

NOT IN

N71-14226

CR-111731

~~NOT IN 170 37812~~

Digital Analysis of Liquid Sloshing
In Rotationally Symmetric Tanks
Under Weak Gravitational Fields

CASE FILE
COPY

**Digital Analysis of Liquid Sloshing
In Rotationally Symmetric Tanks
Under Weak Gravitational Fields**

Volume II

COMPUTATIONAL APPLICATION

by

**T. S. Chandler and L. L. Fontenot
UNIDEV, INC.
Huntsville, Alabama**

March 1970

**Contract No. NAS8-21272
UNIDEV Report UR-00010**

Submitted to

**George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Huntsville, Alabama 35812**

DIGITAL ANALYSIS OF LIQUID SLOSHING
IN ROTATIONALLY SYMMETRIC TANKS
UNDER WEAK GRAVITATIONAL FIELDS

Volume II

COMPUTATIONAL APPLICATION

by

T. S. Chandler and L. L. Fontenot
UNIDEV, INC.
Huntsville, Alabama

March 1970

Contract No. NAS8-21272

UNIDEV Report UR-00010

Submitted to

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Huntsville, Alabama 35812

FOREWORD

This report was prepared by UNIDEV, INC., Huntsville, Alabama, under Contract NAS8-21272 for the George C. Marshall Space Flight Center of the National Aeronautics and Space Administration. The work was administered under technical direction of the Aero-Astrodynamic Laboratory, George C. Marshall Space Flight Center, with Mr. Frank Bugg, acting as Project Manager.

This document deals with the computational aspects of the problem. Theoretical considerations are presented in Volume I.

Technical contributions and helpful suggestions were made by Dr. D. O. Lomen, University of Arizona.

This study was executed under the direction of Dr. L. L. Fontenot, Program Manager and Principal Investigator; Mr. T. S. Chandler developed the Digital Computer Program described herein.

ABSTRACT

This document describes the Digital Computer Program used to determine eigenfrequencies, eigenfunctions, forces, and moments for propellant "sloshing" in rotationally symmetric tanks under weak gravitational fields. The description of the analysis is given in /1/.

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	ix
<u>Sections</u>	
1 Synthesis of Fundamental Equations	1
2 Program Description	8
3 Utilization of the Computer Program	9
3.1 Description of Input	9
3.2 Description of Output	14
4 References	18
5 Sample Input	19
6 Sample Output	21
7 Program Listing	

LIST OF ILLUSTRATIONS

<u>Figure No.</u>		<u>Page</u>
1	Rotationally Symmetric Vessel, Notation	4
2	Coordinate System	9
3	Normal Orientations for Tank Geometries	12
4	Toroid Geometry	14

NOTATIONS*

η	Total free surface displacement.
η'	Free surface displacement in the absence of surface tension.
ζ	Free surface displacement due to surface tension and contact angle effects.
$(\delta \bar{F}_{01})_y$	Total force resulting from action of liquid on the tank wall(s) in the y(or x_2)-direction.
F_2'	Force resulting from action of liquid on the tank wall(s) in the x_2 -direction, in the absence of surface tension.
F_y	Force resulting from action of liquid on the tank wall(s) in the y(or x_2)-direction, due to surface tension and contact angle effects.
$(\delta \bar{F}_{01})_z$	Total force resulting from action of liquid on the tank wall(s) in the z(or x_3)-direction.
F_3'	Force resulting from action of liquid on the tank wall(s) in the x_3 -direction, in the absence of surface tension.
F_z	Force resulting from action of liquid on the tank wall(s) in the x_3 -direction, due to surface tension and contact angle effects.
$(\delta \bar{N}_{01})_x$	Total moment of the forces exerted on the tank wall(s) about the x(or x_1) axis.
T_1'	Moment of the forces exerted on the tank wall(s) about the x_1 -axis, in the absence of surface tension.
T_y	Moment of the forces exerted on the tank wall(s) about the x(or x_1) axis, due to surface tension and contact angle effects.
q_n	Time dependent amplitudes of free surface displacement Fourier expansion.

*

See /2/ for notations not given herein.

ξ_n	Time dependent amplitudes of free surface displacement Fourier expansion, in the absence of surface tension.
λ_n	Time dependent amplitudes of free surface displacement Fourier expansion, corresponding to surface tension and contact angle effects.
α_2	Acceleration in the x_2 -direction.
α_3	Acceleration in the x_3 -direction (assumed constant).
$\ddot{D}$	Angular acceleration of tank about the x_2 -axis.
M	Mass of the liquid.
B_0	Bond Number = $\rho \alpha_3 a^2 / \sigma$
V	Undisturbed volume of the liquid.
σ	Surface tension.
ρ	Density of the liquid.
a	Distance from point B to the z-axis (see Fig. 1).
r, θ	Polar coordinates.
γ	Contact angle.
L	Distance from the center of gravity of the liquid to the undisturbed free surface, in the absence of surface tension.
L_1	Distance from the center of gravity of the liquid to an arbitrary point along an extension of the line AC (see Fig. 1), in the absence of surface tension.
β	Angle between the tangent line and local horizontal (see Fig. 1).
Φ_n	Antisymmetric eigenfunctions associated with the "high-g" eigenoscillation problem.
ω_n	Eigenfrequencies
K_n	Non-dimensional parameter = $L \omega_n^2 / \alpha_3$

$b_n, h_n, \gamma_n, I_{11}$

Liquid parameters associated with the "high-g" slosh problem / 2 / defined in (1.8).

$A_{nm}^*, B_{nm}^*, a_n^*$

b_n^*, c_n^*, e_n^*

f_n^*, g_n^*, h_n^*

Additional liquid parameters resulting from surface tension and contact angle effects defined in (1.4) and (1.5).

ℓ_n^*, B, β_{nm}

$\bar{\beta}_{nm}, \tau_{nm}, \nu_{nm}$

ρ_n

ψ_1

Stokes' potential determined from system (1.7).

$\frac{\omega - \omega_n}{\omega_n}$

Frequency correction = $-\beta_{nn}/2B_0$

INTRODUCTION

In /1/, the equations governing the behavior of liquids in moving tanks under conditions of weightlessness and weak gravitational fields were derived.

The objective of this document is to develop a Digital Computer Program to calculate the pertinent liquid parameters derived in / 2 / for rotationally symmetric tanks, wherein the axis of symmetry coincides with the vector of effective mass forces.

SECTION 1

SYNTHESIS OF FUNDAMENTAL EQUATIONS

The equations which govern the motion of liquids enclosed in moving tanks under weak gravitational fields are derived in /1/. In particular, the equations are specialized to rotationally symmetric vessels, wherein the axis of symmetry coincides with the vector of effective mass forces. The forces, moments and free boundary displacement are given in terms of non-dimensional coefficients, which are expressible by means of the geometry of the vessel, properties of the liquid, contact angle and parameters associated with the "high-g" eigenoscillation problem. In the notation of /2/, these equations are

$$\eta = \eta' + \zeta/B_0 ,$$

$$(\delta \bar{F}_{01})_y = F_2' + F_y/B_0 ,$$

$$(\delta \bar{F}_{01})_z = F_3' + F_z/B_0 , \tag{1.1}$$

$$(\delta \bar{N}_{01})_x = T_1' + T_x/B_0 ,$$

$$q_n = \xi_n + \lambda_n/B_0 ;$$

$$\eta' = \sin\theta \sum_{n=1}^{\infty} \Phi_n(r, L) \xi_n(t) ,$$

$$F_2' = -M\alpha_2 + ML_1(-\ddot{D}) - M \sum_{n=1}^{\infty} \gamma_n c_n^* ,$$

$$F_3' = -M\alpha_3 , \quad (1.2)$$

$$T_1' = ML_1\alpha_2 - I_{11}'(-\ddot{D}) - M \sum_{n=1}^{\infty} \gamma_n [\alpha_3 c_n^* \xi_n + a_n^* \ddot{\xi}_n] ,$$

$$\ddot{\xi}_n + \frac{\alpha_3}{L} \kappa_n \xi_n = -\kappa_n [a_n^* (-\ddot{D}) + c_n^* \alpha_3] ;$$

$$\zeta = \sin\theta \sum_{n=1}^{\infty} [\phi_n(r, L) \lambda_n(t) + \sum_{\substack{m=1 \\ n \neq m}}^{\infty} B_{nm}^* \phi_m(r, L) \xi_n(t)] ,$$

$$F_y = -M \sum_{n=1}^{\infty} \gamma_n c_n^* \ddot{\lambda}_n - M \sum_{n=1}^{\infty} \gamma_n [e_n^* \ddot{\xi}_n + \alpha_3 f_n^* \xi_n] ,$$

$$F_z = 0 , \quad (1.3)$$

$$\begin{aligned} T_x = & -B(-\ddot{D}) - M \sum_{n=1}^{\infty} \gamma_n [\alpha_3 c_n^* \lambda_n + a_n^* \ddot{\lambda}_n] \\ & - M \sum_{n=1}^{\infty} \gamma_n [(b_n^* + g_n^*) \ddot{\xi}_n + \alpha_3 (e_n^* + h_n^* - (L+L_1) f_n^*) \xi_n] , \end{aligned}$$

$$\ddot{\lambda}_n + \frac{\alpha_3}{L} \kappa_n \lambda_n = - \sum_{m=1}^{\infty} (A_{nm}^* \ddot{\xi}_m + \frac{\alpha_3}{L} \kappa_n B_{nm}^* \xi_m) - \kappa_n b_n^* (-\ddot{D}) ;$$

where

$$\begin{aligned} A_{nm}^* &= \frac{\kappa_n}{\kappa_n - \kappa_m} \bar{\beta}_{nm} + \frac{\pi a^3 \kappa_n}{V \gamma_n} v_{nm} \cos \gamma , \quad n \neq m \\ &= \frac{\pi a L^2}{V \gamma_n \kappa_n} (2\tau_{nn} - v_{nn}) \cos \gamma + \frac{\kappa_n a}{L} \left(\frac{\pi a^2 L}{V \gamma_n} v_{nn} - 2 \right) \cos \gamma , \quad n = m , \end{aligned}$$

$$B_{nm}^* = \frac{\kappa_n}{\kappa_n - \kappa_m} \beta_{nm} + \frac{\pi a^2 L}{V \gamma_n} \tau_{nm}, \quad n \neq m,$$

$$= \frac{\pi a^2 L}{V \gamma_n} \tau_{nn}, \quad n=m$$

$$a_n^* = L(b_n - h_n) - L_1 b_n,$$

$$b_n^* = \frac{\pi a L^3}{V \gamma_n \kappa_n} (\phi_n(a, L) \psi_1(a, L) - 2\rho_n) \cos \gamma +$$

$$\frac{\pi a^4}{V \gamma_n} \left(\phi_n(a, L) - \frac{2V \gamma_n}{\pi a^3} b_n \right) \cos \gamma,$$

$$c_n^* = b_n,$$

$$e_n^* = - \sum_{\substack{m=1 \\ n \neq m}}^{\infty} B_{nm}^* c_m^*,$$

(1.4)

$$f_n^* = \frac{\pi a^2}{V \gamma_n} \phi_n(a, L) \cos \gamma,$$

$$g_n^* = - \sum_{\substack{m=1 \\ n \neq m}}^{\infty} B_{nm}^* a_m^*,$$

$$h_n^* = \frac{\pi a^3}{V \gamma_n} \phi_n(a, L),$$

$$\ell_n^* = \frac{\pi a L^4}{V \gamma_n} (\phi_n(a, L) \psi_1(a, L) - 2\rho_n) (b_n + \frac{1}{2} h_n) \cos \gamma,$$

$$B = M \left[\frac{\pi a^5}{2V} \cos \gamma + 2 \sum_{n=1}^{\infty} \gamma_n \ell_n^* \right],$$

with

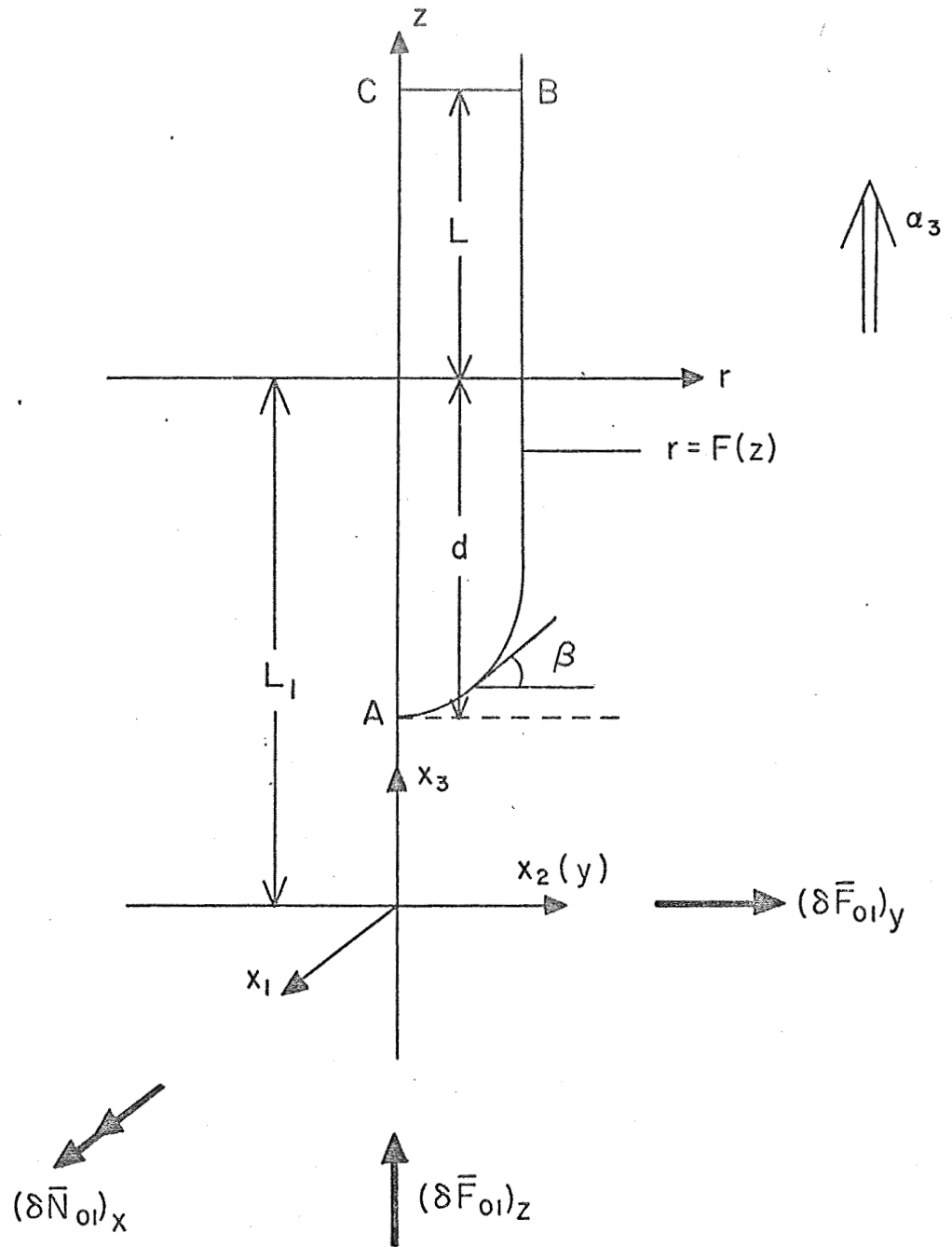


Figure 1. Rotationally Symmetric Vessel, Notation

$$\beta_{nm} = -\frac{\pi a^2 L}{V \gamma_n} \tau_{nm} - \frac{\kappa_m a}{K} \left[2 \sqrt{\frac{\gamma_m}{\gamma_n}} \delta_{nm} - \frac{2 \eta L^3}{V \gamma_n \kappa_n \kappa_m} \tau_{nm} + \frac{\pi L^3}{V \gamma_n} \left(\frac{1}{\kappa_n \kappa_m} - \frac{a^2}{L^2} \right) v_{nm} \right] \cos \gamma ,$$

$$\bar{\beta}_{nm} = \beta_{nm} \text{ with } \kappa_n \text{ replaced by } \kappa_m ,$$

$$\tau_{nm} = \int_C^B \left[\frac{\partial \phi_n(r, L)}{\partial r} \frac{\partial \phi_m(r, L)}{\partial r} + \frac{\phi_n(r, L) \phi_m(r, L)}{r^2} \right] r \, dr , \quad (1.5)$$

$$v_{nm} = \phi_n(a, L) \phi_m(a, L) ,$$

$$\rho_n = \int_C^B \left[\frac{\partial \phi_n(r, L)}{\partial r} \frac{\partial \psi_1(r, L)}{\partial r} + \frac{\phi_n(r, L) \psi_1(r, L)}{r^2} \right] r \, dr .$$

The eigenfrequencies κ_n and functions ϕ_n satisfy the boundary value problem (see Fig. 1)

$$\frac{\partial^2 \phi_n}{\partial r^2} + \frac{1}{r} \frac{\partial \phi_n}{\partial r} - \frac{1}{r^2} \phi_n + \frac{\partial^2 \phi_n}{\partial z^2} = 0, \text{ (interior to ABCOA),}$$

$$\frac{\partial \phi_n}{\partial z} = \frac{\kappa_n}{L} \phi_n , \text{ (along BC),} \quad (1.6)$$

$$\sin \beta \frac{\partial \phi_n}{\partial r} - \cos \beta \frac{\partial \phi_n}{\partial z} = 0, \text{ (along AB),}$$

while the Stokes potential is determined from

$$\frac{\partial^2 \psi_1}{\partial r^2} + \frac{1}{r} \frac{\partial \psi_1}{\partial r} - \frac{1}{r^2} \psi_1 + \frac{\partial^2 \psi_1}{\partial z^2} = 0, \text{ (interior to ABCOA),} \quad (1.7)$$

$$\sin\beta \frac{\partial \Psi_1}{\partial r} - \cos\beta \frac{\partial \Psi_1}{\partial z} = \frac{2z}{L^2} \sin\beta, \quad (\text{on ABC}).$$

Finally, the coefficients

$$\begin{aligned} b_n &= \frac{\pi}{V\gamma_n} \int_C^B r^2 \phi_n(r, L) dr, \\ h_n &= \frac{2\pi}{V\gamma_n \kappa_n} \int_A^B zr \phi_n(r, z) dz, \\ \gamma_n &= \frac{\pi L}{V} \int_C^B r [\phi_n(r, L)]^2 dr, \\ I'_{11} &= \rho \int_{UV} (r^2 \sin^2 \theta + z^2) dV - 4\rho \int_{UV} z^2 dV \\ &\quad + 2\rho L^2 \pi \int_A^B zr \Psi_1(r, z) dz + M L_1^2 \end{aligned} \tag{1.8}$$

are parameters associated with the equations of motion in the absence of surface tension, and therefore can be found once ϕ_n , κ_n and Ψ_1 are known.

In the absence of surface tension ($B_0 = \infty$), the motion of the liquid in the moving tank is completely determined by equations (1.2) whose coefficients are expressible by means of the solutions of (1.6) and (1.7) and parameters (1.8). The NASA-MSFC Digital Computer Program /2/ is available for solving (1.6) and (1.7) and calculating (1.8) for rotationally symmetric vessels. The program and computational procedure are well documented in /2/, and not repeated here.

To find the corrections due to surface tension and contact angle effects, we must compute the additional matrices and vectors in (1.3) which are expressible by means of (1.4) and (1.5). However, an examination of (1.3-5) and the Digital Computer Program /2 / shows that the required quantities can be formed readily if the additional integrations τ_{nn} and ρ_n are known. Therefore, we shall extend the Digital Computer Program to calculate these parameters and the resulting matrices and vectors called for in (1.3). In addition, we shall compute the frequency correction

$$\epsilon^2 \mu_n = - \beta_{nn} / 2B_0. \quad (1.9)$$

SECTION 2

PROGRAM DESCRIPTION

The existing program (Tank Sloshing Version 1, 3712) was modified to accept the equations described in Section 1. To make the necessary changes, five routines were added and the matrix and main routines were modified.

Gauss's Quadrature Method was used as the integration scheme and was written as a new routine, INTG3. The integrands of τ and ρ were placed in the four remaining routines.

The program was written and checked out on the IBM 7094, Mod II, Version 13.

SECTION 3

UTILIZATION OF THE COMPUTER PROGRAM

3.1 DESCRIPTION OF INPUT. There are three sets of required input data, as described in Sections 3.1.1, 3.1.2 and 3.1.4. An optional fourth set is described in Section 3.1.3. Input is subject to the restrictions outlined in Section 3.1.5.

Data is entered in allocated spaces on a coding form. Each digit of a number, a decimal point, or a minus sign occupies 1 space. Unless otherwise specified in the following sections, a number must be entered with a decimal point. A number that must not contain a decimal point must be right-adjusted. That is, all digits must occupy the last of those spaces allocated for the number.

All coordinates required as input must be in inch units, from a system in which the r-coordinate of the tank centerline is zero and the z-coordinate increases upward. The origin may be anywhere on the centerline.

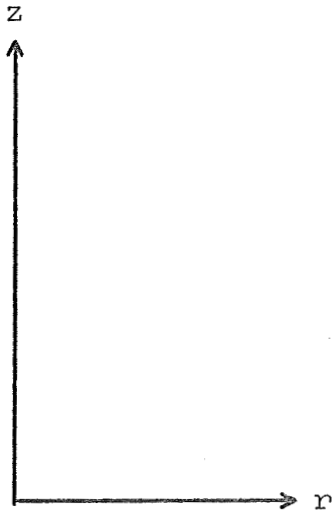


Figure 2. Coordinate System

3.1.1 Problem Input. The first three lines of the coding form contain general information about the problem. On the first line:

<u>Spaces</u>	<u>Data</u>
1 through 72	Enter any desired characters to be used as an identification title.

On the second line:

<u>Spaces</u>	<u>Data</u>
1 through 10	Contact Angle (Radians)
11 through 20	Bond Number

On the third line:

<u>Spaces</u>	<u>Data</u>
1 through 10	Enter the number of segments of the curve. The number must be right-adjusted, without a decimal point.
11 through 20	Enter the number of modes of oscillation desired. The number must be right-adjusted, without a decimal point.
21 through 30	Enter the liquid density (in kilograms per cubic meter).
31 through 40	Enter the r-coordinate (in inches) of the beginning of the segments of the curve.
41 through 50	Enter the z-coordinate (in inches) of the beginning of the segments of the curve.
51 through 60	Enter the number of baffles. The number must be right-adjusted, without a decimal point.

3.1.2 Tank Geometry Input. Beginning with the fourth line of the coding form, a line of the data below is required for each segment of the curve. Segments must be ordered in a continuous, counterclockwise path around the curve. The number of these lines of input must equal the number entered in spaces 1 through 10 of the third line of Problem Input (Section 3.1.1).

<u>Spaces</u>	<u>Data</u>
1 through 10	Enter the r-coordinate (in inches) of the end of the particular segment, having proceeded along the segment in a counterclockwise direction around the curve.
11 through 20	Enter the z-coordinate (in inches) of the end of the particular segment, having proceeded along the segment in a counterclockwise direction around the curve.

Spaces

Data

21 through 30

Leave blank for a straight line segment; otherwise, enter as follows, depending on the type of segment:

Elliptical. Enter the r-coordinate (in inches) of the center of the ellipse.

Circular. Enter the r-coordinate (in inches) of the center of the circle.

Parabolic. Enter the r-coordinate (in inches) of the vertex of the parabola.

31 through 40

Leave blank for a straight line segment; otherwise, enter as follows, depending on the type of segment:

Elliptical. Enter the z-coordinate (in inches) of the center of the ellipse.

Circular. Enter the z-coordinate (in inches) of the center of the circle.

Parabolic. Enter the z-coordinate (in inches) of the vertex of the parabola.

41 through 50

Leave blank for a straight line segment; otherwise enter, depending on the type of the segment, as follows:

Elliptical. Enter the semimajor axis (in inches) of the ellipse.

Circular. Enter the radius (in inches) of the circle.

Parabolic. Enter the directrix (in inches) of the parabola.

51 through 60

Leave blank for a straight line or parabolic segment; otherwise enter, depending on the type of the segment, as follows:

Elliptical. Enter the semiminor axis (in inches) of the ellipse.

Circular. Enter the radius (in inches) of the circle.

SpacesData

61 through 70 Leave blank for a straight line or circular segment; enter the amount of counterclockwise rotation (in degrees) from the norm (as shown in Figure 3) of the ellipse if the segment is elliptical, or of the parabola if the segment is parabolic. It may be left blank for an angle of zero degrees.

3.1.3 Baffle Input. This data is entered only if a number is entered in spaces 51 through 60 of the third line of Problem Input (Section 3.1.1). Beginning with the next line of the coding form, a line of the data below is required for each baffle. The number of these lines must equal the number entered in spaces 51 through 60 of the third line of Problem Input (Section 3.1.1).

SpacesData

1 through 10 Enter the z-coordinate (in inches) of the baffle.

11 through 20 Enter the r-coordinate (in inches) of the inner end of the baffle.

21 through 30 Enter the r-coordinate (in inches) of the outer end of the baffle.

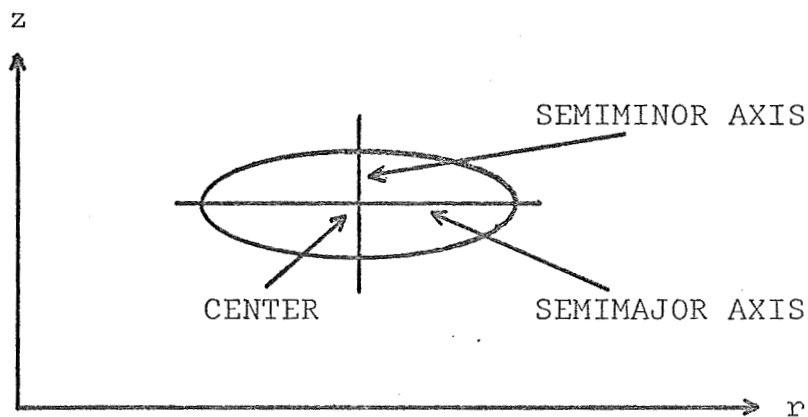
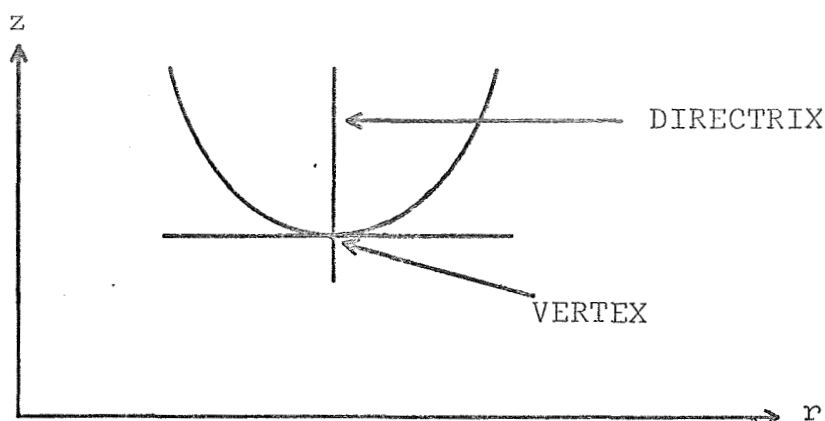
Normal Orientation of an Ellipse

Figure 3. Normal Orientations for Tank Geometries

Normal Orientation of a Parabola



3.1.4 Case Input. Each case to be run requires a line of input as listed below. Any number of cases may be specified.

<u>Space(s)</u>	<u>Data</u>
1 through 10	Enter the z-coordinate (in inches) of the undisturbed free surface.
20	Leave blank to indicate three degrees of freedom; otherwise, enter the digit 6 to indicate six degrees of freedom.
21 through 30	Leave blank to indicate no force distribution coefficients; otherwise, enter the z-coordinate (in inches) of the point dividing the liquid into an UPPER and a LOWER region of partitions.
31 through 40	Leave blank to indicate no force distribution coefficients; otherwise, enter the increment (in inches) along the z-axis between force distribution partitions in the UPPER region of liquid.
41 through 50	Leave blank to indicate no force distribution coefficients; otherwise, enter the increment (in inches) along the z-axis between force distribution partitions in the LOWER region of liquid.

3.1.5 Restrictions on Input.

1. There can be no more than 50 segments.
2. There can be no more than 30 baffles.

3. There can be no more than 5 modes of oscillation.
4. Do not enter the tank centerline as a segment unless it is a segment of the curve.
5. The tank must not be completely full.
6. Each segment must be defined such that for a given r-coordinate, there is only a single value of the z-coordinate.

i.e. A toroid must be described by 4 segments of the same circle:

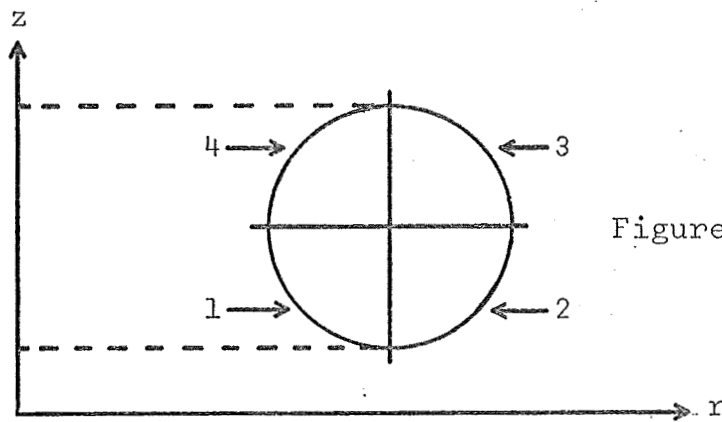


Figure 4. Toroid Geometry

3.2 DESCRIPTION OF OUTPUT. Program output consists of the following printed information:

A. Title Page

1. Name of the computer program
2. Identification title
3. Liquid density (in kilograms per cubic meter)
4. Contact Angle (radians)
5. Bond Number
6. Tank geometry - a description of all segments (in inches and degrees)
7. Lowest point in the tank (in inches)

B. Case Information (each case begins on a new page)

1. Identification title
2. Undisturbed free surface (in inches)
3. Mass of the liquid (in kg)
4. Undisturbed center of gravity (in inches)
5. Number of degrees of freedom
6. Eigenvalue statistics for each mode used - eigenvalue, and eigenvector and its normalizing factor
7. Force and moment equation nondimensional coefficients for each mode used:

K_n	γ_n	b_n	h_n
-------	------------	-------	-------

$a_n^* + L_1 b_n^*$	b_n^*	e_n^*	f_n^*
---------------------	---------	---------	---------

$g_n^* + L_1 e_n^*$	h_n^*	l_n^*
---------------------	---------	---------

$\left. \begin{array}{l} a_n (=b_n) \\ d_n \end{array} \right\}$	for 6 degrees of freedom only
--	-------------------------------

8. Maximum wave height at the outer radius of the undisturbed free surface/ x_1
 9. Kinetic energy/ $\dot{x}_1^2$ (in kg)
 10. I_{11}' and its 4 terms (in kg-m²)
 11. I_{22}' and its 4 terms (in kg-m²)
 12. I_{33}' (in kg-m²)
- | | |
|---|-------------------------------|
| } | for 6 degrees of freedom only |
|---|-------------------------------|
13. Force distribution coefficients for each partition used:

z-coordinates of partition (in inches)

α_2 coefficient (in kg), [$A_{n\alpha}$ of (5.17)]

- $\ddot{u}$ coefficient (in kg-m), [${}_n B_u$ of (5.17)]
 $\alpha_3 \xi_1$ coefficient (in kg/m), [${}_n C_\xi$ of (5.17)]
 $\alpha_3 x_1$ coefficient (in kg/m), [${}_n C_{\xi} K_1 b_1$ of (5.22)]
 $\ddot{\xi}_1$ coefficient (in kg), [Q_n of (5.24)]*
14. Undisturbed center of pressure (in inches) only if force distribution coefficients have been calculated in the LOWER region of the liquid
15. Spring-mass analogy parameters for each mode used:
- m_n (in kg)
 l_n (in inches)
 K_n^*/α_3 (in kg/m)
16. Spring-mass analogy parameters for ALL modes used:
- m_o (in kg)
 l_o (in inches)
 I_o (in kg-m²)
- 17.
- τ
 v
 β
 $\bar{\beta}$
 A^*
 B^*
18. Baffle effect on energy for each baffle used:
- Baffle z-coordinate (in inches)
 Baffle r-coordinates (in inches)
 Instantaneous dE/dt factor (in kg/m^{3/4})
 (see (4.23) of Reference 2)

* See Reference 2

Average dE/dt factor (in $\text{kg/m}^{3/4}$)
(see (4.27) of Reference 2)

C. Diagnostics

If normal program calculations cannot be continued, an appropriate diagnostic comment is printed out and either execution is terminated or continued with assumptions, or a new case is begun.

SECTION 4

REFERENCES

1. On the Motion of Liquids Enclosed in Aerospace Vehicle Tanks Under Weak Gravitational Fields, Vol. I, Theory, UNIDEV Report UR-00010, March 4, 1970.
2. D. O. Lomen, Analysis of Fluid Sloshing, General Dynamics Report GDC-DDE66-018, June 20, 1966.

SECTION 5
SAMPLE INPUT

ATLAS LØX TANK WITH RING BAFFLE
0.7854 10.0

ALSO A BAFFLE TO SHØW BAFFLE ØUTPUT

7	3	114.3363	0.0	43.5	1
60.0	0.0	0.0	00.0	60.0	43.5
60.0	46.0				
56.0	46.0				
56.0	47.0				
60.0	47.0				
60.0	75.0				
0.0	75.0				
50.0	50.0	60.0			
55.0					

SECTION 6
SAMPLE OUTPUT

LIQUID DENSITY = 0.11434E 03 KG/M**3

CONTACT ANGLE = 0.7854DE 00 RADIANS

BOND NUMBER = 0.10000E 02

TANK GEOMETRY
LENGTH UNIT IS INCHES

SEGMENT	TYPE	FROM		TO		CHARACTERISTICS		
		R =	Z =	R =	Z =	SEMI-MAJOR AXIS =	SEMI-MINOR AXIS =	ROTATED BY
1	ELLIPTICAL	R = 0.0000	Z = 43.5000	R = 60.0000	Z = 0.0000	60.0000	43.5000	-0.0 DEGREES
							0.0000 AND Z =	0.0000
2	STRAIGHT LINE	R = 60.0000	Z = 0.0000	R = 60.0000	Z = 46.0000			
3	STRAIGHT LINE	R = 60.0000	Z = 46.0000	R = 56.0000	Z = 46.0000			
4	STRAIGHT LINE	R = 56.0000	Z = 46.0000	R = 56.0000	Z = 47.6000			
5	STRAIGHT LINE	R = 56.0000	Z = 47.0000	R = 60.0000	Z = 47.0000			
6	STRAIGHT LINE	R = 60.0000	Z = 47.0000	R = 60.0000	Z = 75.0000			
7	STRAIGHT LINE	R = 60.0000	Z = 75.0000	R = 0.0000	Z = 75.0000			

LOWEST POINT IN THE TANK IS AT 0.0000 INCHES

UNDISTURBED FREE SURFACE IS AT 55.0000 INCHES

MASS OF THE LIQUID = 0.54822E+03 KG

UNDISTURBED CENTER OF GRAVITY IS AT 39.9459 INCHES

3 DEGREES OF FREQU

EIGENVALUE = 0.10398E 01

EIGENVECTOR NORMALIZING FACTOR = -0.40660E-01

EIGENVECTOR

0.56104E 00
-0.88389E 00
0.10000E 01
-0.65231E 00
0.18270E 00
0.26805E 00
-0.27167E-01
0.16994E-02
-0.94783E-03
0.10035E-02

EIGENVALUE = 0.49273E 01 EIGENVECTOR NORMALIZING FACTOR = -0.12481E 02

EIGENVECTOR

0.96838E-01
-0.52558E 00
0.10000E 01
-0.83875E 00
0.26206E 00
-0.29099E-02
0.12808E 00
-0.30529E-02
0.19078E-02
0.33567E-03

EIGENVALUE = 0.83917E-01

EIGENVECTOR NORMALIZING FACTOR = -0.74610E-01

EIGENVECTOR

0.56773E-01
-0.42327E 00
0.10000E 01
-0.97724E 00
0.34132E 00
0.30848E-02
-0.14342E-01
0.23121E 00
0.20139E-02
0.25987E-02

COEFFICIENTS

MODE

KN

GAMMA

BN

HN

1	0.2608E 00	0.33850E-01	0.81729E 01	0.32318E 01
2	0.12363E 01	0.75811E-03	-0.76397E 01	-0.39208E 01
3	0.21055E 01	0.12456E-02	0.17484E 01	0.18011E 01

COEFFICIENTS

MODE

AN+LIBN

BN*

EN*

FN*

1.	0.18894E 01	0.15029E 02	-0.24434E 02	0.12519E 02
2	-0.14220E 01	-0.16165E 03	0.26267E 03	-0.74753E 02
3	-0.20175E-01	0.13395E 03	-0.31853E 03	0.58406E 02

0.34268251E 01 0.39744631E 00 -0.30680469E 00
-0.11320369E 02 0.23476306E 02 0.27055501E 02
0.492904601E 01 -0.15082991E 02 0.72807970E 02

NATURAL FREQUENCY CORRECTION

0.4467183E-02 - 0.10075340E-01 - 0.22993736E-01
BAFFLE Z-COORDINATE
IN INCHES
R-COORDINATES
INST. DE/DT FACTOR
AVG. DE/DT FACTOR
IN KG/M** (3./4.)

50.0000 50.0000 60.0000 -0.55007E 02 -0.14476E 02

SECTION 7
PROGRAM LISTING

MAIN

- EFN

SOURCE STATEMENT

- IFN(S) -

SLOSH, LOW GRAVITY CORRECTION

DIMENSION TITLE(12),TEMP(3)

COMMON/GAMBO/GAMMAA,BD

COMMON/CON/CON1,CON2,CON3,CON4,CON5,CON6,CON7,CON8,CON9

COMMON/CCON/CCON10,CON11,CON14,CON15,CON16

COMMON /GAUSS/NORD,X(8),W(8)

COMMON/XW/XG(8),WG(8)

COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,ZBOT,IDEGR,

ZG1,G2,G3,ZDIV,DZ(2)

COMMON /BAF/NBAF,ZBAF(30),RBAF1(30),RBAF0(30)

COMMON /ELEM/MORDER,MPOLY,JN(5),NORDER,NPOLY,ACDEF(10,10)

COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS

COMMON /SEG/TYPE(50),BEGRPT,RPOINT(50),BEGZPT,ZPOINT(50),RC(50),

ZZC(50),A(50),B(50),GAMMA(50),GAM(50)

INTEGER TYPE

REAL LEVEL,L,JN

EXTERNAL F1,F2

PI=3.14159265

DO 1 I=1,NORD

XG(I)=0.5*X(I)

WG(I)=0.5*W(I)

1 CONTINUE

PROBLEM DESCRIPTION

READ (5,2) TITLE

2 FORMAT (12A6)

WRITE (6,3) TITLE

3 FORMAT(1H1,24X,12A6)

READ(5,10) GAMMAA,BD

READ (5,5) NSEG,MODES,RHO,BEGRPT,BEGZPT,NBAF

5 FORMAT (2I10,3F10.0,110)

IF (NSEG.GE.2.AND.NSEG.LE.50) GO TO 1005

WRITE (6,8) NSEG

8 FORMAT (1H0,110,22H SEGMENT(5) IS INVALID)

CALL EXIT

1005 WRITE (6,1010) RHO

1010 FORMAT (1H0,17H LIQUID DENSITY = ,E12.5,8H KG/M**3)

WRITE(6,4) GAMMAA,BD

4 FORMAT(717H CONTACT ANGLE = ,E12.5,8H RADIANS//

5 15H BOND NUMBER = ,E12.5)

ZBOT=BEGZPT

WRITE (6,1015)

1015 FORMAT (1H0,52X,13HTANK GEOMETRY/

249X,21HLENGTH UNIT IS INCHES//

31X,7HSEGMENT,6X,4HTYPE,13X,4HFROM,18X,2H10,26X,15HCHARACTERISTICS)

MAIN - EFN SOURCE STATEMENT - IFN(5) -

C TANK GEOMETRY DESCRIPTION

```

C
DO 50 I=1,NSEG
  READ (5,10) RPOINT(1),ZPOINT(1),RC(1),ZC(1),A(1),B(1),GAMMA(1)
10 FORMAT (7F10.0)
  IF (RPOINT(I-1).LT.0.0.OR.RPOINT(1).LT.0.0.OR.RC(1).LT.0.0.OR.
    2A(1).LT.0.0.OR.B(1).LT.0.0.OR.GAMMA(1).LT.0.0.OR.GAMMA(1).GT.360.0
    3) GO TO 65
  IF (ZPOINT(1).LT.ZBOT) ZBOT=ZPOINT(1)
  IF (B(1).EQ.0.0) GO TO 30
  TYPE(1)=1
  IF (A(1).NE.B(1)) GO TO 20
  WRITE (6,15) I,RPOINT(I-1),RPOINT(1),A(1),ZPOINT(I-1),ZPOINT(1),RC
    2(1),ZC(1)
15 FORMAT (1H0,2X,12,7X,8HCIRCULAR,6X,4HR = ,F10.4,7X,
    24HR = ,F10.4,5X,          9HRADIUS = ,F10.4 ,          /26X,4HZ = ,
    3F10.4,7X,          4HZ = ,F10.4,5X,          14HCENTER AT R = ,
    4F10.4,9H AND Z = ,F10.4)
  GO TO 50
20 WRITE (6,25) I,RPOINT(I-1),RPOINT(1),A(1),ZPOINT(I-1),ZPOINT(1),
    ZB(1),GAMMA(1),RC(1),ZC(1)
25 FORMAT (1H0,2X,12,6X,10HELLIPTICAL,5X,4HR = ,F10.4,7X,
    24HR = ,F10.4,5X,          18HSEMI-MAJOR AXIS = ,F10.4,9X,
    310HROTATED BY/26X,4HZ = ,F10.4,7X,          4HZ = ,F10.4,
    45X,18HSEMI-MINOR AXIS = ,F10.4,7X,          F6.1,8H DEGREES/66X,
    514HCENTER AT R = ,F10.4,9H AND Z = ,F10.4)
  GO TO 50
30 IF (A(1).NE.0.0) GO TO 40
  TYPE(1)=2
  WRITE (6,35) I,RPOINT(I-1),RPOINT(1),ZPOINT(I-1),ZPOINT(1)
35 FORMAT (1H0,2X,12,4X,13HSTRAIGHT LINE,4X,4HR = ,F10.4,7X,
    24HR = ,F10.4          /26X,4HZ = ,F10.4,7X,4HZ = ,F10.4)
  GO TO 50
40 TYPE(1)=3
  WRITE (6,45) I,RPOINT(I-1),RPOINT(1),A(1),ZPOINT(I-1),ZPOINT(1),GA
    2MMA(1),RC(1),ZC(1)
45 FORMAT (1H0,2X,12,6X,9HPARABOLIC,6X,4HR = ,F10.4,7X,
    24HR = ,F10.4,5X,          12HDIRECTRIX = ,F10.4,9X,
    310HROTATED BY/26X,4HZ = ,F10.4,7X,          4HZ = ,F10.4,
    436X,F6.1,8H DEGREES/66X,14HVERTEX AT R = ,F10.4,9H AND Z = ,F10.4)
50 CONTINUE
  WRITE (6,55) ZBOT
55 FORMAT (1H0,31HLOWEST POINT IN THE TANK IS AT ,F10.4,7H INCHES)
  IF (BEGZPT.EQ.ZPOINT(NSEG).AND.BEGRPT.EQ.RPOINT(NSEG).OR.
    2BEGZPT.NE.ZPOINT(NSEG).AND.BEGRPT.EQ.0.0.AND.RPOINT(NSEG).EQ.0.0)
    3GO TO 75
  WRITE (6,60)
60 FORMAT (1H0, 18HTANK IS NOT CLOSED)
  CALL EXIT
65 WRITE (6,70) I
70 FORMAT (1H0, 15HSEGMENT NUMBER ,12,11H IS INVALID)
  CALL EXIT
75 ZBOT=ZBOT*0.0254
  BEGRPT=BEGRPT*0.0254
  BEGZPT=BEGZPT*0.0254
  DO 50 I=1,NSEG

```

MAIN

- LFN SOURCE STATEMENT - IFN(5) -

```

RPOINT(1)=RPOINT(1)*0.0254
ZPOINT(1)=ZPOINT(1)*0.0254
RC(1)=RC(1)*0.0254
ZC(1)=ZC(1)*0.0254
A(1)=A(1)*0.0254
B(1)=B(1)*0.0254
GAM(1)=GAMMA(1)*PI/180.0

```

80 CONTINUE

C
C
C
C

READ BAFFLE INPUT CARD(S)

IF (NBAF.LT.1.OR.NBAF.GT.30) GO TO 100

DO 95 I=1,NBAF

READ (5,90) ZBAF(1),RBAFI(1),RBAFO(1)

90 FORMAT (3F10.0)

ZBAF(1)=ZBAF(1)*0.0254

RBAFI(1)=RBAFI(1)*0.0254

RBAFO(1)=RBAFO(1)*0.0254

95 CONTINUE

C
C
C
C

READ A CASE INPUT CARD AND BEGIN A NEW CASE OR TERMINATE RUN

100 READ (5,105) LEVEL,IDEGR,ZDIV,DZ(1),DZ(2)

105 FORMAT (F10.0,I10,3F10.0)

LEVEL=LEVEL*0.0254

ZDIV=ZDIV*0.0254

DZ(1)=DZ(1)*0.0254

DZ(2)=DZ(2)*0.0254

IF (LEVEL.LE.ZBOT) GO TO 170

IF (IDEGR.NE.6) IDEGR=3

C
C
C
C
C

FIND THE OUTER RADIUS AND THE RATIO OF THE INNER RADIUS TO THE

OUTER RADIUS OF THE UNDISTURBED FREE SURFACE

AS=0.0

TEMP(1)=0.0

IFLAG=0

DO 130 I=1,NSEG

IF (ZPOINT(I).EQ.ZPOINT(I-1)) GO TO 130

IF (IFLAG.EQ.0) GO TO 115

IFLAG=0

GO TO 130

115 ARG1=LEVEL-ZPOINT(I)

ARG2=LEVEL-ZPOINT(I-1)

IF (ARG1.NE.0.0) GO TO 120

RR=RPOINT(I)

IFLAG=1

GO TO 125

120 IF (ARG1/ABS(ARG1).EQ.ARG2/ABS(ARG2)) GO TO 130

CALL FINDR (I,LEVEL,RR)

125 IF (AS.LT.RR) AS=RR

IF (TEMP(1).GT.0.0) GO TO 135

TEMP(1)=RR

MAIN

- EFN SOURCE STATEMENT - IFN(S) -

130 CONTINUE

IF (AS.LE.0.0) GO TO 170

EPSL=0.0

GO TO 140

135 IF (TEMP(1).GT.RR) TEMP(1)=RR

EPSL=TEMP(1)/AS

C

C

C

FIND THE UNDISTURBED CENTER OF GRAVITY AND THE LIQUID VOLUME

C

140 ZCG=0.0

CALL INTG1 (F1,LEVEL,ZBOT)

ZCG=ANS

CALL INTG1 (F2,LEVEL,ZBOT)

ZCG=ZCG/ANS

VOL=ANS*PI

C

C

L=LEVEL-ZCG

D=ZCG-ZBOT

H=LEVEL-ZBOT

C

C

C

ESTABLISH THE CENTER OF ROTATION

C

G1=0.0

G2=0.0

G3=0

C

C

C

DETERMINE THE NUMBER OF EIGENFUNCTIONS TO BE USED FOR THE
EIGENVALUE PROBLEM

C

TEMP(1)=88.0296927/(2.0*L/AS)

DO 145 I=1,5

J=6-I

IF (JN(J).LT.TEMP(1)) GO TO 150

145 CONTINUE

GO TO 170

150 MORDER=MPOLY+J

C

C

C

CHECK VALIDITY OF MODES

C

IF (MODES.LT.1.OR.MODES.GT.J) MODES=J

C

C

TEMP(1)=LEVEL/0.0254

TEMP(2)=RH0*VOL

TEMP(3)=ZCG/0.0254

WRITE (6,165) TITLE,TEMP,IDEGR

165 FORMAT (1H1,23X,12A6777

21X,31HUNDISTURBED FREE SURFACE IS AT ,F10.4,7H INCHES///

35X,21HMASS OF THE LIQUID = ,E12.5,3H KG//

45X,36HUNDISTURBED CLNTER OF GRAVITY IS AT ,F10.4,7H INCHES//

55X,11,1X,18HDEGREES OF FREEDOM)

MAIN

- EFN

SOURCE STATEMENT

-

IFN(S)

-

C

C

CON1=AS*COS(GAMMAA)

CON2=PI*AS**2*L/VOL

CON3=CON1/L

CON4=PI*L**3/VOL

CON5=2.0*CON4

CON6=(AS/L)**2

CON7=CON1*PI*L**2/VOL

CON8=PI*AS**2/VOL

CON9=L/2.0

CCON10=PI*AS/VOL

CON11=AS*AS

CON14=COS(GAMMAA)

CON15=L**2

CON16=L**3

CALL MATRIX

GO TO 100

C

C

170 TEMP(1)=LEVEL/0.0254

WRITE (6,175) TITLE,TEMP(1)

175 FORMAT (1H1,23X,12A6///1X,2BHUNDISTURBED FREE SURFACE AT ,F10.4,

218H INCHES IS INVALID)

GO TO 100

C

C

END

\$IBFTC MATRIX LIST,DECK

MATRIX - EFN SOURCE STATEMENT - IFN(S) -

I=N=MPOLY

JNN=JN(1)

CALL INTG2 (F8,LEVEL,ZBOT)

SUMJ=JNN+JNN

AMN(M,N)=-EXP(-SUMJ*L/AS)/SUMJ*ANS

IF (M.EQ.N) GO TO 45

ARG1=JNN*EPSL

CALL BESSEL (ARG1,B1N,B1DN)

ARG2=JNN*EPSL

CALL BESSEL (ARG2,B1M,B1DM)

BMN(M,N)=(ARG2*B1N*B1DM-ARG1*B1M*B1DN)/(ARG2**2-ARG1**2)*EPSL**2

GO TO 50

45 CALL BESSEL (JNN,B1P,B1D)

ARG=JNN*EPSL

CALL BESSEL (ARG,B1,B1D)

BMN(M,N)=((JNN**2-1.)*B1P**2-(ARG*B1D)**2-(ARG**2-1.)*B1**2)/

2*(2.*JNN**2)

50 AMN(N,M)=AMN(M,N)

BMN(N,M)=BMN(M,N)

A(M,N)=AMN(M,N)

A(N,M)=AMN(M,N)

B(M,N)=BMN(M,N)

B(N,M)=BMN(M,N)

60 CONTINUE

C

CALL JACOBI (10,MORDER,IERR,B,A,CNK,LAMBDA,MODES)

IF (IERR.EQ.0) GO TO 100

WRITE (6,80)

80 FORMAT (1H0,35HEIGENVALUE PROBLEM CANNOT BE SOLVED)

RETURN

C

C

C

NON-DIMENSIONAL COEFFICIENTS

C

100 DO 105 N=1,MODES

GAMMAN(N)=0.0

DO 105 K=1,MORDER

DO 105 J=1,MORDER

105 GAMMAN(N)=GAMMAN(N)+PI*L*AS**2/VOL*CNK(K,N)*CNK(J,N)*BMN(K,J)

DO 110 M=1,MORDER

CALL INTG1 (F9,LEVEL,ZBOT)

110 HNSTAR(M)=ANS

DO 115 N=1,MODES

HN(N)=0.0

BN(N)=0.0

KN(N)=L/(LAMBDA(N)*AS)

DO 115 M=1,MORDER

HN(N)=HN(N)+2.*PI*AS**3/(VOL*GAMMAN(N)*KN(N))*CNK(M,N)*HNSTAR(M)

115 BN(N)=BN(N)+PI*AS**3/(VOL*GAMMAN(N))*CNK(M,N)*BHN(1,M)

IF (IDEGR.NE.6) GO TO 295

WRITE (6,130)

130 FORMAT (1H0,4X,21HCOEFFICIENTS MODE,10X,2HKN,14X,6HGAMMAN,12X,

27HAN = BN,13X,2HHN,16X,2HDN//)

DO 140 N=1,MODES

DN(N)=2.0*BN(N)-HN(N)

140 WRITE (6,145) N,KN(N),GAMMAN(N),BN(N),HN(N),DN(N)

MATRIX - EFN SOURCE STATEMENT - IFN(S) -

145 FORMAT (24X,11,5E18.5)

GO TO 750

295 WRITE (6,300)

300 FORMAT (1H0,4X,21HCoefficients MODE,10X,2HKN,14X,6HGAMMAN,14X,
22HBN,16X,2HBN//)

DO 305 N=1,MODES

305 WRITE (6,310) N,KN(N),GAMMAN(N),BN(N),HN(N)

310 FORMAT (24X,11,4E18.5)

C

C

C

C

C

MAXIMUM WAVE HEIGHT AT THE OUTER RADIUS OF THE UNDISTURBED
FREE SURFACE AND THE KINETIC ENERGY

750 TERM(1)=0.0

DO 770 J=1,MORDER

IF (J.GT.MPOLY) GO TO 760

TERM(1)=TERM(1)+CNK(J,1)

GO TO 770

760 I=J-MPOLY

TERM(1)=TERM(1)+CNK(J,1)*JUN(1)

770 CONTINUE

TERM(1)=TERM(1)*KN(1)*BN(1)

TERM(2)=0.0

DO 780 J=1,MORDER

DO 780 K=1,MORDER

780 TERM(2)=TERM(2)+CNK(J,1)*CNK(K,1)*AMN(J,K)

TERM(2)=TERM(2)*RHO*PI*AS*L**2*BN(1)**2/2.0

TERM1=TERM(1)

TERM2=TERM(2)

C

C

C

C

C

CALCULATE THE POLYNOMIAL COEFFICIENTS OF THE EIGENFUNCTIONS
ASSUMED FOR THE BVP NOS. 1 AND 2

DO 800 N=1,NPOLY

800 ACOEF(N,1)=1.0

DO 805 N=2,NPOLY

DO 805 K=2,N,2

805 ACOEF(N,K)=0.0

DO 810 N=3,NPOLY

DO 810 K=3,N,2

810 ACOEF(N,K)=-FLOAT((N-K+2)*(N-K+1))/FLOAT(K**2-1)*ACOE(N,K-2)

C

C

C

C

BOUNDARY VALUE PROBLEM NO. 1

CALL INTG2 (F4,LEVEL,ZBOT)

TERM(1)=-PI*RHO*ANS

C

CALL INTG2 (F5,LEVEL,ZBOT)

TERM(2)=8./3.*PI*RHO*ANS

C

TERM(3)=0.0

DO 223 M=1,NORDER

CALL INTG1 (F18,LEVEL,ZBOT)

RTSIDE(M)=2.0*AS**2/L**2*ANS

MATRIX - EFN SOURCE STATEMENT - IFN(S) -

```

DO 223 N=M,NORDER
  IF (M.GT.NPOLY) GO TO 222
  IF (N.GT.NPOLY) GO TO 220
  CALL INIG2 (F16,LEVEL,ZBOT)
  LIMAT(M,N)=-ANS
  GO TO 223
220 CALL INIG2 (F17,LEVEL,ZBOT)
  LIMAT(M,N)=-ANS
  GO TO 223
222 CALL INIG2 (F15,LEVEL,ZBOT)
  LIMAT(M,N)=ANS
223 LIMAT(N,M)=LIMAT(M,N)
  CALL SOLVE (NORDER,LIMAT,RTSIDE,1,1.0E-8,25,QN,IERR)
  IF (IERR.GT.0) GO TO 245
  WRITE (6,230)
230 FORMAT (1H0,4X,65HTHIRD TERM OF 111P CANNOT BE CALCULATED - IT IS
  2SET EQUAL TO ZERO)
  GO TO 255
245 DO 250 M=1,NORDER
  CALL INIG1 (F18,LEVEL,ZBOT)
250 TERM(3)=TERM(3)+2.0*RHO*PI*L**2*AS**3*QN(N)*ANS
C
255 TERM(4)=(G2**2+G3**2)*RHO*VOL
C
  111P=TERM(1)+TERM(2)+TERM(3)+TERM(4)
  TERM11=TERM(1)
  TERM12=TERM(2)
  TERM13=TERM(3)
  TERM14=TERM(4)
C
C   COMPUTE PS11
  PS11=0.0
  DO 819 N=1,NORDER
  IF (N.GT.NPOLY) GO TO 817
  PSUM=0.0
  DO 815 J=1,N
  PSUM=PSUM+ACOE(N,J)*(L/AS)**(N-J)
  815 CONTINUE
  PS11=PS11+QN(N)*PSUM
  GO TO 819
817 ARG=FLOAT(N-6)*PI/H
  CALL PMODES(ARG,PS,PSBP)
  PS11=PS11+PS+COS(ARG*(L+D))*QN(N)
  819 CONTINUE
C
C   COMPUTE RHON
  DO 827 I=1,MODES
  RHON(I)=0.0
  DO 827 M=1,NORDER
  11=M-NPOLY
  JNM=JN(11)
  RHOSUM=0.0
  DO 825 N=1,NORDER
  IF (N.GT.NPOLY) GO TO 821
  CALL INIG3(F30)

```

MATRIX - EFN SOURCE STATEMENT - IFN(S) -

GO TO 823

821 CALL INTG3(F31)

823 RHOSUM=RHOSUM*QN(N)*ANS

825 CONTINUE

RHON(1)=RHON(1)+CNK(M,I)*RHOSUM

827 CONTINUE

C

C

IF (IDEGR.NE.6) GO TO 2950

C

C

C

BOUNDARY VALUE PROBLEM NO. 2

C

C

C

122P IS THE SAME AS 111P

C

C

C

TERM(1) FOR THE BOUNDARY VALUE PROBLEM NO. 2 IS THE SAME
AS FOR THE BOUNDARY VALUE PROBLEM NO. 1

C

CALL INTG2 (F14,LEVEL,200)

TERM(2)=4.0*RHO*PI*ANS

C

TERM(4)=(G1**2+G3**2)*RHO*VOL

C

IF (IERR.GT.0) GO TO 425

WRITE (6,410)

410 FORMAT (1H0,4X,65HTHIRD TERM OF 122P CANNOT BE CALCULATED - IT IS
2SET EQUAL TO ZERO)

TERM(3)=0.0

GO TO 435

425 TERM(3)=111P-TERM(1)-TERM(2)-TERM(4)

C

435 WRITE (6,450) 111P,(TERM(I),I=1,4)

450 FORMAT (1H0,4X,7H122P = ,E12.5,28H KG-M**2 WHERE THE TERMS ARE,
24E16.5)

C

C

C

BOUNDARY VALUE PROBLEM NO. 3

C

133P=(G1**2+G2**2)*RHO*VOL

WRITE (6,500) 133P

500 FORMAT (1H0,4X,7H133P = ,E12.5,8H KG-M**2)

C

C

RETURN

C

C

C

FORCE DISTRIBUTION COEFFICIENTS

C

2950 IF (DZ(1).EQ.0.0.AND.DZ(2).EQ.0.0) GO TO 200

ARG4=0.0

ARG5=0.0

DO 2955 J=1,5

2955 CCALC(J)=0.0

IF (DZ(1).NE.0.0) GO TO 2960

I=2

MATRIX - EFN SOURCE STATEMENT - IFN(S) -

ZFORCE(1)=ZDIV-DZ(2)

ZFORCE(2)=ZDOT

GO TO 2965

2960 I=1

ZFORCE(1)=LEVEL-DZ(1)

ZFORCE(2)=ZDIV

2965 ZFORCE(3)=LEVEL

ARG3=LEVEL/0.0254

WRITE (6,2970) ARG3

2970 FORMAT (1H0,4X,31HFORCE DISTRIBUTION COEFFICIENTS//

26X,12HZ-COORDINATE,

35X,14HALPHA(2) COEF.,

46X,13HTHETA.. COEF.,

52X,20HALPHA(3)*XI(1) COEF.,

61X,20HALPHA(3)*XN(1) COEF.,

71X,13HXI(1).. COEF./

88X,9HIN INCHES,

911X,5HIN KG,

213X,7HIN KG-M,

321X,7HIN KG/M,

423X,5HIN KG//

5F17.4)

C

3000 M=1

ARG3=ZFORCE(1)/0.0254

DO 3015 N=1,MORDER

IF (N.G1.MPOLY) GO TO 3010

CALL INIG2 (F6,LEVEL,ZFORCE(1))

ALPHIN(N)=-(.4.*FLOAT(N)-2.*FLOAT((N+1)+2.))*ANS

GO TO 3015

3010 J=N-MPOLY

JNN=JN(J)

CALL INIG2 (F7,LEVEL,ZFORCE(1))

ALPHIN(N)=-EXP(-JNN*L/AS)/JNN*ANS

3015 CONTINUE

ARG1=0.0

DO 3020 N=1,MORDER

3020 ARG1=ARG1+PI*AS**2*CNK(N,1)*ALPHIN(N)

CALL INIG1 (F2,LEVEL,ZFORCE(1))

VM=PI*ANS

C

CPRINT(1)=CCALC(1)

CCALC(1)=-RHO*(VM-BN(1)*L*ARG1)

CPRINT(1)=CCALC(1)-CPRINT(1)

C

CPRINT(3)=CCALC(3)

CCALC(3)=RHO*ARG1

CPRINT(3)=CCALC(3)-CPRINT(3)

C

CPRINT(4)=CCALC(4)

CCALC(4)=CCALC(3)*KN(1)*BN(1)

CPRINT(4)=CCALC(4)-CPRINT(4)

C

CPRINT(5)=CCALC(5)

CCALC(5)=-L/KN(1)*CCALC(3)

CPRINT(5)=CCALC(5)-CPRINT(5)

MATRIX - EFN SOURCE STATEMENT - IFN(S) -

```

IF (DZ(2).EQ.0.0) GO TO 3025
CALL INTG1 (F1,ZFORCE(3),ZFORCE(1))
ZCGM=ANS
CALL INTG1 (F2,ZFORCE(3),ZFORCE(1))
ZCGM=ZCGM/ANS+ZCG
ARG4=ARG4+CPRINT(5)*ZCGM
ARG5=ARG5+CPRINT(5)

```

C

```

3025 IF (IERR.GT.0) GO TO 3045
WRITE (6,3030) CPRINT(1),(CPRINT(J),J=3,5),ARG3
3030 FORMAT (17X,E19.5,19X,3E19.5/F17.4)
GO TO 3995
3045 ARG2=0.0
DO 3060 N=1,NORDER
IF (N.GT.NPOLY) GO TO 3050
CALL INTG2 (F16,LEVEL,ZFORCE(1))
GO TO 3060
3050 CALL INTG2 (F17,LEVEL,ZFORCE(1))
3060 ARG2=ARG2-PI*(AS*L)**2*QN(N)*ANS

```

C

```

CPRINT(2)=CCALC(2)
CALL INTG1 (F1,LEVEL,ZFORCE(1))
CCALC(2)=RHO*(PI*ANS-G3*VM-ARG2+(G3*BN(1)-L*(BN(1)-HN(1)))*L*ARG1)
CPRINT(2)=CCALC(2)-CPRINT(2)

```

C

```

WRITE (6,3980) (CPRINT(J),J=1,5),ARG3
3980 FORMAT (17X,5E19.5/F17.4)

```

C

```

3995 ZFORCE(3)=ZFORCE(1)
ZFORCE(1)=ZFORCE(1)-DZ(1)
IF (ZFORCE(1).GE.ZFORCE(2)) GO TO 3000
IF (ZFORCE(2).EQ.ZFORCE(3)) GO TO 4000
ZFORCE(1)=ZFORCE(2)
GO TO 3000
4000 IF (DZ(2).EQ.0.0) GO TO 200
IF (I.EQ.2) GO TO 4005
I=2
ZFORCE(1)=ZDIV-DZ(2)
ZFORCE(2)=ZBOT
GO TO 3000

```

C

```

4005 ZCP=ARG4/ARG5/0.0254
WRITE (6,4010) ZCP
4010 FORMAT (1H0,4X,37HUNDISTURBED CENTER OF PRESSURE IS AT ,F10.4,
27H INCHES)

```

C

C

C

C

SPRING-MASS ANALOGY PARAMETERS

```

200 DO 205 N=1,MODES
LN(N)=G3-L+L*HN(N)/BN(N)
MN(N)=RHO*VOL*GAMMAN(N)*BN(N)**2*KN(N)
205 KNSIAR(N)=MN(N)*KN(N)/L
MO=RHO*VOL
DO 210 N=1,MODES
210 MO=MO-MN(N)

```

```

      LO=G3*RHO*VOL/MO
      DO 215 N=1,MODES
215  LO=LO-RHO*VOL*(GAMMAN(N)*BN(N)*KN(N)*(G3*BN(N)-L*(BN(N)-HN(N))))
      2/MO
      IO=111P-MO*LO**2
      DO 290 N=1,MODES
290  IO=IO-MN(N)*LN(N)**2

C
C   COMPUTE INTEGRAL FOR TAU
C
      DO 75 I=1,MORDER
      XI1=2*I-1
      DO 75 J=1,MORDER
      J1=2*J-1
      XJ1=J1
      IF(I.GT.MPOLY) GO TO 65
      IF(J.GT.MPOLY) GO TO 70
      ANS=((XI1*XJ1+1.0)/(XI1+XJ1))
      XINT(1,J)=ANS
      GO TO 75
65  I1=1-MPOLY
      JNN=JN(I1)
      JJ=J-MPOLY
      JNM=JN(JJ)
      CALL INTG3(F28)
      XINT(1,J)=ANS
      GO TO 75
70  JJ=J-MPOLY
      JNM=JN(JJ)
      CALL INTG3(F29)
      XINT(1,J)=ANS
75  CONTINUE

C
C   COMPUTE TAU AND XNU
C
      DO 95 M=1,MODES
      DO 87 N=M,MODES
      TAU(M,N)=0.0
      DO 87 I=1,MORDER
      DO 87 J=1,MORDER
      IF(I.GT.J) GO TO 81
      XINT2(I,J)=XINT(I,J)
      GO TO 85
81  XINT2(I,J)=XINT(J,I)
85  TAU(M,N)=TAU(M,N)+CNK(I,N)*CNK(J,M)*XINT2(I,J)
87  TAU(N,N)=TAU(M,N)
      SUMM(M)=0.0
      DO 90 I=1,MPOLY
      SUMM(M)=SUMM(M)+CNK(I,M)
90  CONTINUE
      M1=MPOLY+1
      DO 92 I=M1,MORDER
      I1=I-MPOLY
      JNN=JN(I1)
      CALL BESSEL(JNN,B1,B10)
      SUMM(M)=SUMM(M)+CNK(I,M)*B1

```

MATRIX - LFN SOURCE STATEMENT - IFN(S) -

92 CONTINUE

AACOEF(N)=L*(BN(M)-HN(M))

95 CONTINUE

DO 128 N=1,MODES

ECOEF(N)=0.0

GCOEF(N)=0.0

BCCOEF=0.0

CON10=CCON10/GAMMAN(N)

CON13=CON2/GAMMAN(N)

DO 126 M=1,MODES

XNU(N,M)=SUMM(N)*SUMM(M)

IF(N.EQ.M) GO TO 116

DELNM=0.0

CON12=KN(N)/(KN(N)-KN(M))

GO TO 118

116 DELNM=1.0

BETAB(N,M)=0.0

ASCOEF(N,M)=CON10*CON15*CON14*(2.0*TAU(N,N)-XNU(N,N))/KN(N)

S+KN(N)*CON1*(CON13*XNU(N,N)-2.0)/L

BSCOEF(N,M)=CON13*TAU(N,N)

GO TO 120

118 BETAB(N,M)=-CON2*TAU(N,M)/GAMMAN(N)-CON3*KN(M)*

S(-CON5*TAU(N,M)/(KN(M)*KN(M)*GAMMAN(N))+CON4*(1.0/(KN(M)*KN(M))

S-CON6)*XNU(N,M)/GAMMAN(N))

120 BETA(N,M)=-CON2*TAU(N,M)/GAMMAN(N)-CON3*KN(M)*(2.0*SQRT

S(GAMMAN(M)/GAMMAN(N))*DELNM-CON5*TAU(N,M)/(KN(M)*KN(M)*GAMMAN(N))

S+CON4*(1.0/(KN(N)*KN(M))-CON6)*XNU(N,M)/GAMMAN(N))

IF(N.EQ.M) GO TO 124

ASCOEF(N,M)=CON12*BETAB(N,M)+CON10*CON11*KN(N)*XNU(N,M)*CON14

BSCOEF(N,M)=CON12*BETA(N,M)+CON13*TAU(N,M)

124 CONTINUE

ECOEF(N)=-BSCOEF(N,M)*BN(M)+ECOEF(N)

GCOEF(N)=-BSCOEF(N,M)*AACOEF(N)+GCOEF(N)

126 CONTINUE

OMEGAC(N)=-0.5*BETA(N,N)/BO

TEMP=(SUMM(N)*PSII-2.0*RHON(N))*CON15*CON14/AS

BCOEF(N)=CON13 * TEMP/KN(N)+CON10*CON11*CON1*(SUMM(N)-2.0*VOL

S+GAMMAN(N)*BN(N)/(PI*AS**3))

FCOEF(N)=CON10*CON1*SUMM(N)

HCOEF(N)=CON10*CON11*SUMM(N)

XLCOEF(N)=CON13*L*TEMP*(BN(N)+0.5*HN(N))

BCCOEF=BCCOEF+GAMMAN(N)*XLCOEF(N)

128 CONTINUE

BCCOEF=RHO*VOL*(PI*AS**5*CON14/(2.0*VOL)+2.0*BCCOEF)

WRITE(6,11110)

11110 FORMAT(1H0,4X,21HCOEFFICIENTS MODE,8X,8HAN*+L1BN,12X,3HBN*,

S15X,3HEN*,15X,3HFN*//)

DO 829 N=1,MODES

829 WRITE(6,11111)N,AACOEF(N),BCOEF(N),ECOEF(N),FCOEF(N)

11111 FORMAT(24X,11,4E18.5)

WRITE(6,11112)

11112 FORMAT(1H0,4X,21HCOEFFICIENTS MODE,7X,9HGN*+L1EN*,12X,3HNN*,

S15X,3HLN*//)

DO 831 N=1,MODES

831 WRITE(6,11113)N,GCOEF(N),HCOEF(N),XLCOEF(N)

11113 FORMAT(24X,11,3E18.5)

MATRIX - LFN SOURCE STATEMENT - IFN(S) -

WRITE(6,785) TERM1,TERM2

785 FORMAT (1H0,4X,17HMAX. ETA/XN(1) = ,E12.5//

25X,27HKINETIC ENERGY/XN(1).**2 = ,E12.5,3H KG)

WRITE(6,260) 111P,TERM11,TERM12,TERM13,TERM14

260 FORMAT (1H0,4X,7H11P = ,E12.5,28H KG-M**2 WHERE THE TERMS ARE,
24E16.5)

WRITE(6,11116)BCCOEF

11116 FORMAT(1H0,4X,7HB . = ,E12.5)

WRITE (6,315)

315 FORMAT (1H0,4X,31HSPRING-MASS PARAMETERS MODE,9X,2HMN,15X,

22HLN,9X,15HKNSTAR/ALPHA(3)/

343X,5HIN KG,10X,9HIN INCHES,10X,7HIN KG/M//)

DO 320 N=1,MODES

LN(N)=LN(N)/0.0254

320 WRITE (6,325) N,MN(N),LN(N),KNSTAR(N)

325 FORMAT (34X,11,3E17.5)

LO=LO/0.0254

WRITE (6,330) MO,LO,10

330 FORMAT (1H0,31X,5HMO = ,E12.5,3H KG/

232X,5HLO = ,E12.5,7H INCHES/

332X,5HLO = ,E12.5,8H KG-M**2)

WRITE(6,11117) ((TAU(I,J),J=1,MODES),I=1,MODES)

WRITE(6,11118) ((XNU(I,J),J=1,MODES),I=1,MODES)

WRITE(6,11119) ((BETA(I,J),J=1,MODES),I=1,MODES)

WRITE(6,11120) ((BETAB(I,J),J=1,MODES),I=1,MODES)

WRITE(6,11114) ((ASCUEF(I,J),J=1,MODES),I=1,MODES)

WRITE(6,11115) ((BSCUEF(I,J),J=1,MODES),I=1,MODES)

WRITE(6,11123) (OMEGAC(I),I=1,MODES)

11114 FORMAT(/50X,2HA*/(25X,3E16.8))

11115 FORMAT(/50X,2HB*/(25X,3E16.8))

11117 FORMAT(/50X,3HTAU/(25X,3E16.8))

11118 FORMAT(/50X,2HNU/(25X,3E16.8))

11119 FORMAT(/49X,4HBETA/(25X,3E16.8))

11120 FORMAT(/47X,8HBETA BAR/(25X,3E16.8))

11123 FORMAT(/35X,28HNATURAL FREQUENCY CORRECTION/(25X,3E16.8))

C

C

C

C

BAFFLE EFFECT ON ENERGY

IF (NBAF.LT.1.OR.NBAF.GT.30) RETURN

WRITE (6,335)

335 FORMAT (1H0,6X,6HBAFFLE,6X,12HZ-COORDINATE,16X,13HR-COORDINATES,

214X,18HINST. DE/DT FACTOR,4X,17HAVG. DE/DT FACTOR/

321X,9HIN INCHES,19X,9HIN INCHES,28X,16HIN KG/M**2(3./4.))//)

DO 400 I=1,NBAF

ARG3=ZBAF(I)/0.0254

ARG4=RBAFI(I)/0.0254

ARG5=RBAFO(I)/0.0254

IF (ZBAF(I).LT.LEVEL.AND.ZBAF(I).GE.ZBOT.AND.RBAFI(I).NE.RBAFO(I).
2AND.RBAFI(I).GE.0.0.AND.RBAFO(I).GE.0.0) GO TO 345

WRITE (6,340) I,ARG3,ARG4,ARG5

340 FORMAT (9X,I2,3F19.4,21X,9HNO EFFECT)

GO TO 400

345 ARG1=RBAFI(I)

ARG2=RBAFO(I)

IF (ARG1.LT.ARG2) GO TO 350

MATRIX

- EFN

SOURCE STATEMENT

- IFN(S)

-

```
ARG1=RBAFO(1)
ARG2=RBAFI(1)
350 SUM=0.0
DO 390 JJ=1,2
DO 390 J=1,NORD
GO TO (355,360),JJ
355 II=NORD-J+1
T=C.5*W(II)
S=C.5+0.5*X(II)
GO TO 370
360 T=C.5*W(J)
S=C.5+0.5*X(J)
370 RK=(ARG2-ARG1)*S+ARG1
TERM(1)=0.0
KI=NPOLY+1
DO 380 K=KI,NORDER
KK=K-NPOLY
ARG=JN(KK)*RR/AS
CALL BESSEL (ARG,B1,B1D)
380 TERM(1)=TERM(1)+CNK(K,I)*JN(KK)*BT*EXP(-JN(KK)*(LEVEL-ZBAF(I))/AS)
SUM=SUM+ABS(TERM(1))**(5./2.)*RR*T*(ARG2-ARG1)
390 CONTINUE
TERM(1)=-SUM*RHO*18.0*1.23*(KNSTAR(1)/MN(1))**(1./4.)*
ZSQRT((ARG2-ARG1)/PI)*(L*BN(1)/AS)**(5./2.)
TERM(2)=TERM(1)*(KNSTAR(1)/MN(1))**(5./4.)*4.0/3.0/PI
WRITE (6,395) I,ARG3,ARG4,ARG5,TERM(1),TERM(2)
395 FORMAT (9X,12,3F19.4,2E21.5)
400 CONTINUE
C
C
RETURN
END
```

SLUSH

07/2

\$18FTC INTG1 LIST,DECK

SUBROUTINE INTG1 (F,LEV1,LEV2)

```

C
C
C      ROUTINE EVALUATES A CLOSED LINE INTEGRAL WITH RESPECT TO THE
C      Z-COORDINATE FOR THAT PORTION OF THE TANK BETWEEN TWO LEVELS
C      (LEV1 AND LEV2), WHERE F IS THE SUBROUTINE CONTAINING
C      THE INTEGRAND
C
      COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,ZBOT,IDEGR,
      ZG1,G2,G3,ZDIV,DZ(2)
      COMMON /GAUSS/NORD,X(8),W(8)
      COMMON /PARAM/H,N,JNM,JNT,RR,ZZ,FUNC,ANS
      COMMON /SEG/TYPE(50),BEGRPT,RPOINT(50),BEGZPT,ZPOINT(50),RC(50),
      ZZC(50),A(50),B(50),GAMMA(50),GAM(50)
      REAL LEV1,LEV2
C
C
      ANS=0.0
      DO 1999 I=1,NSEG
C
C
C      LOCATE THE PORTION OF THE SEGMENT BETWEEN THE TWO LEVELS
C
      IF (ZPOINT(I-1).EQ.ZPOINT(I)) GO TO 1999
      IF (ZPOINT(I-1).LE.LEV2.AND.ZPOINT(I).LE.LEV2) GO TO 1999
      IF (ZPOINT(I-1).GE.LEV1.AND.ZPOINT(I).GE.LEV1) GO TO 1999
      IF (ZPOINT(I-1).LE.LEV2.AND.ZPOINT(I).GT.LEV2) GO TO 10
      IF (ZPOINT(I-1).LT.LEV1.AND.ZPOINT(I).GE.LEV1) GO TO 20
      IF (ZPOINT(I-1).LT.LEV1.AND.ZPOINT(I).LE.LEV2) GO TO 25
      IF (ZPOINT(I-1).LT.LEV1.AND.ZPOINT(I).LT.LEV1) GO TO 30
      BEG=LEV1
      IF (ZPOINT(I).LE.LEV2) GO TO 5
      END=ZPOINT(I)
      GO TO 1035
5  END=LEV2
      GO TO 1035
10  BEG=LEV2
      IF (ZPOINT(I).GE.LEV1) GO TO 15
      END=ZPOINT(I)
      GO TO 1035
15  END=LEV1
      GO TO 1035
20  BEG=ZPOINT(I-1)
      END=LEV1
      GO TO 1035
25  BEG=ZPOINT(I-1)
      END=LEV2
      GO TO 1035
30  BEG=ZPOINT(I-1)
      END=ZPOINT(I)
C
C
C      EVALUATE THE INTEGRAL ALONG THE SEGMENT BETWEEN THE TWO LEVELS
C
      1035 SUM=0.0

```


SLOSH

05/72

INTGI

- EFW

SOURCE STATEMENT

- IFN(S)

DO 1040 JJ=1,2

DO 1040 J=1,NORD

GO TO (1036,1037),JJ

1036 II=NORD-J+1

T=D.5*W(II)

IF (END.GI.BEG) GO TO 2012

S=D.5+D.5*X(II)

GO TO 1038

2012 S=D.5-D.5*X(II)

GO TO 1038

1037 T=D.5*W(J)

IF (END.GT.BEG) GO TO 2022

S=D.5-D.5*X(J)

GO TO 1038

2022 S=D.5+D.5*X(J)

1038 ZZ=(END-BEG)*S+BEG

CALL FINDR (1,ZZ,RR)

CALL F

1040 SUM=SUM+FUNC*T

ANS=ANS+SUM*(END-BEG)

C

C

1999 CONTINUE

C

C

RETURN

END

SIBFTC INIG2 LIST DECK

IFN(S)

C
C
C
C
C
C
C

C
C
C
C

$\mathbb{R}^n$

C
C
C
C

10

-48

```
IF (IOUT1.EQ.0) GO TO 10
GO TO 5
10 IF (RPOINT(1).EQ.RPOINT(I-1)) GO TO 15
CALL FINDR (1,LEV1,ENDR)
IF (IOUT1.NE.0) GO TO 5
CALL FINDR (1,LEV2,BEGR)
IBEGR=1
BEG=ENDR
END=BEGR
GO TO 80
15 ENDR=RPOINT(1)
IF (IOUT1.NE.0) GO TO 5
BEGR=RPOINT(1)
IBEGR=1
GO TO 110
20 IF (ZPOINT(1).LT.LEV1) GO TO 25
IOUT1=1
IF (RPOINT(1).EQ.RPOINT(I-1)) GO TO 23
CALL FINDR (1,LEV1,IOUT1)
CALL FINDR (1,LEV2,IOUT2)
IF (IBEGR.EQ.0) GO TO 22
21 IBEGR=0
BEG=BEGR
END=IOUT2
ILEV=2
GO TO 80
22 IF (BEGR.EQ.0.0) TEMP=OUT2
BEG=OUT2
END=OUT1
GO TO 80
23 OUT1=RPOINT(1)
24 OUT2=RPOINT(1)
IF (IBEGR.NE.0) GO TO 2,
IF (BEGR.EQ.0.0) TEMP=OUT2
GO TO 110
25 IF (RPOINT(1).EQ.RPOINT(I-1)) GO TO 24
CALL FINDR (1,LEV2,IOUT2)
IF (IBEGR.NE.0) GO TO 21
IF (BEGR.EQ.0.0) TEMP=OUT2
BEG=OUT2
END=RPOINT(1)
GO TO 80
30 IOUT1=1
IF (RPOINT(1).EQ.RPOINT(I-1)) GO TO 35
CALL FINDR (1,LEV1,IOUT1)
BEG=RPOINT(I-1)
END=OUT1
GO TO 80
35 OUT1=RPOINT(1)
GO TO 110
40 IBEGR=1
IF (RPOINT(1).EQ.RPOINT(I-1)) GO TO 45
CALL FINDR (1,LEV2,BEGR)
BEG=RPOINT(I-1)
END=BEGR
GO TO 80
```

INTG2

- EFN

SOURCE STATEMENT

- IFN(S)

45 BEGR=RPOINT(I)

GO TO 110

50 IF (RPOINT(I).EQ.RPOINT(I-1)) GO TO 110

BEG=RPOINT(I-1)

END=RPOINT(I)

C

C

C EVALUATE THE INTEGRAL ALONG THE SEGMENT BETWEEN THE TWO LEVELS

C OR ALONG ONE OF THE TWO LEVELS

C

80 SUM=0.0

DO 90 JJ=1,2

DO 90 J=1,NORD

GO TO (1001,1002),JJ

1001 II=NORD-J+1

T=0.5*W(II)

IF (END.GT.BEG) GO TO 1010

S=0.5+0.5*X(II)

GO TO 85

1010 S=0.5-0.5*X(II)

GO TO 85

1002 T=0.5*W(J)

IF (END.GT.BEG) GO TO 1020

S=0.5-0.5*X(J)

GO TO 85

1020 S=0.5+0.5*X(J)

85 RR=(END-BEG)*S+BEG

IF (ILEV.EQ.0) CALL FINDZ (I,RR,ZZ)

IF (ILEV.EQ.1) ZZ=LEV1

IF (ILEV.EQ.2) ZZ=LEV2

CALL F

SUM=SUM+FUNC*T

90 CONTINUE

ANS=ANS+SUM*(END-BEG)

C

C

C CHECK IF INTEGRATION HAS BEEN COMPLETED ALONG THE SEGMENT

C

IF (ILEV.NE.0.AND.IFLAG.EQ.0) GO TO 1

C

C

C CHECK IF INTEGRATION HAS BEEN PERFORMED ON THE TWO LEVELS

C

110 IF (I.LT.NSEG) GO TO 130

IFLAG=1

IF (IBEGR.EQ.0) GO TO 115

IBEGR=0

BEG=BEGR

END=TEMP

ILEV=2

GO TO 80

115 IF (IOUTI.EQ.0) GO TO 130

IOUTI=0

BEG=OUTI

END=ENDR

ILEV=1

SLOSH

06/2

INT62

- EFW

SOURCE STATEMENT

- IFN(S)

GO TO 80

C

C

130 CONTINUE

C

C

RETURN

END

SLOSH

02/6

SIBFIC INT3 LIST DECK

```
SUBROUTINE IN+G3(F)
COMMON/XW/XG(8),WG(8)
COMMON /GAUSS/NORD,X(8),W(8)
COMMON /PARAM/M,N,JNM,JRN,RR,ZZ,FUNC,ANS
ANS=0.0
DO 10 I=1,2
DO 10 J=1,NORD
GO TO (2,4),I
2 U=-XG(J)
GO TO 5
4 U=XG(J)
5 RR=U+0.5
CALL F
10 ANS=ANS+FUNC*WG(J)
RETURN
END
```


SLUSH

07/2

\$IBFTC FINDR LIST,DECK

```

SUBROUTINE FINDR (I,ZZ,R)
COMMON /SEG/TYPE(50),BEGRPT,RPOINT(50),BEGZPT,ZPOINT(50),RC(50),
ZZC(50),A(50),B(50),GAMMA(50),GAM(50)
INTEGER TYPE,ARG
IF (ZZ-ZPOINT(1)) 2,1,2
1 RR=RPOINT(1)
RETURN
2 IF (ZZ-ZPOINT(1-1)) 4,3,4
3 RR=RPOINT(1-1)
RETURN
4 R1=0.0
R2=0.0
ARG=TYPE(1)
GO TO (5,35,40),ARG
5 C1=ZZ-ZC(1)
C2=SIN(GAM(1))
C3=COS(GAM(1))
C4=(C2*B(1))**2+(C3*A(1))**2
C5=(C3*B(1))**2+(C2*A(1))**2
C6=2.*C2*C3*(B(1)**2-A(1)**2)
C7=(C1*C6)**2-4.*C5*(C4*C1**2-(A(1)*B(1))**2)
IF (C7) 100,10,10
10 R1=RC(1)+(-C1*C6+SQRT(C7))/(2.*C5)
R2=RC(1)+(-C1*C6-SQRT(C7))/(2.*C5)
GO TO 70
35 RR=RPOINT(1-1)+(RPOINT(1)-RPOINT(1-1))/(ZPOINT(1)-ZPOINT(1-1))*(ZZ
2-ZPOINT(1-1))
RETURN
40 C1=ZZ-ZC(1)
IF (GAMMA(1).EQ.90.0) GO TO 50
IF (GAMMA(1).NE.270.0) GO TO 55
RR=RC(1)+C1**2/(2.0*A(1))
RETURN
50 RR=RC(1)-C1**2/(2.0*A(1))
RETURN
55 C2=SIN(GAM(1))
C3=COS(GAM(1))
C4=2.0*C2*C3
C5=C1*C4+2.*A(1)*C2
C6=C5**2-4.*C3**2*((C1**2)-2.*A(1)*C1*C3)
IF (C6) 100,60,60
60 R1=RC(1)+(-C5+SQRT(C6))/(2.*C3**2)
R2=RC(1)+(-C5-SQRT(C6))/(2.*C3**2)
GO TO 70
100 WRITE (6,101) 1
101 FORMAT (1H0,15HSEGMENT NUMBER ,12,11H 15 INVALID)
CALL EXIT
70 IF (R1.LT.0.0) R1=0.0
IF (R2.LT.0.0) R2=0.0
IF ((R1-RPOINT(1))/ABS(R1-RPOINT(1))-(R1-RPOINT(1-1))/ABS(R1-RPOIN
2T(1-1))) 94,96,94
94 RR=R1
RETURN
96 IF ((R2-RPOINT(1))/ABS(R2-RPOINT(1))-(R2-RPOINT(1-1))/ABS(R2-RPOIN
2T(1-1))) 98,100,98

```

SLOSH

FINDR

- EFM

SOURCE STATEMENT

- IFN(S)

05/2

98 RR#R2
RETURN
END

\$1BFTC FINDZ LIST DECK

```

SUBROUTINE FINDZ (I,RR,Z)
COMMON /SEG/TYPE(50),BEGRPT,RPOINT(50),BEGZPT,ZPOINT(50),RC(50),
ZC(50),A(50),B(50),GAMMA(50),GAM(50)
INTEGER TYPE,ARG
IF (RR-RPOINT(I)) 2,1,2
1 ZZ=ZPOINT(I)
RETURN
2 IF (RR-RPOINT(I-1)) 4,3,4
3 ZZ=ZPOINT(I-1)
RETURN
4 Z1=0.0
ZZ=0.0
ARG=TYPE(I)
GO TO (5,20,25),ARG
5 C1=RR-RC(I)
C2=SIN(GAM(I))
C3=COS(GAM(I))
C4=(C2*B(I))*2+(C3*A(I))*2
C5=(C3*B(I))*2+(C2*A(I))*2
C6=2.*C2*C3*(B(I)**2-A(I)**2)
C7=(C1*C6)**2-4.*C4*(C1**2*C5-(A(I)*B(I))*2)
IF (C7) 100,10,10
10 Z1=ZC(I)+(-C1*C6+SQRT(C7))/(2.*C4)
ZZ=ZC(I)+(-C1*C6-SQRT(C7))/(2.*C4)
GO TO 50
20 ZZ=ZPOINT(I-1)+(ZPOINT(I)-ZPOINT(I-1))/(RPOINT(I)-RPOINT(I-1))*(RR
2-RPOINT(I-1))
RETURN
25 C1=RR-RC(I)
IF (GAMMA(I).EQ.0.0+0.0+GAMMA(I).EQ.360.0) GO TO 35
IF (GAMMA(I).NE.180.0) GO TO 40
ZZ=ZC(I)-C1**2/(2.0*A(I))
RETURN
35 ZZ=ZC(I)+C1**2/(2.*A(I))
RETURN
40 C2=SIN(GAM(I))
C3=COS(GAM(I))
C4=2.0*C2*C3
C5=C1*C4-2.*A(I)*C3
C6=C5**2-4.*C2**2*((C1**2+C3)**2+2.*A(I)*C1*C2)
IF (C6) 100,45,45
45 Z1=ZC(I)+(-C5+SQRT(C6))/(2.*C2**2)
ZZ=ZC(I)+(-C5-SQRT(C6))/(2.*C2**2)
GO TO 50
100 WRITE (6,101) I
101 FORMAT (1H0,1SHSEGMENT NUMBER ,12,11H IS INVALID)
CALL EX11
50 IF ((Z1-ZPOINT(I))/ABS(Z1-ZPOINT(I))-(Z1-ZPOINT(I-1))/ABS(Z1-ZPOINT
2(I-1))) 55,60,55
55 ZZ=Z1
RETURN
60 IF ((Z2-ZPOINT(I))/ABS(Z2-ZPOINT(I))-(Z2-ZPOINT(I-1))/ABS(Z2-ZPOINT
2(I-1))) 65,100,65
65 ZZ=Z2
RETURN

```

SLOSH

02/2

FINDZ

-

EFN

SOURCE STATEMENT

-

IFN(S)

-

END

SLOSH

02/2

SIBFTC F1

LIST DECK

SLOSH

02/2

F1

- EFN SOURCE STATEMENT - IFN(S) -

```
SUBROUTINE F1
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,P1,EPST,RHO,MODES,7BOT,IDEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /PARAM/M,N,JNH,JAN,RR,ZZ,FUNC,ANS
FUNC=RR**2*(ZZ-ZCG)
RETURN
END
```


SLOSH

02/2

\$1BFTC F2

LIST DECK

SLOSH

02/2

F2

- EFN SOURCE STATEMENT - IFN(S) -

```
SUBROUTINE F2  
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS  
FUNC=RR**2  
RETURN  
END
```

SLOSH

027

516FTC F4

LIST,DECK

```
SUBROUTINE F4  
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,P1,EPSL,RHO,MODES,7BOT,1DEGR,  
ZG1,G2,G3,ZDIV,DZ(2)  
COMMON /PARAM/H,N,JNM,JNN,RR,ZZ,FUNC,ANS  
FUNC=RR**3*(ZZ-ZCG)+2./3.*RR*(ZZ-ZCG)**3  
RETURN  
END
```

SLOSH

02/2

\$IBFTC F5

LIST DECK

SLOSH

F5

- EFN SOURCE STATEMENT - IFN(S) -

02/2

```
SUBROUTINE F5  
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,7BPT,IDEGR,  
2G1,G2,G3,ZDIV,DZ(2)  
COMMON /PARAM/H,N,JNM,JNN,RR,ZZ,FUNC,ANS  
FUNC=RR*(ZZ-ZCG)**3  
RETURN  
END
```

SLOSH

027

\$IBFTC F6

LIST,DECK

```
SUBROUTINE F6  
COMMON MSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,ZBOT,IOEGR,  
ZG1,G2,G3,ZDIV,DZ(2)  
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS  
FUNC=(RR/AS)**(2*(M+N)-3)*(ZZ-ZCG)/AS**2  
RETURN  
END
```


SLDSH

0272

\$IBFTC F7

LIST DECK

```
SUBROUTINE F7
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,P1,EPSL,RHO,MODELS,ZBOT,IDEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /PARAM/H,N,JNN,JNN,RR,ZZ,FUNC,ANS
REAL JNN
ARG=JNN*RR/AS
CALL BESSEL (ARG,B1,B1D)
FUNC=(FLOAT(2*N-1)*(RR/AS)**(2*N-1)*JNN*B1D+(RR/AS)**(2*N-2)*B1)*
2EXP(JNN*(ZZ-ZCG)/AS)/AS
RETURN
END
```

SLOSH

027.

\$IBFTC F8

LIST,DECK

```
SUBROUTINE F8
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,NODES,7BOT,1DEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS
REAL JNM,JNN
RNON=RR/AS
ARG=JNM*RNON
CALL BESSEL (ARG,B1M,B1DM)
ARG=JNN*RNON
CALL BESSEL (ARG,B1N,B1DN)
FUNC=((ARG*JNM*B1DM*B1DN+B1M*B1N/RNON+ARG*JNM*B1M*B1N)/AS)*
2EXP((JNM+JNN)*(ZZ-ZCG)/AS)
RETURN
END
```

SLOSH

027

SIBFTC F9

LIST DECK

```
SUBROUTINE F9
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPST,RHO,MODES,UBOT,IDEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /ELEM/MORDER,MPOLY,JN(5),NORDER,NPOLY,ACDEF(10,10)
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS
REAL JN,JNM,L
I=M-MPOLY
IF (I) 5,5,10
5 FUNC=((ZZ-ZCG)*RR**(2*M))/AS**(2*M+2)
RETURN
10 JNM=JN(I)
ARG=JNM*RR/AS
CALL BESSEL (ARG,B1,B,D)
FUNC=(ZZ-ZCG)*RR*B1*EXP(JNM*(ZZ-ZCG*L)/AS)/AS**3
RETURN
END
```

SLOSH

0272

\$18FTC F12

LIST,DECK

```
SUBROUTINE F12
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,7BOT,1DEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /ELEM/NORDER,NPOLY,YN(5),NORDER,NPOLY,ACOE(10,10)
COMMON /PARAM/M,N,JNH,JNH,RR,ZZ,FUNC,ANS
IF (M.GT.NPOLY) GO TO 10
FUNC=0.0
IF (ZZ.EQ.ZCG) RETURN
DO 5 I=1,M
5 FUNC=FUNC+ACOE(M,I)*(RR/AS)**(I+2)*((ZZ-ZCG)/AS)**(N-I)/AS
RETURN
10 ARG=FLOAT(H-NPOLY)*PI*RR/H
CALL PMODBS (ARG,PBS,PBSP)
FUNC=PBS*CO$ (FLOAT(H-NPOLY)*PI/H*(ZZ-ZCG+D))*RR**2/AS**3
RETURN
END
```


SLOSH

02/2

\$18BTC F14

LIST DECK

SLOSH

02/2

F14

" EFN SOURCE STATEMENT - IFN(S) -

```
SUBROUTINE F14  
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,7BOT,IDEGR,  
2G1,G2,G3,ZDIV,DZ(2)  
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS  
FUNC=RR**3*(ZZ-ZCG)  
RETURN  
END
```

SLUSH

02/2

\$IBFTC F15

LIST DECK

```

SUBROUTINE FIG
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPST,RHO,MODES,7BOT,IOEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /ELEM/NORDER,MPOLY,JN(5),NORDER,NPOLY,ACOE(10,10)
COMMON /PARAM/H,N,JNH,JNN,RR,ZZ,FUNC,ANS
C=FLOAT(H-NPOLY)*PI/H
ARG=C*RR
CALL PMODBS (ARG,PBSN,PBSPN)
IF (H.EQ.N) GO TO 5
ZARG=(ZZ-ZCG)/AS
DD=FLOAT(H-NPOLY)*PI/H
ARG=DD*RR
CALL PMODBS (ARG,PBSH,PBSPM)
A=(PBSPM*PBSPN*RR*C*DD+PBSH*PBSN/RR)*AS
B=C*DD*RR*AS*PBSH*PBSN
P=SIN(C*D)
Q=SIN(DD*D)
R=COS(C*D)
S=COS(DD*D)
E=A*P*Q+B*R*S
F=A*R*S+B*Q*P
G=B*R*Q-A*P*S
HH=B*P*S-A*R*Q
P=(C+DD)*AS
Q=(C-DD)*AS
R=P*ZARG
S=Q*ZARG
FUNC=((E-F)/P*SIN(R)-(E+F)/Q*SIN(S)+(G+HH)/P*COS(R)+(G-HH)/Q*
2COS(S))/AS/2.0
RETURN
5 A=(RR*PBSPN**2*C**2+PBSN**2/RR)*AS
B=RR*PBSN**2*C**2*AS
FUNC=(-A*(ZZ-ZCG)/AS+(A-B)/(C*AS)*(C*(ZZ-ZCG+0)/2.0-0.25*SIN(2.0*C
2*(ZZ-ZCG+0))))/AS
RETURN
END

```

SFBTC F16

LIST DECK

```
SUBROUTINE F1A
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,7BOT,IDEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /ELEM/MORDER,MPOLY,JN(5),NORDER,NPOLY,ACDEF(10,10)
COMMON /PARAM/M,N,JNH,JNN,RR,ZZ,FUNC,ANS
ZARG=(ZZ-ZCG)/AS
FUNC=0.0
DO 5 K=1,N
DO 5 I=1,M
5 FUNC=FUNC+FLOAT(K*I+1)/FLOAT(M+N-I-K+1)*ACDEF(N,K)*ACDEF(M,I)*
2(RR/AS)**(K+I-1)*ZARG**(M+N-I-K+1)/AS
DO 10 K=1,N
DO 10 I=1,M
IF (N.EQ.K.OR.M.EQ.I) GO TO 10
FUNC=FUNC+FLOAT((N-K)*(M-I))/FLOAT(N+M-K-I-1)*ACDEF(N,K)*
2ACDEF(M,I)*(RR/AS)**(K+I+1)*ZARG**(N+M-K-I-1)/AS
10 CONTINUE
RETURN
END
```

SLOSH

02/2

SIBFTC F17

LIST DECK

```

SUBROUTINE F17
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,7BOT,1DEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /ELEM/MORDER,NPOLY,UN(5),NORDER,NPOLY,ACDEF(10,10)
COMMON /PARAM/N,N,UNH,UNH,RR,ZZ,FUNC,ANS
FUNC=0.0
ZARG=(ZZ-ZCG)/AS
IF (ZARG.EQ.0.0) RETURN
STOR=FLOAT(N-NPOLY)*PI*AS/H
ARG=FLOAT(N-NPOLY)*PI*RR/H
CALL PHOBS(ARG,PBS,PBSP)
ARG=FLOAT(N-NPOLY)*PI/H*(ZZ-ZCG+0)
DO 50 I=1,M
TEMP=0.0
MM=I-1
IF (MM.EQ.0) GO TO 35
GO TO (30,25,20,15,10,5),MM
5 TEMP=TEMP-FLOAT((MM)*(MM-1)*(MM-2)*(MM-3)*(MM-4)*(MM-5))/STOR**7*
  ZZARG**((MM-6)*SIN(ARG)
10 TEMP=TEMP+FLOAT((MM)*(MM-1)*(MM-2)*(MM-3)*(MM-4))/STOR**6*ZARG**
  2(MM-5)*COS(ARG)
15 TEMP=TEMP+FLOAT((MM)*(MM-1)*(MM-2)*(MM-3))/STOR**5*ZARG**((MM-4)*
  2SIN(ARG)
20 TEMP=TEMP-FLOAT((MM)*(MM-1)*(MM-2))/STOR**4*ZARG**((MM-3)*COS(ARG)
25 TEMP=TEMP-FLOAT((MM)*(MM-1))/STOR**3*ZARG**((MM-2)*SIN(ARG)
30 TEMP=TEMP+FLOAT(MM)/STOR**2*ZARG**((MM-1)*COS(ARG)
35 TEMP=TEMP+ZARG**MM/STOR*SIN(ARG)
50 FUNC=FUNC+(STOR*PBSP*FLOAT(1)*(RR/AS)**1+PBS*(RR/AS)**(1-1))*
  ZACDEF(M,1)*TEMP/AS
DO 100 I=1,M
IF (M.EQ.1) RETURN
TEMP=0.0
MM=I-1
IF (MM.EQ.0) GO TO 80
GO TO (75,70,65,60,55),MM
55 TEMP=TEMP+FLOAT((MM)*(MM-1)*(MM-2)*(MM-3)*(MM-4))/STOR**6*ZARG**
  2(MM-5)*SIN(ARG)
60 TEMP=TEMP-FLOAT((MM)*(MM-1)*(MM-2)*(MM-3))/STOR**5*ZARG**((MM-4)*
  2COS(ARG)
65 TEMP=TEMP-FLOAT((MM)*(MM-1)*(MM-2))/STOR**4*ZARG**((MM-3)*SIN(ARG)
70 TEMP=TEMP+FLOAT((MM)*(MM-1))/STOR**3*ZARG**((MM-2)*COS(ARG)
75 TEMP=TEMP+FLOAT(MM)/STOR**2*ZARG**((MM-1)*SIN(ARG)
80 TEMP=TEMP-ZARG**MM/STOR*COS(ARG)
100 FUNC=FUNC-STOR*PBS*ACDEF(M,1)*FLOAT(M-1)*(RR/AS)**(1+1)*TEMP/AS
RETURN
END

```


SLOSH

05/2

\$IBFTC F18 LIST,DECK

BEGIN COMPILING 224018

```
SUBROUTINE F18
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,7BOT,1DEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /ELEM/MORDER,NPOLY,JN(5),NORDER,NPOLY,ACOE(10,10)
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS
IF (M.GT.NPOLY) GO TO 10
FUNC=0.0
DO 5 I=1,M
5 FUNC=FUNC+ACOE(M,I)*(RR/AS)**(I+1)*((ZZ-ZCG)/AS)**(M-I+1)/AS
RETURN
10 ARG=FLOAT(M-NPOLY)*PI*RR/H
CALL PHODBS (ARG,PBS,PBSP)
FUNC=RR/AS*(ZZ-ZCG)/AS*PBS*CO( FLOAT(M-NPOLY)*PI/H*(ZZ-ZCG+D))/AS
RETURN
END
```

SIBFTC-F28D LIST, DECK

```
SUBROUTINE F28
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSE,RHO,MODES,ZBOT,IDEGR,
ZG1,G2,G3,ZDIV,DZ(2)
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS
COMMON/BES/B1I,B1D,B1J,B1DJ,X1I,XJ1,1,J,J1,I1,JJ
COMMON /ELEM/HORDER,MPOLY,JN(5),HORDER,HPOLY,ACDEF(10,10)
REAL JN,JNN,JNM
ARG=JNN*RR
CALL BESSEL(ARG,B1I,B1D)
ARG=JNM*RR
CALL BESSEL(ARG,B1J,B1DJ)
FUNC=(JN(I1)*JN(JJ)*B1D*B1DJ*RR +B1I*B1J /RR)
RETURN
END
```

SIBFTC F29D LIST, DECK

```
SUBROUTINE F29
COMMON NSEG,LEVEL,ZCG,L,D,H,AS,VOL,PI,EPSL,RHO,MODES,ZBOT,IDEGR,
2G1,G2,G3,ZDIV,DZ(2)
COMMON /ELEM/HORDER,MPOLY,JN(5),NORDER,NPOLY,ACOE(10,10)
COMMON /PARAM/M,N,JNM,JNN,RR,ZZ,FUNC,ANS
COMMON/BES/B11,B1D1,B1J,B1DJ,X11,XJ1,I,J,J1,I1,JJ
REAL JN,JNM
ARG=JNM*RR
CALL BESSEL(ARG,B1J,B1DJ)
IF(RR)1,2,1
1 FUNC=(X11*ARG*B1DJ+B1J)*(RR)**(2*I-2)
GO TO 3
2 FUNC=0.0
3 CONTINUE
RETURN
END
```

SLOSH

02/1

\$IBFTC JACOBI LIST,DECK

JACOBI - EFN SOURCE STATEMENT - IFN(5) -

```

CJACOBI      BUFFERING ROUTINE BETWEEN FORTRAN AND EGZF (SHARE)
SUBROUTINE JACOBI (MAX,NACT,NOGO,A,B,VECTOR,VALUE,MODES)
C
C      CALL JACOBI (MAX,NACT,NOGO,A,B,VECTOR,VALUE,MODES)
C      WHERE
C      MAX    DIMENSIONED ORDER OF A AND B MATRICES (LIMIT IS 40)
C      NACT   WORKING ORDER OF A AND B MATRICES
C      NOGO   =0 METHOD SUCCESSFUL
C      NOGO   =1 MATRIX B IS NOT POSITIVE DEFINITE
C      NOGO   =-1 ILL CONDITIONED MATRIX F (SEE SHARE WRITEUP-NYE(V))
C      A      MATRIX A OF AX=(LAMBDA)BX
C      B      MATRIX B OF AX=(LAMBDA)BX
C      VECTOR EIGENVECTOR STORAGE MATRIX - VECTORS BY COLUMNS
C      VALUE  EIGENVALUE STORAGE VECTOR
C      MODES  NUMBER OF EIGENVALUES DESIRED
C
C      THIS ROUTINE SOLVES THE EIGENVALUE PROBLEM
C      AX=(LAMBDA)BX
C      WHERE A AND B MUST BE SYMMETRIC AND B MUST BE POSITIVE DEFINITE
C
C      THE ORIGINAL A AND B MATRICES ARE DESTROYED BY THE
C      EXECUTION OF SUBROUTINE EGZF
C
C      A RESTRICTION IN EGZF LIMITS DIMENSION OF A AND B
C      MATRICES TO 40x40 MAXIMUM. TO CHANGE THIS RESTRICTION
C      CHANGE LOCATION COMMON OF EGZF (NOW SET AT 855 54) (N+14)
C
C      SUBROUTINES INCLUDED IN THIS PACKAGE ARE
C      JACOBI
C      RMEGZF
C      DIMENSION A(MAX,1),B(MAX,1),VECTOR(MAX,1),VALUE(1)
C      N=NACT
C      I1=0
C
C      READY A MATRIX FOR EGZF
C      DO 10 I2=1,N
C      DO 10 I3=1,N
C      I1=I1+1
C      10 A(I1,I1)=A(I3,I2)
C      READY B MATRIX FOR EGZF
C      I4=0
C      DO 20 I5=1,N
C      DO 20 I6=1,I5
C      I4=I4+1
C      20 B(I4,I1)=B(I6,I5)
C      RMEGZF--CALLING SEQUENCE TO EGZF
C      CALL RMEGZF (A,B,VECTOR,N,NOGO)
C      IF(NOGO) 100,25,100
C
C      EGZF STORAGE TO FORTRAN STORAGE
C      25 K1=0
C      DO 27 I7=1,N
C      DO 26 I8=1,I7
C      K1=K1+1
C      26 B(I8,I7)=VECTOR(K1,I)
C      K2=(I7-1)*N+I7
C      27 A(I7,I1)=A(K2,I1)

```


JACOBI - EFF. SOURCE STATEMENT - IFN(5) -

```

C      ARRANGE EIGENVALUES IN DECREASING ORDER
C      REARRANGE EIGENVECTORS TO CORRESPOND TO EIGENVALUES
C      NORMALIZE EIGENVECTORS W.R.T. THEIR LARGEST ELEMENT
DO 90 J1=1,MODES
  VALUE(J1)=0.
  DO 40 J2=J1,N
    IF(VALUE(J1)-A(J2,1)) 30,40,40
30  VALUE(J1)=A(J2,1)
    L1=J2
40  CONTINUE
    A(L1,1)=A(J1,1)
    DO 50 J3=1,N
      VECTOR(J3,J1)=B(J3,L1)
50  B(J3,L1)=B(J3,J1)
    FNORM=0.
    DO 70 J4=1,N
      IF(ABS(FNORM)-ABS(VECTOR(J4,J1))) 60,70,70
60  FNORM=VECTOR(J4,J1)
70  CONTINUE
    DO 80 J5=1,N
      VECTOR(J5,J1)=VECTOR(J5,J1)/FNORM
      VALP=1./VALUE(J1)
      WRITE (6,85) VALP,FNORM,(VECTOR(J5,J1),J5=1,N)
85  FORMAT (1H0,4X,13HEIGENVALUE = ,E12.5,10X,33HEIGENVECTOR NORMALIZI
      2NG FACTOR = ,E12.5,10X,11HEIGENVECTOR/(E106.5))
90  CONTINUE
100 RETURN
    END

```

SIBFTC BLK

LIST DEAK

BLOCK DATA

COMMON /ELEM/NORDER,MPOLY,JN(5),NORDER,NPOLY,ACOE(10,10)

COMMON /GAUSS/NORD,X(8),W(8)

REAL JN

DATA MPOLY/5/

DATA NORDER,NPOLY/9,6/

DATA JN/1.8411837,5.3314427,8.5363163,11.706004,14.863588/

DATA NORD,X,W/8,0.095012509,0.28160355,0.45801677,0.61787624,

20.7554044,0.8656312,0.94457502,0.98940093,0.18945061,0.18260341,

30.16915651,0.14959598,0.12462897,0.095158511,0.062253623,

40.027152459/

END

SLUSH

02/4

518FTC BESSEL LIST,DECK

```

SUBROUTINE BESSSEL(ARG,B1,DBDX)
  DIMENSION A(7),B(7),C(7),D(7),E(7),F(7)
  DOUBLE PRECISION DPARG,DPSUM,DPTH,DPTH1,DPI,DPSUM1
  DATA (A(I),I=1,7)/.5, -.56249985, .21093573, -.03954229, .0044331
  *9, -.00031761, .00001109 /
  DATA (B(I),I=1,7)/.79788456, .00000156, .01659667, .00017105, -.00
  *249511, .00113653, -.00020033 /
  DATA (C(I),I=1,7)/-2.35619449, .12499612, .00005650, -.00637879, .
  *00074348, .00079824, -.00029166 /
  DATA (D(I),I=1,7)/1.0, -2.2499997, 1.2656208, -.3163846, .0444479
  *, -.0039444, .0002100 /
  DATA (E(I),I=1,7)/.79788456, -.00000077, -.00552740, -.00009512,
  *.00137237, -.00072805, .00014476 /
  DATA (F(I),I=1,7)/-.78539816, -.04166397, -.00003954, .00262573,
  *-.00054125, -.00029333, .00013558 /
  DPTH=0.0
  DPSUM=0.0
  IF (ARG)9990,1000,100
100 IF (ARG<3.) 200,200,500
200 DPARG=ARG/3.0
  DO 300 I=1,7
    DPTH=DPTH+D(I)*DPARG**(2*I-2)
300 DPSUM=DPSUM+A(I)*DPARG**(2*I-2)
  B1= ARG*DPSUM
  DBDX=DPTH-B1/ARG
  GO TO 10000
500 DPARG=3.0/ARG
  DPSUM1=0.0
  DPTH1=0.0
  DO 600 I=1,7
    DPI=DPARG**(I-1)
    DPTH1=DPTH1+F(I)*DPI
    DPSUM1=DPSUM1+E(I)*DPI
    DPTH=DPTH+C(I)*DPI
600 DPSUM=DPSUM+B(I)*DPI
  DUM=SQRT(1.0/ARG)
  DPTH=ARG+DPTH
  DPTH1=DPTH1+ARG
  B1=DUM *DPSUM*ACOS(DPTH)
  DBDX=DUM*DPSUM1*UCOS(DPTH1)-B1/ARG
  GO TO 10000
1000 B1=0.0
  DBDX=.5
  GO TO 10000
9990 WRITE (6,9991) ARG
9991 FORMAT (1H0,4X,20HBESSEL FUNCTION ARGUMENT OF ,E12.5.
  21H IS INVALID)
  CALL EXIT
10000 RETURN
  END

```

\$IBFTC PNODES LIST DECK

PMDDBS

- EFN

SOURCE STATEMENT

- IFN(S)

-

```

SUBROUTINE PMDDBS( X,BM1,BM1D )
C   COMPUTES MODIFIED BESSEL FCN 1 AND ITS DERIVATIVE BY POLY. APPROX.
  DIMENSION A(6),B(6),C(9),D(9)
  DOUBLE PRECISION T,S,Z,BMD
  DATA A/.00032411,.00301532,.02658733,.15084934,.51498869,.8789059
  *4 /,B /.0045813,.0360768,.2659732,1.2067492,3.0899424,3.5156229 /
  DATA C/-.00420059,.01787654,-.02895312,.02282967,-.01031555,
  *.00163801,-.00362018,-.03988024,.39894228 /,      D/.00392377,-.
  *.01647633,.02635537,-.02057706,.00916261,-.00157565,.00225319,
  *.01328592,.39894228 /
  IF ( X .EQ. 0. ) GO TO 600
  100 T = X/3.75
  120 S = 1*T
  150 IF ( X .GT. 3.75 ) GO TO 200
  160 BM1 =X*(((A(1) )*S+A(2))*S+A(3))*S+A(4))*S+A(5))*S+A(6))*S+.5
  *)
  BMD= (((B(1) )*S+B(2))*S+B(3))*S+B(4))*S+B(5))*S+B(6))*S+1.
  GO TO 400
  200 Z=X
  BM1=DEXP (Z)/DSQRT (Z) * ( (((C(1)/T +C(2))/T+C(3))/T+C(4))
  */T+C(5))/T+C(6))/T+C(7))/T+C(8))/T+C(9) )
  BMD=DEXP (Z)/DSQRT (Z) * ( (((D(1)/T +D(2))/T+D(3))/T+D(4))
  */T+D(5))/T+D(6))/T+D(7))/T+D(8))/T+D(9) )
  400 BM1D = BMD-BM1/X
  500 RETURN
  600 BM1 = 0.
  BM1D = .5
  GO TO 500
  END

```

SLOSH

09/2

SIBFTC SOLVE LIST, DECK

SOLVE - EFM SOURCE STATEMENT - IFN(S) -

```

CSOLVE  LINEAR EQUATION SOLVER WITH ITERATIVE IMPROVEMENT  VERSION IV  SLV
SUBROUTINE SOLVE(NN,A,B,IN,EPS,ITMAX,X,IT)  SLV
C SOLVES AX=B WHERE A IS NNxN MATRIX AND B IS Nx1 VECTOR  SLV
C IN=  SLV
C 1 FOR FIRST ENTRY  SLV
C 2 FOR SUBSEQUENT ENTRIES WITH NEW B  SLV
C 3 TO RESTORE A AND B  SLV
C EPS AND ITMAX ARE PARAMETERS IN THE ITERATION  SLV
C IT=  SLV
C -1 IF A IS SINGULAR  SLV
C 0 IF NOT CONVERGENT  SLV
C NUMBER OF ITERATIONS IF CONVERGENT  SLV
C CALLS MAP SUBROUTINES ILOG2,DOT,SDOT AND DAD  SLV
C  SLV
C TO MODIFY DIMENSIONS, CHANGE THE NEXT 3 (NOT 2 BUT 3) CARDS.  SLV
DIMENSION A(10,10),B(10),X(10),AA(10,10),DX(10),R(10),
2      Z(10),RM(10),IRP(10)
MA=10
C MA MUST = DECLARED DIMENSION OF SYSTEM  SLV
EQUIVALENCE(R,DX)  SLV
GO TO (1000,2000,3000),IN  SLV
1000 N=NN  SLV
NM1=N-1  SLV
NP1=N+1  SLV
C  SLV
C EQUILIBRATION  SLV
C  SLV
DO 510 I=1,N  SLV
KTOP=ILOG2(A(I,1))  SLV
DO 503 J=2,N  SLV
503 KTOP=MAX0(KTOP,ILOG2(A(I,J)))  SLV
RM(I)=2.0**(-KTOP)  SLV
DO 509 J=1,N  SLV
509 A(I,J)=A(I,J)*RM(I)  SLV
510 CONTINUE  SLV
C  SLV
C SAVE EQUILIBRATED DATA  SLV
C  SLV
DO 548 I=1,N  SLV
DO 548 J=1,N  SLV
548 AA(I,J)=A(I,J)  SLV
C  SLV
C GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING  SLV
C  SLV
DO 99 M=1,NM1  SLV
TOP=ABS (A(M,M))  SLV
IMAX=M  SLV
DO 12 I=M,N  SLV
IF (TOP-ABS (A(I,M)))10,12,12  SLV
10 TOP=ABS (A(I,M))  SLV
IMAX=I  SLV
12 CONTINUE  SLV
IF (TOP)14,13,14  SLV
13 IT=-1  SLV
C *SINGULAR*  SLV

```

SOLVE

- EFL

SOURCE STATEMENT

- IFN(5) -

```

      RETURN
14      IRP(N)=IMAX
23      IF (IMAX=N) 20,29,24
24      DO 25 J=1,N
          TEMP=A(N,J)
          A(N,J)=A(IMAX,J)
          A(IMAX,J)=TEMP
25      MP1=N+1
29      DO 33 I=MP1,N
          EM=A(I,M)/A(M,M)
          A(I,M)=EM
          IF (EM) 31,33,31
31      DO 32 J=MP1,N
32      A(I,J)=A(I,J)-A(M,J)*EM
33      CONTINUE
99      CONTINUE
      IRP(N)=N
      IF (A(N,N)) 120,113,120
113      IT=-1
      RETURN
120      CONTINUE
C      STORAGE FOR A NOW CONTAINS TRIANGULAR L AND U SO THAT (L+I)*U=A
C
C      DUPLICATE INTERCHANGES IN DATA
C
      DO 229 I=1,N
          IP=IRP(I)
          IF (I=IP) 221,229,221
221      DO 222 J=1,N
          TEMP=AA(I,J)
          AA(I,J)=AA(IP,J)
          AA(IP,J)=TEMP
222
229      CONTINUE
C
C      PROCESS RIGHT HAND SIDE
C
2000      CONTINUE
      DO 601 I=1,N
601      B(I)=B(I)+RM(I)
      DO 609 I=1,NM1
          IP=IRP(I)
          TEMP=B(I)
          B(I)=B(IP)
          B(IP)=TEMP
609      CONTINUE
C
C      SOLVE FOR FIRST APPROXIMATION TO X
C
199      DO 200 I=1,N
200      Z(I)=-SDOT(I-1,A(I,1),NA,Z(I),1,-B(I))
          DO 201 K=1,N
              I=NP1-K
201      X(I)=-SDOT(N-1,A(I,I+1),NA,X(I+1),1,-Z(I))/A(I,I)
C
C      ITERATIVE IMPROVEMENT
C

```

SOLVE

- EFN

SOURCE STATEMENT

- IFN(S)

```

IF(ITMAX/370,370,300)
300 TOP=0.0
DO 303 I=1,N
303 TOP=AMAX1(TOP,ABS(X(I)))
EPSX=EPS*TOP
DO 369 IT=1,ITMAX
C      FIND RESIDUALS
DO 319 I=1,N
319 R(I)=-DOT(N,AA(I,1),MA,X(I),1,-B(I))
C      FIND INCREMENT
DO 329 I=1,N
329 Z(I)=-SDOT(I-1,A(I,1),MA,Z(I),1,-R(I))
DO 339 K=1,N
I=NP1-K
339 DX(I)=-SDOT(N-I,A(I,1+1),MA,DX(I+1),1,-Z(I))/A(I,1)
C      INCREMENT AND TEST CONVERGENCE
TOP=0.0
DO 342 I=1,N
TEMP=X(I)
X(I)=DAD(X(I),DX(I))
DELX=ABS(X(I)-TEMP)
TOP=AMAX1(TOP,DELX)
342 CONTINUE
IF(TOP-EPSX)381,381,369
369 CONTINUE
370 IT=0
381 RETURN
C
C      RESTORE A AND B
C
3000 CONTINUE
DO 709 K=1,N
I=NP1-K
IP=IKP(I)
IF(1-IP)701,709,701
701 TEMP=B(I)
B(I)=B(IP)
B(IP)=TEMP
DO 702 J=1,N
TEMP=AA(I,J)
AA(I,J)=AA(IP,J)
702 AA(IP,J)=TEMP
709 CONTINUE
DO 729 I=1,N
B(I)=B(I)/RH(I)
DO 729 J=1,N
A(I,J)=AA(I,J)/RH(I)
729 CONTINUE
RETURN
END

```

BIBMAP DØT	84,DECK,LIST		
* DØT AND FRIENDS	RØUTINES FØR USE WITH SØLVE		DØT40010
ENTRY	DØT (N,A(1),MA,B(1),MB,C)	DOUBLE INNER PRØDUCT	DØT40020
ENTRY	SDØT (N,A(1),MA,B(1),MB,C)	INNER PRØDUCT	DØT40030
ENTRY	ILØG2 (A)	FLØATING PØINT EXPØNENT	DØT40040
ENTRY	DAD (A,B)	ADD WITH RØUND	DØT40050
* SNAD	MACRØ M	STØRE NEGATIVE ØF ADDRESS IN DECREMENT	DØT40060
	SUB	=Ø100000 CØMPLEMENT IF PØSITIVE	DØT40070
	ALS	18	DØT40080
	STD	M	DØT40090
	ENDM	SNAD	DØT40100
* DØT	SAVE	1,2,4	DØT40110
	STZ	S	DØT40120
	STZ	S+1	DØT40130
	CLA*	8,4 C	DØT40140
	LDQ	C+1	DØT40150
	STØ	C	DØT40160
	CLA*	3,4 N	DØT40170
	TZE	NØNE SKIP LØØP IF N = 0	DØT40180
	STØ	N	DØT40190
	CLA	4,4 BASE ADDRESS ØF A	DØT40200
	PAC	,1 X1=-(BASE ØF A)	DØT40210
	CLA*	5,4 MA	DØT40220
	SNAD	MA	DØT40230
	CLA	6,4 BASE ADDRESS ØF B	DØT40240
	PAC	,2 X2=-(BASE ØF B)	DØT40250
	CLA*	7,4 MB	DØT40260
	SNAD	MB	DØT40270
	LXA	N,4 X4=A	DØT40280
LØØP	LDQ	0,1 A(1)	DØT40290
	FMP	0,2 B(1)	DØT40300
	DFAD	S	DØT40310
	DST	S	DØT40320
MA	TXI	**+1,1,** (X1)=(X1)+MA	DØT40330
MB	TXI	**+1,2,** (X2)=(X2)+MB	DØT40340
	TIX	LØØP,4,1 END ØF MAIN LØØP	DØT40350
NØNE	DFAD	C	DØT40360
	FRN		DØT40370
	RETURN	DØT	DØT40380
			DØT40390
			DØT40400
			DØT40410
SDØT	SAVE	1,2,4	DØT40420
	STZ	S	DØT40430
	CLA*	8,4	DØT40440
	STØ	C	DØT40450
	CLA*	3,4	DØT40460
	TZE	SNØNE	DØT40470
	STØ	N	DØT40480
	CLA	4,4	DØT40490
	PAC	,1	DØT40500
	CLA*	5,4	DØT40510
	SNAD	SMA	DØT40520
	CLA	6,4	DØT40530
	PAC	,2	DØT40540
	CLA*	7,4	DØT40550
	SNAD	SMB	DØT40560
	LXA	N,4	DØT40570
SLØØP	LDQ	0,1	DØT40580
	FMP	0,2	DØT40590
	FAD	S	DØT40600

STØ	S			DØT40610
SMA TXI	++1,1,***			DØT40620
SMB TXI	++1,2,***			DØT40630
SNØNE TIX	SLEØP,4,1			DØT40640
FAD	C			DØT40650
RETURN	SDØT			DØT40660
				DØT40670
ILØG2 CAL*	3,4			DØT40680
ANA	=Ø377000000000			DØT40690
SUB	=Ø200000000000			DØT40700
ARS	27			DØT40710
				DØT40720
TRA	1,4			DØT40730
				DØT40740
DAD CLA*	3,4			DØT40750
FAD*	4,4			DØT40760
FRN				DØT40770
TRA	1,4			DØT40780
				DØT40790
				DØT40800
				DØT40810
				DØT40820
				DØT40830
				DØT40840
				DØT40850
\$IRMAP RWEØ2F	800,LIST,REF,DECK,M90,RELMØD,NØ(),NØMFTC			RWEG2F0000
*RWEØ2F	RØUTINE BETWEEN JACØ81 AND EG2F			RWEG2F0000
	TØ SET UP SYMBOLIC CALLING SEQUENCE			RWEG2F0000
ENTRY	RWEØ2F			RWEG2F0000
RWEØ2F SAVE	(1,2,4)	SAVE INDEX REGISTERS.		RWEG2F0000
CLA	3,4			RWEG2F0000
STA	SUB+1			RWEG2F0000
CLA	4,4	B.		RWEG2F0000
PAX	,2			RWEG2F0000
SXD	SUB+1,2	A.		RWEG2F0000
CLA	5,4	VECTØR.		RWEG2F0000
STA	SUB+2			RWEG2F0000
CLA*	6,4	N.		RWEG2F0000
PAX	,2			RWEG2F0000
SXD	SUB+2,2			RWEG2F0000
CLA	7,4	NØGØ.		RWEG2F0000
STA	TEMP			RWEG2F0000
ØCT	4760000000016	EMTM ØN 7094 CØMPUTER, NØP ØN 7090		RWEG2F0000
SUB TSX	EG2F,4	EG2F CALLING SEQUENCE.		RWEG2F0000
PZE	***,***	A,,B.		RWEG2F0000
PZE	***,***	V,,N.		RWEG2F0000
TRA	ERRØR	ERRØR EXIT.		RWEG2F0000
STZ*	TEMP	NØRMAL EXIT.		RWEG2F0000
ØCT	Ø760000000016	LMTM ØN 7094 CØMPUTER, NØP ØN 7090		RWEG2F0000
RETURN	RWEØ2F			RWEG2F0000
TEMP PZE				RWEG2F0000
ERRØR TZE	ACC			RWEG2F0000
CLS	=1	NØGØ		RWEG2F0000
STØ*	TEMP	=-1.		RWEG2F0000
TRA	RETI			RWEG2F0000
ACC CLA	=1	NØGØ		RWEG2F0000
STØ*	TEMP	=1.		RWEG2F0000
TRA	RETI			RWEG2F0000
EJECT				RWEG2F0000
TIL	SHARE SUBRØUTINE EG2F			RWEG2F0000
EG2F	SXD EG2F+Ø209,4			EG2F0003

SXD EG2F+0210,1		EG2F0004
SXD EG2F+0211,2		EG2F0005
CLA 1,4		EG2F0006
STA EG2F+0215	LCC A MATRIX	EG2F0007
STA EG2F+0042		EG2F0008
STA EG2F+0066		EG2F0009
STA EG2F+0068		EG2F0010
ARS 18		EG2F0011
STA EG2F+0029	LCC B MATRIX	EG2F0012
STA EG2F+0214		EG2F0013
CLA 2,4		EG2F0014
STA EG2F+0216	LCC V MATRIX	EG2F0015
STA EG2F+0030		EG2F0016
STD EG2F+0207		EG2F0017
ALS 18		EG2F0018
STD EG2F+0042		EG2F0019
LDQ EG2F+0207		EG2F0020
MPY EG2F+0207		EG2F0021
ALS 17		EG2F0022
CQM		EG2F0023
ADD EG2F+0208		EG2F0024
STD EG2F+0213	-(N SQUARED)	EG2F0025
CLA EG2F+0207		EG2F0026
CQM		EG2F0027
ADD EG2F+0208		EG2F0028
STD EG2F+0029		EG2F0029
STD EG2F+0212		EG2F0030
TSX EG2F+0217,4		EG2F0031
PZE *,0,*		EG2F0032
PZE *		EG2F0033
TRA EG2F+0053		EG2F0034
TSX EG2F+0072,4	FORM M(-1)TAM(-1)	EG2F0035
LXD EG2F+0209,4		EG2F0036
CLA 1,4		EG2F0037
TPL EG2F+0038		EG2F0038
CLS EG2F+0207		EG2F0039
TRA EG2F+0039		EG2F0040
CLA EG2F+0207		EG2F0041
ARS 18		EG2F0042
STD EG2F+0043		EG2F0043
TSX EG2F+0370,4		EG2F0044
PZE *,0,*		EG2F0045
PZE *		EG2F0046
TRA EG2F+0055		EG2F0047
TRA EG2F+0060		EG2F0048
TSX EG2F+0181,4		EG2F0049
TSX EG2F+0132,4		EG2F0050
TSX EG2F+0191,4		EG2F0051
LXD EG2F+0209,4		EG2F0052
LXD EG2F+0210,1		EG2F0053
LXD EG2F+0211,2		EG2F0054
TRA 4,4		EG2F0055
PXD		EG2F0056
TRA EG2F+0056		EG2F0057
CLA EG2F+0344		EG2F0058
LXD EG2F+0210,1		EG2F0059
LXD EG2F+0211,2		EG2F0060
LXD EG2F+0209,4		EG2F0061
TRA 3,4		EG2F0062
STD COMMON+000		EG2F0063
CLA EG2F+0207		EG2F0064

PDX 0,1	N	EG2F0065
CØM		EG2F0066
STD EG2F+0069		EG2F0067
LXD EG2F+0342,2		EG2F0068
CLA *,2		EG2F0069
FDP CØMMØN+000		EG2F0070
STQ *,2		EG2F0071
TXI EG2F+0070,2,*		EG2F0072
TIX EG2F+0066,1,1		EG2F0073
TRA EG2F+0049		EG2F0074
SXD CØMMØN+004,4		EG2F0075
CLA EG2F+0213		EG2F0076
STD EG2F+0101		EG2F0077
STD EG2F+0124		EG2F0078
STD EG2F+0127		EG2F0079
CLA EG2F+0212		EG2F0080
STD EG2F+0126		EG2F0081
CLA EG2F+0214		EG2F0082
STA EG2F+0116		EG2F0083
CLA EG2F+0215		EG2F0084
STA EG2F+0117		EG2F0085
CLA EG2F+0216		EG2F0086
STA EG2F+0099		EG2F0087
STA EG2F+0118		EG2F0088
STA EG2F+0119		EG2F0089
TSX EG2F+0097,4		EG2F0090
CLA EG2F+0216		EG2F0091
STA EG2F+0117		EG2F0092
CLA EG2F+0215		EG2F0093
STA EG2F+0099		EG2F0094
STA EG2F+0118		EG2F0095
STA EG2F+0119		EG2F0096
TSX EG2F+0097,4		EG2F0097
LXD CØMMØN+004,4		EG2F0098
TRA 1,4		EG2F0099
SXD CØMMØN+003,4		EG2F0100
LXA EG2F+0342,4		EG2F0101
STZ *,4		EG2F0102
TXI EG2F+0101,4,-1		EG2F0103
TXH EG2F+0099,4,*		EG2F0104
CLA EG2F+0341		EG2F0105
STØ CØMMØN+002		EG2F0106
LXA EG2F+0342,4		EG2F0107
SXD EG2F+0122,4		EG2F0108
CLA EG2F+0122		EG2F0109
STD CØMMØN+001		EG2F0110
ADD CØMMØN+002		EG2F0111
STD EG2F+0122		EG2F0112
CLA CØMMØN+002		EG2F0113
ADD EG2F+0341		EG2F0114
STD CØMMØN+002		EG2F0115
STZ CØMMØN+000		EG2F0116
LXD CØMMØN+000,2	00	EG2F0117
LXD CØMMØN+001,1		EG2F0118
LDQ *,1		EG2F0119
FMP *,2		EG2F0120
FAD *,4		EG2F0121
STØ *,4		EG2F0122
TXI EG2F+0121,1,-1		EG2F0123
TXI EG2F+0122,2,-1		EG2F0124
TXH EG2F+0116,1,*		EG2F0125

TXI EG2F+0124,4,-1	EG2F0126
TXL EG2F+0130,4,*	EG2F0127
LXD CØMMØN+000,2 00	EG2F0128
TXI EG2F+0127,2,*	EG2F0129
TXL EG2F+0106,2,*	EG2F0130
SXD CØMMØN+000,2 00	EG2F0131
TRA EG2F+0114	EG2F0132
LXD CØMMØN+003,4	EG2F0133
TRA 1,4	EG2F0134
SXD CØMMØN+004,4	EG2F0135
CLA EG2F+0214	EG2F0136
STA EG2F+0154	EG2F0137
CLA EG2F+0216	EG2F0138
STA EG2F+0155	EG2F0139
STA EG2F+0165	EG2F0140
CLA EG2F+0213	EG2F0141
STD EG2F+0163	EG2F0142
STD EG2F+0168	EG2F0143
CLA EG2F+0212	EG2F0144
STD EG2F+0162	EG2F0145
STD EG2F+0169	EG2F0146
CLA EG2F+0341	EG2F0147
STD CØMMØN+002	EG2F0148
STZ CØMMØN+000	EG2F0149
STZ CØMMØN+001	EG2F0150
LXA EG2F+0342,4	EG2F0151
LXD CØMMØN+001,1	EG2F0152
LXD CØMMØN+000,2 00	EG2F0153
CLA CØMMØN+002	EG2F0154
STD EG2F+0158	EG2F0155
STZ CØMMØN+003	EG2F0156
LDQ *,1	EG2F0157
FMP *,2	EG2F0158
FAD CØMMØN+003	EG2F0159
STØ CØMMØN+003	EG2F0160
TXI EG2F+0159,1,*	EG2F0161
CLA EG2F+0158	EG2F0162
ADD EG2F+0341	EG2F0163
STD EG2F+0158	EG2F0164
TXI EG2F+0163,2,*	EG2F0165
TXH EG2F+0154,2,*	EG2F0166
CLA CØMMØN+003	EG2F0167
STØ *,4	EG2F0168
TXI EG2F+0167,4,-1	EG2F0169
SXD CØMMØN+000,4 00	EG2F0170
TXL EG2F+0179,4,*	EG2F0171
TXH EG2F+0149,4,*	EG2F0172
CLA EG2F+0169	EG2F0173
ADD EG2F+0212	EG2F0174
STD EG2F+0169	EG2F0175
CLA CØMMØN+002	EG2F0176
ADD EG2F+0341	EG2F0177
STD CØMMØN+002	EG2F0178
ADD CØMMØN+001	EG2F0179
STD CØMMØN+001	EG2F0180
TRA EG2F+0149	EG2F0181
LXD CØMMØN+004,4	EG2F0182
TRA 1,4	EG2F0183
CLA EG2F+0212	EG2F0184
STD EG2F+0199	EG2F0185
CLA EG2F+0213	EG2F0186

STD EG2F+0200	EG2F0187
STD EG2F+0205	EG2F0188
CLA EG2F+0216	EG2F0189
STA EG2F+0194	EG2F0190
STA EG2F+0195	EG2F0191
STA EG2F+0196	EG2F0192
STA EG2F+0197	EG2F0193
STZ COMMON+000	EG2F0194
LXD COMMON+000,1 00	EG2F0195
LXD COMMON+000,2 00	EG2F0196
CLA *,1	EG2F0197
LDQ *,2	EG2F0198
STQ *,1	EG2F0199
STØ *,2	EG2F0200
TXI EG2F+0199,1,-1	EG2F0201
TXI EG2F+0200,2,*	EG2F0202
TXH EG2F+0194,2,*	EG2F0203
CLA COMMON+000	EG2F0204
ADD EG2F+0212	EG2F0205
ADD EG2F+0341	EG2F0206
STD COMMON+000	EG2F0207
TXH EG2F+0192,1,*	EG2F0208
TRA 1,4	EG2F0209
PZE	EG2F0210
PZE 0,0,1	EG2F0211
PZE	EG2F0212
PZE	EG2F0213
PZE	EG2F0214
PZE	EG2F0215
PZE	EG2F0216
PZE	EG2F0217
PZE	EG2F0218
PZE	EG2F0219
SXD COMMON+010,1	EG2F0220
SXD COMMON+011,2	EG2F0221
CLA 2,4	EG2F0222
STA EG2F+0295	EG2F0223
STA EG2F+0296	EG2F0224
SUB EG2F+0344	EG2F0225
STA EG2F+0266	EG2F0226
STA EG2F+0283	EG2F0227
CLA 1,4	EG2F0228
STD EG2F+0311	EG2F0229
SXD COMMON+004,4	EG2F0230
LXD EG2F+0341,2	EG2F0231
STA EG2F+0282	EG2F0232
STA EG2F+0236	EG2F0233
STA EG2F+0239	EG2F0234
STA EG2F+0326	EG2F0235
STA EG2F+0328	EG2F0236
STA EG2F+0258	EG2F0237
STA EG2F+0259	EG2F0238
CLA *	EG2F0239
STØ COMMON+013,2	EG2F0240
CLA EG2F+0343	EG2F0241
STØ *	EG2F0242
CLA EG2F+0342	EG2F0243
STØ COMMON+005	EG2F0244
CLA COMMON+005	EG2F0245
ADD EG2F+0344	EG2F0246
STØ COMMON+005	EG2F0247

ADD EG2F+0259		EG2FC248
STA EG2F+0259		EG2FC249
STA EG2F+0273		EG2FC250
STA EG2F+0285		EG2FC251
STA EG2F+0284		EG2FC252
STA EG2F+0308		EG2FC253
STA EG2F+0304		EG2FC254
SXD EG2F+0267,2	COMP LAMBDA	EG2FC255
LXD EG2F+0342,4		EG2FC256
LXD EG2F+0341,1		EG2FC257
SXD EG2F+0264,1		EG2FC258
LXD EG2F+0342,1		EG2FC259
STZ COMMON+006		EG2FC260
LDQ *,4		EG2FC261
FMP *,1		EG2FC262
FAD COMMON+006		EG2FC263
STØ COMMON+006		EG2FC264
TXI EG2F+0263,4,-1		EG2FC265
TXI EG2F+0264,1,-1		EG2FC266
TXH EG2F+0258,1,*		EG2FC267
FDP COMMON+013,1		EG2FC268
STQ *,1		EG2FC269
TXH EG2F+0269,1,*		EG2FC270
TRA EG2F+0271		EG2FC271
LXD EG2F+0264,1		EG2FC272
TXI EG2F+0255,1,-1		EG2FC273
SXD EG2F+0275,2		EG2FC274
LXD EG2F+0342,4		EG2FC275
STZ *,4		EG2FC276
TXI EG2F+0275,4,-1		EG2FC277
TXH EG2F+0273,4,*		EG2FC278
TXI EG2F+0277,2,-1		EG2FC279
SXD EG2F+0290,2		EG2FC280
LXD EG2F+0342,4		EG2FC281
LXD EG2F+0341,2		EG2FC282
LXD EG2F+0342,1		EG2FC283
SXD EG2F+0288,2		EG2FC284
LDQ *,4	A LAMBDA TIMES A ROW	EG2FC285
FMP *,2		EG2FC286
FAD *,1	VECTOR WHICH WILL BE	EG2FC287
STØ *,1	THE NEW ROW	EG2FC288
TXI EG2F+0287,4,-1		EG2FC289
TXI EG2F+0288,1,-1		EG2FC290
TXH EG2F+0282,1,*		EG2FC291
TXI EG2F+0290,2,-1		EG2FC292
TXH EG2F+0280,2,*	END COMP. NEW ROW	EG2FC293
STZ COMMON+008		EG2FC294
LXD EG2F+0288,2		EG2FC295
SXD EG2F+0303,2		EG2FC296
LXD EG2F+0342,4		EG2FC297
LDQ *,4		EG2FC298
FMP *,4		EG2FC299
STØ COMMON+007		EG2FC300
LDQ COMMON+007		EG2FC301
FMP COMMON+014,4		EG2FC302
FAD COMMON+008		EG2FC303
STØ COMMON+008		EG2FC304
TXI EG2F+0303,4,-1		EG2FC305
TXH EG2F+0295,4,*		EG2FC306
CLA *,1		EG2FC307
FSB COMMON+008		EG2FC308

STØ CØMMØN+014,2		EG2F0309
CLA EG2F+0343		EG2F0310
STØ *,1		EG2F0311
TXI EG2F+0310,2,-1		EG2F0312
LXD EG2F+0267,1		EG2F0313
TXH EG2F+0242,2,*		EG2F0314
SXD EG2F+0321,2		EG2F0315
TXI EG2F+0314,2,-2	-(N+2)	EG2F0316
SXD EG2F+0334,2		EG2F0317
LXD EG2F+0342,1		EG2F0318
CLA CØMMØN+014,1		EG2F0319
TSX EG2F+0345,4		EG2F0320
TRA EG2F+0339		EG2F0321
STØ CØMMØN+014,1		EG2F0322
TXI EG2F+0321,1,-1		EG2F0323
TXH EG2F+0316,1,*		EG2F0324
LXD EG2F+0341,6		EG2F0325
LXD EG2F+0342,1		EG2F0326
TXI EG2F+0325,2,-1		EG2F0327
SXD EG2F+0330,4		EG2F0328
CLA *,1		EG2F0329
FDP CØMMØN+012,2		EG2F0330
STØ *,1		EG2F0331
TXI EG2F+0330,1,-1		EG2F0332
TXH EG2F+0326,1,*		EG2F0333
SXD EG2F+0332,2		EG2F0334
TXI EG2F+0333,4,*		EG2F0335
TXI EG2F+0334,2,-1		EG2F0336
TXH EG2F+0325,2,*		EG2F0337
LXD CØMMØN+010,1		EG2F0338
LXD CØMMØN+011,2		EG2F0339
LXD CØMMØN+004,4		EG2F0340
TRA 4,4		EG2F0341
LXD CØMMØN+004,4		EG2F0342
TRA 3,4		EG2F0343
PZE 1,0,-1		EG2F0344
PZE		EG2F0345
DEC -1.		EG2F0346
PZE 1		EG2F0347
STØ CØMMØN+000	SAVE ARGUMENT SIGN	EG2F0348
SSP	N	EG2F0349
TZE EG2F+0365	GØ TØ EXIT IF ZERØ	EG2F0350
STØ CØMMØN+001	SAVE N	EG2F0351
ANA EG2F+0368	CØMPUTE TRIAL VALUE, X	EG2F0352
LRS 1	X	EG2F0353
ADD CØMMØN+001	X	EG2F0354
LRS 1	X	EG2F0355
ADD EG2F+0369	X	EG2F0356
SXD CØMMØN+000,4	00 RETURN ADDRESS	EG2F0357
LXA EG2F+0346,4	SET INDEX FØR 3 ITERATIØNS	EG2F0358
STØ CØMMØN+002	SAVE X	EG2F0359
CLA CØMMØN+001	CØMPUTE SQUARE RØØT	EG2F0360
FDH CØMMØN+002	X N/X	EG2F0361
STØ CØMMØN+003	X	EG2F0362
CLA CØMMØN+003	X N/X	EG2F0363
FAD CØMMØN+002	X + X	EG2F0364
SUB EG2F+0368	X DIVIDE BY 2	EG2F0365
TXI EG2F+0356,4,1	REPEAT LØØP	EG2F0366
LXD CØMMØN+000,4	00ØRE EXIT ADDRESS	EG2F0367
LDQ CØMMØN+000	TEST SIGN ØF ARGUMENT	EG2F0368
TQP 2,4	IF + , SKIP ØNE	EG2F0369

TRA 1,4	IF - , DO NOT SKIP ONE	EG2F0370
DEC 134217728,8657043456 0*2 EXP -127, 1/2 * EXP -64		EG2F0371
REM NYEVV 328 CARDS EIGENVALUES AND EIGENVECTORS		EG2F0372
SXD CØMMØN+000,1 00 SAVE IRS		EG2F0373
SXD CØMMØN+001,2		EG2F0374
SXD CØMMØN+002,4		EG2F0375
PXD	SAVE SLTS	EG2F0376
LXA EG2F+0694,1		EG2F0377
ALS 1		EG2F0378
SLT 5,1		EG2F0379
ADD EG2F+0375		EG2F0380
TIX EG2F+0375,1,1		EG2F0381
STA EG2F+0373		EG2F0382
CLA 1,4	M	EG2F0383
PAX 0,1		EG2F0384
STA EG2F+0488		EG2F0385
STA EG2F+0534		EG2F0386
STA EG2F+0536		EG2F0387
STA EG2F+0538		EG2F0388
STA EG2F+0558		EG2F0389
ADM 2,4	M+N	EG2F0390
STA EG2F+0482		EG2F0391
STA EG2F+0489		EG2F0392
STA EG2F+0539		EG2F0393
STA EG2F+0543		EG2F0394
CAL 2,4	N	EG2F0395
STA EG2F+0381		EG2F0396
ADD EG2F+0375	N+1	EG2F0397
STA EG2F+0464		EG2F0398
ALS 18		EG2F0399
STD EG2F+0604		EG2F0400
SUB EG2F+0695	N-1	EG2F0401
STD EG2F+0465		EG2F0402
CØM	-N	EG2F0403
STD EG2F+0560		EG2F0404
STD EG2F+0561		EG2F0405
SUB EG2F+0696	-(N+1)	EG2F0406
STD EG2F+0567		EG2F0407
LDQ 2,4	N**2	EG2F0408
MPY 2,4		EG2F0409
STQ CØMMØN+003		EG2F0410
CLA 1,4	M+N**2	EG2F0411
ADD CØMMØN+003		EG2F0412
STA EG2F+0441		EG2F0413
STA EG2F+0442		EG2F0414
STA EG2F+0611		EG2F0415
STA EG2F+0623		EG2F0416
STA EG2F+0625		EG2F0417
STA EG2F+0638		EG2F0418
STA EG2F+0641		EG2F0419
CØM	-V	EG2F0420
ADD EG2F+0696		EG2F0421
STØ CØMMØN+004		EG2F0422
PXD 0,1		EG2F0423
ADD CØMMØN+004	-(V-M)	EG2F0424
STD CØMMØN+004		EG2F0425
ADM EG2F+0567	-(V-M)-N	EG2F0426
STD CØMMØN+005		EG2F0427
CLA 2,4	FLØAT N	EG2F0428
SSP		EG2F0429
ADD EG2F+0697		EG2F0430

FAD EG2F+0697		EG2F0431
STØ CØMMØN+006		EG2F0432
CLA 2,4	SET NØ VECTØRS	EG2F0433
TPL EG2F+0594		EG2F0434
CLA EG2F+0689		EG2F0435
STØ EG2F+0528		EG2F0436
CLA EG2F+0704		EG2F0437
STØ CØMMØN+007		EG2F0438
CLA EG2F+0690		EG2F0439
STØ EG2F+0593		EG2F0440
TØV EG2F+0439		EG2F0441
STZ CØMMØN+008	CALCULATE NØRM	EG2F0442
LXA CØMMØN+003,7		EG2F0443
LDQ 0,1		EG2F0444
FMP 0,1		EG2F0445
TØV EG2F+0611		EG2F0446
FAD CØMMØN+008		EG2F0447
TØV EG2F+0651		EG2F0448
STØ CØMMØN+008		EG2F0449
TIX EG2F+0441,1,1		EG2F0450
SLF		EG2F0451
TSX EG2F+0655,4		EG2F0452
FDH CØMMØN+006		EG2F0453
STQ CØMMØN+006		EG2F0454
CLA CØMMØN+006		EG2F0455
SUB EG2F+0698		EG2F0456
STØ CØMMØN+008	FINAL NØRM	EG2F0457
CLA CØMMØN+006		EG2F0458
SUB EG2F+0699		EG2F0459
STØ CØMMØN+006	PRØVISIONAL NØRM	EG2F0460
LXA EG2F+0569,1	SET $-(J-2)N-(K-1)=0$	EG2F0461
CLA EG2F+0534	UNDERSET J=1	EG2F0462
STA EG2F+0487		EG2F0463
STA EG2F+0522		EG2F0464
CLA EG2F+0696	SET J-1=1	EG2F0465
STØ CØMMØN+003		EG2F0466
PDX 0,2	SET J-K=1	EG2F0467
TXH EG2F+0576,2	J-K GREATER THAN N-1	EG2F0468
CLA EG2F+0522	J TØ J+1	EG2F0469
ADD EG2F+0381		EG2F0470
STA EG2F+0522		EG2F0471
STA EG2F+0525		EG2F0472
STA EG2F+0557		EG2F0473
CLA EG2F+0487		EG2F0474
ADM EG2F+0464		EG2F0475
STA EG2F+0487		EG2F0476
STA EG2F+0540		EG2F0477
STA EG2F+0542		EG2F0478
LXA EG2F+0569,4	SET $(1-K)(N+1)=0$	EG2F0479
CLA EG2F+0534	SET K=1	EG2F0480
STA EG2F+0521		EG2F0481
STA EG2F+0524		EG2F0482
STA EG2F+0555		EG2F0483
CLA CØMMØN+006	NØRM	EG2F0484
SBM 0,1	-A(J,K)	EG2F0485
TPL EG2F+0564	GREATER THAN 0	EG2F0486
SLN 4	SPALLER	EG2F0487
SXD EG2F+0689,2	SAVE IRS	EG2F0488
SXD EG2F+0654,4		EG2F0489
CLA	A(J,J)	EG2F0490
FSB 0,4	-A(K,K)	EG2F0491

FDH 0,1	/A(J,K)	EG2F0492
STQ COMMON+011		EG2F0493
CLA COMMON+011		EG2F0494
TMI EG2F+0494		EG2F0495
SLN 3	SL3 0N IF G POSITIVE	EG2F0496
SSP		EG2F0497
SUB EG2F+0699	G	EG2F0498
STQ COMMON+011		EG2F0499
LDQ COMMON+011	G**2	EG2F0500
FMP COMMON+011		EG2F0501
FAD EG2F+0704	1+G**2	EG2F0502
STQ COMMON+012		EG2F0503
TSX EG2F+0655,4	(1+G**2)**1/2	EG2F0504
LRS 35		EG2F0505
FMP COMMON+011	G(1+G**2)**1/2	EG2F0506
FAD COMMON+012	+1+G**2	EG2F0507
STQ COMMON+011		EG2F0508
CLA EG2F+0705	(SINT)**2	EG2F0509
FDH COMMON+011		EG2F0510
STQ COMMON+011		EG2F0511
CLA COMMON+011		EG2F0512
TSX EG2F+0655,4	SIN T	EG2F0513
SLT 3		EG2F0514
SSM		EG2F0515
STQ COMMON+012		EG2F0516
CLA EG2F+0704		EG2F0517
FSB COMMON+011		EG2F0518
TSX EG2F+0655,4	COST	EG2F0519
STQ COMMON+011		EG2F0520
LXA EG2F+0569,2	SET MAIN MATRIX	EG2F0521
CLA EG2F+0560		EG2F0522
STD EG2F+0527	SET I-1=0	EG2F0523
CLA 0,2	A(K,I)	EG2F0524
LDQ 0,2	A(J,I)	EG2F0525
TSX EG2F+0673,4	B(K,I),B(J,I)	EG2F0526
STQ 0,2		EG2F0527
STQ 0,2		EG2F0528
TXI EG2F+0527,2,-1	I TO I+1	EG2F0529
TXH EG2F+0521,2	I=N	EG2F0530
HTR 0		EG2F0531
TRA EG2F+0607		EG2F0532
CAL EG2F+0689	J-K	EG2F0533
COM		EG2F0534
ADM EG2F+0654	+(1-K)(N+1)	EG2F0535
PDX 0,4		EG2F0536
LDQ 0,4	A(K,J) IN MQ	EG2F0537
LXD EG2F+0654,2	(1-K)(N+1)	EG2F0538
CLA 0,2	A(K,K) IN AC	EG2F0539
TSX EG2F+0673,4		EG2F0540
STQ 0,2	B(K,K)	EG2F0541
CLA 0,1	A(J,K) IN AC	EG2F0542
LDQ	A(J,J) IN MQ	EG2F0543
TSX EG2F+0673,4		EG2F0544
STQ	B(J,J)	EG2F0545
STQ 0,1	B(J,K)	EG2F0546
CLA COMMON+003	J	EG2F0547
COM		EG2F0548
ADD EG2F+0696		EG2F0549
PDX 0,2		EG2F0550
CAL EG2F+0689	K-J	EG2F0551
COM		EG2F0552

ADD EG2F+0696	M+K-J	EG2F0553
ARS 18		EG2F0554
ADD EG2F+0534		EG2F0555
STA EG2F+0556		EG2F0556
LXA EG2F+0569,4	SET I-1=0	EG2F0557
CLA 0,4	A(K,I)	EG2F0558
ST0 0,2	A(I,K)	EG2F0559
CLA 0,4	A(J,I)	EG2F0560
ST0 0,2	A(I,J)	EG2F0561
TXI EG2F+0560,4,-1	I T0 I+1	EG2F0562
TXL EG2F+0562,4	I=N	EG2F0563
TXI EG2F+0555,2		EG2F0564
LXD EG2F+0689,2	RESTORE IRS	EG2F0565
LXD EG2F+0654,4		EG2F0566
TNX EG2F+0569,2,1	J-K=1	EG2F0567
CLA EG2F+0521	K T0 K+1	EG2F0568
ADD EG2F+0381		EG2F0569
TXI EG2F+0568,4		EG2F0570
TXI EG2F+0478,1,-1		EG2F0571
PXD 0,1	M T0 M+1	EG2F0572
ADD C0MM0N+003	AND K=1	EG2F0573
ADD EG2F+0567		EG2F0574
PDX 0,1		EG2F0575
CLA C0MM0N+003		EG2F0576
ADD EG2F+0696		EG2F0577
TRA EG2F+0463		EG2F0578
SLT 4	IS ITERATION IDLE	EG2F0579
TRA EG2F+0579		EG2F0580
TRA EG2F+0458		EG2F0581
CLA C0MM0N+006	IS NORM FINAL	EG2F0582
LDQ C0MM0N+008		EG2F0583
TLQ EG2F+0456		EG2F0584
LXA EG2F+0694,1	RESTORE SLTS	EG2F0585
CLA EG2F+0373		EG2F0586
LRS 3		EG2F0587
LBT		EG2F0588
SLN 5,1		EG2F0589
LLS 1		EG2F0590
TIK EG2F+0585,1,1		EG2F0591
LXD C0MM0N+000,1	00 RESTORE IRS	EG2F0592
LXD C0MM0N+001,2		EG2F0593
LXD C0MM0N+002,4		EG2F0594
CLA C0MM0N+007		EG2F0595
HTR 0		EG2F0596
CLA 1,4	SETTING FOR VECTORS	EG2F0597
ARS 18		EG2F0598
ADD C0MM0N+003		EG2F0599
STA EG2F+0600		EG2F0600
STA EG2F+0603		EG2F0601
LXA C0MM0N+003,3	SET IDENTITY MATRIX	EG2F0602
STZ 0,1		EG2F0603
TIK EG2F+0600,1,1		EG2F0604
CLA EG2F+0704		EG2F0605
ST0 0,2		EG2F0606
TIK EG2F+0603,2		EG2F0607
CLA EG2F+0691		EG2F0608
TRA EG2F+0433		EG2F0609
SLN 2	SET MT T0 VECTORS	EG2F0610
LXD C0MM0N+004,2		EG2F0611
CLA C0MM0N+005		EG2F0612
TRA EG2F+0520		EG2F0613

CAL 0,1	UNDERFLOW	EG2F0614
ANA EG2F+0701		EG2F0615
SUB EG2F+0702		EG2F0616
TPL EG2F+0646		EG2F0617
SLN 3	SLN3 FØR UNDERFLOW	EG2F0618
SLT 2	TEST FØR PREVIOUS ØVERFLOW	EG2F0619
TRA EG2F+0632		EG2F0620
PXD 3		EG2F0621
STD EG2F+0627		EG2F0622
CLA EG2F+0692	SET ERROR RETURN	EG2F0623
STØ EG2F+0593		EG2F0624
SLF		EG2F0625
CLA 0,4	UNSCALE MATRIX	EG2F0626
FDH CØMMØN+007		EG2F0627
STQ 0,4		EG2F0628
TNX EG2F+0582,4,1		EG2F0629
TXH EG2F+0623,4		EG2F0630
CLA CØMMØN+007		EG2F0631
FDH CØMMØN+013		EG2F0632
STQ CØMMØN+007		EG2F0633
TRA EG2F+0618		EG2F0634
CHS	SCALE MATRIX	EG2F0635
ADD EG2F+0704		EG2F0636
STØ CØMMØN+013		EG2F0637
LDQ CØMMØN+013		EG2F0638
FMP CØMMØN+007		EG2F0639
STØ CØMMØN+007		EG2F0640
LDQ 0,2		EG2F0641
FMP CØMMØN+013		EG2F0642
TØV EG2F+0653		EG2F0643
STØ 0,2		EG2F0644
TIX EG2F+0638,2,1		EG2F0645
CLA EG2F+0693		EG2F0646
STØ EG2F+0593		EG2F0647
TRA EG2F+0439		EG2F0648
SUB EG2F+0703	FMP ØVERFLOW	EG2F0649
SLN 2		EG2F0650
SLT 3		EG2F0651
TRA EG2F+0632	SCALE	EG2F0652
TRA EG2F+0618	UNSCALE	EG2F0653
CLA EG2F+0699	FAD ØVERFLOW	EG2F0654
TRA EG2F+0647		EG2F0655
SXD EG2F+0627,2	SCALING ØVERFLOW	EG2F0656
TXL EG2F+0623		EG2F0657
SXD EG2F+0694,4	SQUARE RØØT	EG2F0658
LXA EG2F+0618,4		EG2F0659
STØ CØMMØN+013		EG2F0660
ANA EG2F+0699		EG2F0661
ARS 1		EG2F0662
ADD CØMMØN+013		EG2F0663
ARS 1		EG2F0664
ADD EG2F+0700		EG2F0665
STØ CØMMØN+009		EG2F0666
CLA CØMMØN+013		EG2F0667
FDH CØMMØN+009		EG2F0668
CLA CØMMØN+009		EG2F0669
STQ CØMMØN+009		EG2F0670
FAD CØMMØN+009		EG2F0671
SUB EG2F+0699		EG2F0672
TIX EG2F+0663,4,1		EG2F0673
LXD EG2F+0694,4		EG2F0674

TRA 1,4		EG2F0675
ST0 COMMON+013	MATRIX TRANSFORMATION	EG2F0676
ST0 COMMON+009		EG2F0677
FMP COMMON+011		EG2F0678
ST0 COMMON+010		EG2F0679
LDQ COMMON+013		EG2F0680
FMP COMMON+012		EG2F0681
FAD COMMON+010		EG2F0682
ST0 COMMON+010		EG2F0682
LDQ COMMON+009		EG2F0684
FMP COMMON+012		EG2F0685
ST0 COMMON+009		EG2F0686
LDQ COMMON+013		EG2F0687
FMP COMMON+011		EG2F0688
FSB COMMON+009		EG2F0689
LDQ COMMON+010		EG2F0690
TRA 1,4		EG2F0691
TXL EG2F+0530		EG2F0692
TRA 5,4		EG2F0693
SLT 2		EG2F0694
TRA 3,4		EG2F0695
TRA 4,4		EG2F0696
ØCT 4,2000000,1000000,233000000000,330000000000,1000000000		EG2F0697
ØCT 100400000000,377000000000,101000000000,176000000000		EG2F0698
DEC 1.,.5		EG2F0691
COMMON BSS 54		RWEG2F1000
END		RWEG2F100-