YLOCAL*STX
113             IF (IFCW .EQ. 1 .OR. IPLANE .EQ. NPLANE) EUL(J,1)=TX
114             EUL(J,3) = XYTZ(IGP,3)
115             IF (IPLANE .EQ. NPLANE .OR. IGP .EQ. IGPCYL) GO TO 127
116             IDOF = IDOF+1
117             JDOF(J,1) = IDOF
118             IF (IFCW .EQ. 1) GO TO 127
119             IDOF = IDOF+1
120             JDOF(J,2) = IDOF
121         127 IF (IPLANE .EQ. 1) GO TO 129
122             IDOF = IDOF+1
123             JDOF(J,3) = IDOF
124         129 CONTINUE
125             NJ = J
126             WRITE (NUTXYZ) NJ,I3,NJ,I6,NJ,I3
127             WRITE (NUTXYZ) ((JDOF(I,J),I=1,NJ),J=1,6)
128             WRITE (NUTXYZ) ((XYZ (I,J),I=1,NJ),J=1,3)
129             WRITE (NUTXYZ) ((EUL (I,J),I=1,NJ),J=1,3)
130       C
131       C   GET REAL JOINT NUMBERS OF GRID POINTS.
132             IGP = 0
133             DO 205 J=1,NJ
134             IF (ABS(XYZ(J,3)-ZAXIS) .GT. EPS) GO TO 205
135             IGP = IGP+1
136             JRN(IGP) = J
137         205 CONTINUE
138             CALL WRITIM (JRN, NGP,1, 3HJRN, KGP)
139       C
140       C   CALCULATE ELEMENT JOINT NUMBERS FOR SUBROUTINE FLUID.
141             WRITE (NUTEL,1005) NAMEF
142             WRITE (NUTEL,1021) NAMEM,NAMEK,IBLANK,IBLANK
143             WRITE (NUTEL,2022) RO,BKM
144       C   NUMBER OF GIVEN JOINTS ON XY PLANE DETERMINES TYPE OF ELEMENT.
145       C   FOR ELEMENTS ON X-AXIS, 3 JOINTS=TETRAHEDRON, 4 JOINTS=PENTAHEDRON.
146       C   FOR OTHER ELEMENTS, 3 JOINTS=PENTAHEDRON, 4 JOINTS=HEXAHEDRON.
147             NEL = 0
148       C   GRID POINT NUMBERING FOR ELEMENTS MUST BE CLOCKWISE.
149       C   IF ELEMENT IS ON X-AXIS, FIRST TWO GRID POINTS MUST BE ON X-AXIS.
150       C   ELEMENTS CANNOT HAVE JUST ONE GRID POINT ON X-AXIS.
151         210 READ (NIT,1010) IGP1,IGP2,IGP3,IGP4
152             IF (IGP1 .EQ. 0) GO TO 269
153             KEL = 3
154             IF (IGP4 .GT. 0) KEL=4
155             IF (IGP1 .GT. NPAX) GO TO 250
156       C   ELEMENT IS ON X-AXIS.
157             J1 = IGP1
158             J4 = IGP2
159             IF (KEL .EQ. 4) GO TO 230
160             DO 225 ISECT=1,NSECT
161             J2 = JRN(IGP3)+ISECT-1
162             J3 = J2+1
163             NEL = NEL+1
164         225 WRITE (NUTEL,2010) NEL,J1,J2,J3,J4,IZ,IZ,IZ,IZ
165             GO TO 210
166         230 DO 235 ISECT=1,NSECT
167             J2 = JRN(IGP4)+ISECT-1
```

```
      J5 = JRN(IGP3)+ISECT-1
      J3 = J2+1
      J6 = J5+1
      NEL = NEL+1
  235 WRITE (NUTEL,2010) NEL,J1,J2,J3,J4,J5,J6,IZ,IZ
      GO TO 215
C  ELEMENT IS NOT ON X-AXIS.
  252 IF (KEL .EQ. 4) GO TO 260
      DO 255 ISECT=1,NSECT
      J1 = JRN(IGP1)+ISECT-1
      J2 = JRN(IGP2)+ISECT-1
      J3 = JRN(IGP3)+ISECT-1
      J4 = J1+1
      J5 = J2+1
      J6 = J3+1
      NEL = NEL+1
  255 WRITE (NUTEL,2010) NEL,J1,J2,J3,J4,J5,J6,IZ,IZ
      GO TO 215
  260 DO 265 ISECT=1,NSECT
      J1 = JRN(IGP1)+ISECT-1
      J2 = JRN(IGP2)+ISECT-1
      J3 = JRN(IGP3)+ISECT-1
      J4 = JRN(IGP4)+ISECT-1
      J5 = J1+1
      J6 = J2+1
      J7 = J3+1
      J8 = J4+1
      NEL = NEL+1
  265 WRITE (NUTEL,2010) NEL,J1,J2,J3,J4,J5,J6,J7,J8
      GO TO 215
  269 WRITE (NUTEL,2010) IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ,IZ
C
C  CALCULATE ELEMENT JOINT NUMBERS FOR SUBROUTINE GRAVTY.
      WRITE (NUTEL,1005) NAMEG
      WRITE (NUTEL,1021) IBLANK,NAMEK
      WRITE (NUTEL,2022) RO
      WRITE (NUTEL,2022) GX,GY,GZ
      NEL = 0
      J1 = NPAX
      IGP2 = IFS(2)
      DO 275 ISECT=1,NSECT
      J2 = JRN(IGP2)+ISECT-1
      J3 = J2+1
      NEL = NEL+1
  275 WRITE (NUTEL,2010) NEL,J1,J2,J3,IZ
      DO 285 IS=3,NFS
      IGP1 = IFS(IS-1)
      IGP2 = IFS(IS)
      DO 285 ISECT=1,NSECT
      J1 = JRN(IGP1)+ISECT-1
      J2 = JRN(IGP2)+ISECT-1
      J3 = J2+1
      J4 = J1+1
      NEL = NEL+1
  285 WRITE (NUTEL,2010) NEL,J1,J2,J3,J4
      WRITE (NUTEL,2015)IZ,IZ,IZ,IZ,IZ
```

```
      C
      C  CALCULATE ELEMENT JOINT NUMBERS FOR SUBROUTINE SURFTN.
           WRITE (NUTEL,1005) NAMEST
           WRITE (NUTEL,1021) NAMEK
           WRITE (NUTEL,2022) ST
           NEL = 2
           J1 = NPAX
           IGP2 = IFS(2)
           DO 292 ISECT=1,NSECT
           J2 = JRN(IGP2)+ISECT-1
           J3 = J2+1
           NEL = NEL+1
       292 WRITE (NUTEL,2010) NEL,J1,J2,J3,IZ
           DO 295 IS=3,NFS
           IGP1 = IFS(IS-1)
           IGP2 = IFS(IS)
           DO 295 ISECT=1,NSECT
           J1 = JRN(IGP1)+ISECT-1
           J2 = JRN(IGP2)+ISECT-1
           J3 = J2+1
           J4 = J1+1
           NEL = NEL+1
       295 WRITE (NUTEL,2010) NEL,J1,J2,J3,J4
           WRITE (NUTEL,2010)IZ,IZ,IZ,IZ,IZ
      C
      C  RETURN CARD FOR SUBROUTINE FINELB.
           WRITE (NUTEL,1005) IRTN
           RETURN
      C
           END
```

```
@PRT  FI,LBFINE
```

```
PHILIPBIN207*F1.LBFINE
      1            COMPILER (XM=1),(EQUIV=CMN)
      2            SUBROUTINE LBFINE (NUTEL,NUTXYZ,NUTM,NUTK,NUTLT,NUTST,NUTMX,NUTKX,
      3           *                   NUTBX,NUT1,NUT2,NUT3)
      4      C
      5      C  MAIN PROGRAM TO READ (XYZ), (JDOF), (EUL) AND CALCULATE (ON OPTION)
      6      C  ASSEMBLED FINITE ELEMENT MASS, STIFFNESS MATRICES.
      7      C  CALLS FORMA SUBROUTINES FINELB,YIN    ,YWRITE.
      8      C  DEVELOPED BY W BENFIELD, C BODLEY, R PHILIPPUS, R WOHLEN.  JULY 1973.
      9      C  LAST REVISION BY RL WOHLEN, FEBRUARY 1975.
     10      C
     11            DOUBLE PRECISION V
     12            COMMON / DOUBLE / V(12000), LV(12000)
     13            DIMENSION XYZ(2000,3), JDOF(2000,6), EUL(2000,3)
     14            EQUIVALENCE (XYZ(1),V(1)), (EUL(1),V(6001)), (JDOF(1),LV(1))
     15            DATA  KRX, KCX,  KRJ, KCJ,  KRE, KCE,   KV /
     16           *     2000,   3, 2000,   6, 2000,   3,12000 /
     17      C  READ XYZ,JDOF,EUL FROM NUTXYZ CREATED IN DATA GENERATOR LBDGEN.
     18            REWIND NUTXYZ
     19            READ (NUTXYZ) NJ,NCX,NRJ,NCJ,NRE,NCE
     20                                                          NERROR=1
     21            IF (NCX .NE. 3) GO TO 999
     22                                                          NERROR=2
     23            IF (NRJ .NE. NJ .OR. NCJ .NE. 6) GO TO 999
     24                                                          NERROR=3
     25            IF (NRE .NE. NJ .OR. NCE .NE. 3) GO TO 999
     26                                                          NERROR=4
     27            IF ( NJ.GT.KRX .OR. NCX.GT.KCX .OR.
     28           *     NRJ.GT.KRJ .OR. NCJ.GT.KCJ .OR.
     29           *     NRE.GT.KRE .OR. NCE.GT.KCE) GO TO 999
     30            READ (NUTXYZ) ((JDOF(I,J),I=1,NRJ),J=1,NCJ)
     31            READ (NUTXYZ) (( XYZ(I,J),I=1, NJ),J=1,NCX)
     32            READ (NUTXYZ) (( EUL(I,J),I=1,NRE),J=1,NCE)
     33            CALL FINELB (XYZ,JDOF,EUL,NUTEL,NJ,
     34           *             NUTM,NUTK,
     35           *             V,LV,KV,KRX,KRJ,KRE,
     36           *             NUTMX,NUTKX,NUT1,NUT2,NUT3)
     37            CALL YWRITE (NUTM,4HMASS  ,V,LV,KV)
     38            CALL YWRITE (NUTK,4HSTIF  ,V,LV,KV)
     39            RETURN
     40      C
     41        999 CALL ZZBOMB (6HLBFINE,NERROR)
     42            END
```

```
@PRT  F1.MDTITL
```

```
PHILIPBIN207*F1.MDTITL
      1                COMPILER (XM=1),(EQUIV=CMN)
      2                SUBROUTINE MDTITL (PTITLE,MODE,FREQ,NRWT,KPTITL)
      3          C
      4          C  SUBROUTINE TO FORM MODE NUMBER AND FREQUENCY TITLES FOR PLOT3.
      5          C  DEVELOPED BY WA BENFIELD.  FEBRUARY 1974.
      6          C  LAST REVISION BY R A PHILIPPUS.  MARCH 1975.
      7          C
      8                DIMENSION PTITLE(KPTITL)
      9          C
     10           1001 FORMAT (3A10)
     11           1002 FORMAT (5A6)
     12           2001 FORMAT (4HMODE,I4,6H,  F =,F10.6,6H HZ.    )
     13          C
     14                REWIND NRWT
     15                WRITE (NRWT,2001) MODE,FREQ
     16                REWIND NRWT
     17                IF (KPTITL.EQ. 8) READ (NRWT,1001) (PTITLE(I),I=1,3)
     18                IF (KPTITL.EQ.13) READ (NRWT,1002) (PTITLE(I),I=1,5)
     19                RETURN
     20                END


@PRT  F1.MODED
```

```
PHILIPBIN207*F1.MODED
      1              COMPILER (XM=1),(EQUIV=CMN)
      2              SUBROUTINE MODED  (NUTM,NUTK,NUTP,NUTF,NRSVT1,NUTB1,NUTR1)
      3        C
      4        C  SUBROUTINE   TO COMPUTE MODES USING SPARSE MASS AND STIF MATRIGES
      5        C  WITH DENSE MODE SUBROUTINE.
      6        C  DEVELOPED BY WA BENFIELD.  MAY 1974.
      7        C  LAST REVISION BY RL MOHLEN. FEBRUARY 1975.
      8        C
      9              DOUBLE PRECISION V
     10              COMMON / DOUBLE / A(115,115), S(115,115), W2(115), W(115),
     11             *                  FREQ(115), V(3000), LV(3000), INORDS(205)
     12        C
     13              DATA KA,KV / 115, 3000 /
     14        C
     15              CALL YSTOD  (NUTM,A,NRA,NCA,KA ,KA ,V,LV,KV,NUTB1)
     16              CALL YSTOD  (NUTK,S,NRMS,NCS,KA ,KA ,V,LV,KV,NUTB1)
     17              CALL MODE1  (A,S,W2,W,FREQ,NRMS,0.0,KA ,NUTR1)
     18              CALL WRITE  (W2,NRMS,1,2HW2,KA)
     19              CALL WRITE  (FREQ,NRMS,1,4HFREQ,KA)
     20              CALL WRITE  (A,NRMS,NRMS,5HMODES,KA)
     21        C  CONVERT DENSE TO SPARSE FOR PLOTS.
     22              CALL YDTOS  (FREQ,NUTF,NRMS,1,KA,1,V,LV,KV,NUTB1)
     23              CALL YDTOS  (A,NUTP,NRMS,NRMS,KA,KA,V,LV,KV,NUTB1)
     24              IF (NRSVT1 .LE. 0) RETURN
     25              REWIND NRSVT1
     26              CALL WTAPE  (W2,NRMS,1,2HW2,KA ,NRSVT1)
     27              CALL WTAPE  (FREQ,NRMS,1,4HFREQ,KA ,NRSVT1)
     28              CALL YWTAPE (NUTP,5HMODES ,V,LV,KV,NRSVT1)
     29              CALL LTAPE  (NRSVT1)
     30              RETURN
     31              END

@PRT  F1.MODES
```

```
PHILIPBIN207*F1,MODES
      1            COMPILER (XM=1),(EQUIV=CMN)
      2            SUBROUTINE MODES  (NUTM,NUTK,NUTZ,NUTF,NRSVT1,
      3           *                   NUT1,NUT2,NUT3,NUT4,NUT5,NUT6,NUT7)
      4      C
      5      C  ITERATIVE RAYLEIGH-RITZ METHOD OF DR. JOHN ADMIRE.
      6      C  TECHNIQUE = COMPOSITE STRUCTURE.
      7      C  VERSION = NON-SWEEPING.
      8      C  PROGRAMMING LOGIC = SPARSE.
      9      C  MAXIMUM SIZE OF MASS,STIF = 1920.
     10      C  MAXIMUM NU = 70
     11      C  DEVELOPED BY R L WOHLEN AND R A PHILIPPUS.  MARCH 1972.
     12      C  LAST REVISION BY RL WOHLEN. FEBRUARY 1975.
     13      C
     14      C  ***************************************************************
     15      C  INPUT DATA READ IN THIS PROGRAM.
     16      C     NW                                          FORMAT (10X,I5)
     17      C     NU                                          FORMAT (10X,I5)
     18      C     SHIFT                                       FORMAT (10X,E10)
     19      C     MAXIT                                       FORMAT (10X,I5)
     20      C
     21      C  DEFINITION OF INPUT VARIABLES.
     22      C  NW      = NUMBER OF MODES WANTED.
     23      C  NU      = NUMBER OF RAYLEIGH-RITZ MODES TO USE.
     24      C  SHIFT   = SHIFT VALUE TO USE.
     25      C  MAXIT   = MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED.
     26      C
     27            DOUBLE PRECISION V
     28            COMMON / DOUBLE / V(10920), LV(10920), W2( 70), W( 70), FREQ( 70),
     29           *                  IWORDS(3030)
     30            DIMENSION A( 70, 70), S( 70, 70)
     31      C
     32            EQUIVALENCE (V(3641),S(1)), (LV(3641),A(1))
     33      C
     34            DATA NIT,NOT / 5,6 /
     35            DATA    KV, KA /
     36           *     10920, 70 /
     37            DATA NITER1, NITER2,   TOLZ,  TOLW2/
     38           *          0,      1, 1.E-06, 1.E-04/
     39            DATA IFPRNT/1000/
     40       1001 FORMAT (10X, 4I5)
     41       1010 FORMAT (10X, E10.0)
     42      C
     43            READ (NIT,1001) NW
     44            READ (NIT,1001) NU
     45            READ (NIT,1010) SHIFT
     46            READ (NIT,1001) MAXIT
     47      C
     48            CALL YZERO  (NUTZ,1,1)
     49            CALL YMODE2 (NUTM,NUTK,NUTZ,W2,W,FREQ,NW, V,LV,A,S, KV,KA,
     50           *             NUT1,NUT2,NUT3,NUT4,NUT5,NUT6,NUT7,
     51           *             IFPRNT,MAXIT,
     52           *             NU,NITER1,NITER2,SHIFT,TOLZ,TOLW2)
     53      C
     54            CALL WRITE  (W2,NU,1,2HW2,KA)
     55            CALL WRITE  (FREQ,NU,1,4HFREQ,KA)
```

```
      CALL YWRITE (NUTZ,5HMODES,V,LV,KV)
C  CONVERT DENSE TO SPARSE FOR PLOTS.
      CALL YDTOS  (FREQ,NUTF,NU,1,KA,1,V,LV,KV,NUT7)
      IF (NRSVT1 .LE. 0) RETURN
      REWIND NRSVT1
      CALL WTAPE  (W2,NU,1,2HW2,KA,NRSVT1)
      CALL WTAPE  (FREQ,NU,1,4HFREQ,KA,NRSVT1)
      CALL YWTAPE (NUTZ,5HMODES,V,LV,KV,NRSVT1)
      CALL LTAPE  (NRSVT1)
C
      RETURN
      END
```

```
@PRT  F1.STF2ST
```

```
PHILIPBIN207*F1.STF2ST
      1            COMPILER (XM=1),(EQUIV=CMN)
      2            SUBROUTINE STF2ST (CJ,EJ,ST,NAMEK,S,WI,KCJ,KEJ,KS,KWI)
      3            DIMENSION CJ(KCJ,1), EJ(KEJ,1), S(KS,1), WI(KWI,1)
      4      C
      5      C  SUBROUTINE TO CALCULATE FINITE ELEMENT...
      6      C     STIFFNESS MATRIX
      7      C  FOR A SURFACE TENSION TRIANGLE ELEMENT WITH UNRESTRAINED BOUNDARIES.
      8      C  STIFFNESS MATRIX IS IN GLOBAL COORDINATE DIRECTIONS.
      9      C  GLOBAL COORDINATE ORDER IS
     10      C     (U,V,W) JOINT 1, THEN JOINT 2, 3.
     11      C  WHERE U,V,W ARE TRANSLATIONS.
     12      C  EULER ANGLE CONVENTION IS GLOBAL X,Y,Z PERMUTATION.
     13      C  CALLS FORMA SUBROUTINES BTABA,DCOS2,K2ST1,ZZBOMB.
     14      C  DEVELOPED BY RL WOHLEN.  FEBRUARY 1975.
     15      C
     16      C     SUBROUTINE ARGUMENTS
     17      C  CJ      = INPUT   MATRIX OF GLOBAL X,Y,Z COORDINATES AT TRIANGLE JOINTS
     18      C                    ROWS 1,2,3 CORRESPOND TO X,Y,Z COORDINATES.
     19      C                    COLS 1,2,3 CORRESPOND TO JOINTS 1,2,3. SIZE(3,3).
     20      C  EJ      = INPUT   MATRIX OF EULER ANGLES (DEGREES) AT TRIANGLE JOINTS.
     21      C                    ROWS 1,2,3 CORRESPOND TO GLOBAL X,Y,Z PERMUTATION.
     22      C                    COLS 1,2,3 CORRESPOND TO JOINTS 1,2,3. SIZE(3,3).
     23      C  ST      = INPUT   SURFACE TENSION. (FORCE/LENGTH).
     24      C  NAMEK   = INPUT   TYPE OF STIF MATRIX WANTED.
     25      C                    = K1, USES K2ST1, LINEAR DISPLACEMENT FIELD.
     26      C  S       = OUTPUT  STIFFNESS MATRIX . SIZE(9,9).
     27      C  WI      = INPUT   WORKSPACE MATRIX. SIZE(18,18).
     28      C  KCJ     = INPUT   ROW DIMENSION OF CJ IN CALLING PROGRAM.
     29      C  KEJ     = INPUT   ROW DIMENSION OF EJ IN CALLING PROGRAM.
     30      C  KS      = INPUT   ROW DIMENSION OF S IN CALLING PROGRAM. MIN=9.
     31      C  KWI     = INPUT   ROW DIMENSION OF WI IN CALLING PROGRAM. MIN=18.
     32      C
     33                                                                  NERROR=1
     34           IF (KS .LT. 9 .OR. KWI .LT. 18) GO TO 999
     35           SL12 = SQRT((CJ(1,2)-CJ(1,1))**2 + (CJ(2,2)-CJ(2,1))**2
     36          *                                 + (CJ(3,2)-CJ(3,1))**2)
     37           SL23 = SQRT((CJ(1,3)-CJ(1,2))**2 + (CJ(2,3)-CJ(2,2))**2
     38          *                                 + (CJ(3,3)-CJ(3,2))**2)
     39           SL13 = SQRT((CJ(1,3)-CJ(1,1))**2 + (CJ(2,3)-CJ(2,1))**2
     40          *                                 + (CJ(3,3)-CJ(3,1))**2)
     41           X3   = (SL13**2+SL12**2-SL23**2)/(2.0*SL12)
     42           Y3   = SQRT(SL13**2-X3**2)
     43           IF (NAMEK .EQ. 6HK1    ) GO TO 110
     44                                                                  NERROR=2
     45           GO TO 999
     46      C
     47      C  K2ST1 = LINEAR DISPLACEMENT FIELD.
     48        110 CALL K2ST1  (SL12,X3,Y3,ST,S,KS)
     49           CALL DCOS2  (CJ,EJ,WI,KCJ,KEJ,KWI)
     50      C  SELECT DZ ROWS.
     51           DO 210 J=1,9
     52           DO 210 I=1,3
     53        210 WI(I,J) = 0.0
     54           DO 215 J=1,3
     55           WI(I,J) = WI(10,J)
```

```
      W1(2,J+3) = W1(13,J+6)
  215 W1(3,J+6) = W1(16,J+12)
      CALL BTABA  (S,W1, 3,9, KS,KW1)
      RETURN
C
  999 CALL ZZBOMB (6HSTF25T,NERROR)
      END
```

```
@PRT  F1,STF35T
```

```
PHILIPBIN207*F1.STF3ST
      1                COMPILER (XM=1),(EQUIV=CMN)
      2                SUBROUTINE STF3ST (CJ,EJ,ST,NAMEK,S,W1,W2,KCJ,KEJ,KS,KW1,KW2)
      3                DIMENSION CJ(KCJ,1),EJ(KEJ,1),S(KS,1),W1(KW1,1),W2(KW2,1)
      4                DIMENSION CW(3,3), EW(3,3), IV1(9), IV2(9), IV3(9), IV4(9)
      5                DATA IV1/ 1, 2, 3, 4, 5, 6, 7, 8, 9/,
      6               *     IV2/ 1, 2, 3, 7, 8, 9,10,11,12/,
      7               *     IV3/ 1, 2, 3, 4, 5, 6,10,11,12/,
      8               *     IV4/ 4, 5, 6, 7, 8, 9,10,11,12/
      9            C
     10            C  SUBROUTINE TO CALCULATE FINITE ELEMENT...
     11            C     STIFFNESS MATRIX
     12            C  FOR A SURFACE TENSION QUADRILATERAL ELEMENT WITH
     13            C  UNRESTRAINED BOUNDARIES.
     14            C  STIFFNESS MATRIX IS IN GLOBAL COORDINATE DIRECTIONS.
     15            C  GLOBAL COORDINATE ORDER IS
     16            C     (U,V,W) JOINT 1, THEN JOINT 2, 3, 4.
     17            C  WHERE U,V,W ARE TRANSLATIONS.
     18            C  EULER ANGLE CONVENTION IS GLOBAL X,Y,Z PERMUTATION.
     19            C  CALLS FORMA SUBROUTINES REVADD,STF2ST,ZZBOMB.
     20            C  DEVELOPED BY  RL WOHLEN.  FEBRUARY 1975.
     21            C
     22            C     SUBROUTINE ARGUMENTS
     23            C  CJ       = INPUT  MATRIX OF GLOBAL X,Y,Z COORDINATES AT QUAD JOINTS.
     24            C                    ROWS 1,2,3 CORRESPOND TO X,Y,Z COORDINATES.
     25            C                    COLS 1,2,3,4 CORRESPOND TO JOINTS 1,2,3,4. SIZE(3,4).
     26            C  EJ       = INPUT  MATRIX OF EULER ANGLES (DEGREES) AT QUAD JOINTS.
     27            C                    ROWS 1,2,3 CORRESPOND TO GLOBAL X,Y,Z PERMUTATION.
     28            C                    COLS 1,2,3,4 CORRESPOND TO JOINTS 1,2,3,4. SIZE(3,4).
     29            C  ST       = INPUT  SURFACE TENSION (FORCE/LENGTH).
     30            C  NAMEK    = INPUT  TYPE OF STIF MATRIX WANTED.
     31            C                    = K1, USES 4 TRIANGLES, OVERLAP AVERAGE.
     32            C  S        = OUTPUT STIFFNESS MATRIX SIZE(12,12).
     33            C  W1       = INPUT  WORKSPACE MATRIX. SIZE(9,9).
     34            C  W2       = INPUT  WORKSPACE MATRIX. SIZE(18,18).
     35            C  KCJ      = INPUT  ROW DIMENSION OF CJ IN CALLING PROGRAM. MIN=3.
     36            C  KEJ      = INPUT  ROW DIMENSION OF EJ IN CALLING PROGRAM. MIN=3.
     37            C  KS       = INPUT  ROW DIMENSION OF S IN CALLING PROGRAM. MIN=12.
     38            C  KW1      = INPUT  ROW DIMENSION OF W1 IN CALLING PROGRAM. MIN=9.
     39            C  KW2      = INPUT  ROW DIMENSION OF W2 IN CALLING PROGRAM. MIN=18.
     40            C
     41                                                                    NERROR=1
     42                IF (KS .LT. 12 .OR. KW1 .LT. 9 .OR. KW2 .LT. 18) GO TO 999
     43                DO 5 J=1,12
     44                DO 5 I=1,12
     45              5 S(I,J) = 0.0
     46                IF (NAMEK .EQ. 6HK1     ) GO TO 110
     47                                                                    NERROR=2
     48                GO TO 999
     49            C
     50          110 DO 200 I=1,3
     51                CW(I,1) = CJ(I,1)
     52                EW(I,1) = EJ(I,1)
     53                CW(I,2) = CJ(I,2)
     54                EW(I,2) = EJ(I,2)
     55                CW(I,3) = CJ(I,3)
```

```
      200 EW(I,3) = EJ(I,3)
          CALL STF2ST (CW,EW,ST,NAMEK,W1,W2,3,3,KW1,KW2)
          CALL REVADD (.5,W1,IV1,IV1,S, 9,9,12,12, KW1,KS)
          DO 201 I=1,3
          CW(I,1) = CJ(I,1)
          EW(I,1) = EJ(I,1)
          CW(I,2) = CJ(I,3)
          EW(I,2) = EJ(I,3)
          CW(I,3) = CJ(I,4)
      201 EW(I,3) = EJ(I,4)
          CALL STF2ST (CW,EW,ST,NAMEK,W1,W2,3,3,KW1,KW2)
          CALL REVADD (.5,W1,IV2,IV2,S, 9,9,12,12, KW1,KS)
          DO 203 I=1,3
          CW(I,1) = CJ(I,1)
          EW(I,1) = EJ(I,1)
          CW(I,2) = CJ(I,2)
          EW(I,2) = EJ(I,2)
          CW(I,3) = CJ(I,4)
      203 EW(I,3) = EJ(I,4)
          CALL STF2ST (CW,EW,ST,NAMEK,W1,W2,3,3,KW1,KW2)
          CALL REVADD (.5,W1,IV3,IV3,S, 9,9,12,12, KW1,KS)
          DO 205 I=1,3
          CW(I,1) = CJ(I,2)
          EW(I,1) = EJ(I,2)
          CW(I,2) = CJ(I,3)
          EW(I,2) = EJ(I,3)
          CW(I,3) = CJ(I,4)
      205 EW(I,3) = EJ(I,4)
          CALL STF2ST (CW,EW,ST,NAMEK,W1,W2,3,3,KW1,KW2)
          CALL REVADD (.5,W1,IV4,IV4,S, 9,9,12,12, KW1,KS)
          RETURN
C
      999 CALL ZZBOMB (6HSTF3ST,NERROR)
          END
```

@PRT FI.SURFTN

```
PHILIPBIN207*FI.SURFTN
      1              COMPILER (XM=I),(EQUIV=CMN)
      2              SUBROUTINE SURFTN (XYZ,JDOF,EUL,NUTEL,NJ,
      3             *                   NUTKX,V,T,S,KX,KJ,KE,KW)
      4              DIMENSION XYZ(KX,1),JDOF(KJ,1),EUL(KE,1),W(KW,1),T(KW,1),S(KW,1)
      5              DIMENSION CJ(3,4),CJ(3,4),IVI(12)
      6              DATA NIT,NOT/ 5,6 /
      7              DATA NAMEL/6HSURFTN/, IBLNK/6H      /, KCJ/3/
      8        C
      9        C  SUBROUTINE TO CALCULATE (ON OPTION) FINITE ELEMENT ...
     10        C     STIFFNESS MATRICES AND IVECS (ON NUTKX),
     11        C  FOR SURFACE TENSION ELEMENTS. TRIANGULAR (JOINT 4 = 0) OR
     12        C  QUADRILATERAL (JOINT 4 .GT. 0).
     13        C  STIFFNESS MATRICES ARE IN GLOBAL COORDINATE DIRECTIONS.
     14        C  GLOBAL COORDINATE ORDER IS
     15        C     (U,V,W) JOINT 1, THEN JOINT 2,3,(4).
     16        C  WHERE U,V,W ARE TRANSLATIONS.
     17        C  IVEC GIVES ELEMENT DOF INTO GLOBAL DOF. EXAMPLES...
     18        C     IVEC(6)=834 PLACES ELEMENT DOF 6 INTO GLOBAL DOF 834.
     19        C     IVEC(3)=0   OMITS ELEMENT DOF 3 FROM GLOBAL DOF. THIS CONSTRAINS
     20        C                 ELEMENT DOF 3 TO ZERO MOTION.
     21        C  DATA ARRANGEMENT ON NUTKX FOR EACH FINITE ELEMENT IS (W=K)
     22        C     WRITE (NUTWX) NAMEW,NEL,NR,NC,NAMEL,(IBLNK,I=1,5),
     23        C                   ((W(I,J),I=1,NR),J=1,NC),(IVEC(I),I=1,NC)
     24        C  CALLS FORMA SUBROUTINES PAGEHD,STF2ST,STF3ST,ZZBOMB.
     25        C  DEVELOPED BY RL WOHLEN. FEBRUARY 1975.
     26        C
     27        C  ***************************************************************
     28        C  INPUT DATA READ IN THIS SUBROUTINE FROM NUTEL. IF NUTEL = NIT, DATA I
     29        C  READ FROM CARDS.
     30        C     NAMEK                                          FORMAT (A6)
     31        C     ST                                             FORMAT (5X,E10)
     32        C  20 NEL,J1,J2,J3,J4                                FORMAT (5I5)
     33        C     IF (J1 .EQ. 0) RETURN
     34        C     GO TO 20
     35        C
     36        C  DEFINITION OF INPUT VARIABLES.
     37        C  NAMEK  = TYPE OF STIFFNESS MATRIX WANTED.
     38        C           = K1, LINEAR DISPLACEMENT ASSUMED.
     39        C           = 6H       OR 6HNOSTIF, NO STIFFNESS MATRIX CALCULATED.
     40        C  ST     = SURFACE TENSION (FORCE/LENGTH).
     41        C  NEL    = FINITE ELEMENT NUMBER. FOR REFERENCE ONLY, NOT USED IN
     42        C           CALCULATIONS. WRITTEN ON NUTKX.
     43        C  J1     = JOINT NUMBER AT ELEMENT VERTEX 1.
     44        C  J2     = JOINT NUMBER AT ELEMENT VERTEX 2.
     45        C  J3     = JOINT NUMBER AT ELEMENT VERTEX 3.
     46        C  J4     = JOINT NUMBER AT ELEMENT VERTEX 4. (USED FOR QUADRILATERAL).
     47        C  THE ELEMENT MAY BE NUMBERED CLOCKWISE OR COUNTER-CLOCKWISE.
     48        C
     49        C  EXPLANATION OF INPUT FORMATS. NUMBER INDICATES CARD COLUMNS USED.
     50        C     I = INTEGER DATA, RIGHT ADJUSTED.
     51        C     E = DECIMAL POINT DATA, ANYWHERE IN FIELD. EXPONENT RIGHT ADJUSTED
     52        C     X = CARD COLUMNS SKIPPED.
     53        C  ***************************************************************
     54        C
     55        C     SUBROUTINE ARGUMENTS (ALL INPUT)
```

```
      C   XYZ     = MATRIX OF JOINT GLOBAL X,Y,Z LOCATIONS. ROWS CORRESPOND
      C             TO JOINT NUMBERS. COLUMNS 1,2,3 CORRESPOND TO THE JOINT
      C             X,Y,Z LOCATIONS RESPECTIVELY. SIZE(NJ,3).
      C   JDOF    = MATRIX OF JOINT GLOBAL DEGREES OF FREEDOM. ROWS CORRESPOND
      C             TO JOINT NUMBERS. COLUMNS 1,2,3 CORRESPOND TO THE JOINT
      C             TRANSLATION DOFS AND COLUMNS 4,5,6 CORRESPOND TO THE JOINT
      C             ROTATION DOFS. SIZE(NJ,6).
      C   EUL     = MATRIX OF JOINT EULER ANGLES (DEGREES). ROWS CORRESPOND
      C             TO JOINT NUMBERS. COLUMNS 1,2,3 CORRESPOND TO THE
      C             GLOBAL X,Y,Z PERMUTATION. SIZE(NJ,3).
      C   NUTEL   = LOGICAL NUMBER OF TAPE CONTAINING ELEMENT INPUT DATA FOR
      C             THIS SUBROUTINE. IF NUTEL = NIT, DATA IS READ FROM CARDS.
      C   NJ      = NUMBER OF JOINTS OR ROWS IN MATRICES (XYZ), (JDOF), (EUL).
      C   NUTKX   = LOGICAL NUMBER OF UTILITY TAPE ON WHICH ELEMENT
      C             STIFFNESS MATRICES AND IVECS ARE OUTPUT.
      C             NUTKX MAY BE ZERO IF STIFFNESS MATRIX IS NOT FORMED.
      C             USES FORTRAN READ, WRITE.
      C   W       = MATRIX WORK SPACE. MIN SIZE(12,12).
      C   T       = MATRIX WORK SPACE. MIN SIZE( 9, 9).
      C   S       = MATRIX WORK SPACE. MIN SIZE(18,18).
      C   KX      = ROW DIMENSION OF XYZ IN CALLING PROGRAM.
      C   KJ      = ROW DIMENSION OF JDOF IN CALLING PROGRAM.
      C   KE      = ROW DIMENSION OF EUL IN CALLING PROGRAM.
      C   KW      = ROW DIMENSION OF W, T, AND S IN CALLING PROGRAM. MIN=18.
      C
     1001 FORMAT (A6)
     1002 FORMAT (5X,E10.3)
     1003 FORMAT (5I5)
     2001 FORMAT (//25X 41HINPUT DATA FOR SURFACE TENSION STIFFNESS
     *                  36H(TRIANGLE OR QUADRILATERAL) ELEMENTS)
     2002 FORMAT (//20X 41HINPUT DATA FOR SURFACE TENSION STIFFNESS
     *                  48H(TRIANGLE OR QUADRILATERAL) ELEMENTS (CONTINUED))
     2003 FORMAT (/12X7HSTIF = A6, /15X4HST = 1PE10.4,
     *          //15X7HELEMENT 13X7HJOINT 1 13X7HJOINT 2 13X7HJOINT 3
     *                       13X7HJOINT 4
     *           /15X6HNUMBER)
     2004 FORMAT (18X,5(I5,15X))
      C
           NLINE = 0
           CALL PAGEHD
           WRITE (NOT,2001)
           READ (NUTEL,1001) NAMEK
           READ (NUTEL,1002) ST
           WRITE (NOT,2003) NAMEK,ST
      C
        20 READ (NUTEL,1003) NEL,J1,J2,J3,J4
           IF (J1 .LE. 0) RETURN
           NLINE = NLINE + 1
           IF (NLINE .LE. 42) GO TO 30
           CALL PAGEHD
           WRITE (NOT,2002)
           WRITE (NOT,2003) NAMEK,ST
           NLINE = 0
        30 WRITE (NOT,2004) NEL,J1,J2,J3,J4
                                                                    NERROR=1
           IF (J1.GT.NJ .OR. J2.GT.NJ .OR. J3.GT.NJ .OR. J4.GT.NJ) GO TO 999
```

```
      C
      C  FORM FINITE ELEMENT COORDINATE LOCATIONS,EULER ANGLES, REVADD IVEC.
            DO 42 I=1,3
            CJ(I,1) = XYZ(J1,I)
            CJ(I,2) = XYZ(J2,I)
            CJ(I,3) = XYZ(J3,I)
            EJ(I,1) = EUL(J1,I)
            EJ(I,2) = EUL(J2,I)
            EJ(I,3) = EUL(J3,I)
            IVI(I  ) = JDOF(J1,I)
            IVI(I+3) = JDOF(J2,I)
         42 IVI(I+6) = JDOF(J3,I)
            IF (J4 .GT. 0) GO TO 44
            NCOL = 9
            CALL STF2ST (CJ,EJ,ST,NAMEK,W,S,KCJ,KCJ,KW,KW)
            GO TO 110
      C
         44 DO 45 I=1,3
            CJ(I,4) = XYZ(J4,I)
            EJ(I,4) = EUL(J4,I)
         45 IVI(I+9) = JDOF(J4,I)
            NCOL = 12
            CALL STF3ST (CJ,EJ,ST,NAMEK,W,T,S,KCJ,KCJ,KW,KW,KW)
      C
        110 IF (NAMEK .EQ. 6H        .OR. NAMEK .EQ. 6HNOSTIF) GO TO  20
                                                                NERROR=2
            IF (NUTKX .LE. 0) GO TO 999
            WRITE (NUTKX) NAMEK,NEL,NCOL,NCOL,NAMEL,(IBLNK,I=1,5),
           *     ((W(I,J),I=1,NCOL),J=1,NCOL), (IVI(I),I=1,NCOL)
      C
            GO TO 20
      C
        999 CALL ZZBOMB (6HSURFTN,NERROR)
            END
```

```
@PRT  F1.MAPCDS
```

6.2 Computer Program Input Requirements

Input requirements for the Free Surface Static Equilibrium Shape Computer Program are given in Section 6.2.1. Input requirements for the Vibration Analysis Computer Program are given in Section 6.2.2.

6.2.1 Input Requirements - Free Surface Static Equilibrium Shape Program - A description of the input requirements to the free surface static equilibrium shape program along with listings of sample input are presented in this section. The first three cards required are to satisfy subroutine START (as explained in Reference (3)). Data input to either the search mode or the survey mode is accomplished by means of the NAMELIST facility available in Univac 1108 Fortran V. Section 6.4 of Reference (6) contains a detailed explanation of this facility. Input values which must be specified for the survey mode are:

ACOFO — the base value for the A sweep, i.e., the first A value used will be ACOFO+DACOF

BONDNO — the nondimensional ratio of inertial forces to surface tension forces, based on container length

DACOF — the increment applied to ACOFO to generate successive values of A

DELTAT — the arc length increment used in the numerical integration algorithm to generate a solution trajectory

NA — the number of A values to be computed, i.e., NA trajectories will be generated ranging from ACOFO+DACOF to ACOFO+NA*DACOF

IPRNT — trajectory results from the numerical integration will be printed every IPRNT integration intervals, ignored if PRINT=.FALSE.

NX — the number of tank axis intercepts used to generate solutions from a given trajectory (A value), i.e., for a given value of A NX solutions will be generated ranging from XUP-DX to XLO+DX where DX=(XUP-XLO)/NX

PRINT — controls the printing of intermediate results, PRINT=.TRUE. prints all intermediate results, PRINT=.FALSE. prints only the solution summary (Figure 3-1)

RMAX — the maximum radius of the container

SEARCH — controls the selection of program mode, SEARCH=.FALSE. for the survey mode

XLO — specifies the lower limit of the tank axis intercept for which solutions are generated from a given trajectory

XMAX — length of the container measured along the axis of symmetry

XUP — specifies the upper limit of the tank axis intercept for which solutions are generated from a given trajectory

Figure 6-2 shows a listing of a sample problem input data for the survey mode. Note that BONDNO is not given in the input data, thus the default value specified in the computer program (Section 6.1.1) is used.

Input values which must be specified for the search mode are:

ACOFO — initial value of A

BONDNO — same description as survey mode

DACOF — initial increment to be applied to ACOFO, the program tries both ACOFO+DACOF and ACOFO-DACOF in searching for an improved solution; if none is found, DACOF is halved and the search repeated. The value of A corresponding to the desired solution must lie in the range ACOFO$\pm$DACOF for this procedure to work

DELTAT — same description as survey mode

EPSC — value of the error function at which convergence is established

IPRNT — same description as survey mode

NX — same description as survey mode

PHID — the desired value of contact angle in degrees

PRINT — same description as survey mode

RMAX — same description as survey mode

SEARCH — same description as survey mode, however, SEARCH=.TRUE. for the search mode

ULPCT — desired value of ullage volume percentage

XLO — same description as survey mode

XMAX — same description as survey mode

XUP — same description as survey mode

Figure 6-3 shows a listing of a sample problem input data for the search mode. Note that SEARCH is not given in the input data, thus the default value specified in the computer program (Section 6.1.1) is used.

Multiple runs may be made in either mode by repeating the cards required by subroutine START and the NAMELIST data as many times as desired. The run is terminated when START reads the word STOP in the run number field of the first card.

```
SVYTUG
SAMPLE SURVEY MODE RUN FOR
TUG-LIKE TANK
 $INDATA
      ACOF0=0.110D00,
      DACOF=-0.010D00,
      DELTAT=0.20D00,
      NA=15,
      NX=40,
      PRINT=.FALSE.,
      RMAX=32.2D00,
      SEARCH=.FALSE.,
      XLO=0.0D00,
      XMAX=146.5D00,
      XUP=146.5D00,
 $END
STOP
```


Figure 6-2 SAMPLE PROBLEM INPUT DATA - FREE SURFACE STATIC EQUILIBRIUM SHAPE PROGRAM, SURVEY MODE

```
BONDTUG   WARNER
TUG-LIKE CONTAINER, BOND NUMBER = 1.0, CONTACT ANGLE = 0.3        115
ULLAGE VOLUME = 80.0 PCT
 $INDATA
       ACOF0=0.0615625D00,
       BONDNO=1.0D00,
       DACOF=0.00003125D00,
       DELTAT=0.20D00,
       EPSC=0.8D-05,
       NX=50,
       PHID=0.3000,
       PRINT=.TRUE.,
       RMAX=32.2D00,
       ULPCT=80.0D00,
       XLO=46.0D00,
       XMAX=146.5D00,
       XUP=48.0,
 $END
STOP
```


Figure 6-3 SAMPLE PROBLEM INPUT DATA - FREE SURFACE STATIC EQUILIBRIUM SHAPE PROGRAM, SEARCH MODE

6.2.2 Input Requirements - Vibration Analysis Program - An explanation of the input to the vibration analysis program, along with a listing of input data to a sample problem (see Figure 6-4) are given here. Input formats to subroutines START, READ and READIM are explained in Reference (3).

Card No. (Ref. Figure 6-4)	Input	
1	Run no., cols. 1-6; name, cols. 11-28	Three cards to satisfy subroutine "START"
2	Title 1, cols. 1-78	
3	Title 2, cols. 1-78	
4	'INITIL' or 'NOINIT', cols. 1-6	To initialize or not to initialize the reserve tape
5	Number of sectors in 90° model	Format (10X,I5)
6	Matrix name (XYTZ), no. of grid points, no. of cols. (3).	Subroutine READ
7-30	Grid point X, Y coordinates and θ_z values	
31	Ten zeros	
32	Z coordinate of longitudinal axis	Format (10X,E10)
33	Grid point of corner if container is a cylinder	Format (10X,I5)
34	Matrix name (GP-CW), no. of rows (1), no. of grid points on container wall	Subroutine READIM
35	Container wall grid point numbers	
36	Ten zeros	
37	Matrix name (GP-FS), no. of rows (1), no. of grid points on fluid surface	Subroutine READIM
38	Fluid surface grid point numbers	
39	Ten zeros	
40	Mass option; M1 for lumped, M2 for consistent	Format (A2)
41	Fluid mass density, scale factor x bulk modulus, surface tension	Format (3(5X,E10))
42	Acceleration	Format (5X,E10)
43-57	Element number, grid point numbers for elements (CW numbering)	Format (4I5)
58	Ten zeros	
59	Mode calculation option; MODES for large sparse, MODED for small dense	Format (A5)

60	No. of modes wanted	Format (10X,I5)
61	No. of modes used	Format (10X,I5)
62	Shift value for ω^2 (convergence will be about this value)	Format (10X,E17.0)
63	No. of maximum iteration allowed (cards 60-63 are omitted if MODED option used)	Format (10X,I5)
64	Plot option; PLOT or NOPLOT (if NOPLOT, omit cards 65-87)	Format (A6)
65	First mode plotted (usually 1)	Format (10X,I5)
66	Last mode plotted ($\leq$ no. of modes used)	Format (10X,I5)
67	No. of plot views	Format (10X,I5)
68	Stereo plot option (1 for stereo, 0 if not)	Format (10X,I5)
69	No. of tracings of line	Format (10X,I5)
70	Print option (1 = yes, 0 = no)	Format (10X,I5)
71	Plot title	Format (13A6)
72	Elements plotted; GRAVTY plots surface joints, FLUID plots all joints	Format (A6)
73	No. of view positions	Format (10X,I5)
74	Roll angle	Format (10X,E10)
75	Matrix name (COELOC), no. of rows (1), no. of cols. (3)	Subroutine READ
76	X, Y, Z coordinates of center of eyes	
77	Ten zeros	
78	Matrix name (VPLOC), no. of rows (1), no. of columns (3)	Subroutine READ
79	X, Y, Z coordinates of viewpoint	
80	Ten zeros	
81	Read plot data option (1 = yes, 0 = no)	Format (10X,I5)
82	Cross-section option (1 = yes, 0 = no)	Format (10X,I5)
83	Read modal data option (1 = yes, 0 = no)	Format (10X,I5)
84	Actual mode number of first mode calculated	Format (10X,I5)
85	Scale factor on modal displacements	Format (10X,E10)

86	Option on superposition of undeformed and deformed joints (1=yes, 0=no)	Format (10X,I5)
87	Symmetry option. Use XSA in cols. 15-17	
88	Plot option; PLOT or NOPLOT (if PLOT repeat cards 65-87)	Format (A6)
89	End of data. STOP in cols. 1-4	

```
G2/4-S     RL WOHLEN                                                    119
LATERAL SLOSH. GRID 2, 4 SECTORS. SPARSE. BKM*E-8.
IUS STRETCHED TRANSTAGE OXTOTZER (NITROGEN TETROXIDE) TANK. BN=1. ULL VOL=80
INITTL
NSECT          4
XYTZ     24    3
     1     1    0.                  0.                  90.
     2     1   18.                  0.
     3     1   30.                  0.
     4     1   40.                  0.
     5     1   47.4                 0.
     6     1   27.5                15.8                 30.
     7     1   36.5                13.0
     8     1   43.3                11.2
     9     1   49.1                10.0
    10     1   40.7                23.5                 30.
    11     1   44.5                21.5
    12     1   48.0                19.7
    13     1   52.4                17.0
    14     1   47.0                27.0                 25.
    15     1   51.7                28.9                 20.
    16     1   54.5                25.0
    17     1   57.3                23.0
    18     1   62.0                31.5                  6.
    19     1   62.9                29.9
    20     1   64.0                27.9
    21     1   66.5                32.0                  3.
    22     1   71.4                32.0                  0.0
    23     1   71.6                31.0
    24     1   79.4                32.0                  0.0
0000000000
Z-AXIS         0.0
NOT CYL        0
GP-CW     1    9
     1     1    1    6   10   14   15   18   21   22   24
0000000000
GP-FS     1    7
     1     1    5    9   13   17   20   23   24
0000000000
M2
  RO= 1.36 E-04 BKM= 1.31 E-03  ST= 1.56 E-04
  GX=-1.12 E-03
     1     1    2    6
     2     2    3    7    6
     3     3    4    8    7
     4     4    5    9    8
     5     6    7   11   10
     6     7    8   12   11
     7     8    9   13   12
     8    10   11   15   14
     9    11   12   16   15
    10    12   13   17   16
    11    15   16   19   18
    12    16   17   20   19
    13    18   19   22   21
    14    19   20   23   22
    15    22   23   24
0000000000
MODES
NW             3
```

Figure 6-4 SAMPLE PROBLEM INPUT DATA - VIBRATION ANALYSIS PROGRAM

```
      120
NU                6
SHIFT             3.0E-6
MAXIT            20
PLOT
MN START          1
MN END            6
NVIEWS            1
IF STER           0
NTRACE            1
IF PRINT          0
IUS OX TANK. BN=1. ULL VOL = 80. GRID 2, 4 SECTORS.
GRAVTY
N VIEW PT         1
ROLL ANGLE      180.
COELOC    1       3
    1     1     70.                0.              -200.
0000000000
VPLOC     1       3
    1     1     50.                0.                0.
0000000000
IF READ PD        0
CROSS SECT        0
IF READ MD        0
REAL MN1          1
SCALE FACT        5.
IF SUPRPOS        1
SYM               XSA
NOPLOT
STOP
```


Figure 6-4 SAMPLE PROBLEM INPUT DATA VIBRATION ANALYSIS PROGRAM (cont'd)