

NRC Publications Archive Archives des publications du CNRC

Seismic Hydrodynamic Forces Acting on 2D Models of Gravity Dams Jablonski, A. M.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.4224/20358989>

Internal Report (National Research Council of Canada. Institute for Research in Construction), 1990-11-01

NRC Publications Archive Record / Notice des Archives des publications du CNRC :

<https://nrc-publications.canada.ca/eng/view/object/?id=0ccde4a7-f554-43d9-b69b-f79b37f5fe18>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=0ccde4a7-f554-43d9-b69b-f79b37f5fe18>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

Ref
Ser
TH1
R427
no. 602
v. 1
BLDG.

IRC PUB

**National Research
Council Canada**

Institute for
Research in
Construction

**Conseil national
de recherches Canada**

Institut de
recherche en
construction

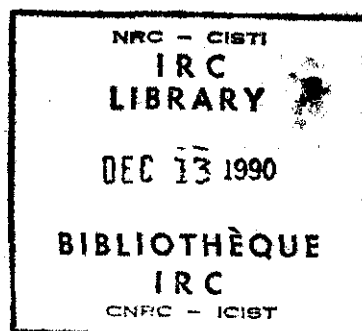
Seismic Hydrodynamic Forces Acting on 2D Models of Gravity Dams Volume 1

by A.M. Jablonski

ANALYZED

Internal Report No. 602

Date of Issue: November 1990



This is an internal report of the Institute for Research in Construction. It is not to be cited as a reference in other publications.

CONTENTS

1.	Introduction	4
1.1	General Comments	4
1.2	Description of Procedure	4
1.3	List of Computer Programs	5
2.	User's Guide to Programs	6
2.1	General Comments	6
2.2	Earthquake Input Data - ACCELN	6
2.2.1	Description of ACCELN Program	6
2.2.2	Input Data	6
2.2.3	Output	6
2.2.4	Procedure to run ACCELN on VM	7
2.3	Harmonic Response - BEMC2DN	7
2.3.1	Description of the BEMC2DN Program	7
2.3.2	Input Data	9
2.3.3	Output	12
2.3.4	Procedure to run BEMC2DN on VM	12
2.4	Earthquake Response - FOURTN	13
2.4.1	Description of the FOURTN Program	13
2.4.2	Input Data	15
2.4.3	Output	16
2.4.4	Procedure to run FOURTN on VM	16
2.5	Plotting Program - SPLOTN2	16
2.5.1	Description of the SPLOTN2 Program	16
2.5.2	Input Data	18
2.5.3	Output	19
2.5.4	Procedure to run SPLOTN2 on VM	19
2.6	Combined Response - SUM	19
2.6.1	Description of the SUM Program	19
2.6.2	Input Data	20
2.6.3	Output Data	20
2.6.4	Procedure to run SUM on VM	20
3.	References	21
Appendices in Additional Volume		
Appendix A. Numerical Example		3

Appendix B.	Listings of Computer Programs	24
B.1	Program ACCELN FORTRAN A1	25
B.2	Program BEMC2DN FORTRAN A1	30
B.3	Program FOURTN FORTRAN A1	84
B.4	Program SLPOTN2 FORTRAN A1	91
B.5	Program SUM FORTRAN A1	102

1. INTRODUCTION

1.1 General Comments

This document describes a procedure to be followed to evaluate seismic hydrodynamic forces acting on 2D models of gravity dams. To meet objectives of this procedure a number of computer programs had to be developed together with a modification of the core program BEMC.2D based on the constant boundary element formulation. At the beginning of this chapter, the short description of the procedure is presented. The rest of this document covers user's guides to these programs together with a numerical example. Listings of them and references are also included.

1.2 Description of Procedure

The procedure is divided into four main steps. These steps should be followed according to the presented description.

Step 1 - Earthquake Input Data

First preparation of an input data file DATAS INPUT A1 must be completed based on an available earthquake record. Depending on the time step and the required number of points in the Fast Fourier Transformation subroutines the specific total time (representative duration time of an earthquake record) should be chosen. Creation of the file DATAS INPUT A1 is made through XEDIT (editor on VM) as e.g. a modification of a copied earthquake record file (in acceleration history).

In order to obtain a file INPS1 INPUT A1 run the program ACCELN FORTRAN A1. After screening INPS1 INPUT A1 w.r.t. number of points and chosen part of the original earthquake file, it can be used as an input file to the programs in Step 3 - Earthquake Response. Execution file EQN EXEC A1 is used to run the program ACCELN FORTRAN A1. Details are included in Section 2.2 - Earthquake Input Data.

Step 2 - Harmonic Response

The response of a dam-reservoir-foundation system subjected to a horizontal or vertical excitation is obtained with the use of the previously developed computer program BEMC2DN based on the constant boundary element formulation. The running of this program is fulfilled using a batch file BATCHS BATCH A1. Input data file DATAM INPUT A1 will be described in detail in subsection 2.3.2. The program BEMC2DN FORTRAN A1 can create three output files:

OUTM OUTPUT A1	a complete output file for each frequency step;
EIGENM OUTPUT A1	an empty file, which may be used in obtaining data from an eigenvalue subroutine;
INFTM INPUT A1	a vital output file, which is used as an input file in step 3 - Earthquake Response. Its name should be changed to INFT1 INPUT A1.

Step 3 - Earthquake Response

The Fast Fourier Transformation technique is used to obtain the response of the system to a horizontal or vertical component of the given earthquake acceleration record.

In order to obtain file OUTFFT OUTPUT A1 run program FOURTN FORTRAN A1 with two input files: INPS1 INPUT A1 from the ACCEL program and INFT1 INPUT A1 from the BEMC2DN program.

Execution file FOURTN2 EXEC A1 is used to run the program FOURTN FORTRAN A1 (NE = 4096 pts) or file FOURTN 1 EXEC A1 is used to run a modified program FOURTNE FORTRAN A1 (NE = 8192 pts).

Details are included in Section 2.4.

Step 4 - Plotting

The set of DISPPLA subroutines is used to obtain several output plots. They could be first reviewed online and later be dumped onto the laser printer associated with VM.

In order to obtain output plots run the program SPLOTN2 FORTRAN A1. Execution file DUM2 EXEC A1 is used to run this program. There are two input files: INPS1 INPUT A1 obtained from ACCELN FORTRAN A1 and OUTFFT OUTPUT A1 obtained from FOURTN FORTRAN A1 (or FOURTNE FORTRAN A1), for this program.

The program SPLOTN2 FORTRAN A1 using DISSPLA creates eight files:

- | | | | |
|----|---------------------------------------|----|---------|
| 1. | Acceleration record (ACCEL vs TIME) | by | PLOT1; |
| 2. | FFT of Acceleration (Re) | by | PLOT2; |
| 3. | FFT of Acceleration (Im) | by | PLOT3; |
| 4. | FFT of Acceleration (Abs) | by | PLOT31; |
| 5. | Harmonic response vs. frequency (Re) | by | PLOT4; |
| 6. | Harmonic response vs. frequency (Im) | by | PLOT5; |
| 7. | Harmonic response vs. frequency (Abs) | by | PLOT51; |
| 8. | Earthquake response vs. frequency | by | PLOT 6. |

Note: The above listed 4-step procedure should be performed for the horizontal and the vertical excitations. In order to share results of the earthquake response by PLOT6 a file named SUM1 OUTPUT A1 is produced. Rename this file to SUM1H OUTPUTA1 for the horizontal response and to SUM1V OUTPUT A1 for the vertical one. These files are later used by an additional program SUM FORTRAN A1. An execution file SUM1 EXEC A1 runs this program. The program SUM FORTRAN A1 produces the combined response from the superposition of two components (horizontal and vertical).

1.3 List of Computer Programs

ACCELN FORTRAN A1 - to prepare an earthquake input data
BEMC2DN FORTRAN A1 - to calculate a harmonic response
FOURTN FORTRAN A1 - to calculate earthquake response
SPLOTN2 FORTRAN A1 - to plot results
SUM FORTRAN A1 - to plot a combined response

2. *USER'S GUIDES TO PROGRAMS*

2.1 General Comments

In this chapter five computer programs are presented. The most important programs are: a core program BEMC2DN FORTRAN A1, a FFT program FOURTN FORTRAN A1 and a plotting program SPLOTN2 FORTRAN A1. Each step of this procedure was planned to run separately in allowing a prospective user to make appropriate changes in input and output statements.

First each program is described and later its input and output. A description of a procedure to run a specific program follows. A numerical example is presented in Appendix A. The listings of those programs are included in Appendix B. All programs are written in FORTRAN77.

2.2 Earthquake Input Data - ACCELN

2.2.1 Description of the ACCELN Program

This FORTRAN computer program can be used to calculate the input data file INPS1. The program could accommodate any number of points used in the original earthquake record. It can read also any specified part from this record.

2.2.2 Input Data

The following parameters could be changed at the beginning and before running the ACCELN program:

NT = number of time step (now set at 4096)

NI = initial step number to be read from the record file (now set at 0)

NL = last step number to be read from the same record (any number depending on number of steps in the earthquake record).

Input data is stored in the vector AC (9000). The dimension of the vector can be also changed. Communication between input data file and the program is maintained by the OPEN statement in the form:

OPEN (LEC, FORM = 'FORMATTED', STATUS = 'OLD', FILE = DATAS).

The input data file DATAS INPUT A1 is a modified version of the considered original earthquake data (in acceleration history). Depending on the time step and the number of points in the Fast Fourier Transformation subroutines, the specific total time (duration) is chosen. Creation of that file may be done using XEDIT (online editor on VM). READ statement for the DATAS should be changed according to the format used in the earthquake record.

2.2.3 Output

The program ACCELN creates an output file called INPS1 INPUT A1. Output data is stored in the vector A (9000). WRITE statement for the INPS1 should be changed accordingly to the required format.

2.2.4 Procedure to run ACCELN on VM

To run the program ACCELN simply invoke the execution file EQN EXEC A1. The execution file EQN EXEC A1 is given below:

```
/* EXEC FILE EQN */  
SETLEVEL FORTRAN NEW  
'FI * CLEAR'  
FORTVSL  
FORTVS2 ACCELN '( NOFIPS NOPRI'  
LOAD ACCELN '( CLEAR START'  
SENDFILE INPS1 INPUT A1 account name  
EXIT
```

2.3 Harmonic Response - BEMC2DN

2.3.1 Description of the BEMC2DN Program

The constant boundary element program for 2D reservoirs - BEMC2DN FORTRAN A1 is a modified version of the previously developed program BEMC2D FORTRAN A1. The program BEMC2D FORTRAN A1 constitutes a VM/IBM version of the original program BEMC.2D developed for the APOLLO/DOMAIN computer [1].

This program produces harmonic responses due to horizontal or vertical excitation in hydrodynamic forces. Those results are later used as an input in the Fast Fourier Transform analyses included in the next program FOURTN FORTRAN A1.

The BEMC2DN program can be used to calculate the hydrodynamic pressure on a rigid dam storing a two-dimensional finite or infinite water reservoir and subjected to a horizontal or/and vertical harmonic excitation.

The program uses the constant boundary elements. The pressures and the pressure derivatives are assumed to have constant values over the element and equal to their values at the mid-node of each element. The development of this 2D boundary element model of a gravity dam-reservoir-foundation system subjected to ground motion has been presented by Liu and Cheng [2], Humar and Jablonski [3], and Jablonski [4].

There are no limitations on the shape of the dam face or the finite reservoir. The infinite reservoir, however, has to be divided into two parts: one finite and irregular and other regular but infinite. The upstream face of the dam, the reservoir surface, the bottom and the upstream and of the reservoir (in case of the infinite reservoir this end is an interface between irregular and regular parts), are to be covered by a constant boundary element mesh.

The main program can handle up to 100 boundary elements. However, this could be extended to any number (depending on the computer capacity).

The main program calls the eight following subroutines:

INPUT - to read all data and print the first part of the output.

- EIGEN - to set and calculate the complex eigenvalue problem in case of infinite radiation with foundation damping included.
- EIGENR - to set and calculate the real eigenvalue program in case of infinite radiation only and without foundation damping.
- FMAT - to form matrices **G** and **H** and rearrange them accordingly to the boundary conditions on the top of the reservoir.
- ADDFBC - to modify the matrix **H** for the boundary conditions on the bottom of the reservoir and at its far boundary (transmitting boundary), and to compute the vector of known values of the R.H.S. of the system of equations. This subroutine forms the system of equations for the damped case.
- ADDUND - to form the system of equations and apply all B.C.'s for the undamped case.
- SLNPD - the solver of linear equations systems (Gauss elimination)
- OUTPUT - to reorder the vector of computed unknowns and print out the rest of the output file.

The subroutine EIGEN calls the subroutine:

- FSFEM - to compute the real and imaginary portions of the eigenvector matrix and at the same time to create the real and imaginary parts of the eigenvalue vector.

The subroutine FSFEM calls the following subroutines:

- SET - to form the complex matrices for the eigenvalue problem, which governs the damped vibrations in the infinite part of the reservoir $(A + i\omega\gamma B) p = \lambda^2 F p$, where matrices **A**, **B** and **F** are described in detail in Refs. 4 and 5.
- CQZHES - this subroutine is a complex analogue of the first step of the QZ algorithm for solving generalized real matrix eigenvalue problems [5]. This subroutine accepts a pair of complex general matrices and reduces one of them to upper Hessenberg form with real (and non-negative) subdiagonal elements and the other to upper triangular form using unitary transformations.
- CQZVAL - this subroutine is a complex analogue of steps 2 and 3 of the QZ algorithm for solving generalized real matrix eigenvalue problems. This subroutine accepts a pair of complex matrices, one of them in upper Hessenberg form and the other in upper triangular form. The Hessenberg matrix must further have real subdiagonal elements. It reduces the Hessenberg matrix to triangular form using unitary transformations while maintaining the triangular form of the other matrix and making its diagonal elements real and non-negative. It then returns quantities whose ratios give the generalized eigenvalues.
- CQZVEC - is a complex analogue of the fourth step for solving generalized real matrix eigenvalue problems. This subroutine accepts a pair of complex matrices in

upper triangular form, where one of them further must have real diagonal elements. It computes the eigenvectors of the triangular problem and transforms the result back to the original system. This is the last step of QZ algorithm originally developed by Stewart and Moler [5] and later modified by B.S. Garbov from Applied Mathematics Division, Argonne National Laboratory, Illinois, U.S.A.

NORMA - normalizes the eigenvectors obtained from the QZ algorithm w.r.t matrix F.

The subroutine EIGENR calls the following subroutines:

- SETR - forms the real matrices for the eigenvalue problem, which governs the undamped vibrations in the infinite part of the reservoir: $A p = \lambda^2 F p$.
- JACOBI - uses the generalized Jacobi method to solve the generalized real matrix eigenvalue problem [6]. Eigenvectors and eigenvalues are presented in a normalized form in its output.

The subroutine FMAT calls the following subroutines:

- INTE - computes the G and H matrix elements except those on the diagonal by means of numerical integration along the boundary elements. The numerical integration uses the 4-point Gauss quadrature scheme.
- INLO - calculates the diagonal elements of matrix G.

The subroutine INTE calls the subroutine:

- BFUNCT - computes the values of Bessel function $Y_0(kr)$ and its derivative using the series representation of the Bessel functions [7,8].

Subroutines ADDFBC, ADDUND and OUTPUT call the subroutine:

- TP - calculates the transformation matrix T_p for adjustment of linearly formed for end condition into content element formulation.

2.3.2 *Input data*

Input data is read from the file DATAM INPUT A1 (unit No. LEC = 2).

The following table shows the order of presentation of data in the program BEMC2DN FORTRAN A1:

TABLE 2-1

Input Data for the BEMC2DN FORTRAN

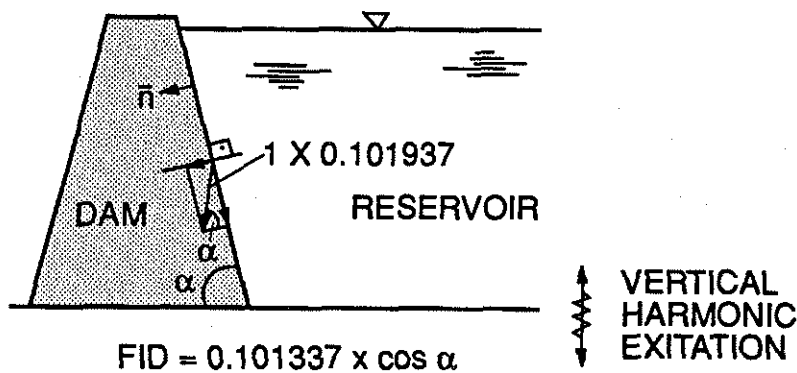
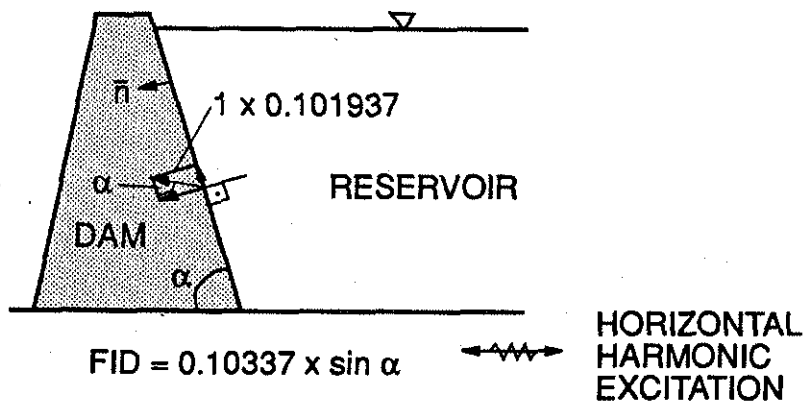
Card Group	No. of Cards	Format	FORTTRAN Variables
A	1	18A4	TITLE
B1	1	General Format	N = number of boundary elements N0 = last element number at top near the dam N1 = last element number at the base of the dam N2 = last element number at the reservoir bottom near the end NOUT = 1, flag for detailed output, 0, flag for output at the dam only
B2	1	General Format	HY = height of the dam C = velocity of sound in water ALFA = wave reflection coefficient DF = vertical acceleration of the first mode at the far end
C	N	General Format	X,Y = coordinates of nodal points
D	N1-N0	General Format	FID = boundary conditions on the dam face
E	N2-N1	General Format	FIB = boundary conditions on the foundation base
F	No. of cases	General Format	OMEGA = excitation frequency

The node numbers must start from the element at the top of the reservoir adjacent to the far end and numbered in counter-clockwise direction.

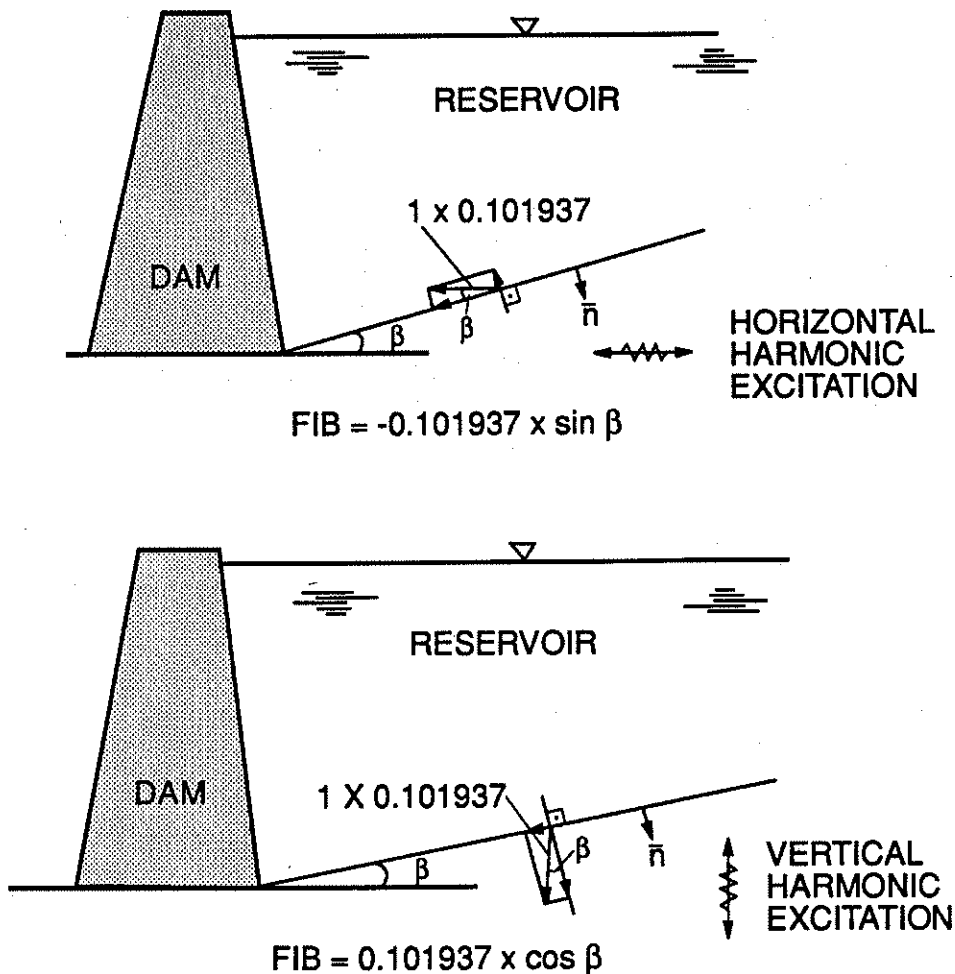
Units:

- HY = height of the dam in m;
 C = 1440 m/sec, velocity of sound in water;
 ALFA = wave reflection coefficient, dimensionless value from 0 to 1.0;
 DF = 0.101937, unit vertical ground acceleration (w/g , where $w = 1$ metric tonne and $g = 9.81 \text{ m/s}^2$);
 X,Y = coordinates of modal points in m;
 *FID = 0, for vertical excitation; 0.101937, unit horizontal acceleration (w/g);
 *F1B = 0, for a rigid reservoir bottom; 0.101937, unit vertical ground acceleration;
 OMEGA = in rad/s

BOUNDARY CONDITIONS ON INCLINED DAM FACE



BOUNDARY CONDITIONS ON INCLINED RESERVOIR BOTTOM



Whenever the direction of excitation component is the same as the one of the outward normal to the reservoir boundary, the input value should be positive, and negative otherwise.

2.3.3 Output

The program BEMC2DN FORTRAN A1 creates three output files: OUTM OUTPUT A1, EIGENM OUTPUT A1 and INFTM INPUT A1. After changing the name of the last file to INFT1 INPUT A1, it can be used as an input to calculate the specific earthquake response.

2.3.4 Procedure to run BEMC2DN on VM

To run the program BEMC2DN FORTRAN A1 simply invoke the batch file BATCHS BATCH A1. The batch file BATCHS BATCH A1 is given below:

```
/* BATCHS */
/* TESTING BEMC2DN PROGRAM */
EXEC SYSPROF
ADDRESS COMMAND CP LINK account name 191 196 RR password
ACCESS 196 'B/A'
TAGPRINT
TRACE C
SETLEVEL FORTRAN NEW
F1 * CLEAR'
TEMPDISK C10
FI 3 DISK OUTM OUTPUT C
FI 6 DISK EIGENM OUTPUT C
```

```

FI 7 DISK INFTM INPUT A1
FORTVSL
FORTVS2 BEMC2DN '(NOPRI NOFIPS'
LOAD BEMC2DN '(CLEAR START'
/* SENDFILE OUTM OUTPUT C account name */
/* TSSPRT OUTM OUTPUT C '(CL D DEFAULT' */
SENDFILE INFTM INPUT A1 account name

```

2.4 Earthquake Response - FOURTN

2.4.1 Description of the FOURTN Program

The program FOURTN calculates the response in hydrodynamic pressures of the linear dam-reservoir-foundation system to the excitation history given in the discrete form. The program FOURTN FORTRAN A1 employs the FFT routines originally developed by Paul Swarztrauber, National Center for Atmospheric Research, Denver, Colorado, USA, and in the form provided by the new version 1.0 of the math Library, IMSL [9].

For obtaining the response to an arbitrary ground motion through the frequency domain analysis, the standard procedure can be summarized in the following steps:

1. The excitation history has to be expressed in terms of the harmonic components.
2. The evaluation of the response of the linear dam-reservoir-foundation system is done next.
3. The superposition of the harmonic responses to obtain the total response completes the solution.

When the excitation history is given in terms of function of the acceleration $a(t)$ and $F(\omega)$ represents the harmonic response of the system to unit excitation $e^{i\omega t}$, the so-called harmonic synthesis is usually expressed as:

$$F(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} A(\omega) \bar{F}(\omega) e^{i\omega t} d\omega \quad (1)$$

where

$$A(\omega) = \int_{-\infty}^{\infty} a(t) e^{-i\omega t} dt \quad (2)$$

Equation 2 represents the Fourier Transform of the acceleration function $a(t)$, while Eq. 1 is the Inverse Fourier Transform of the product of the frequency functions $A(\omega)$ and $F(\omega)$.

Both equations can be solved by a numerical technique based on the Discrete Fourier Transform (DFT).

The first step is to assure the periodic type of excitation, which implies an approximation of an arbitrary motion, which is not periodic. The infinite integrals 1 and 2 are replaced by the

finite sums. The selected period of excitation, T_0 , serves to define the lowest frequency in the analysis.

$$\Delta\omega = \frac{2\pi}{T_0} \quad (3)$$

The excitation period is then divided into N equal time steps Δt , and the acceleration function $a(t)$ is defined for the discrete time steps $t_m = m\Delta t$.

The discrete forms of the expressions given by Eqs. 1 and 2 can be rewritten as follow:

$$F(m\Delta t) = \frac{\Delta\omega}{2\pi} \sum_{n=0}^{N-1} [A(n\Delta\omega) \bar{F}(n\Delta\omega)] e^{i\frac{2\pi n m}{N}} \quad (4)$$

and

$$A(n\Delta\omega) = \Delta t \sum_{m=0}^{N-1} [a(m\Delta t)] e^{-i\frac{2\pi n m}{N}} \quad (5)$$

Then, the Fast Fourier Transform (FFT) technique is used to evaluate Eqs. 4 and 5. The computer program FOURTN FORTRAN A1 calculates the response of the system subjected to an arbitrary excitation history given in the discrete form $a(m\Delta t)$.

Using the DFT implies that both the excitation history and the response are periodic functions of period T_0 . This results in errors in evaluation of the transient response of the system. The excitation period is elongated by adding an additional grace band of zero excitations to the end. It ensures that the system will have a period of free vibration so that the system will eventually come to the rest at the end of each period.

The program FOURTN FORTRAN A1 has three parameters, which can be easily changed.

NT = 4096, Number of steps to be employed in the FFT;
NC = 200, Initial number of frequency steps employed to calculate harmonic response;
ND = 10, Number of nodes at dam face introduced in the BEMC2DN program.

This program uses two input files: INPS1 INPUT A1 and INFT1 INPUT A1. Reading statements should be checked for both of them and replaced by required formats used in these files.

The following steps are included in the main program:

1. Read an input data created by program ACCELN FORTRAN A1;
2. Divide the acceleration vector by 100 to have it in m/s^2 ;
3. Carry out forward FFT;
4. Read required harmonic responses;
5. Modify harmonic response to have it in the form ready for FFT routine;
6. Carry out inverse FFT for a chosen harmonic response vector.

The main program calls the three following subroutines.

IWKIN (224611) - to change the amount of space allocated. FORTRAN subroutines that work with matrices as input or output often require extra arrays

for use as workspace. IMSL routines usually do not require the user explicitly to allocate such arrays for use as workspace (it is done automatically). By default, the total amount of space allocated in the common area for storage of numeric state is 5000 numeric storage units. This space is allocated as needed for all variables. The user can change the allocation by supplying the following FORTRAN statements in conjunction with this subroutine:

```
COMMON /WORKSR/ RWKSP
```

```
REAL RWKSP (24611)
```

```
CALL IWKIN (24611)    It will request 24611 units (this number can be changed if required).
```

If an IMSL routine attempts to allocate workspace in excess of the amount available in common stack (5000), the routine issues an error message that indicates how much space is required and prints statements like above. Other details may be found in Ref. 9.

FFTCF (NT, AC, AN) - computes the Fourier coefficients of a complex periodic vector AC.
Where:

NT = length of the sequence or number of steps to be employed in the FFT;
AC(I) = processed acceleration vector from earthquake records of length NT (input);
AN(I) = complex vector of length NT containing Fourier coefficients (output)

FFTCF computes the discrete complex Fourier transform of a complex vector of size NT. The method used is a variant of the Cooley-Tukey algorithm which is most efficient when NT is a product of small prime factors. If NT satisfies this condition then the computational effort is proportional to NT Log (NT) [9].

FFTCB (NT, Y2, Y2N) - computes the inverse Fourier transform of the vector $Y2(I) = AN(I) * Y1(I)$. AN(I) is a complex vector containing Fourier coefficients of the acceleration record. Y1(I) is a complex vector of total hydrodynamic forces normalized w.r.t. the hydrostatic force (in this case 5000 for the dam height 100 m). The method used by the FFTCB subroutine is the same as above [9].

2.4.2 Input Data

The following parameters could be changed at the beginning and before running the FOURTN program:

NT = number of steps in the FFT;
NC = initial number of frequency steps to calculate harmonic response;
ND = number of nodes at dam face (in the BEMC2DN program).

Two input files along with one output file are required and they are linked through the OPEN statements in the form:


```
DATA INPS1//INPS1 INPUT A1/  
DATA INFT1//INFT1 INPUT A1/  
DATA OUTFFT//OUTFFT OUTPUT A1/
```

```
OPEN (2, FORM = 'FORMATTED', STATUS = 'OLD', FILE = INPS1)  
OPEN (3, FORM = 'FORMATTED', STATUS = 'OLD', FILE = INFT1)  
OPEN (5, FORM = 'FORMATTED', STATUS = 'NEW', FILE = OUTFFT)
```

The input file INPS1 INPUT A1 is an output from the ACCELN program, and INFT1 INPUT A1 is an output created by the BEMC2DN program. READ statements for the INPS1 should be changed according to the format used in this particular file.

2.4.3 Output

The program FOURTN FORTRAN A1 creates an output file called OUTFFT OUTPUT (very large file). That file should be deleted after being used in the next step (i.e. in plotting graphs).

2.4.4 Procedure to run FOURTN on VM

To run the program FOURTN simply invoke the execution file FOURTN2 EXEC A1. The execution file FOURTN2 EXEC A1 is given below:

```
/* FOURT */  
'FI * CLEAR'  
FORTVSL IMSL1 IMSL2  
FORTVS2 FOURTN '(NOPRI NOFIPS'  
FI 2 DISK INPS1 INPUT A1  
FI 3 DISK INFT1 INPUT A1  
FI 5 DISK OUTFFT OUTPUT A1 '(RECFM F LRECL 80'  
LOAD FOURTN '(CLEAR START'  
EXIT
```

2.5 Plotting Program - SPLOTN2

2.5.1 Description of the SPLOTN2 Program

The program SPLOTN2 produces following graphs (on screen of a terminal or printed by a laser printer):

1. Acceleration record (time history) - PLOT1
2. Real part of DFT of acceleration - PLOT2
3. Imaginary part of DFT of acceleration - PLOT3
4. Absolute value of DFT of acceleration - PLOT31
5. Real part of harmonic response - PLOT4
6. Imaginary part of harmonic response - PLOT5
7. Absolute value of harmonic response - PLOT51
8. Earthquake response in hydrodynamic pressure (time history) - PLOT6

The SPLOTN2 FORTRAN A1 uses the routines from the DISSPLA package installed on VM [10].

The program has two parameters:

NT = 4086, number of frequency steps used in the FFT procedure;
NE = 4096, maximum dimension of arrays used in the program.

The reading of the acceleration history vector should be based on an appropriate format in READ statement.

The main program calls the nine following subroutines:

- NOMDEV(1) - to nominate a graphics device named in the DISSPLA command in the execution file DUM2 EXEC A1 (level 0 → 1);
- PLOT1 (NE, X, Y) - to plot acceleration history record;
- PLOT2 (NE, X, ANR) - to plot real part of DFT of acceleration history record;
- PLOT3 (NE, X, ANI) - to plot imaginary part of DFT of acceleration history record;
- PLOT4(NX, X, Y1R) - to plot real part of harmonic response (NX = 200 steps);
- PLOT5 (NX, X, Y1I) - to plot imaginary part of harmonic response;
- PLOT51 (NX, X, Z) - to plot absolute value of harmonic response;
- PLOT31 (NE, X, AA) - to plot absolute value of DFT of acceleration history;
- PLOT6 (NE, X, Y2NR) - to plot hydrodynamic force time history (earthquake response);
- DONEPL - to terminate DISSPLA (level 1 → 0);

Each of the plotting subroutines uses the DISSPLA hierarchy along with a number of DISSPLA (old and new) routines.

The subroutine PLOT1 is described in detail as an example. The rest of subroutines follows this example.

The subroutine PLOT1 (N, X, Y) has a number of steps. First, it sets steps and maximum values on X and Y axes. It finds also maximum and minimum of the particular acceleration history record.

The subroutine PLOT1 calls the following DISSPLA routines:

- PAGE (11.0, 8.5) - to set the page limits (DISSPLA sets the page limits to the standard 8.5 by 11 inch page size). This subroutine allows a change in the page limits;
- NOBRDR - to suppress the default page border in order to draw an enhanced border;

- PHYSOR (1.0, 1.0) - to specify a physical origin on the page;
- TRIPLX - to specify character style (fonts) - TRIPLX is one from eight options;
- TITLE ('MANIC-3 STA2-A HOR.', 19, 'STEPS, 1 STEP = 0.02 SEC', 21, 'ACCELERATION M/SEC/SEC', 22, 9.0, 6.5)
- to specify titles and captions;
- GRAF (XMIN, XSTP, XMAX, YMIN, YSTP, YMAX) - to draw axes;
- MESSAG ('MAX = ', 6, 4.0, 6) - to write specific captions and messages relative to physical origin (example is given);
- REALNO (RMAX, +2, 4.8, 6) - to place real numbers anywhere on a plot relative to the physical origin (example is given);
- INTNO (I1, 6.6, 6) - to position integer numbers anywhere on a plot relative to the physical origin (example is given);
- STRTP (0.0, 3.25) - to move to the point without drawing the line from the physical origin to desired point (example is given);
- CONNPT (9.0, 3.25) - to connect successive points (in this case - (9.0, 3.25)) with straight lines starting from the previous position (in this case - (0.0, 3.25));
- CURVE (X, Y, N, O) - to plot data (after the axis system has been defined and CADISSPLA). IMARK = 0 (marker specification means points connected without symbols drawn).

2.5.2 Input Data

Two initial parameters NT = 4096 and NE = 4096 may be changed depending on the number of frequency steps used in the FFT procedure.

The SPLOTN2 FORTRAN A1 uses two input files and creates one output file. they are linked to the main program through three OPEN statements in the following form:

```
DATA INPS1//INPS1 INPUT A1/  
DATA OUTFFT//OUTFFT OUTPUT A1/  
DATA SUM1//SUM1 OUTPUT A1/.
```

```
OPEN (2, FORM = 'FORMATTED', STATUS = 'OLD', FILE = INPS1)  
OPEN (3, FORM = 'FORMATTED', STATUS = 'OLD', FILE = OUTFFT)  
OPEN (4, FORM = 'FORMATTED', STATUS = 'NEW', FILE = SUM1)
```

All input data is provided by two earlier programs ACCELN and FOURTN.

When all graphs are obtained, the output file OUTFFT can be deleted.

2.5.3 Output

The program SPLOTN2 FORTRAN A1 creates an output file called SUM1 OUTPUT A1. For horizontal response this name should be changed to SUM1H OUTPUT A1 and for vertical to SUM1V OUTPUT A1, respectively.

2.5.4 Procedure to run SPLOTN2 on VM

To run the program SPLOTN2 FORTRAN A1 invoke the given execution file DUM2 EXEC A1 in two versions.

First version of the DUM2 EXEC A1 is directed to review plots on screen of the terminal interfacing VM:

```
/* DUM2 EXEC FILE */  
'FI * CLEAR'  
SETLEVEL FORTRAN NEW  
FORTVS2  
FORTVS2 SPLOTN2'(NOFIPS NOPRINT'  
DISSPLA SPLOTN2 FILE '(DUM2)  
PLOT DUM2 TEKALL' (4010, 960)  
/* PLOT DUM2 LASER */  
'FI * CLEAR'  
EXIT
```

Second version of the DUM2 EXEC A1 plots the final sets of graphs on the laser printer located in the computer center.

```
/* DUM2 EXEC FILE */  
'FI * CLEAR'  
SETLEVEL FORTRAN NEW  
FORTVSL  
FORTVS2 SPLOTN2 '(NOFIPS NOPRINT'  
DISSPLA SPLOTN2 FILE '(DUM2)  
/* PLOT DUM2 TEKALL '(4010, 960) */  
PLOT DUM2 LASER  
'FI * CLEAR'  
EXIT
```

2.6 Combined Response - SUM

2.6.1 Description of the SUM Program

The program SUM FORTRAN A1 combines horizontal and vertical time histories of the hydrodynamic force and plots them in the form of the combined response.

This program has one parameter:

NT = 4096, number of time steps in the time history of the hydrodynamic force.

It reads two vectors and adds them together. It uses also one subroutine PLOT1 (N,X,Y) which is similar to one used in the program SPLOTN2.

The main program calls the three following subroutines:

- NOMDEV(1) - to nominate a graphics device named in the DISSPLA command in the execution file SUM1 EXEC A1 (level 0 → 1);
- PLOT1 (NT, X, E) - to plot the final combined response in hydrodynamic forces;
- DONEPL - to terminate DISSPLA (level 1 → 0).

The subroutine PLOT1 uses the DISSPLA hierarchy along with a number of DISSPLA routines (see Section 2.5.1).

2.6.2 Input Data

The initial parameter, NT = 4096 can be changed depending on the needs and based on calculations carried out previously.

The SUM FORTRAN A1 uses two input files and creates one plot. They are linked to the main program through three OPEN statements in the following form:

```
DATA SUM1H//SUM1H OUTPUT A1/  
DATA SUM1V//SUM1V OUTPUT A1/
```

```
OPEN (2, FORM = 'FORMATTED', STATUS = 'OLD', FILE = SUM1H)  
OPEN (3, FORM = 'FORMATTED', STATUS = 'OLD', FILE = SUM1V)
```

All input data is provided by the program SPLOTN2.

2.6.3 Output

The program SUM produces one plot only for a combined response in hydrodynamic pressure (time history of hydrodynamic pressure). It does not create any output file.

2.6.4 Procedure to run SUM on VM

To run the program SUM1 invoke the given below execution file SUM1 EXEC A1.

```
/* SUM 1 EXEC FILE */  
'F1 * CLEAR'  
SETLEVEL FORTRAN NEW  
FORTVSL  
FORTVS2 SUM '(NOFIPS NOPRINT'  
DISSPLA SUM FILE '(SUM1)  
PLOT SUM1 TEKALL '(4010, 9600)  
/* PLOT SUM 1 LASER */  
'FI * CLEAR'  
EXIT
```

The above version serves for checking actual graph on screen of the terminal interfacing VM. In order to dump it on the laser printer change two following statements to the given form:

```
/* PLOT SUM1 TEKALL '(4010, 960)'  
PLOT SUM1 LASER
```

3. REFERENCES

- [1] Jablonski, A.M., "Boundary Element Analysis of Earthquake Induced Hydrodynamic Pressures in a Water Reservoir", Vol. 2, User's Guides to Computer Programs, Ph.D. Thesis Submitted to Carleton University, Ottawa, Canada, 1988.
- [2] Liu, P.L.-F., Cheng, A.H.-D., "Boundary Solutions for Fluid-Structure Interaction", Journal of Hydraulic Engineering, ASCE, Vol. 110, 1984, pp. 51-64.
- [3] Humar, J.L., Jablonski, A.M., "Boundary Element Reservoir Model for Seismic Analysis of Gravity Dams", Earthquake Engineering and Structural Dynamics, Vol. 16, 1988, pp. 1129-1156.
- [4] Jablonski, A.M., "Boundary Element Analysis of Earthquake Induced Hydrodynamic Pressures in a Water Reservoir", Vol. 1, Ph.D. Thesis Submitted to Carleton University, Ottawa, Canada, 1988.
- [5] Moler, C.B., Stewart, G.W., "An Algorithm for Generalized Matrix Eigenvalues Problems", Journal of Numerical Analysis, SIAM, Vol. 10, 1973, pp. 241-265.
- [6] Bathe, K.-J., "Finite Element Procedures in Engineering Analysis", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
- [7] MacLachlan, N.W., "Bessel Functions for Engineers", Oxford at the Clarendon Press, Oxford, England, 1955.
- [8] Hanna, Y.G., "Application of Boundary Element Method to Certain Problems in Structural Dynamics", M.Eng. Thesis Submitted to Carleton University, Ottawa, Canada, 1980.
- [9] User's Manual, Math/Library, Fortran Subroutines for Mathematical Applications, Version 1.0, IMSL, Problem-Solving Software Systems, Houston, Texas, 1987.
- [10] CA-DISSPLA User Manual, Version 11.0, Volume 1 and 2, Computer Associates International, Inc., NY, USA, 1988.

Ref
Ser
TH1
R427
no. 602
v. 2
BLDG.

**National Research
Council Canada**

Institute for
Research in
Construction

**Conseil national
de recherches Canada**

Institut de
recherche en
construction

IRC PUB

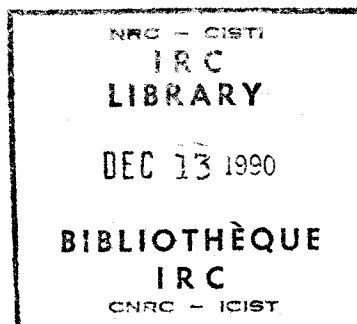
Seismic Hydrodynamic Forces Acting on 2D Models of Gravity Dams

Volume 2 - Appendices

by A.M. Jablonski

Internal Report No. 602

Date of issue: November 1990



This is an internal report of the Institute for Research in Construction. It is not to be cited as a reference in other publications.

**SEISMIC HYDRODYNAMIC FORCES ACTING ON
2D MODELS OF GRAVITY DAMS**

User's Guide to Computer Programs

Appendices

by

Alexander M. Jablonski

September 1990

Institute for Research in Construction
National Research Council of Canada
Ottawa, Ontario, Canada K1A 0R6

CONTENTS

Appendix A:	Numerical Example	3
Appendix B:	Listings of Computer Programs	24
B.1	Program ACCELN FORTRAN A1	25
B.2	Program BEMC2DN FORTRAN A1	30
B.3	Program FOURTN FORTRAN A1	84
B.4	Program SPLOTN2 FORTRAN A1	91
B.5	Program SUM FORTRAN A1	102

APPENDIX A

NUMERICAL EXAMPLE

Input Data

For the purpose of illustration, a simple example is presented in this Appendix.

A two-dimensional infinite reservoir impounded by a straight gravity dam is subjected to an El Centro horizontal acceleration (file INPSHE INPUT A1). The reservoir is modelled using 55 constant boundary element mesh as shown in Fig. A.1

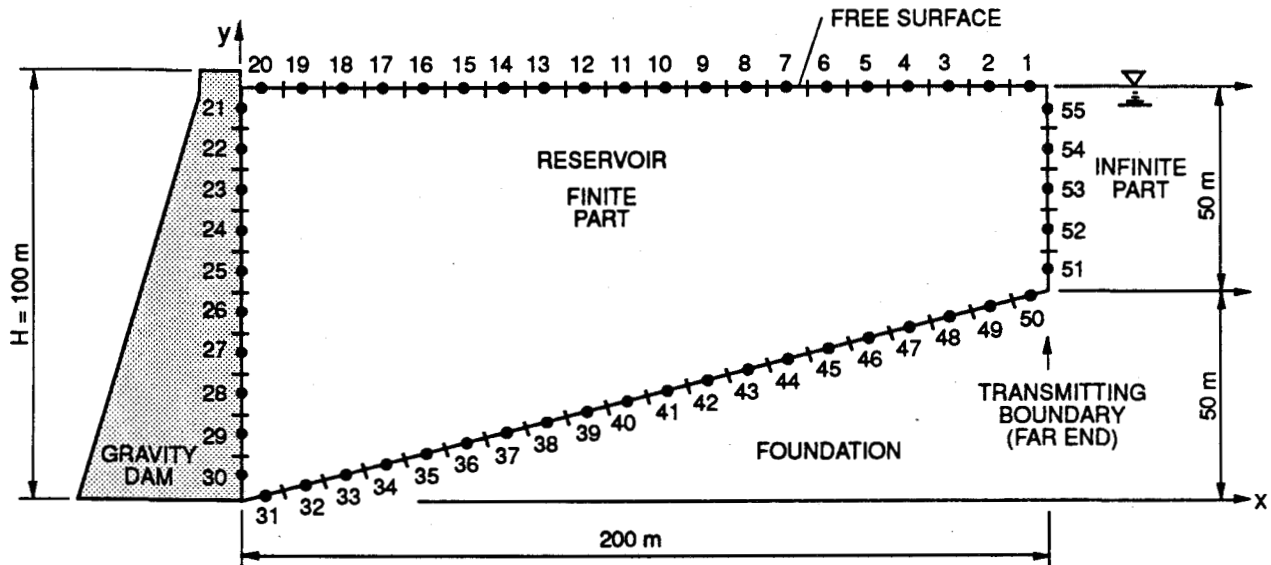


Fig. A.1 The Constant Boundary Element Model of Infinite 2D Reservoir - $N = 55$

The list of the parameters used in the input data file, TESTM INPUT A1 is as follows:

- $N = 55$ (elements)
- $N0 = 20$ (last elem. number at the top near a dam)
- $N1 = 30$ (last elem. number at the base of the dam)
- $N2 = 50$ (last elem. number at the base near far end)
- $NOUT = 1$ (flag for detailed output)
- $HY = 50$ (height of the infinite part of the reservoir in meters)
- $C = 1440$ (velocity of sound in water in m/s^2)
- $ALFA = 0.0$ (coefficient of reflection in full damping case)
- $DF = 0.0$ (vertical acceleration of the first mode at the far end)

Coordinates of the nodal points together with boundary conditions on the dam face and on the foundation base together with excitation frequencies are listed in a copy of the original file TESTM INPUT A1.

Listing of the file TESTM INPUT A1

2D INFINITE RESERVOIR BY THE BEMC

55,20,30,50,1	TES00010
50,1440,0.00,0.0	TES00020
200.0,100.0	TES0003
190.0,100.0	TES00040
180.0,100.0	TES00050
170.0,100.0	TES00060
160.0,100.0	TES00070
150.0,100.0	TES00080
140.0,100.0	TES00090
130.0,100.0	TES00100
120.0,100.0	TES00110
110.0,100.0	TES00120
100.0,100.0	TES00130
90.0,100.0	TES00140
80.0,100.0	TES00150
70.0,100.0	TES00160
60.0,100.0	TES00170
50.0,100.0	TES00180
40.0,100.0	TES00190
30.0,100.0	TES00200
20.0,100.0	TES00210
10.0,100.0	TES00220
0.0,100.0	TES00230
0.0,90.0	TES00240
0.0,80.0	TES00250
0.0,70.0	TES00260
0.0,60.0	TES00270
0.0,50.0	TES00280
0.0,40.0	TES00290
0.0,30.0	TES00300
0.0,20.0	TES00310
0.0,10.0	TES00320
0.0,0.0	TES00330
10.0,2.5	TES00340
20.0,5.0	TES00350
30.0,7.5	TES00360
40.0,10.0	TES00370
50.0,12.5	TES00380
60.0,15.0	TES00390
70.0,17.5	TES00400
80.0,20.0	TES00410
90.0,22.5	TES00420
100.0,25.0	TES00430
110.0,27.5	TES00440
120.0,30.0	TES00450
130.0,32.5	TES00460
140.0,35.0	TES00470
150.0,37.5	TES00480
160.0,40.0	TES00490
170.0,42.5	TES00500
180.0,45.0	TES00510
190.0,47.5	TES00520
200.0,50.0	TES00530
200.0,60.0	TES00540
	TES00550

7.0564	TES01120
7.3632	TES01130
7.6700	TES01140
7.9768	TES01180
8.2836	TES01190
8.5904	TES01200
8.8972	TES01210
9.2040	TES01220
9.5108	
9.8176	
10.1244	
10.4312	
10.7380	
11.0448	TES01230
11.3516	TES01240
11.6584	TES01250
11.9652	TES01260
12.2720	TES01270
12.5788	TES01280
12.8856	TES01290
13.1924	TES01300
13.4992	TES01310
13.8060	TES01320
14.1128	TES01330
14.4196	TES01340
14.7264	TES01350
15.0332	TES01360
15.3400	TES01370
15.6468	TES01380
15.9536	TES01390
16.2604	TES01400
16.5672	TES01420
16.8740	TES01430
17.1808	TES01440
17.4876	
17.7944	
18.1012	
18.4080	
18.7148	
19.0216	
19.3284	
19.6352	
19.9420	
20.2488	
20.5556	
20.8624	
21.1692	
21.4760	
21.7828	
22.0896	
22.3964	
22.7032	
23.0100	
23.3168	
23.6236	

23.9304
24.2372
24.5440
24.8508
25.1576
25.4644
25.7712
26.0780
26.3848
26.6916
26.9984
27.3052
27.6120
27.9188
28.2256
28.5324
28.8392
29.1460
29.4528
29.7596
30.0664
30.3732
30.6800
30.9868
31.2936
31.6004
31.9072
32.2140
32.5208
32.8276
33.1344
33.4412
33.7480
34.0548
34.3616
34.6684
34.9752
35.2820
35.5888
35.8956
36.2024
36.5092
36.8160
37.1228
37.4296
37.7364
38.0432
38.3500
38.6568
38.9636
39.2704
39.5772
39.8840
40.1908
40.4976

TES00900
TES00910
TES00920
TES00930
TES00940
TES00950
TES00960
TES00970
TES00980
TES00990
TES01000
TES01010
TES01020
TES01030
TES01040
TES01050
TES01060
TES01070
TES01080
TES01090
TES01100
TES01110
TES01120
TES01130
TES01140
TES01180
TES01190
TES01200
TES01210
TES01220

40.8044	
41.1112	
41.4180	
41.7248	TES01230
42.0316	TES01240
42.3384	TES01250
42.6452	TES01260
42.9520	TES01270
43.2588	TES01280
43.5656	TES01290
43.8724	TES01300
44.1792	TES01310
44.4860	TES01320
44.7928	TES01330
45.0996	TES01340
45.4064	TES01350
45.7132	TES01360
46.0200	TES01370
46.3268	TES01380
46.6336	TES01390
46.9404	TES01400
47.2472	TES01410
47.5540	TES01420
47.8608	TES01430
48.1676	TES01440
48.4744	
48.7812	
49.0880	
49.3948	
49.7016	
50.0084	
50.3152	
50.6220	
50.9288	
51.2356	
51.5424	
51.8492	
52.1560	
52.4628	
52.7696	
53.0764	
53.3832	
53.6900	
53.9968	
54.3036	
54.6104	
54.9172	
55.2240	
55.5308	
55.8376	
56.1444	
56.4512	
56.7580	
57.0648	
57.3716	

PAGE 00006

VM/HPD 4.2 CMS SL422

...

A1 NATIONAL RESEARCH COUNCIL OF CANADA

INPUT

FILE: TESTM

57.6784
57.9852
58.2920
58.5988
58.9056
59.2124
59.5192
59.8260
60.1328
60.4396
60.7464
61.0532
61.3600

RRRRRRRRRR	SSSSSSSSSS	CCCCCCCC	SSSSSSSSSS	888888888	333333333	222222222	00000000
RRRRRRRRRR	SSSSSSSSSS	CCCCCCCC	SSSSSSSSSS	88888888888	33333333333	22222222222	000000000
RR RR	SS SS	CC CC	SS SS	88 88	33 33	22 22	00 0000
RR RR	SS	CC	SS	88 88	33	22	00 00 00
RR RR	SSS	CC	SSS	88 88	33	22	00 00 00
RRRRRRRRRR	SSSSSSSS	CC	SSSSSSSS	8888888	3333	22	00 00 00
RRRRRRRRRR	SSSSSSSS	CC	SSSSSSSS	8888888	3333	22	00 00 00
RR RR	SSS	CC	SSS	88 88	33	22	00 00 00
RR RR	SS	CC	SS	88 88	33	22	0000 00
RR RR	SS SS	CC CC	SS SS	88 88	33 33	22	000 000
RR RR	SSSSSSSSSS	CCCCCCCC	SSSSSSSSSS	88888888888	33333333333	22222222222	0000000000
RR RR	SSSSSSSS	CCCCCCCC	SSSSSSSS	888888888	333333333	22222222222	00000000

EEEEEEEEEE	NN	NN	DDDDDDDD	888888888	333333333	222222222	00000000
EEEEEEEEEE	NNN	NN	DDDDDDDD	88888888888	33333333333	22222222222	000000000
EE	NNNN	NN	DD DD	88 88	33 33	22 22	00 0000
EE	NN NN	NN	DD DD	88 88	33	22	00 00 00
EE	NN NN	NN	DD DD	88 88	33	22	00 00 00
EEEEEEEE	NN NN	NN	DD DD	8888888	3333	22	00 00 00
EEEEEEEE	NN NN	NN	DD DD	8888888	3333	22	00 00 00
EE	NN NN	NN	DD DD	88 88	33	22	00 00 00
EE	NN NNN	NN	DD DD	88 88	33 33	22	0000 00
EE	NN NNN	NN	DD DD	88 88	33 33	22	000 00
EEEEEEEEEE	NN	NN	DDDDDDDD	88888888888	33333333333	22222222222	0000000000
EEEEEEEEEE	NN	N	DDDDDDDD	888888888	333333333	22222222222	00000000

Final Output Graphs from SPLOTN2 FORTRAN A1

The El Centro horizontal component of the acceleration record and the response to it in the hydrodynamic pressures (hydrodynamic pressure history) and other graphs.

α_R = 0.00 (full damping)
NT = 4096 (number of points in the acceleration record and also number of steps used in the FFT analysis)
 Δw = 0.3068 rad/s (frequency step)

The El Centro component was recorded during the Imperial Valley earthquake in California on May 18, 1940. It had a magnitude of M6.6. The measured peak acceleration was $PHA = -3.41705 \text{ m/s}^2$ and number of points in the record was equal to 2687.

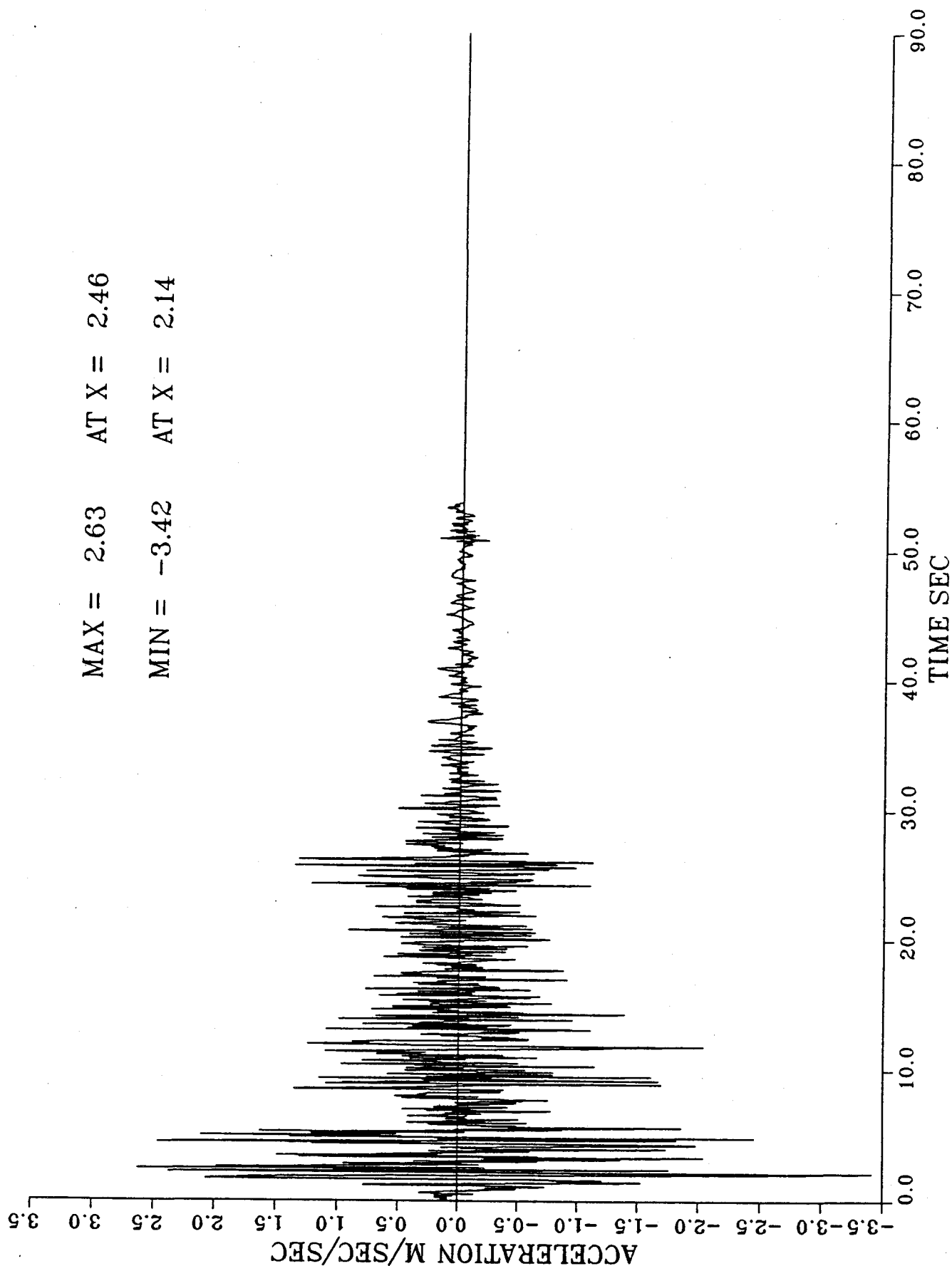
JABLONSK

[illegible][illegible]

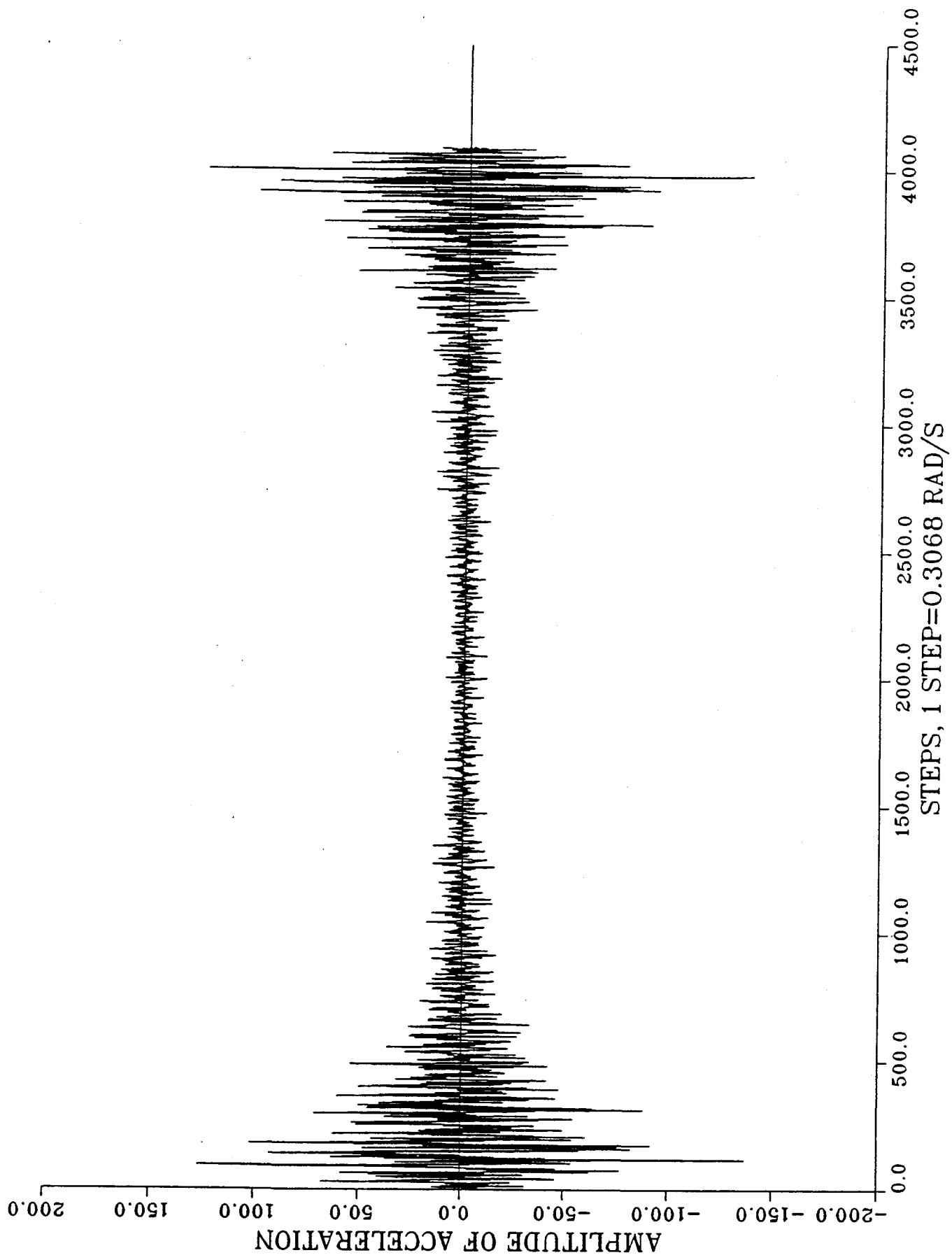
EL CENTRO HOR

MAX = 2.63 AT X = 2.46

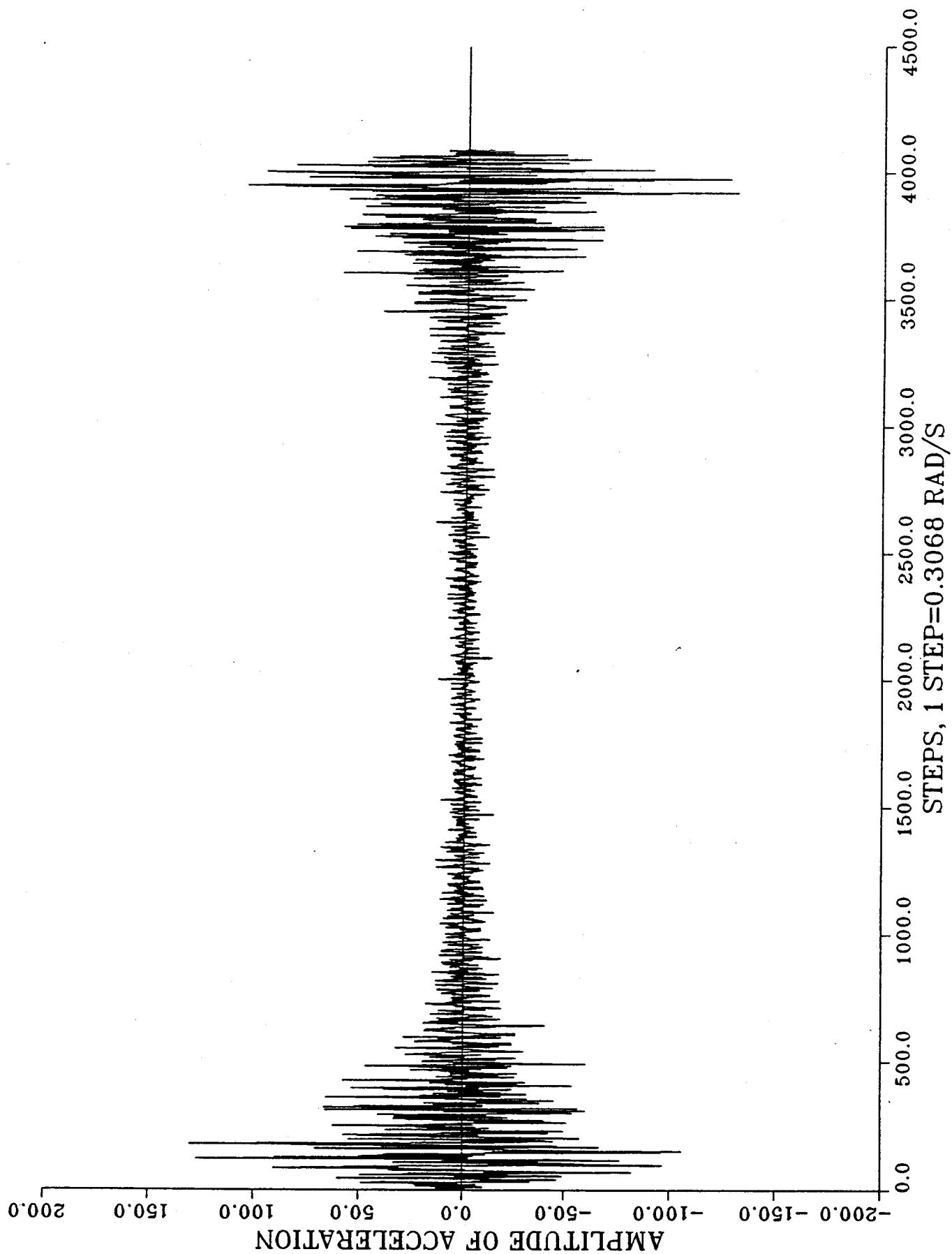
MIN = -3.42 AT X = 2.14



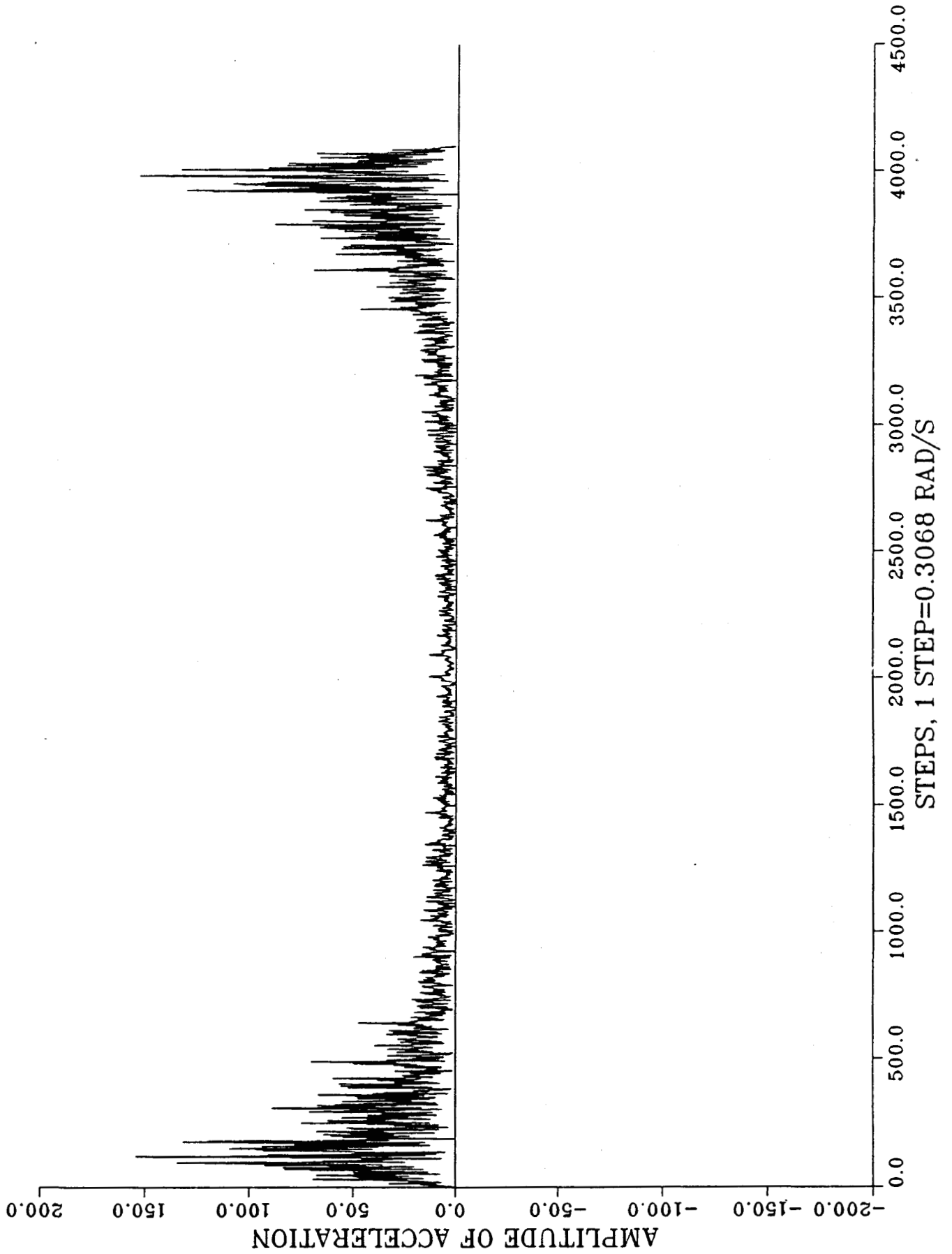
DFT OF ACCEL. - REAL PART



DFT OF ACCEL. - IMAG. PART



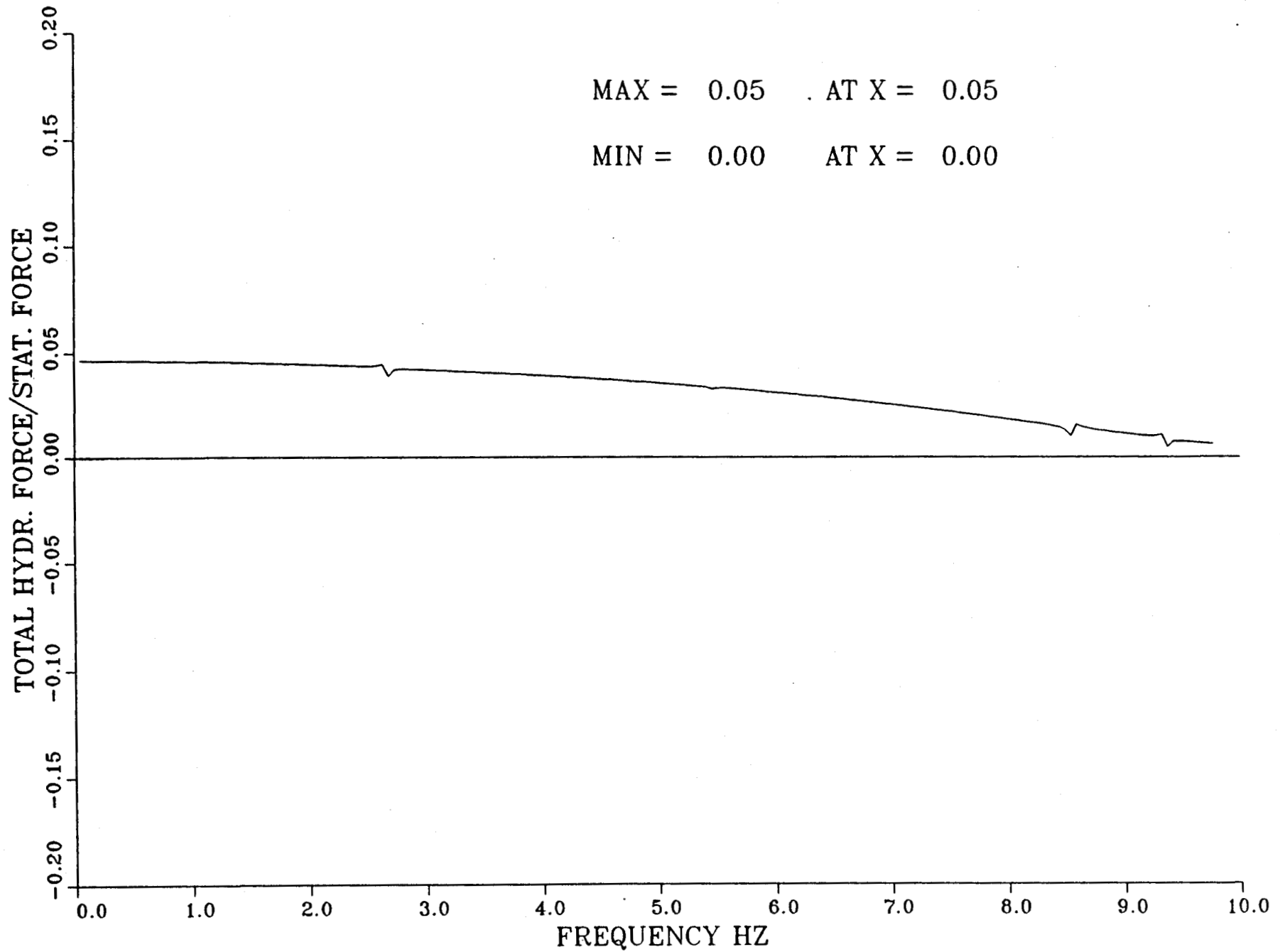
DFT OF ACCEL. - ABS. VALUE



HARM. RESPONSE – REAL PART

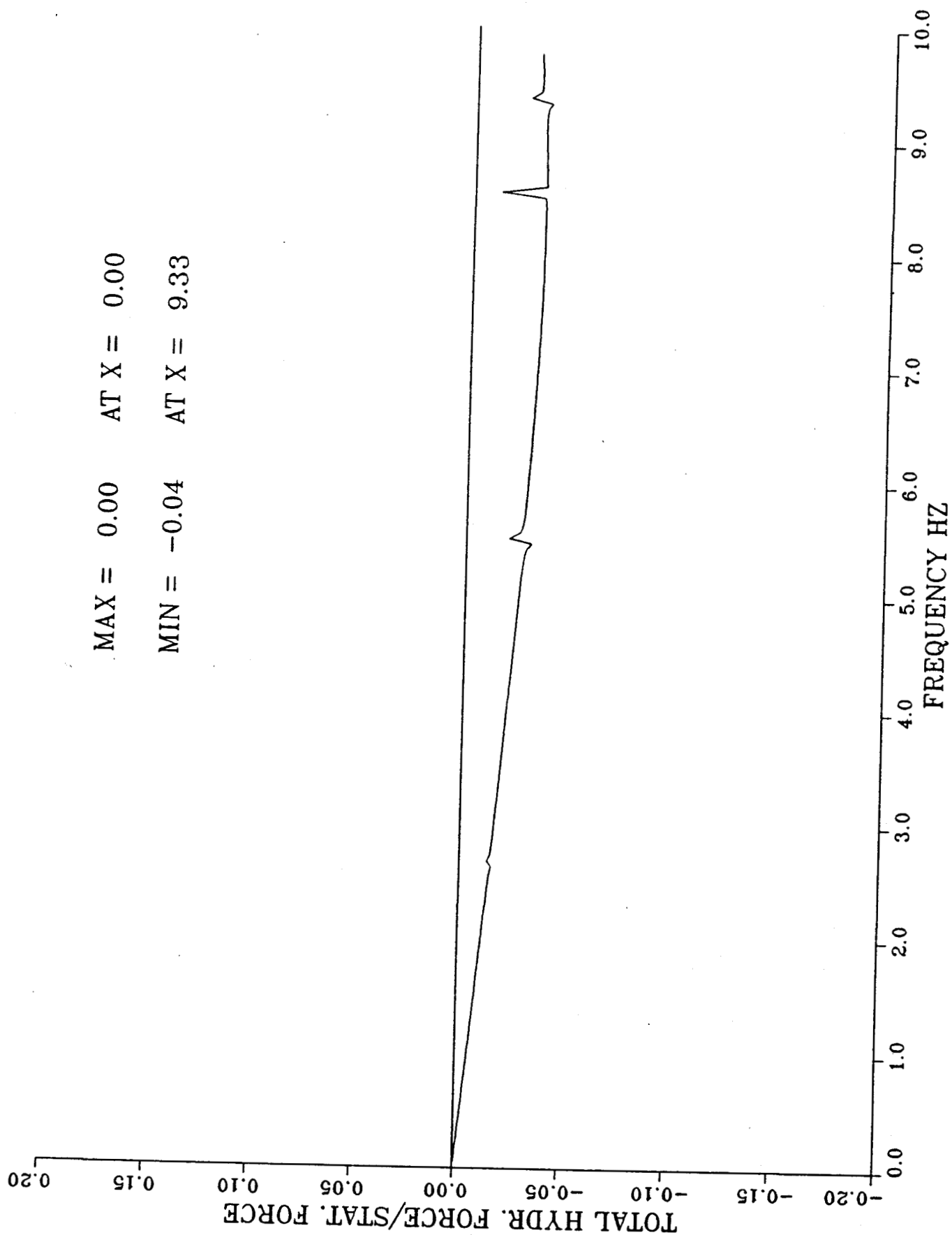
MAX = 0.05 . AT X = 0.05

MIN = 0.00 AT X = 0.00

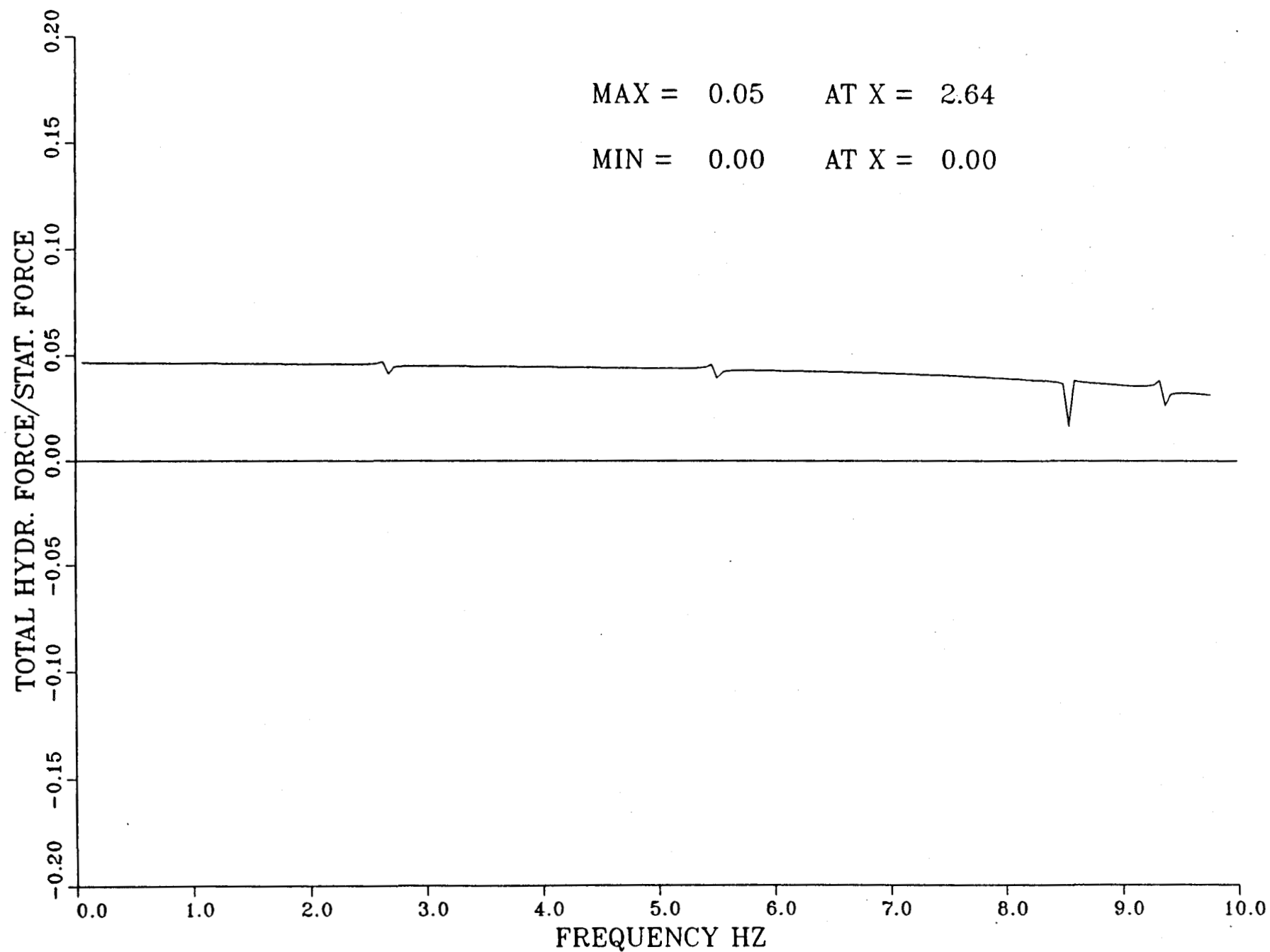


HARM. RESPONSE - IMAG. PART

MAX = 0.00 AT X = 0.00
MIN = -0.04 AT X = 9.33

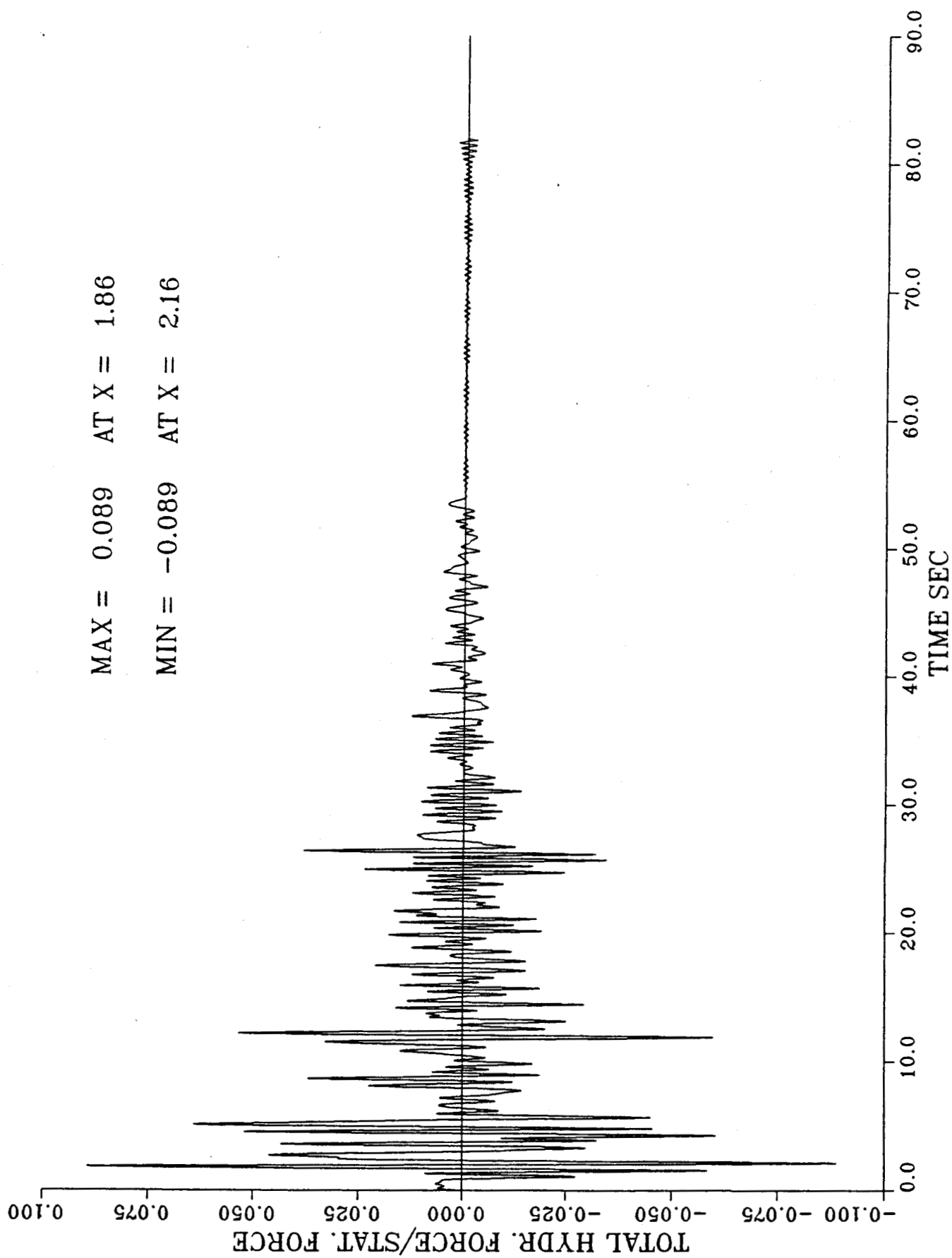


HARM. RESPONSE - ABS. VALUE



EL CENTRO RESPONSE

- 22 -



USERID: ALEXNMP

CLASS: A

SPoolID: 4366

PAGES PRINTED: 00010

DEVICE: 221LASER

KILOBYTES: 00372

FORM: PAPER11

COMPLETED: 90/09/04 15:44:58

APPENDIX B

LISTINGS OF COMPUTER PROGRAMS

B.1	Program ACCELN FORTRAN A1	25
B.2	Program BEMC2DN FORTRAN A1	30
B.3	Program FOURTN FORTRAN A1	84
B.4	Program SPLOTN2 FORTRAN A1	91
B.5	program SUM FORTRAN A1	102

B.1 Program ACCELN FORTRA A1


```

RRRRRRRRRR MM MM
RRRRRRRRRRR MMM MM
RR RR MMMM MMMM
RR RR MM MM MM MM
RR RR MM MMMM MM
RRRRRRRRRRR MM MM MM
RRRRRRRRRRR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM

```

```

11 5555555555 11
111 5555555555 111
1111 55 1111
11 55 11
11 55 11
11 55555555 11
11 5555555555 11
11 55 11
11 55 11
11 55 11
1111111111 5555555555 1111111111
1111111111 5555555555 1111111111

```

```

MM MM
MMM MMM
MMMM MMMM
MM MM MM MM
MM MMMM MM
MM MM MM
MM MM
MM MM
MM MM
MM MM
MM MM
MM MM

```

```

2222222222 00000000
22222222222 0000000000
22 22 00 0000
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 0000 00
22 000 00
22222222222 0000000000
22222222222 00000000

```

```

JJJJJJJJJ AAAAAAAAAA BBBB BBBB LL
JJJJJJJJJ AAAAAAAAAA BBBB BBBB LL
JJ AA AA BB BB LL
JJ AA AA BB BB LL
JJ AA AA BB BB LL
JJ AAAAAAAAAA BBBB BBBB LL
JJ AAAAAAAAAA BBBB BBBB LL
JJ AA AA BB BB LL
JJ JJ AA AA BB BB LL
JJ JJ AA AA BB BB LL
JJJJJJJ AA AA BBBB BBBB LLLLLLLLLL
JJJJJ AA AA BBBB BBBB LLLLLLLLLL

```

```

00000000000 NN NN SSSSSSSSS KK KK
00000000000 NNN NN SSSSSSSSSSS KK KK
00 00 NNNN NN SS SS KK KK
00 00 NN NN NN SS KK KK
00 00 NN NN NN SSS KK KK
00 00 NN NN NN SSSSSSSSS KKKKKKK
00 00 NN NN NN SSSSSSSSS KKKKKKK
00 00 NN NN NN SSS KK KK
00 00 NN NNNN SS KK KK
00 00 NN NNN SS SS KK KK
00000000000 NN NN SSSSSSSSSSS KK KK
00000000000 NN N SSSSSSSSS KK KK

```

```

RRRRRRRRRRR SSSSSSSSSS CCCCCCCCC SSSSSSSSSS 00000000 333333333 888888888 11
RRRRRRRRRRR SSSSSSSSSSS CCCCCCCCCC SSSSSSSSSSS 0000000000 33333333333 88888888888 111
RR RR SS SS CC CC SS SS 00 0000 33 33 88 88 1111
RR RR SS CC SS 00 00 00 33 88 88 11
RR RR SSS CC SSS 00 00 00 33 88 88 11
RRRRRRRRRRR SSSSSSSSS CC SSSSSSSSS 00 00 00 333 8888888 11
RRRRRRRRRRR SSSSSSSSS CC SSSSSSSSS 00 00 00 3333 88888888 11
RR RR SSS CC SSS 00 00 00 33 88 88 11
RR RR SS CC SS 0000 00 33 33 88 88 11
RR RR SS CC SS 000 00 33 33 88 88 11
RR RR SSSSSSSSSS CCCCCCCCCC SSSSSSSSSS 0000000000 33333333333 88888888888 1111111111
RR RR SSSSSSSSSS CCCCCCCCCC SSSSSSSSSS 00000000 3333333333 8888888888 1111111111

```

DATE/TIME= SEP 14, 1990 13:39:12
 USER ACCOUNT= ALEXNMP
 NODES D\E\C=NRCVM01 \NRCVM01 \NRCVM01
 JOB = J 381 START

| PRINT: CLASS= E
 | FORMS= S1PT
 | FCB= STD8
 | UCS= GT15

| DR. A.JABLONSKI
 | RM. 151
 | M-20

```
C      SUBROUTINE ACCEL(NT,DT,AC,A)                                ACC00010
C                                                                ACC00020
C      PROGRAMMER: ALEXANDER M. JABLONSKI                        ACC00030
C                                                                ACC00040
C      DATE: JULY 5, 1989                                       ACC00050
C      MODIFIED: JULY 6, 1989                                    ACC00060
C                                                                ACC00070
C      CENTER: STRUCTURES SECTION                                ACC00080
C               INSTITUTE FOR RESEARCH IN CONSTRUCTION          ACC00090
C               NATIONAL RESEARCH COUNCIL, CANADA               ACC00100
C                                                                ACC00110
C      NOTE: THIS SUBROUTINE READS A PROCESSED EARTHQUAKE DATA ACC00120
C            AND CONVERTS IT INTO INPUT DATA FILE FOR USE IN   ACC00130
C            FFT PROGRAM FOURT21                                 ACC00140
C                                                                ACC00150
C                                                                ACC00160
C      PROGRAM ACCELN                                           ACC00170
C      PARAMETER(NT=4096)                                       ACC00180
C      PARAMETER(NI=0)                                          ACC00190
C      PARAMETER(NL=2324)                                       ACC00200
C      NT=NUMBER OF TIME STEP IN FFT                            ACC00210
C      NI=INITIAL STEP NUMBER TO BE READ FROM THE RECORD FILE  ACC00220
C      NL=LAST STEP NUMBER TO BE READ FROM THE SAME RECORD     ACC00230
C                                                                ACC00240
C      REAL AC(9000)                                           ACC00250
C      REAL A(9000)                                             ACC00260
C                                                                ACC00270
C      CHARACTER*22 DATAS,INPS1                                ACC00280
C                                                                ACC00290
C      DATA DATAS/'/DATAS INPUT A1'/                          ACC00300
C      DATA INPS1/'/INPS1 INPUT A1'/                          ACC00310
C      LEC=2                                                    ACC00320
C      IMP=3                                                    ACC00330
C      OPEN(LEC,FORM='FORMATTED',STATUS='OLD',FILE=DATAS)      ACC00340
C      OPEN(IMP,FORM='FORMATTED',STATUS='NEW',FILE=INPS1)      ACC00350
C                                                                ACC00360
C      READ DATA FROM INPUT FILE DATAS                         ACC00370
C                                                                ACC00380
C                                                                ACC00390
C                                                                ACC00400
C      DO 10 I=1,NL                                             ACC00410
C      READ(LEC,200)AC(I)                                       ACC00420
C  10  CONTINUE                                                 ACC00430
C  200 FORMAT(E13.6)                                           ACC00440
C                                                                ACC00450
C      DO 11 I=1,NT                                             ACC00460
C      A(I)=0.0                                                 ACC00470
C  11  CONTINUE                                                 ACC00480
C                                                                ACC00490
C      DO 20 I=1,NT                                             ACC00500
C      J=I+NI                                                   ACC00510
C      IF(J.GT.NL) THEN                                         ACC00520
C      A(I)=0.0                                                 ACC00530
C      GO TO 20                                                 ACC00540
C      END IF                                                  ACC00550
```

```
      A(I)=AC(J)
      20 CONTINUE
C
      NN=1+(NT/8)
      DO 30 J=1,NN
      NX=(J-1)*8
      WRITE(IMP,100)(A(I+NX),I=1,8)
      30 CONTINUE
      100 FORMAT(8F9.3)
      STOP
      END
```

ACC00560
ACC00570
ACC00580
ACC00590
ACC00600
ACC00610
ACC00620
ACC00630
ACC00640
ACC00650
ACC00660

RRRRRRRRRR	SSSSSSSSSS	CCCCCCCCCC	SSSSSSSSSS	00000000	333333333	888888888	11
RRRRRRRRRR	SSSSSSSSSS	CCCCCCCCCC	SSSSSSSSSS	000000000	3333333333	8888888888	111
RR	RR	SS	SS	CC	SS	SS	00
RR	RR	SS	CC	SS	SS	00	00 00
RR	RR	SSS	CC	SSS	SS	00	00 00
RRRRRRRRRR	SSSSSSSSSS	CC	SSSSSSSSSS	00	00	00	3333
RRRRRRRRRR	SSSSSSSSSS	CC	SSSSSSSSSS	00	00	00	3333
RR	RR	SSS	CC	SSS	SS	00	00 00
RR	RR	SS	CC	SS	SS	0000	00
RR	RR	SS	SS	CC	SS	000	00
RR	RR	SSSSSSSSSS	CCCCCCCCCC	SSSSSSSSSS	000000000	3333333333	8888888888
RR	RR	SSSSSSSSSS	CCCCCCCCCC	SSSSSSSSSS	00000000	3333333333	8888888888

EEEEEEEEEE	NN	NN	DDDDDDDD	333333333	888888888	11
EEEEEEEEEE	NNN	NN	DDDDDDDD	3333333333	8888888888	111
EE	NNNN	NN	DD	33	33	88
EE	NN NN	NN	DD	33	88	88
EE	NN NN	NN	DD	33	88	88
EEEEEEEE	NN	NN	DD	3333	88888888	11
EEEEEEEE	NN	NN NN	DD	3333	88888888	11
EE	NN	NN NN	DD	33	88	88
EE	NN	NNNN	DD	33	88	88
EE	NN	NNN	DD	33	88	88
EEEEEEEEEE	NN	NN	DDDDDDDD	3333333333	8888888888	111111111
EEEEEEEEEE	NN	N	DDDDDDDD	333333333	888888888	111111111

B.2 Program BEMC2DN FORTRAN A1

RRRRRRRRRR	MM	MM	11	5555555555	11
RRRRRRRRRR	MMM	MMM	111	5555555555	111
RR	RR	MMM	1111	55	1111
RR	RR	MM MM	11	55	11
RR	RR	MM MMM	11	55	11
RRRRRRRRRR	MM	MM	11	55555555	11
RRRRRRRRRR	MM	MM	11	55555555	11
RR	RR	MM	11	55	11
RR	RR	MM	11	55	11
RR	RR	MM	11	55	11
RR	RR	MM	1111111111	5555555555	1111111111
RR	RR	MM	1111111111	5555555555	1111111111

MM	MM	222222222	00000000
MMM	MMM	22222222222	0000000000
MMMM	MMMM	22	22 00 0000
MM MM	MM MM	22	22 00 00 00
MM MMMM	MM	22	22 00 00 00
MM MM	MM	22	22 00 00 00
MM	MM	22	22 00 00 00
MM	MM	22	22 00 00 00
MM	MM	22	22 0000 00
MM	MM	22	22 000 00
MM	MM	22222222222	0000000000
MM	MM	22222222222	00000000

JJJJJJJJJ	AAAAAAAAA	BBBBBBBBBB	LL	00000000000	NN	NN	SSSSSSSSSS	KK	KK
JJJJJJJJJ	AAAAAAAAA	BBBBBBBBBB	LL	00000000000	NNN	NN	SSSSSSSSSSSS	KK	KK
JJ	AA	AA	BB	LL	00	00	NNNN	NN	SS
JJ	AA	AA	BB	BB	LL	00	00	NN NN	NN
JJ	AA	AA	BB	BB	LL	00	00	NN NN	NN
JJ	AAAAAAAAA	BBBBBBBBBB	LL	00	00	NN NN	NN	SSSSSSSS	KKKKKK
JJ	AAAAAAAAA	BBBBBBBBBB	LL	00	00	NN NN	NN	SSSSSSSS	KKKKKK
JJ	AA	AA	BB	BB	LL	00	00	NN NN	NN
JJ	AA	AA	BB	BB	LL	00	00	NN NN	NN
JJ	AA	AA	BB	BB	LL	00	00	NN NN	NN
JJJJJJJ	AA	AA	BBBBBBBBBB	LLLLLLLLLLLL	00000000000	NN	NN	SSSSSSSSSSSS	KK
JJJJJ	AA	AA	BBBBBBBBBB	LLLLLLLLLLLL	00000000000	NN	N	SSSSSSSSSS	KK

RRRRRRRRRR	SSSSSSSSSS	CCCCCCCCC	SSSSSSSSSS	444	444	222222222	666666666
RRRRRRRRRR	SSSSSSSSSSSS	CCCCCCCCCCC	SSSSSSSSSSSS	4444	4444	22222222222	66666666666
RR	RR	SS	SS	CC	CC	SS	SS
RR	RR	SS	SS	CC	CC	SS	SS
RR	RR	SSS	SSS	CC	CC	SSS	SSS
RRRRRRRRRR	SSSSSSSSSS	CC	SSSSSSSSSS	44444444444	44444444444	22	66666666666
RRRRRRRRRR	SSSSSSSSSS	CC	SSSSSSSSSS	44444444444	44444444444	22	66666666666
RR	RR	SSS	SSS	CC	CC	SSS	SSS
RR	RR	SS	SS	CC	CC	SS	SS
RR	RR	SS	SS	CC	CC	SS	SS
RR	RR	SSSSSSSSSSSS	CCCCCCCCCCC	SSSSSSSSSSSS	SSSSSSSSSSSS	44	44
RR	RR	SSSSSSSSSS	CCCCCCCCC	SSSSSSSSSS	SSSSSSSSSS	44	44

DATE/TIME= JUL 31, 1990 11:21:02	PRINT: CLASS= E	DR. A.JABLONSKI
USER ACCOUNT= ALEXNMP	FORMS= S1PT	RM. 151
NODES D\E\C=NRCVM01 \NRCVM01	FCB= STD8	M-20
JOB = J4426 START	UCS= GT15	

```

C                                     BEM00010
C *****BEM00020
C                                     BEM00030
C PROGRAM BEMC.2D.I.FTN                                     BEM00040
C                                     BEM00050
C PROGRAM FOR THE SOLUTION OF A TWO-DIMENSIONAL RESERVOIR BEM00060
C PROBLEM WITH ONE INFINITE BOUNDARY AND DAMPING INCLUDED BEM00070
C BY CONSTANT BOUNDARY ELEMENT METHOD BEM00080
C SPECIAL B.C. AT FAR BOUNDARY THROUGH FEM, DAMPING INCLUDED BEM00090
C                                     BEM00100
C PROGRAMMER: ALEXANDER M. JABLONSKI BEM00110
C                                     BEM00120
C DATE: JULY 9, 1987, BEM00130
C SUBSTANTIALLY MODIFIED AUGUST 27, 1987 BEM00140
C MODIFIED JUNE 27,1989 BEM00150
C MODIFIED AUGUST 1, 1989 BEM00160
C                                     BEM00170
C CENTER: CASCADE, CARLETON UNIVERSITY BEM00180
C                                     BEM00190
C NOTE: THIS PROGRAM IS BASED ON PROGRAM BEM1.2D.I.FTN BEM00200
C                                     BEM00210
C *****BEM00220
C N = NUMBER OF BOUNDARY ELEMENTS, READ THROUGH INPUT BEM00230
C X = ARRAY CONTAINING X COORDINATES OF THE NODES BEM00240
C Y = ARRAY CONTAINING Y COORDINATES OF THE NODES BEM00250
C XM = ARRAY CONTAINING X COORDINATES OF THE MIDPOINTS OF THE ELEMENBEM00270
C YM = ARRAY CONTAINING Y COORDINATES OF THE MIDPOINTS OF THE ELEMENBEM00280
C G = MATRIX IN THE BOUNDARY ELEMENT EQUATION BEM00290
C H = MATRIX IN THE BOUNDARY ELEMENT EQUATION BEM00300
C FID= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS BEM00310
C ON THE DAM FACE BEM00320
C FIB= ARRAY OF PRESCRIBED VALUES OF BOUNDARY CONDITIONS BEM00330
C ON THE BASE BEM00340
C DFI = RIGHT HAND SIDE VECTOR IN BOUNDARY INTEGRAL EQUATION; BEM00350
C AFTER SOLUTION, IT CONTAINS THE VALUES AT THE BOUNDARY BEM00360
C THAT WERE PREVIOUSLY UNKNOWN BEM00370
C LEC = UNIT NO. CONTAINING INPUT BEM00380
C IMP = UNIT NO. ON WHICH OUTPUT IS WRITTEN BEM00390
C NX = 2*N = DIMENSION OF THE SYSTEM OF EQUATIONS BEM00400
C NU = DIMENSION OF H AND G MATRICES IN THE MAIN PROGRAM BEM00410
C ND = DIMENSION OF THE SPECIAL BOUNDARY CONDITION MATRIX AT BEM00420
C FAR BOUNDARY IN THE MAIN PROGRAM BEM00430
C D = VALUE OF THE DETERMINANT OF COEFFICIENT MATRIX BEM00440
C CONST = OMEGA/C BEM00450
C OMEGA = EXCITING FREQUENCY BEM00460
C C = VELOCITY OF SOUND IN WATER BEM00470
C CONST1=OMEGA BEM00480
C CONST2=FOUNDATION DAMPING COEFFICIENT GAMMA BEM00490
C AFR=REAL PART OF EIGENVECTOR SI BEM00500
C AFI=IMAGINARY PART OF EIGENVECTOR SI BEM00510
C AKR=REAL PART OF KAPPA MATRIX BEM00520
C AKI=IMAGINARY PART OF KAPPA MATRIX BEM00530
C NO=LAST ELEMENT NUMBER AT TOP OF RESERVOIR NEAR THE DAM BEM00540
C NI=LAST ELEMENT NUMBER AT THE BASE OF THE DAM BEM00550

```

C	N2=LAST ELEMENT NUMBER AT THE BASE NEAR FAR END	BEM00560
C	MF=NUMBER OF ELEMENTS ON BASE	BEM00570
C	ME=NUMBER OF ELEMENTS ON THE FAR BOUNDARY	BEM00580
C	HY=HEIGHT OF THE RESERVOIR	BEM00590
C		BEM00600
	PROGRAM BEMC2D	BEM00610
	IMPLICIT DOUBLE PRECISION (A-H,O-Z)	BEM00620
	COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND	BEM00630
	COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2	BEM00640
	DOUBLE PRECISION X(101),Y(101),G(200,200),H(200,200),	BEM00650
	&DFI(200),FID(50),FIB(50),HOLD(200,200),XM(101),YM(101)	BEM00660
	REAL AFI(20,20),AFR(20,20),AKR(20),AKI(20)	BEM00670
	REAL YR(20,500),YI(20,500)	BEM00680
	COMPLEX YC(20,500),TX(500),HX(500)	BEM00690
	LOGICAL FLAG	BEM00700
	CHARACTER*22 TESTM,OUTM,EIGENM,INFTM	BEM00710
	DATA TESTM '/'TESTM INPUT A1'/'	BEM00720
	DATA OUTM '/'OUTM OUTPUT C'/'	BEM00730
	DATA EIGENM '/'EIGENM OUTPUT C'/'	BEM00740
	DATA INFTM '/'INFTM INPUT A1'/'	BEM00750
	LEC=2	BEM00760
	IMP=3	BEM00770
	IWRITE=4	BEM00780
	NO=20	BEM00790
	NU=200	BEM00800
	PI=3.141592653	BEM00810
	OPEN (LEC,FORM='FORMATTED',STATUS='OLD',FILE=TESTM)	BEM00820
	OPEN (IMP,FORM='FORMATTED',STATUS='NEW',FILE=OUTM)	BEM00830
	OPEN (IWRITE,FORM='FORMATTED',STATUS='NEW',FILE=EIGENM)	BEM00840
	OPEN (7,FORM='FORMATTED',STATUS='NEW',FILE=INFTM)	BEM00850
C		BEM00860
C	READ DATA FROM INPUT FILE	BEM00870
C		BEM00880
	CALL INPUT(X,Y,FID,FIB,NO,N1,N2,NDUT,C,DF)	BEM00890
	NX=2*N	BEM00900
	NC=0	BEM00910
5	WRITE(IMP,300)	BEM00920
	READ(LEC,*,END=15)OMEGA	BEM00930
	NC=NC+1	BEM00940
	CONST=OMEGA/C	BEM00950
	WRITE(IMP,350)OMEGA,CONST	BEM00960
	CONST1=OMEGA	BEM00970
C		BEM00980
C	COMPUTE EIGENVALUES AND EIGENVECTORS FOR THE FAR BOUNDARY	BEM00990
C		BEM01000
	IF(ME.GT.0)THEN	BEM01010
	IF(CONST2.GT.1.0E-08)THEN	BEM01020
	FLAG=.TRUE.	BEM01030
	CALL EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI)	BEM01040
	ELSE	BEM01050
	CALL EIGENR(AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI,IMP,FLAG)	BEM01060
	END IF	BEM01070
*	WRITE(6,2001)	BEM01080
*2001	FORMAT(' ')	BEM01090
		BEM01100


```

*      DO 10 I=1,ME
*      WRITE(6,2010)AKR(I),AKI(I)
*      10 CONTINUE
*      write (6,2001)
*      do 2003 i=1,me
*      write(6,2010)(afr(i,j),j=1,me)
*      2003 continue
*      write(6,2001)
*      END IF
C
C      FORM SYSTEM OF EQUATIONS
C
C      CALL FMAT(X,Y,XM,YM,G,H,HOLD,FID,FIB,DFI,NO,N1)
*      do 2000 i=1,n
*      write(6,2010)(h(i,j),j=1,n)
*      2000 continue
*      write(6,2001)
*      write(6,2001)
*      do 2002 i=1,n
*      write(6,2010)(g(i,j),j=1,n)
*      2002 continue
C
C      MODIFY H MATRIX FOR BOUNDARY CONDITIONS AT BASE AND FAR END
C      AND COMPUTE DFI
C
      IF(CONST2.GT.1.0E-08)THEN
      CALL ADDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,NO,N1,N2,DFI,FID,FIB,DF)
      ELSE
      CALL ADDUND(H,G,HOLD,AFR,AKR,AKI,NO,N1,N2,DFI,FID,FIB,FLAG,DF)
      END IF
*      write(6,2001)
*      write(6,2001)
*      do 2005 i=1,nx
*      write(6,2010)(h(i,j),j=1,nx)
*      2005 continue
*      write(6,2001)
*      write(6,2001)
*      write(6,2010)(dfr(i),i=1,nx)
C
C      SOLVE SYSTEM OF EQUATIONS
C
      IF(FLAG)THEN
      CALL SLNPD(H,DFI,D,NX,NU,IMP)
      ELSE
      CALL SLNPD(H,DFI,D,N,NU,IMP)
      END IF
*      DO 999 I=1,N
*      WRITE(IMP,*IDFI(I)
*      999 CONTINUE
*      IF(FLAG)THEN
*      DO 1000 I=1,N
*      WRITE(IMP,*IDFI(I+N)
*      1000 CONTINUE

```

BEM01110
 BEM01120
 BEM01130
 BEM01140
 BEM01150
 BEM01160
 BEM01170
 BEM01180
 BEM01190
 BEM01200
 BEM01210
 BEM01220
 BEM01230
 BEM01240
 BEM01250
 BEM01260
 BEM01270
 BEM01280
 BEM01290
 BEM01300
 BEM01310
 BEM01320
 BEM01330
 BEM01340
 BEM01350
 BEM01360
 BEM01370
 BEM01380
 BEM01390
 BEM01400
 BEM01410
 BEM01420
 BEM01430
 BEM01440
 BEM01450
 BEM01460
 BEM01470
 BEM01480
 BEM01490
 BEM01500
 BEM01510
 BEM01520
 BEM01530
 BEM01540
 BEM01550
 BEM01560
 BEM01570
 BEM01580
 BEM01590
 BEM01600
 BEM01610
 BEM01620
 BEM01630
 BEM01640
 BEM01650

```

*      END IF
C
C
C      OUTPUT RESULTS
C
      CALL OUTPUT(XM,YM,FID,FIB,DFI,AFR,AFI,AKR,AKI,
* NO,N1,N2,NOUT,FLAG,DF,NC,TPR,TPI,HXR,HXI,YR,YI)
      TX(NC)=CMPLX(TPR,TPI)
      HX(NC)=CMPLX(HXR,HXI)
      WRITE(7,*)NC,TX(NC),HX(NC)
      WRITE(7,300)
      DO 30 I=1,N1-N0
      YCR=YR(I,NC)
      YCI=YI(I,NC)
      YC(I,NC)=CMPLX(YCR,YCI)
      WRITE(7,*)YC(I,NC)
30  CONTINUE
      GO TO 5
C
2010 format(6e13.5)
300 FORMAT(//,1X,70('*'),//)
350 FORMAT(1X,'EXCITING FREQUENCY',E14.5,/,1X,
* 'WAVE CONSTANT',E14.5,/)
15 STOP
      END
C
C
C      -----
C
      SUBROUTINE INPUT(X,Y,FID,FIB,N0,N1,N2,NOUT,C,DF)
*****
C
C      READS INPUT DATA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
      DOUBLE PRECISION X(1),Y(1),FID(1),FIB(1),C
      CHARACTER*4 TITLE(18)
      WRITE (IMP,100)
C
C      READ TITLE
C
      READ (LEC,150)TITLE
C
C      READ BASIC PARAMETERS
C
      READ(LEC,*)N,N0,N1,N2,NOUT
      READ(LEC,*)HY,C,ALF,DF
      CONST2=((1-ALF)/(1+ALF))*(1/C)
C
      WRITE(IMP,300)N,N0,N1,N2,NOUT
      WRITE(IMP,350)HY,C,ALF,CONST2
C
C
C      READ COORDINATES OF NODAL POINTS OF THE

```

```

BEM01660
BEM01670
BEM01680
BEM01690
BEM01700
BEM01710
BEM01720
BEM01730
BEM01740
BEM01750
BEM01760
BEM01770
BEM01780
BEM01790
BEM01800
BEM01810
BEM01820
BEM01830
BEM01840
BEM01850
BEM01860
BEM01870
BEM01880
BEM01890
BEM01900
BEM01910
BEM01920
BEM01930
BEM01940
BEM01950
BEM01960
BEM01970
BEM01980
BEM01990
BEM02000
BEM02010
BEM02020
BEM02030
BEM02040
BEM02050
BEM02060
BEM02070
BEM02080
BEM02090
BEM02100
BEM02110
BEM02120
BEM02130
BEM02140
BEM02150
BEM02160
BEM02170
BEM02180
BEM02190
BEM02200

```

```

C      BOUNDARY ELEMENTS IN ARRAY X AND Y      BEM02210
C      WRITE(IMP,500)      BEM02220
      DO 10 I=1,N      BEM02230
      READ(LEC,*)X(I),Y(I)      BEM02240
      10 WRITE(IMP,700)I,X(I),Y(I)      BEM02250
C      BEM02260
C      BEM02270
C      BEM02280
C      READ BOUNDARY CONDITION ON THE DAM FACE      BEM02290
C      BEM02300
      WRITE(IMP,760)      BEM02310
      WRITE(IMP,800)      BEM02320
      ILAST=N1-N0      BEM02330
      DO 20 I=1,ILAST      BEM02340
      READ(LEC,*)FID(I)      BEM02350
      WRITE(IMP,750)I,FID(I)      BEM02360
      20 CONTINUE      BEM02370
C      BEM02380
C      BEM02390
C      READ BOUNDARY CONDITION ON THE BASE OF THE RESERVOIR      BEM02400
C      BEM02410
      WRITE(IMP,770)      BEM02420
      WRITE(IMP,800)      BEM02430
      ILAST=N2-N1      BEM02440
      DO 30 I=1,ILAST      BEM02450
      READ(LEC,*)FIB(I)      BEM02460
      WRITE(IMP,750)I,FIB(I)      BEM02470
      30 CONTINUE      BEM02480
      ME=N-N2      BEM02490
      MF=N2-N1      BEM02500
C      BEM02510
      100 FORMAT(1X,60('*'))      BEM02520
      150 FORMAT(18A4)      BEM02530
      300 FORMAT(1X,'DATA',/,1X,'NUMBER OF BOUNDARY ELEMENTS',      BEM02540
      *I3,/,1X,'ELEMENT NUMBER AT TOP OF RESERVOIR NEAR THE DAM',I3,/,      BEM02550
      *1X,'ELEMENT NUMBER AT BASE OF DAM',I3,/,1X,      BEM02560
      *'ELEMENT NUMBER AT BOTTOM NEAR FAR END',I3,/,1X,      BEM02570
      *'FLAG FOR DETAILED OUTPUT',I3,/)      BEM02580
      350 FORMAT(1X,'HEIGHT OF DAM',E14.5,/,1X,      BEM02590
      *'VELOCITY OF SOUND IN WATER',E14.5,/,1X,      BEM02600
      *'WAVE REFLECTION COEFFICIENT',E14.5,/,1X,      BEM02610
      *'FOUNDATION DAMPING COEFFICIENT',E14.5//)      BEM02620
      400 FORMAT(2(2X,E14.7,2X))      BEM02630
      500 FORMAT(' COORDINATES OF THE NODAL POINTS',      BEM02640
      *//,4X,'POINT',10X,'X',18X,'Y')      BEM02650
      700 FORMAT(5X,I3,2(5X,E14.7))      BEM02660
      800 FORMAT(5X,'NODE',6X,      BEM02670
      *'PRESCRIBED VALUE')      BEM02680
      750 FORMAT(5X,I3,8X,E14.7)      BEM02690
      760 FORMAT(///' BOUNDARY CONDITIONS ON THE DAM FACE',/)      BEM02700
      770 FORMAT(///' BOUNDARY CONDITIONS ON THE RESERVOIR BOTTOM',/)      BEM02710
C      BEM02720
C      BEM02730
      RETURN      BEM02740
      END      BEM02750

```

```

C ----- BEM02760
C ----- BEM02770
C ----- BEM02780
SUBROUTINE EIGEN(AFI,AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI) BEM02790
C ***** BEM02800
DOUBLE PRECISION CONST,CONST1,CONST2,HY BEM02810
REAL AFI(ND,ND),AFR(ND,ND),AKR(ND),AKI(ND) BEM02820
C BEM02830
DY=SNGL(HY/ME) BEM02840
T=SNGL(CONST) BEM02850
T1=SNGL(CONST1) BEM02860
T2=SNGL(CONST2) BEM02870
C BEM02880
CALL FSFEMI(AFI,AFR,ME,ND,T,T1,T2,DY,AKR,AKI) BEM02890
C BEM02900
RETURN BEM02910
END BEM02920
C BEM02930
C ----- BEM02940
C ----- BEM02950
SUBROUTINE FSFEMI(AFI,AFR,N,NM,T,T1,T2,DY,AKR,AKI) BEM02960
C ***** BEM02970
C BEM02980
AFR= REAL PART OF THE EIGENVECTOR MATRIX BEM02990
AFI= IMAGINARY PART OF EIGENVECTOR MATRIX BEM03000
AKR=REAL PART OF KAPPA MATRIX DIAGONAL BEM03010
AKI=IMAGINARY PART OF KAPPA MATRIX DIAGONAL BEM03020
C BEM03030
N=ME= ND. OF ELEMENTS ON THE FAR BOUNDARY BEM03040
NM=ND= MAX. DIMENSION OF TNE EIGENVECTOR MATRIX BEM03050
DY= PASSED LENGTH OF AN ELEMENT ( DP IN THE MAIN PROGRAM ) BEM03060
C BEM03070
T=CONST= OMEGA/C ( DP IN THE MAIN PROGRAM ) BEM03080
C BEM03090
T1=CONST1= OMEGA ( DP IN THE MAIN PROGRAM ) BEM03100
C BEM03110
T2=CONST2=GAMMA BEM03120
C BEM03130
REAL A( 20, 20),B( 20, 20),Z( 20, 20),BHOLD( 20, 20), BEM03140
$ AHOLD( 20, 20),AHOLDI( 20, 20),BHOLDI( 20,20), BEM03150
$ ALFR( 20),ALFI( 20),BETA( 20), BEM03160
$ EPS1, AI(20,20),BI(20,20),ZI(20,20) BEM03170
REAL AMAX1 BEM03180
COMPLEX FI(20,20),XM(20,20) BEM03190
COMPLEX XKAP(20),XK2(20) BEM03200
REAL AFR(NM,NM),AFI(NM,NM),AKR(NM),AKI(NM) BEM03210
INTEGER ERROR BEM03220
DIMENSION INDEX(20) BEM03230
DATA IWRITE/4/ BEM03240
C BEM03250
CALL SET(DY,T1,T2,N,NM,A,AI,B,BI) BEM03260
DO 717 I=1,N BEM03270
DO 718 J=1,N BEM03280
AHOLD(I,J)=A(I,J) BEM03290
BHOLD(I,J)=B(I,J) BEM03300
AHOLDI(I,J)=AI(I,J) BEM03310
BHOLDI(I,J)=BI(I,J) BEM03320
718 CONTINUE BEM03330
717 CONTINUE BEM03340

```

```

      WRITE(IWRITE,20)
      $      N
20  FORMAT(30H1THE FULL MATRIX A  OF ORDER,
      $      I4,22H IS (PRINTED BY ROWS)/)
      DO 30 I = 1,N
30  WRITE(IWRITE,40)
      $      (A(I,J),AI(I,J),J=1,N)
40  FORMAT(5(2F6.2,1HI,3X))
      WRITE(IWRITE,41)
      $      N
41  FORMAT(30H0THE FULL MATRIX B  OF ORDER,
      $      I4,22H IS (PRINTED BY ROWS)/)
      DO 42 I = 1,N
42  WRITE(IWRITE,43)
      $      (B(I,J),BI(I,J),J=1,N)
43  FORMAT(5(2F6.2,1HI,3X))

C
C
C
C
C      EIGENVALUES AND EIGENVECTORS USING CQZVAL AND CQZVEC
C
      EPS1 = 0.0
      CALL CQZHES(NM,N,A,AI,B,BI,.TRUE.,Z,ZI)
      CALL CQZVAL(NM,N,A,AI,B,BI,EPS1,ALFR,ALFI,BETA,
      $      .TRUE.,Z,ZI,ERROR)
      CALL CQZVEC(NM,N,A,AI,B,BI,ALFR,ALFI,BETA,Z,ZI)
      DO 719 I=1,N
      DO 720 J=1,N
      A(I,J)=AHOLD(I,J)
      B(I,J)=BHOLD(I,J)
      AI(I,J)=AHOLDI(I,J)
      BI(I,J)=BHOLDI(I,J)
720 CONTINUE
719 CONTINUE
      WRITE(IWRITE,292)
292  FORMAT(/15X,7HALFR(I),19X,7HALFI(I),19X,7HBETA(I))
      DO 295 I = 1,N
      WRITE(IWRITE,293)
      $      I,ALFR(I),ALFI(I),BETA(I)
293  FORMAT(I2,3(1PE23.6,3X))
295  CONTINUE
      WRITE(IWRITE,300)
300  FORMAT(/14X,35HCOMPUTED EIGENVALUE AND EIGENVECTOR,20X)
C
      DO 510 K = 1, N
      BETA(K) = AMAX1(BETA(K),1.0E-50)
      ALFR(K) = ALFR(K) / BETA(K)
      ALFI(K) = ALFI(K) / BETA(K)
C
C      ONE EIGENVECTOR.
C
340  WRITE(IWRITE,350)
      $      K,ALFR(K),ALFI(K),(Z(I,K),
      $      ZI(I,K),I=1,N)

```

```

BEM03310
BEM03320
BEM03330
BEM03340
BEM03350
BEM03360
BEM03370
BEM03380
BEM03390
BEM03400
BEM03410
BEM03420
BEM03430
BEM03440
BEM03450
BEM03460
BEM03470
BEM03480
BEM03490
BEM03500
BEM03510
BEM03520
BEM03530
BEM03540
BEM03550
BEM03560
BEM03570
BEM03580
BEM03590
BEM03600
BEM03610
BEM03620
BEM03630
BEM03640
BEM03650
BEM03660
BEM03670
BEM03680
BEM03690
BEM03700
BEM03710
BEM03720
BEM03730
BEM03740
BEM03750
BEM03760
BEM03770
BEM03780
BEM03790
BEM03800
BEM03810
BEM03820
BEM03830
BEM03840
BEM03850

```

```

350  FORMAT(/I2,1P2E23.6/(5X,2E23.6))
510  CONTINUE
      DO 610 K=1,N
      DO 620 L=1,N
      FI(K,L)=CMPLX(Z(K,L),ZI(K,L))
620  CONTINUE
610  CONTINUE
      DO 611 I=1,N
      WRITE(IWRITE,*) (FI(I,J),J=1,N)
611  CONTINUE
      CALL NGRMA(FI,B,N,NM,XM)
      DO 81 I=1,N
      WRITE(IWRITE,*) (FI(I,J),J=1,N)
81  CONTINUE
      DO 613 I=1,N
      DO 612 J=1,N
      AFR(I,J)=REAL(FI(I,J))
      AFI(I,J)=AIMAG(FI(I,J))
612  CONTINUE
613  CONTINUE
      DO 171 I=1,N
      ALFR(I)=ALFR(I)-T*T
      XK2(I)=CMPLX(ALFR(I),ALFI(I))
      XKAP(I)=CSQRT(XK2(I))
      WRITE(IWRITE,*)XKAP(I)
      AKR(I)=REAL(XKAP(I))
      AKI(I)=AIMAG(XKAP(I))
171  CONTINUE
      RETURN
      END
C
C -----
C
SUBROUTINE EIGENR(AFR,ND,ME,CONST,CONST1,CONST2,HY,AKR,AKI
1 ,IMP,FLAG)
C *****
C DOUBLE PRECISION CONST,CONST1,CONST2,HY
C REAL AFR(ND,ND),AKR(ND),AKI(ND)
C REAL ALFR(20),A(20,20),F(20,20)
C LOGICAL FLAG
C
C DO 10 I=1,ME
C AKR(I)=0
C AKI(I)=0
10 CONTINUE
C DY=SNGL(HY/ME)
C T=SNGL(CONST)
C T1=SNGL(CONST1)
C T2=SNGL(CONST2)
C FLAG=.FALSE.
C CALL SETR(DY,ME,ND,A,F)
C CALL JACOBI(A,F,ALFR,AFR,ME,ND)
C DO 30 I=1,ME
C TEMP=ALFR(I)-T*T
C TX=TEMP

```

```

BEM03860
BEM03870
BEM03880
BEM03890
BEM03900
BEM03910
BEM03920
BEM03930
BEM03940
BEM03950
BEM03960
BEM03970
BEM03980
BEM03990
BEM04000
BEM04010
BEM04020
BEM04030
BEM04040
BEM04050
BEM04060
BEM04070
BEM04080
BEM04090
BEM04100
BEM04110
BEM04120
BEM04130
BEM04140
BEM04150
BEM04160
BEM04170
BEM04180
BEM04190
BEM04200
BEM04210
BEM04220
BEM04230
BEM04240
BEM04250
BEM04260
BEM04270
BEM04280
BEM04290
BEM04300
BEM04310
BEM04320
BEM04330
BEM04340
BEM04350
BEM04360
BEM04370
BEM04380
BEM04390
BEM04400

```

```

      IF(TEMP.LT.0) THEN
      FLAG=.TRUE.
      AKI(I)=SQRT(-TX)
      ELSE IF(TEMP.GT.0) THEN
      AKR(I)=SQRT(TX)
      ELSE
      WRITE(IMP,100)
100  FORMAT(1X,'RESONANCE HAS TAKEN PLACE')
      STOP
      END IF
30  CONTINUE
      RETURN
      END
C
C -----
C
C SUBROUTINE CQZHES(NM,N,AR,AI,BR,BI,MATZ,ZR,ZI)
C *****
C THIS ALGORITHM QZ FOR THE CLASS OF COMPLEX GENERALIZED MATRIX
C SYSTEMS EXHIBITING THE USE OF QZ TO FIND ALL THE EIGENVALUES
C AND EIGENVECTORS FOR THE EIGENPROBLEM  $A*X = (\text{LAMBDA})*B*X$ .
C
C THE DIMENSION OF A,AI,B,BI,Z, AND ZI SHOULD BE NM BY NM.
C THE DIMENSION OF ALFR,ALFI,BETA, AND NORM SHOULD BE NM.
C THE DIMENSION OF AHOLD AND BHOLD SHOULD BE NM BY NM.
C THE DIMENSION OF AHOLDI AND BHOLDI SHOULD BE NM BY NM.
C HERE NM = 20.
C
C INTEGER I,J,K,L,N,K1,LB,L1,NM,NK1,NM1
C REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ZR(NM,N),ZI(NM,N)
C REAL R,S,T,TI,U1,U2,XI,XR,YI,YR,RHO,U1I
C REAL SQRT,CABS,ABS
C LOGICAL MATZ
C COMPLEX CMPLX
C
C THIS SUBROUTINE IS A COMPLEX ANALOGUE OF THE FIRST STEP OF THE
C QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,
C SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART.
C
C THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX GENERAL MATRICES AND
C REDUCES ONE OF THEM TO UPPER HESSENBERG FORM WITH REAL (AND NON-
C NEGATIVE) SUBDIAGONAL ELEMENTS AND THE OTHER TO UPPER TRIANGULAR
C FORM USING UNITARY TRANSFORMATIONS. IT IS USUALLY FOLLOWED BY
C CQZVAL AND POSSIBLY CQZVEC.
C
C ON INPUT-
C
C NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C DIMENSION STATEMENT,
C
C N IS THE ORDER OF THE MATRICES,
C
C A=(AR,AI) CONTAINS A COMPLEX GENERAL MATRIX,

```

```

BEM04410
BEM04420
BEM04430
BEM04440
BEM04450
BEM04460
BEM04470
BEM04480
BEM04490
BEM04500
BEM04510
BEM04520
BEM04530
BEM04540
BEM04550
BEM04560
BEM04570
BEM04580
BEM04590
BEM04600
BEM04610
BEM04620
BEM04630
BEM04640
BEM04650
BEM04660
BEM04670
BEM04680
BEM04690
BEM04700
BEM04710
BEM04720
BEM04730
BEM04740
BEM04750
BEM04760
BEM04770
BEM04780
BEM04790
BEM04800
BEM04810
BEM04820
BEM04830
BEM04840
BEM04850
BEM04860
BEM04870
BEM04880
BEM04890
BEM04900
BEM04910
BEM04920
BEM04930
BEM04940
BEM04950

```

```

C                                     BEM04960
C      B=(BR,BI) CONTAINS A COMPLEX GENERAL MATRIX,          BEM04970
C                                     BEM04980
C      MATZ SHOULD BE SET TO .TRUE. IF THE RIGHT HAND TRANSFORMATIONS BEM04990
C      ARE TO BE ACCUMULATED FOR LATER USE IN COMPUTING        BEM05000
C      EIGENVECTORS, AND TO .FALSE. OTHERWISE.                BEM05010
C                                     BEM05020
C      ON OUTPUT-                                              BEM05030
C                                     BEM05040
C      A HAS BEEN REDUCED TO UPPER HESSENBERG FORM. THE ELEMENTS BEM05050
C      BELOW THE FIRST SUBDIAGONAL HAVE BEEN SET TO ZERO, AND THE BEM05060
C      SUBDIAGONAL ELEMENTS HAVE BEEN MADE REAL (AND NON-NEGATIVE), BEM05070
C                                     BEM05080
C      B HAS BEEN REDUCED TO UPPER TRIANGULAR FORM. THE ELEMENTS BEM05090
C      BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,          BEM05100
C                                     BEM05110
C      Z=(ZR,ZI) CONTAINS THE PRODUCT OF THE RIGHT HAND        BEM05120
C      TRANSFORMATIONS IF MATZ HAS BEEN SET TO .TRUE.          BEM05130
C      OTHERWISE, Z IS NOT REFERENCED.                          BEM05140
C                                     BEM05150
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOV, BEM05160
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY BEM05170
C                                     BEM05180
C      -----BEM05190
C      ***** INITIALIZE Z ***** BEM05200
C      IF (.NOT. MATZ) GO TO 10 BEM05210
C                                     BEM05220
C      DO 3 I = 1, N BEM05230
C                                     BEM05240
C          DO 2 J = 1, N BEM05250
C              ZR(I,J) = 0.0 BEM05260
C              ZI(I,J) = 0.0 BEM05270
C          2 CONTINUE BEM05280
C                                     BEM05290
C          ZR(I,I) = 1.0 BEM05300
C          3 CONTINUE BEM05310
C      ***** REDUCE B TO UPPER TRIANGULAR FORM WITH BEM05320
C      TEMPORARILY REAL DIAGONAL ELEMENTS ***** BEM05330
C      10 IF (N .LE. 1) GO TO 170 BEM05340
C          NM1 = N - 1 BEM05350
C                                     BEM05360
C          DO 100 L = 1, NM1 BEM05370
C              L1 = L + 1 BEM05380
C              S = 0.0 BEM05390
C                                     BEM05400
C              DO 20 I = L, N BEM05410
C                  S = S + ABS(BR(I,L)) + ABS(BI(I,L)) BEM05420
C              20 CONTINUE BEM05430
C                                     BEM05440
C              IF (S .EQ. 0.0) GO TO 100 BEM05450
C              RHO = 0.0 BEM05460
C                                     BEM05470
C              DO 25 I = L, N BEM05480
C                  BR(I,L) = BR(I,L) / S BEM05490
C                  BI(I,L) = BI(I,L) / S BEM05500

```



```

      BI(I,L) = BI(I,L) / S
      RHO = RHO + BR(I,L)**2 + BI(I,L)**2
25  CONTINUE
C
      R = SQRT(RHO)
      XR = CABS(CMPLX(BR(L,L),BI(L,L)))
      IF (XR .EQ. 0.0) GO TO 27
      RHO = RHO + XR * R
      UI1 = -BR(L,L) / XR
      UII = -BI(L,L) / XR
      YR = R / XR + 1.0
      BR(L,L) = YR * BR(L,L)
      BI(L,L) = YR * BI(L,L)
      GO TO 28
C
27  BR(L,L) = R
      UI1 = -1.0
      UII = 0.0
C
28  DO 50 J = L1, N
      T = 0.0
      TI = 0.0
      DO 30 I = L, N
      T = T + BR(I,L) * BR(I,J) + BI(I,L) * BI(I,J)
      TI = TI + BR(I,L) * BI(I,J) - BI(I,L) * BR(I,J)
30  CONTINUE
C
      T = T / RHO
      TI = TI / RHO
C
      DO 40 I = L, N
      BR(I,J) = BR(I,J) - T * BR(I,L) + TI * BI(I,L)
      BI(I,J) = BI(I,J) - T * BI(I,L) - TI * BR(I,L)
40  CONTINUE
C
      XI = UI1 * BI(L,J) - UII * BR(L,J)
      BR(L,J) = UI1 * BR(L,J) + UII * BI(L,J)
      BI(L,J) = XI
50  CONTINUE
C
      DO 80 J = 1, N
      T = 0.0
      TI = 0.0
      DO 60 I = L, N
      T = T + BR(I,L) * AR(I,J) + BI(I,L) * AI(I,J)
      TI = TI + BR(I,L) * AI(I,J) - BI(I,L) * AR(I,J)
60  CONTINUE
C
      T = T / RHO
      TI = TI / RHO
C
      DO 70 I = L, N
      AR(I,J) = AR(I,J) - T * BR(I,L) + TI * BI(I,L)
      AI(I,J) = AI(I,J) - T * BI(I,L) + TI * AR(I,L)
70  CONTINUE
C
      BEM05510
      BEM05520
      BEM05530
      BEM05540
      BEM05550
      BEM05560
      BEM05570
      BEM05580
      BEM05590
      BEM05600
      BEM05610
      BEM05620
      BEM05630
      BEM05640
      BEM05650
      BEM05660
      BEM05670
      BEM05680
      BEM05690
      BEM05700
      BEM05710
      BEM05720
      BEM05730
      BEM05740
      BEM05750
      BEM05760
      BEM05770
      BEM05780
      BEM05790
      BEM05800
      BEM05810
      BEM05820
      BEM05830
      BEM05840
      BEM05850
      BEM05860
      BEM05870
      BEM05880
      BEM05890
      BEM05900
      BEM05910
      BEM05920
      BEM05930
      BEM05940
      BEM05950
      BEM05960
      BEM05970
      BEM05980
      BEM05990
      BEM06000
      BEM06010
      BEM06020
      BEM06030
      BEM06040
      BEM06050
```

```

      AI(I,J) = AI(I,J) - T * BI(I,L) - TI * BR(I,L)      BEM06060
70    CONTINUE                                           BEM06070
C                                           BEM06080
      XI = U1 * AI(L,J) - U1I * AR(L,J)                  BEM06090
      AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)              BEM06100
      AI(L,J) = XI                                         BEM06110
80    CONTINUE                                           BEM06120
C                                           BEM06130
      BR(L,L) = R * S                                       BEM06140
      BI(L,L) = 0.0                                         BEM06150
C                                           BEM06160
      DO 90 I = L1, N                                       BEM06170
        BR(I,L) = 0.0                                       BEM06180
        BI(I,L) = 0.0                                       BEM06190
90    CONTINUE                                           BEM06200
C                                           BEM06210
100  CONTINUE                                           BEM06220
C ***** REDUCE A TO UPPER HESSENBERG FORM WITH REAL SUBDIAGONAL BEM06230
C      ELEMENTS, WHILE KEEPING B TRIANGULAR ***** BEM06240
      DO 160 K = 1, NM1                                     BEM06250
        K1 = K + 1                                         BEM06260
C ***** SET BOTTOM ELEMENT IN K-TH COLUMN OF A REAL ***** BEM06270
      IF (AI(N,K) .EQ. 0.0) GO TO 105                       BEM06280
      R = CABS(CMPLX(AR(N,K),AI(N,K)))                     BEM06290
      U1 = AR(N,K) / R                                     BEM06300
      U1I = AI(N,K) / R                                    BEM06310
      AR(N,K) = R                                           BEM06320
      AI(N,K) = 0.0                                         BEM06330
C                                           BEM06340
      DO 103 J = K1, N                                       BEM06350
        XI = U1 * AI(N,J) - U1I * AR(N,J)                  BEM06360
        AR(N,J) = U1 * AR(N,J) + U1I * AI(N,J)              BEM06370
        AI(N,J) = XI                                         BEM06380
103  CONTINUE                                           BEM06390
C                                           BEM06400
      XI = U1 * BI(N,N) - U1I * BR(N,N)                    BEM06410
      BR(N,N) = U1 * BR(N,N) + U1I * BI(N,N)                BEM06420
      BI(N,N) = XI                                           BEM06430
105  IF (K .EQ. NM1) GO TO 170                             BEM06440
      NK1 = NM1 - K                                         BEM06450
C ***** FOR L=N-1 STEP -1 UNTIL K+1 DO -- ***** BEM06460
      DO 150 LB = 1, NK1                                     BEM06470
        L = N - LB                                         BEM06480
        L1 = L + 1                                         BEM06490
C ***** ZERO A(L+1,K) ***** BEM06500
      S = ABS(AR(L,K)) + ABS(AI(L,K)) + AR(L1,K)           BEM06510
      IF (S .EQ. 0.0) GO TO 150                             BEM06520
      U1 = AR(L,K) / S                                       BEM06530
      U1I = AI(L,K) / S                                       BEM06540
      U2 = AR(L1,K) / S                                       BEM06550
      R = SQRT(U1*U1+U1I*U1I+U2*U2)                         BEM06560
      U1 = U1 / R                                           BEM06570
      U1I = U1I / R                                          BEM06580
      U2 = U2 / R                                           BEM06590
      AR(L,K) = R * S                                       BEM06600

```

```

      AI(L,K) = 0.0
      AR(L1,K) = 0.0
C
      DO 110 J = K1, N
        XR = AR(L,J)
        XI = AI(L,J)
        YR = AR(L1,J)
        YI = AI(L1,J)
        AR(L,J) = U1 * XR + U1I * XI + U2 * YR
        AI(L,J) = U1 * XI - U1I * XR + U2 * YI
        AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110    CONTINUE
C
      XR = BR(L,L)
      BR(L,L) = U1 * XR
      BI(L,L) = -U1I * XR
      BR(L1,L) = -U2 * XR
C
      DO 120 J = L1, N
        XR = BR(L,J)
        XI = BI(L,J)
        YR = BR(L1,J)
        YI = BI(L1,J)
        BR(L,J) = U1 * XR + U1I * XI + U2 * YR
        BI(L,J) = U1 * XI - U1I * XR + U2 * YI
        BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
        BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
120    CONTINUE
C ***** ZERO B(L+1,L) *****
      S = ABS(BR(L1,L1)) + ABS(BI(L1,L1)) + ABS(BR(L1,L))
      IF (S .EQ. 0.0) GO TO 150
      U1 = BR(L1,L1) / S
      U1I = BI(L1,L1) / S
      U2 = BR(L1,L) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      BR(L1,L1) = R * S
      BI(L1,L1) = 0.0
      BR(L1,L) = 0.0
C
      DO 130 I = 1, L
        XR = BR(I,L1)
        XI = BI(I,L1)
        YR = BR(I,L)
        YI = BI(I,L)
        BR(I,L1) = U1 * XR + U1I * XI + U2 * YR
        BI(I,L1) = U1 * XI - U1I * XR + U2 * YI
        BR(I,L) = U1 * YR - U1I * YI - U2 * XR
        BI(I,L) = U1 * YI + U1I * YR - U2 * XI
130    CONTINUE
C
      DO 140 I = 1, N

```

```

BEM06610
BEM06620
BEM06630
BEM06640
BEM06650
BEM06660
BEM06670
BEM06680
BEM06690
BEM06700
BEM06710
BEM06720
BEM06730
BEM06740
BEM06750
BEM06760
BEM06770
BEM06780
BEM06790
BEM06800
BEM06810
BEM06820
BEM06830
BEM06840
BEM06850
BEM06860
BEM06870
BEM06880
BEM06890
BEM06900
BEM06910
BEM06920
BEM06930
BEM06940
BEM06950
BEM06960
BEM06970
BEM06980
BEM06990
BEM07000
BEM07010
BEM07020
BEM07030
BEM07040
BEM07050
BEM07060
BEM07070
BEM07080
BEM07090
BEM07100
BEM07110
BEM07120
BEM07130
BEM07140
BEM07150

```

BEM07160 BEM07170 BEM07180 BEM07190 BEM07200 BEM07210 BEM07220 BEM07230 BEM07240 BEM07250 BEM07260 BEM07270 BEM07280 BEM07290 BEM07300 BEM07310 BEM07320 BEM07330 BEM07340 BEM07350 BEM07360 BEM07370 BEM07380 BEM07390 BEM07400 BEM07410 BEM07420 BEM07430 BEM07440 BEM07450 BEM07460 BEM07470 BEM07480 BEM07490 BEM07500 BEM07510 BEM07520 BEM07530 BEM07540 BEM07550 BEM07560 BEM07570 BEM07580 BEM07590 BEM07600 BEM07610 BEM07620 BEM07630 BEM07640 BEM07650 BEM07660 BEM07670 BEM07680 BEM07690 BEM07700

```

C      THIS SUBROUTINE IS A COMPLEX ANALOGUE OF STEPS 2 AND 3 OF THE      BEM07710
C      QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,    BEM07720
C      SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART,        BEM07730
C      AS MODIFIED IN TECHNICAL NOTE NASA TN E-7305(1973) BY HARD.        BEM07740
C                                                                           BEM07750
C      THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX MATRICES, ONE OF THEM    BEM07760
C      IN UPPER HESSENBERG FORM AND THE OTHER IN UPPER TRIANGULAR FORM,    BEM07770
C      THE HESSENBERG MATRIX MUST FURTHER HAVE REAL SUBDIAGONAL ELEMENTS.  BEM07780
C      IT REDUCES THE HESSENBERG MATRIX TO TRIANGULAR FORM USING           BEM07790
C      UNITARY TRANSFORMATIONS WHILE MAINTAINING THE TRIANGULAR FORM       BEM07800
C      OF THE OTHER MATRIX AND FURTHER MAKING ITS DIAGONAL ELEMENTS        BEM07810
C      REAL AND NON-NEGATIVE. IT THEN RETURNS QUANTITIES WHOSE RATIOS     BEM07820
C      GIVE THE GENERALIZED EIGENVALUES. IT IS USUALLY PRECEDED BY        BEM07830
C      CQZHEs AND POSSIBLY FOLLOWED BY CQZVEC.                             BEM07840
C                                                                           BEM07850
C      ON INPUT-                                                           BEM07860
C                                                                           BEM07870
C      NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL             BEM07880
C      ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM                 BEM07890
C      DIMENSION STATEMENT,                                               BEM07900
C                                                                           BEM07910
C      N IS THE ORDER OF THE MATRICES,                                    BEM07920
C                                                                           BEM07930
C      A=(AR,AI) CONTAINS A COMPLEX UPPER HESSENBERG MATRIX               BEM07940
C      WITH REAL SUBDIAGONAL ELEMENTS,                                    BEM07950
C                                                                           BEM07960
C      B=(BR,BI) CONTAINS A COMPLEX UPPER TRIANGULAR MATRIX,             BEM07970
C                                                                           BEM07980
C      EPS1 IS A TOLERANCE USED TO DETERMINE NEGLIGIBLE ELEMENTS.        BEM07990
C      EPS1 = 0.0 (OR NEGATIVE) MAY BE INPUT, IN WHICH CASE AN            BEM08000
C      ELEMENT WILL BE NEGLECTED ONLY IF IT IS LESS THAN ROUNDOFF         BEM08010
C      ERROR TIMES THE NORM OF ITS MATRIX. IF THE INPUT EPS1 IS           BEM08020
C      POSITIVE, THEN AN ELEMENT WILL BE CONSIDERED NEGLIGIBLE            BEM08030
C      IF IT IS LESS THAN EPS1 TIMES THE NORM OF ITS MATRIX. A            BEM08040
C      POSITIVE VALUE OF EPS1 MAY RESULT IN FASTER EXECUTION,             BEM08050
C      BUT LESS ACCURATE RESULTS,                                          BEM08060
C                                                                           BEM08070
C      MATZ SHOULD BE SET TO .TRUE. IF THE RIGHT HAND TRANSFORMATIONS     BEM08080
C      ARE TO BE ACCUMULATED FOR LATER USE IN COMPUTING                   BEM08090
C      EIGENVECTORS, AND TO .FALSE. OTHERWISE,                            BEM08100
C                                                                           BEM08110
C      Z=(ZR,ZI) CONTAINS, IF MATZ HAS BEEN SET TO .TRUE., THE            BEM08120
C      TRANSFORMATION MATRIX PRODUCED IN THE REDUCTION                    BEM08130
C      BY CQZHEs, IF PERFORMED, OR ELSE THE IDENTITY MATRIX.             BEM08140
C      IF MATZ HAS BEEN SET TO .FALSE., Z IS NOT REFERENCED.              BEM08150
C                                                                           BEM08160
C      ON OUTPUT                                                           BEM08170
C                                                                           BEM08180
C      A HAS BEEN REDUCED TO UPPER TRIANGULAR FORM. THE ELEMENTS          BEM08190
C      BELOW THE MAIN DIAGONAL HAVE BEEN SET TO ZERO,                     BEM08200
C                                                                           BEM08210
C      B IS STILL IN UPPER TRIANGULAR FORM, ALTHOUGH ITS ELEMENTS         BEM08220
C      HAVE BEEN ALTERED. IN PARTICULAR, ITS DIAGONAL HAS BEEN SET      BEM08230
C      REAL AND NON-NEGATIVE. THE LOCATION BRIN,1) IS USED TO             BEM08240
C      STORE EPS1 TIMES THE NORM OF B FOR LATER USE BY CQZVEC,            BEM08250

```

```

C
C      ALFR AND ALFI CONTAIN THE REAL AND IMAGINARY PARTS OF THE
C      DIAGONAL ELEMENTS OF THE TRIANGULARIZED A MATRIX,
C
C      BETA CONTAINS THE REAL NON-NEGATIVE DIAGONAL ELEMENTS OF THE
C      CORRESPONDING B. THE GENERALIZED EIGENVALUES ARE THEN
C      THE RATIOS ((ALFR+I*ALFI)/BETA),
C
C      Z CONTAINS THE PRODUCT OF THE RIGHT HAND TRANSFORMATIONS
C      (FOR BOTH STEPS) IF MATZ HAS BEEN SET TO .TRUE.,
C
C      IERR IS SET TO
C      ZERO      FOR NORMAL RETURN,
C      J          IF AR(J,J-1) HAS NOT BECOME
C                ZERO AFTER 50 ITERATIONS.
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARROW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C      -----
C
C      IERR = 0
C      ***** COMPUTE EPSA,EPSB *****
C      ANORM = 0.0
C      BNORM = 0.0
C
C      DO 30 I = 1, N
C        ANI = 0.0
C        IF (I .NE. 1) ANI = ABS(AR(I,I-1))
C        BNI = 0.0
C
C        DO 20 J = I, N
C          ANI = ANI + ABS(AR(I,J)) + ABS(AI(I,J))
C          BNI = BNI + ABS(BR(I,J)) + ABS(BI(I,J))
C        20 CONTINUE
C
C        IF (ANI .GT. ANORM) ANORM = ANI
C        IF (BNI .GT. BNORM) BNORM = BNI
C      30 CONTINUE
C
C      IF (ANORM .EQ. 0.0) ANORM = 1.0
C      IF (BNORM .EQ. 0.0) BNORM = 1.0
C      EP = EPS1
C      IF (EP .GT. 0.0) GO TO 50
C      ***** COMPUTE ROUNDOFF LEVEL IF EPS1 IS ZERO *****
C      EP = 1.0
C      40 EP = EP / 2.0
C      IF (1.0 + EP .GT. 1.0) GO TO 40
C      50 EPSA = EP * ANORM
C      EPSB = EP * BNORM
C      ***** REDUCE A TO TRIANGULAR FORM, WHILE
C      KEEPING B TRIANGULAR *****
C
C      LOR1 = 1
C      ENORN = N
C      EN = N

```

```

BEM08260
BEM08270
BEM08280
BEM08290
BEM08300
BEM08310
BEM08320
BEM08330
BEM08340
BEM08350
BEM08360
BEM08370
BEM08380
BEM08390
BEM08400
BEM08410
BEM08420
BEM08430
BEM08440
BEM08450
BEM08460
BEM08470
BEM08480
BEM08490
BEM08500
BEM08510
BEM08520
BEM08530
BEM08540
BEM08550
BEM08560
BEM08570
BEM08580
BEM08590
BEM08600
BEM08610
BEM08620
BEM08630
BEM08640
BEM08650
BEM08660
BEM08670
BEM08680
BEM08690
BEM08700
BEM08710
BEM08720
BEM08730
BEM08740
BEM08750
BEM08760
BEM08770
BEM08780
BEM08790
BEM08800

```

```

C ***** BEGIN QZ STEP *****
60 IF (EN .EQ. 0) GO TO 1001
   IF (.NOT. MATZ) ENORN = EN
   ITS = 0
   NA = EN - 1
   ENM2 = NA - 1
C ***** CHECK FOR CONVERGENCE OR REDUCIBILITY.
C      FOR L=EN STEP -1 UNTIL 1 DO -- *****
70 DO 80 LL = 1, EN
   LM1 = EN - LL
   L = LM1 + 1
   IF (L .EQ. 1) GO TO 95
   IF (ABS(AR(L,LM1)) .LE. EPSA) GO TO 90
80 CONTINUE
C
90 AR(L,LM1) = 0.0
C ***** SET DIAGONAL ELEMENT AT TOP OF B REAL *****
95 B11 = CABS(CMPLX(BR(L,L),BI(L,L)))
   IF (B11 .EQ. 0.0) GO TO 98
   U1 = BR(L,L) / B11
   U1I = BI(L,L) / B11
C
   DO 97 J = L, ENORN
      XI = U1 * AI(L,J) - U1I * AR(L,J)
      AR(L,J) = U1 * AR(L,J) + U1I * AI(L,J)
      AI(L,J) = XI
      XI = U1 * BI(L,J) - U1I * BR(L,J)
      BR(L,J) = U1 * BR(L,J) + U1I * BI(L,J)
      BI(L,J) = XI
97 CONTINUE
C
   BI(L,L) = 0.0
98 IF (L .NE. EN) GO TO 100
C ***** 1-BY-1 BLOCK ISOLATED *****
   ALFR(EN) = AR(EN,EN)
   ALFI(EN) = AI(EN,EN)
   BETA(EN) = B11
   EN = NA
   GO TO 60
C ***** CHECK FOR SMALL TOP OF B *****
100 L1 = L + 1
   IF (B11 .GT. EPSB) GO TO 120
   BR(L,L) = 0.0
   S = ABS(AR(L,L)) + ABS(AI(L,L)) + ABS(AR(L1,L))
   U1 = AR(L,L) / S
   U1I = AI(L,L) / S
   U2 = AR(L1,L) / S
   R = SQRT(U1*U1+U1I*U1I+U2*U2)
   U1 = U1 / R
   U1I = U1I / R
   U2 = U2 / R
   AR(L,L) = R * S
   AI(L,L) = 0.0
C
   DO 110 J = L1, ENORN

```

```

BEM08810
BEM08820
BEM08830
BEM08840
BEM08850
BEM08860
BEM08870
BEM08880
BEM08890
BEM08900
BEM08910
BEM08920
BEM08930
BEM08940
BEM08950
BEM08960
BEM08970
BEM08980
BEM08990
BEM09000
BEM09010
BEM09020
BEM09030
BEM09040
BEM09050
BEM09060
BEM09070
BEM09080
BEM09090
BEM09100
BEM09110
BEM09120
BEM09130
BEM09140
BEM09150
BEM09160
BEM09170
BEM09180
BEM09190
BEM09200
BEM09210
BEM09220
BEM09230
BEM09240
BEM09250
BEM09260
BEM09270
BEM09280
BEM09290
BEM09300
BEM09310
BEM09320
BEM09330
BEM09340
BEM09350

```

```

      XR = AR(L,J)
      XI = AI(L,J)
      YR = AR(L1,J)
      YI = AI(L1,J)
      AR(L,J) = U1 * XR + U1I * XI + U2 * YR
      AI(L,J) = U1 * XI - U1I * XR + U2 * YI
      AR(L1,J) = U1 * YR - U1I * YI - U2 * XR
      AI(L1,J) = U1 * YI + U1I * YR - U2 * XI
      XR = BR(L,J)
      XI = BI(L,J)
      YR = BR(L1,J)
      YI = BI(L1,J)
      BR(L1,J) = U1 * YR - U1I * YI - U2 * XR
      BR(L,J) = U1 * XR + U1I * XI + U2 * YR
      BI(L,J) = U1 * XI - U1I * XR + U2 * YI
      BI(L1,J) = U1 * YI + U1I * YR - U2 * XI
110 CONTINUE
C
      LM1 = L
      L = L1
      GO TO 90
C ***** ITERATION STRATEGY *****
120 IF (ITS .EQ. 50) GO TO 1000
      IF (ITS .EQ. 10) GO TO 135
C ***** DETERMINE SHIFT *****
      B33 = BR(NA,NA)
      B33I = BI(NA,NA)
      IF (CABS(CMPLX(B33,B33I)) .GE. EPSB) GO TO 122
      B33 = EPSB
      B33I = 0.0
122 B44 = BR(EN,EN)
      B44I = BI(EN,EN)
      IF (CABS(CMPLX(B44,B44I)) .GE. EPSB) GO TO 124
      B44 = EPSB
      B44I = 0.0
124 B3344 = B33 * B44 - B33I * B44I
      B3344I = B33 * B44I + B33I * B44
      A33 = AR(NA,NA) * B44 - AI(NA,NA) * B44I
      A33I = AR(NA,NA) * B44I + AI(NA,NA) * B44
      A34 = AR(NA,EN) * B33 - AI(NA,EN) * B33I
      X - AR(NA,NA) * BR(NA,EN) + AI(NA,NA) * BI(NA,EN)
      A34I = AR(NA,EN) * B33I + AI(NA,EN) * B33
      X - AR(NA,NA) * BI(NA,EN) - AI(NA,NA) * BR(NA,EN)
      A43 = AR(EN,NA) * B44
      A43I = AR(EN,NA) * B44I
      A44 = AR(EN,EN) * B33 - AI(EN,EN) * B33I - AR(EN,NA) * BR(NA,EN)
      A44I = AR(EN,EN) * B33I + AI(EN,EN) * B33 - AR(EN,NA) * BI(NA,EN)
      SH = A44
      SHI = A44I
      XR = A34 * A43 - A34I * A43I
      XI = A34 * A43I + A34I * A43
      IF (XR .EQ. 0.0 .AND. XI .EQ. 0.0) GO TO 140
      YR = (A33 - SH) / 2.0
      YI = (A33I - SHI) / 2.0
      Z3 = CSQRT(CMPLX(YR**2-YI**2+XR,2.0*YR*YI+XI))

```

```

BEM09360
BEM09370
BEM09380
BEM09390
BEM09400
BEM09410
BEM09420
BEM09430
BEM09440
BEM09450
BEM09460
BEM09470
BEM09480
BEM09490
BEM09500
BEM09510
BEM09520
BEM09530
BEM09540
BEM09550
BEM09560
BEM09570
BEM09580
BEM09590
BEM09600
BEM09610
BEM09620
BEM09630
BEM09640
BEM09650
BEM09660
BEM09670
BEM09680
BEM09690
BEM09700
BEM09710
BEM09720
BEM09730
BEM09740
BEM09750
BEM09760
BEM09770
BEM09780
BEM09790
BEM09800
BEM09810
BEM09820
BEM09830
BEM09840
BEM09850
BEM09860
BEM09870
BEM09880
BEM09890
BEM09900

```



```

      U1 = REAL(Z3)
      U1I = AIMAG(Z3)
      IF (YR * U1 + YI * U1I .GE. 0.0) GO TO 125
      U1 = -U1
      U1I = -U1I
125  Z3 = (CMPLX(SH,SHI) - CMPLX(XR,XI) / CMPLX(YR+U1,YI+U1I))
      X / CMPLX(B3344,B3344I)
      SH = REAL(Z3)
      SHI = AIMAG(Z3)
      GO TO 140
C ***** AD HOC SHIFT *****
135  SH = AR(EN,NA) + AR(NA,ENM2)
      SHI = 0.0
C ***** DETERMINE ZEROth COLUMN OF A *****
140  A1 = AR(L,L) / B11 - SH
      A1I = AI(L,L) / B11 - SHI
      A2 = AR(L1,L) / B11
      ITS = ITS + 1
      IF (.NOT. MATZ) LOR1 = L
C ***** MAIN LOOP *****
      DO 260 K = L, NA
        K1 = K + 1
        K2 = K + 2
        KM1 = MAX0(K-1,L)
C ***** ZERO A(K+1,K-1) *****
        IF (K .EQ. L) GO TO 170
        A1 = AR(K,KM1)
        A1I = AI(K,KM1)
        A2 = AR(K1,KM1)
170  S = ABS(A1) + ABS(A1I) + ABS(A2)
        U1 = A1 / S
        U1I = A1I / S
        U2 = A2 / S
        R = SQRT(U1*U1+U1I*U1I+U2*U2)
        U1 = U1 / R
        U1I = U1I / R
        U2 = U2 / R
C
      DO 180 J = KM1, ENORN
        XR = AR(K,J)
        XI = AI(K,J)
        YR = AR(K1,J)
        YI = AI(K1,J)
        AR(K,J) = U1 * XR + U1I * XI + U2 * YR
        AI(K,J) = U1 * XI - U1I * XR + U2 * YI
        AR(K1,J) = U1 * YR - U1I * YI - U2 * XR
        AI(K1,J) = U1 * YI + U1I * YR - U2 * XI
        XR = BR(K,J)
        XI = BI(K,J)
        YR = BR(K1,J)
        YI = BI(K1,J)
        BR(K,J) = U1 * XR + U1I * XI + U2 * YR
        BI(K,J) = U1 * XI - U1I * XR + U2 * YI
        BR(K1,J) = U1 * YR - U1I * YI - U2 * XR
        BI(K1,J) = U1 * YI + U1I * YR - U2 * XI

```

```

BEM09910
BEM09920
BEM09930
BEM09940
BEM09950
BEM09960
BEM09970
BEM09980
BEM09990
BEM10000
BEM10010
BEM10020
BEM10030
BEM10040
BEM10050
BEM10060
BEM10070
BEM10080
BEM10090
BEM10100
BEM10110
BEM10120
BEM10130
BEM10140
BEM10150
BEM10160
BEM10170
BEM10180
BEM10190
BEM10200
BEM10210
BEM10220
BEM10230
BEM10240
BEM10250
BEM10260
BEM10270
BEM10280
BEM10290
BEM10300
BEM10310
BEM10320
BEM10330
BEM10340
BEM10350
BEM10360
BEM10370
BEM10380
BEM10390
BEM10400
BEM10410
BEM10420
BEM10430
BEM10440
BEM10450

```

```

180    CONTINUE
C
      IF (K .EQ. L) GO TO 240
      AI(K,KM1) = 0.0
      AR(K1,KM1) = 0.0
      AI(K1,KM1) = 0.0
C ***** ZERO B(K+1,K) *****
240    S = ABS(BR(K1,K1)) + ABS(BI(K1,K1)) + ABS(BR(K1,K))
      U1 = BR(K1,K1) / S
      U1I = BI(K1,K1) / S
      U2 = BR(K1,K) / S
      R = SQRT(U1*U1+U1I*U1I+U2*U2)
      U1 = U1 / R
      U1I = U1I / R
      U2 = U2 / R
      IF (K .EQ. NA) GO TO 245
      XR = AR(K2,K1)
      AR(K2,K1) = U1 * XR
      AI(K2,K1) = -U1I * XR
      AR(K2,K) = -U2 * XR
C
245    DO 250 I = LOR1, K1
      XR = AR(I,K1)
      XI = AI(I,K1)
      YR = AR(I,K)
      YI = AI(I,K)
      AR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      AI(I,K1) = U1 * XI - U1I * XR + U2 * YI
      AR(I,K) = U1 * YR - U1I * YI - U2 * XR
      AI(I,K) = U1 * YI + U1I * YR - U2 * XI
      XR = BR(I,K1)
      XI = BI(I,K1)
      YR = BR(I,K)
      YI = BI(I,K)
      BR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      BI(I,K1) = U1 * XI - U1I * XR + U2 * YI
      BR(I,K) = U1 * YR - U1I * YI - U2 * XR
      BI(I,K) = U1 * YI + U1I * YR - U2 * XI
C
250    CONTINUE
C
      BI(K1,K1) = 0.0
      BR(K1,K) = 0.0
      BI(K1,K) = 0.0
      IF (.NOT. MATZ) GO TO 260
C
      DO 255 I = 1, N
      XR = ZR(I,K1)
      XI = ZI(I,K1)
      YR = ZR(I,K)
      YI = ZI(I,K)
      ZR(I,K1) = U1 * XR + U1I * XI + U2 * YR
      ZI(I,K1) = U1 * XI - U1I * XR + U2 * YI
      ZR(I,K) = U1 * YR - U1I * YI - U2 * XR
      ZI(I,K) = U1 * YI + U1I * YR - U2 * XI
255    CONTINUE

```

```

BEM10460
BEM10470
BEM10480
BEM10490
BEM10500
BEM10510
BEM10520
BEM10530
BEM10540
BEM10550
BEM10560
BEM10570
BEM10580
BEM10590
BEM10600
BEM10610
BEM10620
BEM10630
BEM10640
BEM10650
BEM10660
BEM10670
BEM10680
BEM10690
BEM10700
BEM10710
BEM10720
BEM10730
BEM10740
BEM10750
BEM10760
BEM10770
BEM10780
BEM10790
BEM10800
BEM10810
BEM10820
BEM10830
BEM10840
BEM10850
BEM10860
BEM10870
BEM10880
BEM10890
BEM10900
BEM10910
BEM10920
BEM10930
BEM10940
BEM10950
BEM10960
BEM10970
BEM10980
BEM10990
BEM11000

```

```

C
260 CONTINUE
C ***** SET LAST A SUBDIAGONAL REAL AND END QZ STEP *****
IF (AI(EN,NA) .EQ. 0.0) GO TO 70
R = CABS(CMPLX(AR(EN,NA),AI(EN,NA)))
U1 = AR(EN,NA) / R
U1I = AI(EN,NA) / R
AR(EN,NA) = R
AI(EN,NA) = 0.0
C
DO 270 J = EN, ENORN
XI = U1 * AI(EN,J) - U1I * AR(EN,J)
AR(EN,J) = U1 * AR(EN,J) + U1I * AI(EN,J)
AI(EN,J) = XI
XI = U1 * BI(EN,J) - U1I * BR(EN,J)
BR(EN,J) = U1 * BR(EN,J) + U1I * BI(EN,J)
BI(EN,J) = XI
270 CONTINUE
C
GO TO 70
C ***** SET ERROR -- BOTTOM SUBDIAGONAL ELEMENT HAS NOT
C BECOME NEGLIGIBLE AFTER 50 ITERATIONS *****
1000 IERR = EN
C ***** SAVE EPSB FOR USE BY CQZVEC *****
1001 IF (N .GT. 1) BR(N,1) = EPSB
RETURN
C ***** LAST CARD OF CQZVAL *****
END
C
C -----
C
SUBROUTINE CQZVEC(NM,N,AR,AI,BR,BI,ALFR,ALFI,BETA,ZR,ZI)
C *****
C
INTEGER I,J,K,M,N,EN,II,JJ,NA,NM,NN
REAL AR(NM,N),AI(NM,N),BR(NM,N),BI(NM,N),ALFR(N),ALFI(N),
X BETA(N),ZR(NM,N),ZI(NM,N)
REAL R,T,RI,TI,XI,ALMI,ALMR,BETM,EPSB
REAL CABS
COMPLEX Z3
COMPLEX CMPLX
REAL REAL,AIMAG
C
C
C
C
C THIS SUBROUTINE IS A COMPLEX ANALOGUE OF THE FOURTH STEP OF THE
C QZ ALGORITHM FOR SOLVING GENERALIZED MATRIX EIGENVALUE PROBLEMS,
C SIAM J. NUMER. ANAL. 10, 241-256(1973) BY MOLER AND STEWART.
C
C THIS SUBROUTINE ACCEPTS A PAIR OF COMPLEX MATRICES IN UPPER
C TRIANGULAR FORM, WHERE ONE OF THEM FURTHER MUST HAVE REAL DIAGONAL
C ELEMENTS. IT COMPUTES THE EIGENVECTORS OF THE TRIANGULAR PROBLEM
C AND TRANSFORMS THE RESULTS BACK TO THE ORIGINAL COORDINATE SYSTEM.

```

```

BEM11010
BEM11020
BEM11030
BEM11040
BEM11050
BEM11060
BEM11070
BEM11080
BEM11090
BEM11100
BEM11110
BEM11120
BEM11130
BEM11140
BEM11150
BEM11160
BEM11170
BEM11180
BEM11190
BEM11200
BEM11210
BEM11220
BEM11230
BEM11240
BEM11250
BEM11260
BEM11270
BEM11280
BEM11290
BEM11300
BEM11310
BEM11320
BEM11330
BEM11340
BEM11350
BEM11360
BEM11370
BEM11380
BEM11390
BEM11400
BEM11410
BEM11420
BEM11430
BEM11440
BEM11450
BEM11460
BEM11470
BEM11480
BEM11490
BEM11500
BEM11510
BEM11520
BEM11530
BEM11540
BEM11550

```

- 53 -

```

      M = I + 1
C      DO 610 J = M, EN
      T = BETM * AR(I,J) - ALMR * BR(I,J) + ALMI * BI(I,J)
      TI = BETM * AI(I,J) - ALMR * BI(I,J) - ALMI * BR(I,J)
      IF (J .EQ. EN) GO TO 605
      XI = T * BI(J,EN) + TI * BR(J,EN)
      T = T * BR(J,EN) - TI * BI(J,EN)
      TI = XI
605    R = R + T
      RI = RI + TI
610    CONTINUE
C      T = ALMR * BETA(I) - BETM * ALFR(I)
      TI = ALMI * BETA(I) - BETM * ALFI(I)
      IF (T .EQ. 0.0 .AND. TI .EQ. 0.0) T = EPSB
      Z3 = CMPLX(R,RI) / CMPLX(T,TI)
      BR(I,EN) = REAL(Z3)
      BI(I,EN) = AIMAG(Z3)
700    CONTINUE
C      800 CONTINUE
C      ***** END BACK SUBSTITUTION.
C      TRANSFORM TO ORIGINAL COORDINATE SYSTEM.
C      FOR J=N STEP -1 UNTIL 2 DO -- *****
      DO 880 JJ = 2, N
      J = N + 2 - JJ
      M = J - 1
C      DO 880 I = 1, N
C      DO 860 K = 1, M
      ZR(I,J) = ZR(I,J) + ZR(I,K) * BR(K,J) - ZI(I,K) * BI(K,J)
      ZI(I,J) = ZI(I,J) + ZR(I,K) * BI(K,J) + ZI(I,K) * BR(K,J)
860    CONTINUE
C      880 CONTINUE
C      ***** NORMALIZE SO THAT MODULUS OF LARGEST
C      COMPONENT OF EACH VECTOR IS 1 *****
      DO 950 J = 1, N
      T = 0.0
C      DO 930 I = 1, N
      R = CABS(CMPLX(ZR(I,J),ZI(I,J)))
      IF (R .GT. T) T = R
930    CONTINUE
C      DO 940 I = 1, N
      ZR(I,J) = ZR(I,J) / T
      ZI(I,J) = ZI(I,J) / T
940    CONTINUE
C      950 CONTINUE
C      1001 RETURN

```

BEM12110
 BEM12120
 BEM12130
 BEM12140
 BEM12150
 BEM12160
 BEM12170
 BEM12180
 BEM12190
 BEM12200
 BEM12210
 BEM12220
 BEM12230
 BEM12240
 BEM12250
 BEM12260
 BEM12270
 BEM12280
 BEM12290
 BEM12300
 BEM12310
 BEM12320
 BEM12330
 BEM12340
 BEM12350
 BEM12360
 BEM12370
 BEM12380
 BEM12390
 BEM12400
 BEM12410
 BEM12420
 BEM12430
 BEM12440
 BEM12450
 BEM12460
 BEM12470
 BEM12480
 BEM12490
 BEM12500
 BEM12510
 BEM12520
 BEM12530
 BEM12540
 BEM12550
 BEM12560
 BEM12570
 BEM12580
 BEM12590
 BEM12600
 BEM12610
 BEM12620
 BEM12630
 BEM12640
 BEM12650

```
C ***** LAST CARD OF CQZVEC ***** BEM12660
END BEM12670
C ----- BEM12680
C BEM12690
BEM12700
C BEM12710
SUBROUTINE NORMA(FI,G,N,NM,XM) BEM12720
C ***** BEM12730
C BEM12740
COMPLEX FI(NM,NM),XM(NM,NM),TEMP(20),SUM1,SUM2 BEM12750
COMPLEX XI(20,20),X(20,20),X1,X2 BEM12760
REAL G(NM,NM) BEM12770
DATA IWRITE/4/ BEM12780
C BEM12790
DO 30 I=1,N BEM12800
DO 40 J=1,N BEM12810
SUM1=(0.0,0.0) BEM12820
DO 50 K=1,N BEM12830
SUM1=SUM1+CMPLX(G(J,K))*FI(K,I) BEM12840
50 CONTINUE BEM12850
TEMP(J)=SUM1 BEM12860
40 CONTINUE BEM12870
SUM2=(0.0,0.0) BEM12880
DO 41 J=1,N BEM12890
SUM2=SUM2+FI(J,I)*TEMP(J) BEM12900
41 CONTINUE BEM12910
SUM2=CSQRT(SUM2) BEM12920
DO 51 J=1,N BEM12930
FI(J,I)=FI(J,I)/SUM2 BEM12940
51 CONTINUE BEM12950
30 CONTINUE BEM12960
C BEM12970
C FOR CHECK CARRY OUT [FI]*[G]*[FI]=[I] BEM12980
C BEM12990
DO 100 I=1,N BEM13000
DO 110 J=1,N BEM13010
X1=(0.0,0.0) BEM13020
DO 120 K=1,N BEM13030
X1=X1+FI(K,I)*CMPLX(G(K,J)) BEM13040
120 CONTINUE BEM13050
X(I,J)=X1 BEM13060
110 CONTINUE BEM13070
100 CONTINUE BEM13080
DO 130 I=1,N BEM13090
DO 140 J=1,N BEM13100
X2=(0.0,0.0) BEM13110
DO 150 K=1,N BEM13120
X2=X2+X(I,K)*FI(K,J) BEM13130
150 CONTINUE BEM13140
XI(I,J)=X2 BEM13150
140 CONTINUE BEM13160
130 CONTINUE BEM13170
DO 161 I=1,N BEM13180
WRITE(IWRITE,*) (XI(I,J),J=1,N) BEM13190
161 CONTINUE BEM13200
```

```
      RETURN                                BEM13210
      END                                  BEM13220
C -----BEM13230
C -----BEM13240
C -----BEM13250
      SUBROUTINE SET(DY,T1,T2,N,NM,H,BO,G,BI) BEM13260
C *****BEM13270
C SET CALCULATES MATRIX [A]p=[H]+i[BO] BEM13280
C AND MATRIX [B]p=[G] BEM13290
      REAL H(NM,NM),BO(NM,NM),G(NM,NM) BEM13300
      REAL BI(NM,NM) BEM13310
      DO 10 I=1,N BEM13320
      I1=I-1 BEM13330
      I2=I+1 BEM13340
      DO 20 J=1,N BEM13350
      IF(I.EQ.1.AND.J.EQ.1) THEN BEM13360
      H(I,J)=1.0 BEM13370
      G(I,J)=1.0 BEM13380
      ELSE IF (I.NE.1.AND.I.EQ.J) THEN BEM13390
      H(I,J)=2.0 BEM13400
      G(I,J)=2.0 BEM13410
      ELSE IF (J.EQ.I1.OR.J.EQ.I2) THEN BEM13420
      H(I,J)=-1.0 BEM13430
      G(I,J)=0.5 BEM13440
      ELSE BEM13450
      H(I,J)=0.0 BEM13460
      G(I,J)=0.0 BEM13470
      END IF BEM13480
      H(I,J)=H(I,J)/DY BEM13490
      G(I,J)=G(I,J)*(DY/3) BEM13500
20 CONTINUE BEM13510
10 CONTINUE BEM13520
      DO 30 I=1,N BEM13530
      DO 40 J=1,N BEM13540
      IF(I.EQ.1.AND.J.EQ.1) THEN BEM13550
      BO(I,J)=T2*T1 BEM13560
      ELSE BEM13570
      BO(I,J)=0.0 BEM13580
      END IF BEM13590
40 CONTINUE BEM13600
30 CONTINUE BEM13610
      DO 80 I=1,N BEM13620
      DO 90 J=1,N BEM13630
      BI(I,J)=0.0 BEM13640
90 CONTINUE BEM13650
80 CONTINUE BEM13660
      RETURN BEM13670
      END BEM13680
C -----BEM13690
C -----BEM13700
C -----BEM13710
C -----BEM13720
      SUBROUTINE SETR(DY,ME,ND,A,F) BEM13730
C *****BEM13740
C *****BEM13750
```

```

      REAL A(ND,ND),F(ND,ND)
      DO 9 I=1,ME
      DO 8 J=1,ME
      A(I,J)=0.
      F(I,J)=0.
8    CONTINUE
9    CONTINUE
      A(1,1)=1/DY
      A(1,2)=-1./DY
      F(1,1)=DY/3
      F(1,2)=.5*DY/3
      DO 10 I=2,ME-1
      A(I,I)=2/DY
      F(I,I)=2*DY/3
      A(I,I-1)=-1/DY
      A(I,I+1)=-1/DY
      F(I,I-1)=.5*DY/3
      F(I,I+1)=.5*DY/3
10   CONTINUE
      A(ME,ME)=2/DY
      A(ME,ME-1)=-1/DY
      F(ME,ME)=2*DY/3
      F(ME,ME-1)=.5*DY/3
      RETURN
      END
C
C -----
C
C SUBROUTINE FMAT(X,Y,XM,YM,G,H,HOLD,FID,FIB,DFI,NO,N1)
C *****
C
C FMAT COMPUTES THE G AND H MATRICES AND FORMS
C THE SYSTEMS OF EQUATIONS AX=F
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
C COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C DOUBLE PRECISION X(1),Y(1),G(NU,NU),H(NU,NU),
1 FID(1),FIB(1),DFI(1),HOLD(NU,NU),XM(1),YM(1)
C
C NX=2*N
C DO 10 I=1,NX
C DFI(I)=0
C DO 11 J=1,NX
C G(I,J)=0.0
C H(I,J)=0.0
11  CONTINUE
10  CONTINUE
C
C X(N+1)=X(1)
C Y(N+1)=Y(1)
C
C DO 12 I=1,N
C XM(I)=(X(I)+X(I+1))/2
C YM(I)=(Y(I)+Y(I+1))/2

```

```

BEM13760
BEM13770
BEM13780
BEM13790
BEM13800
BEM13810
BEM13820
BEM13830
BEM13840
BEM13850
BEM13860
BEM13870
BEM13880
BEM13890
BEM13900
BEM13910
BEM13920
BEM13930
BEM13940
BEM13950
BEM13960
BEM13970
BEM13980
BEM13990
BEM14000
BEM14010
BEM14020
BEM14030
BEM14040
BEM14050
BEM14060
BEM14070
BEM14080
BEM14090
BEM14100
BEM14110
BEM14120
BEM14130
BEM14140
BEM14150
BEM14160
BEM14170
BEM14180
BEM14190
BEM14200
BEM14210
BEM14220
BEM14230
BEM14240
BEM14250
BEM14260
BEM14270
BEM14280
BEM14290
BEM14300

```



```

12 CONTINUE
C
DO 20 I=1,N
DO 30 J=1,N
IF(I-J)40,45,40
40 CALL INTE(XM(I),YM(I),X(J),Y(J),X(J+1),Y(J+1),A,B)
H(I,J)=A
G(I,J)=B
GO TO 30
45 CALL INLO(X(J),Y(J),X(J+1),Y(J+1),B)
G(I,J)=B
H(I,J)=-0.5
30 CONTINUE
20 CONTINUE
C
C ARRANGE THE SYSTEM OF EQUATIONS
C
DO 150 J=1,N0
DO 145 I=1,N
CH=G(I,J)
G(I,J)=-H(I,J)
H(I,J)=-CH
145 CONTINUE
150 CONTINUE
C
DO 160 I=1,N
DO 170 J=1,N
HOLD(I,J)=H(I,J)
170 CONTINUE
160 CONTINUE
C
RETURN
END
C
C -----
C
SUBROUTINE INTE(XP,YP,X1,Y1,X2,Y2,H,G)
*****
C
THIS SUBROUTINE COMPUTES THE VALUES OF THE H AND GMATRIX
C OFF DIAGONAL ELEMENTS BY MEANS OF NUMERICAL INTEGRATION
C ALONG AN ELEMENT
C USING 4-POINT GAUSS QUADRATURE
C RA=DISTANCE FROM THE POINT UNDER CONSIDERATION
C TO THE INTEGRATION POINT ON THE BOUNDARY ELEMENT
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
DOUBLE PRECISION XCO(4),YCO(4),GI(4),OME(4),
&LEN,K
K=CONST
GI(1)=0.86113631
GI(2)=-GI(1)
GI(3)=0.33998104

```

```

BEM14310
BEM14320
BEM14330
BEM14340
BEM14350
BEM14360
BEM14370
BEM14380
BEM14390
BEM14400
BEM14410
BEM14420
BEM14430
BEM14440
BEM14450
BEM14460
BEM14470
BEM14480
BEM14490
BEM14500
BEM14510
BEM14520
BEM14530
BEM14540
BEM14550
BEM14560
BEM14570
BEM14580
BEM14590
BEM14600
BEM14610
BEM14620
BEM14630
BEM14640
BEM14650
BEM14660
BEM14670
BEM14680
BEM14690
BEM14700
BEM14710
BEM14720
BEM14730
BEM14740
BEM14750
BEM14760
BEM14770
BEM14780
BEM14790
BEM14800
BEM14810
BEM14820
BEM14830
BEM14840
BEM14850

```

```

      GI(4)=-GI(3)
      OME(1)=0.34785485
      OME(2)=OME(1)
      OME(3)=0.65214515
      OME(4)=OME(3)
      AX=(X2-X1)/2
      BX=(X2+X1)/2
      AY=(Y2-Y1)/2
      BY=(Y2+Y1)/2
      IF(AX.EQ.0)THEN
        DIST=DABS(XP-X1)
      ELSE
        TA=AY/AX
        DIST=DABS((TA*XP-YP+Y1-TA*X1)/DSQRT(TA**2+1))
      END IF
      SIG=(X1-XP)*(Y2-YP)-(X2-XP)*(Y1-YP)
      IF(SIG.LT.0) DIST=-DIST
      H=0
      G=0
      LEN=DSQRT(AX*AX+AY*AY)
      DO 40 I=1,4
        XCO(I)=AX*GI(I)+BX
        YCO(I)=AY*GI(I)+BY
        RA=DSQRT((XP-XCO(I))**2+(YP-YCO(I))**2)
        CON=K*RA
        CALL BFUNCT(CON,YOX,Y1X)
        G=G+YOX*OME(I)*LEN*0.25
40    H=H+Y1X*OME(I)*DIST*K*(0.25/RA)*LEN
      RETURN
      END
C
C -----
C
      SUBROUTINE INLO(X1,Y1,X2,Y2,G)
      *****
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/MAIN/CONST,N,LEC,IMP,NU,ND
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
C
C
C
      THIS SUBROUTINE COMPUTES THE VALUES OF THE DIAGONAL
      ELEMENTS OF THE G MATRIX
C
      AX=(X2-X1)/2
      AY=(Y2-Y1)/2
      SR=DSQRT(AX**2+AY**2)
      XXX=SR*CONST
      G=(SR/3.1415927)*(DLOG(XXX)-1.0)
      RETURN
      END
C
C -----
C
      SUBROUTINE BFUNCT(XX,YOX,Y1X)

```

```

BEM14860
BEM14870
BEM14880
BEM14890
BEM14900
BEM14910
BEM14920
BEM14930
BEM14940
BEM14950
BEM14960
BEM14970
BEM14980
BEM14990
BEM15000
BEM15010
BEM15020
BEM15030
BEM15040
BEM15050
BEM15060
BEM15070
BEM15080
BEM15090
BEM15100
BEM15110
BEM15120
BEM15130
BEM15140
BEM15150
BEM15160
BEM15170
BEM15180
BEM15190
BEM15200
BEM15210
BEM15220
BEM15230
BEM15240
BEM15250
BEM15260
BEM15270
BEM15280
BEM15290
BEM15300
BEM15310
BEM15320
BEM15330
BEM15340
BEM15350
BEM15360
BEM15370
BEM15380
BEM15390
BEM15400

```

```

C *****BEM15410
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)          BEM15420
COMMON/MAIN/CONST,N,LEC,IMP,NU,ND          BEM15430
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2      BEM15440
REAL*8 XULTO,XOSHO,XOSH1                   BEM15450
INTEGER XF,ZO,Z                             BEM15460
C BESSEL FUNCTION CALCULATIONS              BEM15470
IF(XX-15.9)1000,1000,2000                  BEM15480
1000 A=0.6366197724                          BEM15490
      B=0.5772156646                          BEM15500
      XULTO=1.0                              BEM15510
      SIGMA=0.0                              BEM15520
      TYO=0.0                                BEM15530
      TY1=0.0                                BEM15540
      DO 7 I=1,500                           BEM15550
      XF=I                                    BEM15560
      Z=-1.0                                 BEM15570
      ZO=XF+1.0                              BEM15580
      SINYO=Z**XF                            BEM15590
      SINY1=Z**(ZO)                          BEM15600
      XOSHO=(XX/2.0)**(2*XF)                 BEM15610
      XOSH1=(XX/2.0)**(2*XF-1.0)             BEM15620
      XULTO=XULTO*XF                         BEM15630
      XULT1=XULTO/XF                         BEM15640
      SIGMA=SIGMA+(1.0/XF)                   BEM15650
      VALUO=DLOG(XX/2.0)+B-SIGMA              BEM15660
      VALU1=VALUO+(1.0/(2.0*XF))             BEM15670
      TRMYO=((SINYO*XOSHO)/(XULTO**2))*(VALUO) BEM15680
      TRMY1=((SINY1*XOSH1)/(XULTO*XULT1))*(VALU1) BEM15690
      TYO=TYO+TRMYO                          BEM15700
      TY1=TY1+TRMY1                          BEM15710
      ERYO=DABS(TRMYO/TYO)                   BEM15720
      ERY1=DABS(TRMY1/TY1)                   BEM15730
      IF(ERYO-ERY1)1,1,2                     BEM15740
1      ER=ERY1                               BEM15750
      GO TO 10                               BEM15760
2      ER=ERYO                               BEM15770
10 IF(ER-.001)11,11,7                       BEM15780
7 CONTINUE                                  BEM15790
11 YOX=A*(DLOG(XX/2.0)+B+TYO)                BEM15800
      Y1X=A*((1.0/XX)-TY1)                  BEM15810
      GO TO 100                              BEM15820
2000 AH=3.1415926536/4.0                     BEM15830
      BH=XX-AH                               BEM15840
      CH=XX+AH                               BEM15850
      S1H=SIN(BH)                            BEM15860
      S2H=SIN(CH)                            BEM15870
      DHYO=S2H/(8.0*XX)                     BEM15880
      DHY1=S1H*3.0/(8.0*XX)                 BEM15890
      HH=XX-(3.0*AH)                         BEM15900
      S3H=SIN(HH)                           BEM15910
      EH=2.0/(3.1415926536*XX)              BEM15920
      FH=DSQRT(EH)                           BEM15930
      XMYO=S1H-DHYO                          BEM15940

```

```

XMY1=S3H+DHY1
Y0X=XMY0*FH
Y1X=-XMY1*FH
100 RETURN
END
C
C -----
C
SUBROUTINE ADDFBC(H,G,HOLD,AFR,AFI,AKR,AKI,N0,N1,N2,
1 DFI,FID,FIB,DF)
C
C *****
C
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND)
DOUBLE PRECISION H(NU,NU),G(NU,NU),T(100,20)
DOUBLE PRECISION FID(NU),FIB(NU),DFI(NU),HOLD(NU,NU)
REAL UL(20,20),VB(20,20),VL(20,20),UB(20,20)
COMPLEX XKAP(20),PSI(20,20),TEMP1(20,20),TEMP2(20,20)
COMPLEX SUM3,DX(20)
REAL DXR(20),DXI(20)
REAL ULT(20,20),VLT(20,20),UBT(20,20),VBT(20,20)
REAL AFRT(20,20),AFIT(20,20),TP(20,20)
C
C REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
C CONDITIONS INTO ACCOUNT
C
C ARRANGE H MATRIX
C REAL PART
C
IF(ME.GT.0)THEN
DO 20 J=1,ME
DO 10 I=1,ME
UL(I,J)=AFR(I,J)*AKR(J)
VB(I,J)=AFI(I,J)*AKI(J)
VL(I,J)=AFI(I,J)*AKR(J)
UB(I,J)=AFR(I,J)*AKI(J)
10 CONTINUE
20 CONTINUE
C
C CREATE MATRIX TP WHICH TRANSFORMS VECTORS OF PRESSURES AND THEIR
C DERIVATIVES AT FAR END FROM LINEAR FORMULATION INTO CONSTANT ONE
C ( FROM NODAL POINTS INTO MIDPOINTS )
C
CALL TRANSP(TP)
C
C
DO 21 I=1,ME
DO 22 J=1,ME
TEM1=0.0
TEM2=0.0
TEM3=0.0
TEM4=0.0
DO 23 K=1,ME

```

```

BEM15960
BEM15970
BEM15980
BEM15990
BEM16000
BEM16010
BEM16020
BEM16030
BEM16040
BEM16050
BEM16060
BEM16070
BEM16080
BEM16090
BEM16100
BEM16110
BEM16120
BEM16130
BEM16140
BEM16150
BEM16160
BEM16170
BEM16180
BEM16190
BEM16200
BEM16210
BEM16220
BEM16230
BEM16240
BEM16250
BEM16260
BEM16270
BEM16280
BEM16290
BEM16300
BEM16310
BEM16320
BEM16330
BEM16340
BEM16350
BEM16360
BEM16370
BEM16380
BEM16390
BEM16400
BEM16410
BEM16420
BEM16430
BEM16440
BEM16450
BEM16460
BEM16470
BEM16480
BEM16490
BEM16500

```

```
      TEM1=TEM1+TP(I,K)*UL(K,J)
      TEM2=TEM2+TP(I,K)*VB(K,J)
      TEM3=TEM3+TP(I,K)*VL(K,J)
      TEM4=TEM4+TP(I,K)*UB(K,J)
23  CONTINUE
      ULT(I,J)=TEM1
      VBT(I,J)=TEM2
      VLT(I,J)=TEM3
      UBT(I,J)=TEM4
22  CONTINUE
21  CONTINUE
C
      DO 41 I=1,ME
      DO 31 J=1,ME
      ULT(I,J)=ULT(I,J)-VBT(I,J)
      VLT(I,J)=VLT(I,J)+UBT(I,J)
31  CONTINUE
41  CONTINUE
C
      DO 61 I=1,ME
      DO 62 J=1,ME
      TEM1=0.0
      TEM2=0.0
      DO 63 K=1,ME
      TEM1=TEM1+TP(I,K)*AFR(K,J)
      TEM2=TEM2+TP(I,K)*AFI(K,J)
63  CONTINUE
      AFRT(I,J)=TEM1
      AFIT(I,J)=TEM2
62  CONTINUE
61  CONTINUE
C
      DO 70 I=1,N
      DO 60 J=1,ME
      TEM1=0.0
      TEM2=0.0
      DO 50 K=1,ME
      KK=K+N2
      TEM1=TEM1+H(I,KK)*DBLE(AFRT(K,J))
      TEM2=TEM2+G(I,KK)*DBLE(ULT(K,J))
50  CONTINUE
      T(I,J)=TEM1+TEM2
60  CONTINUE
70  CONTINUE
C
      DO 75 I=1,N
      DO 72 J=1,ME
      JJ=J+N2
      H(I,JJ)=T(I,J)
72  CONTINUE
75  CONTINUE
C
C IMAGINARY PART
C
C
```

```
BEM16510
BEM16520
BEM16530
BEM16540
BEM16550
BEM16560
BEM16570
BEM16580
BEM16590
BEM16600
BEM16610
BEM16620
BEM16630
BEM16640
BEM16650
BEM16660
BEM16670
BEM16680
BEM16690
BEM16700
BEM16710
BEM16720
BEM16730
BEM16740
BEM16750
BEM16760
BEM16770
BEM16780
BEM16790
BEM16800
BEM16810
BEM16820
BEM16830
BEM16840
BEM16850
BEM16860
BEM16870
BEM16880
BEM16890
BEM16900
BEM16910
BEM16920
BEM16930
BEM16940
BEM16950
BEM16960
BEM16970
BEM16980
BEM16990
BEM17000
BEM17010
BEM17020
BEM17030
BEM17040
BEM17050
```

```
DO 120 I=1,N
DO 110 J=1,ME
TEM1=0.0
TEM2=0.0
DO 100 K=1,ME
KK=K+N2
TEM1=TEM1+G(I,KK)*DBLE(VLT(K,J))
TEM2=TEM2+H(I,KK)*DBLE(AFIT(K,J))
100 CONTINUE
T(I,J)=TEM1+TEM2
110 CONTINUE
120 CONTINUE
C
DO 140 I=1,N
DO 130 J=1,ME
H(N+I,J+N2)=T(I,J)
130 CONTINUE
140 CONTINUE
END IF
C
IF(CONST2.GT.1.E-08) THEN
TEMP=CONST1*CONST2
DO 90 I=1,N
DO 80 J=N1+1,N2
H(N+I,J)=G(I,J)*TEMP
80 CONTINUE
90 CONTINUE
END IF
C
C FILL IN THE REST OF H MATRIX
C
C
DO 160 I=1,N
DO 150 J=1,N
H(I,J+N)=H(I+N,J)
H(I+N,J+N)=H(I,J)
150 CONTINUE
160 CONTINUE
C
C FORM RIGHT HAND VECTOR
C
DO 180 I=1,N
DO 170 J=NO+1,N1
DFI(I)=DFI(I)+G(I,J)*FID(J-NO)
170 CONTINUE
180 CONTINUE
C
DO 200 I=1,N
DO 190 J=N1+1,N2
DFI(I)=DFI(I)+G(I,J)*FIB(J-N1)
190 CONTINUE
200 CONTINUE
C
IF(DF.GT.0.0) THEN
DO 211 I=1,ME
```

```
BEM17060
BEM17070
BEM17080
BEM17090
BEM17100
BEM17110
BEM17120
BEM17130
BEM17140
BEM17150
BEM17160
BEM17170
BEM17180
BEM17190
BEM17200
BEM17210
BEM17220
BEM17230
BEM17240
BEM17250
BEM17260
BEM17270
BEM17280
BEM17290
BEM17300
BEM17310
BEM17320
BEM17330
BEM17340
BEM17350
BEM17360
BEM17370
BEM17380
BEM17390
BEM17400
BEM17410
BEM17420
BEM17430
BEM17440
BEM17450
BEM17460
BEM17470
BEM17480
BEM17490
BEM17500
BEM17510
BEM17520
BEM17530
BEM17540
BEM17550
BEM17560
BEM17570
BEM17580
BEM17590
BEM17600
```

```

      XKAP(I)=CMPLX(AKR(I),AKI(I))
      DO 212 J=1,ME
      PSI(I,J)=CMPLX(AFRT(I,J),AFIT(I,J))
212  CONTINUE
211  CONTINUE
      DO 210 I=1,ME
      DO 220 J=1,ME
      TEMP1(I,J)=PSI(I,J)/(XKAP(J)*XKAP(J))
220  CONTINUE
210  CONTINUE
      DO 230 I=1,ME
      DO 240 J=1,ME
      SUM3=(0.0,0.0)
      DO 250 K=1,ME
      SUM3=SUM3+TEMP1(J,K)*PSI(I,K)
250  CONTINUE
      TEMP2(J,I)=SUM3
240  CONTINUE
230  CONTINUE
      DO 260 I=1,ME
      DX(I)=-TEMP2(I,1)*SNGL(DF)
260  CONTINUE
      DO 270 I=1,ME
      DXR(I)=REAL(DX(I))
      DXI(I)=AIMAG(DX(I))
270  CONTINUE
      WRITE(IMP,101)(DXR(I),DXI(I),I=1,ME)
101  FORMAT(2(4X,E14.5))
      DO 280 I=1,N
      DO 290 J=1,ME
      K1=J-ME+N
      DFI(I)=DFI(I)+HOLD(I,K1)*DBLE(DXR(J))
      DFI(I+N)=DFI(I+N)+HOLD(I,K1)*DBLE(DXI(J))
290  CONTINUE
280  CONTINUE
      END IF
      RETURN
      END
C
C -----
C
C SUBROUTINE SLNPD(A,B,D,NX,NU,IMP)
C *****
C
C SOLUTION OF LINEAR SYSTEMS OF EQUATIONS
C BY THE GAUSS ELIMINATION METHOD PROVIDING
C FOR INTERCHANGING ROWS WHEN ENCOUNTERING
C A ZERO DIAGONAL COEFFICIENT
C
C A :SYSTEM MATRIX
C B :ORIGINALLY IT CONTAINS THE INDEPENDENT
C COEFFICIENTS. AFTER SOLUTION IT CONTAINS THE
C VALUES OF THE SYSTEM UNKNOWNNS
C

```

```

BEM17610
BEM17620
BEM17630
BEM17640
BEM17650
BEM17660
BEM17670
BEM17680
BEM17690
BEM17700
BEM17710
BEM17720
BEM17730
BEM17740
BEM17750
BEM17760
BEM17770
BEM17780
BEM17790
BEM17800
BEM17810
BEM17820
BEM17830
BEM17840
BEM17850
BEM17860
BEM17870
BEM17880
BEM17890
BEM17900
BEM17910
BEM17920
BEM17930
BEM17940
BEM17950
BEM17960
BEM17970
BEM17980
BEM17990
BEM18000
BEM18010
BEM18020
BEM18030
BEM18040
BEM18050
BEM18060
BEM18070
BEM18080
BEM18090
BEM18100
BEM18110
BEM18120
BEM18130
BEM18140
BEM18150

```

C	NX : ACTUAL NUMBER OF UNKNOWNNS	BEM18160
C		BEM18170
C	NU : ROW AND COLUMN DIMENSION OF A	BEM18180
C		BEM18190
	IMPLICIT DOUBLE PRECISION (A-H,O-Z)	BEM18200
	DOUBLE PRECISION A(NU,NU),B(NU)	BEM18210
	N1=NX-1	BEM18220
	DO 100 K=1,N1	BEM18230
	K1=K+1	BEM18240
	C=A(K,K)	BEM18250
	IF(DABS(C)-.000001)1,1,3	BEM18260
	1 DO 7 J=K1,NX	BEM18270
C		BEM18280
C	TRY TO INTERCHANGE ROWS TO GET NON ZERO	BEM18290
C	DIAGONAL COEFFICIENT	BEM18300
C		BEM18310
	IF (DABS(A(J,K))-0.000001)7,7,5	BEM18320
5	DO 6 LL=K,NX	BEM18330
	C=A(K,LL)	BEM18340
	A(K,LL)=A(J,LL)	BEM18350
6	A(J,LL)=C	BEM18360
	C=B(K)	BEM18370
	B(K)=B(J)	BEM18380
	B(J)= C	BEM18390
	C=A(K,K)	BEM18400
	GO TO 3	BEM18410
7	CONTINUE	BEM18420
8	WRITE(IMP,2)K	BEM18430
	D=0.0	BEM18440
	GO TO 300	BEM18450
C		BEM18460
C	DIVIDE ROW BY DIAGONAL COEFFICIENT	BEM18470
C		BEM18480
3	C=A(K,K)	BEM18490
	DO 4 J=K1,NX	BEM18500
4	A(K,J)=A(K,J)/C	BEM18510
	B(K)=B(K)/C	BEM18520
C		BEM18530
C	ELIMINATE UNKNOWN X(K) FROM ROW I	BEM18540
C		BEM18550
	DO 10 I=K1,NX	BEM18560
	C=A(I,K)	BEM18570
	DO 9 J=K1,NX	BEM18580
9	A(I,J)=A(I,J)-C*A(K,J)	BEM18590
10	B(I)=B(I)-C*B(K)	BEM18600
100	CONTINUE	BEM18610
C		BEM18620
C	COMPUTE LAST UNKNOWN	BEM18630
C		BEM18640
	IF (DABS(A(NX,NX))-0.000001)101,101,102	BEM18650
101	WRITE(IMP,2)NX	BEM18660
	D=0.	BEM18670
	GO TO 300	BEM18680
102	B(NX)=B(NX)/A(NX,NX)	BEM18690
C		BEM18700


```

C      APPLY BACKSUBSTITUTION PROCESS TO COMPUTE REMAINING      BEM18710
C      UNKNOWNNS                                              BEM18720
C                                                              BEM18730
      DO 200 LL=1,N1                                          BEM18740
      K=NX-LL                                              BEM18750
      K1=K+1                                              BEM18760
      DO 200 J=K1,NX                                          BEM18770
200    B(K)=B(K)-A(K,J)*B(J)                                BEM18780
C                                                              BEM18790
C      COMPUTE VALUE OF DETERMINATE                          BEM18800
      D=1.0                                              BEM18810
      DO 250 I=1,NX                                          BEM18820
250    D=D*A(I,I)                                          BEM18830
      WRITE(IMP,333)D                                        BEM18840
333    FORMAT('THE DETERMINANT =',E14.8)                  BEM18850
      2 FORMAT('**** SINGULARITY IN ROW ****',I5)          BEM18860
300    RETURN                                              BEM18870
      END                                                  BEM18880
C                                                              BEM18890
C      -----BEM18900
C                                                              BEM18910
      SUBROUTINE JACOBI(AS,BM,EIGV,X,ME,ND)                BEM18920
C      *****BEM18930
C                                                              BEM18940
      REAL EIGV(ND),X(ND,ND), BM(ND,ND)                  BEM18950
      REAL AS(ND,ND),D(150)                                BEM18960
C                                                              BEM18970
C      AS=A : STIFNESS MATRIX                                BEM18980
C      BM=F : MASS MATRIX                                    BEM18990
C      NT=ME : TOTAL DOF                                     BEM19000
C      X : FINAL EIGENVECTOR                                BEM19010
C      EIGV: FINAL EIGENVALUE                                BEM19020
C      IFPR=0 :FOR PRINTOUT OF INTERM. RESULTS              BEM19030
C      IFPR=1 :FOR PRINTOUT OF INTERM. RESULTS              BEM19040
C                                                              BEM19050
C      CONSTANSTS                                           BEM19060
C                                                              BEM19070
      NSMAX = 15                                           BEM19080
      RTOL = 10.**(-12)                                     BEM19090
      IFPR = 0                                              BEM19100
C                                                              BEM19110
C                                                              BEM19120
      NT= ME                                              BEM19130
C                                                              BEM19140
C      INITIALIZE EIGENVALUE AND EIGENVECTOR MATRICES      BEM19150
C                                                              BEM19160
      DO 10 I = 1,NT                                       BEM19170
      IF (AS(I,I) .GT. 0. .AND. BM(I,I) .GT. 0.) GOTO 9    BEM19180
      WRITE (*,2020)                                       BEM19190
      STOP                                                  BEM19200
9      D(I) = AS(I,I)/BM(I,I)                               BEM19210
      EIGV(I) = D(I)                                       BEM19220
10     CONTINUE                                           BEM19230
      DO 30 I = 1,NT                                       BEM19240
      DO 20 J = 1,NT                                       BEM19250

```

```

20  X(I,J) = 0.
30  X(I,I) = 1.
    IF (NT.EQ. 1) THEN
C    WRITE (*,*) 'EIGENVALUE : ',EIGV(1)
      RETURN
    ELSE
      ENDIF

C    INITIALIZE SWEEP COUNTER AND EIGEN ITERATION
C
    NSWEEP = 0
    NR = NT - 1
40  NSWEEP = NSWEEP + 1
    IF (IFPR .EQ. 1) WRITE (*, 2000) NSWEEP

C
C    CHECK IF PRESENT OFF-DIAGONAL ELEMENTS ARE LARGE ENOUGH TO
C    REQUIRE ZEROING
C
    EPS = (.01*NSWEEP)**2
    DO 210 J = 1,NR
      JJ = J + 1
      DO 210 K = JJ,NT
        EPTOLA = (AS(J,K)*AS(J,K))/(AS(J,J)*AS(K,K))
        EPTOLB = (BM(J,K)*BM(J,K))/(BM(J,J)*BM(K,K))
        IF((EPTOLA .LT. EPS) .AND. (EPTOLB .LT. EPS)) GOTO 210

C
C    IF ZEROING IS REQUIRED CALCULATE THE ROT MAT ELM CA ,CG
C
    AKK = AS(K,K) * BM(J,K) - BM(K,K) * AS(J,K)
    AJJ = AS(J,J) * BM(J,K) - BM(J,J) * AS(J,K)
    AB = AS(J,J) * BM(K,K) - AS(K,K) * BM(J,J)
    CHECK = (AB*AB + 4.* AKK * AJJ)/4.
    IF (CHECK .LT. 0.) THEN
      WRITE (*,2020)
      STOP
    ELSE
      ENDIF
60  SQCH = SQRT(CHECK)
    D1 = AB/2. + SQCH
    D2 = AB/2. - SQCH
    DEN = D1
    IF (ABS(D2) .GT. ABS(D1)) THEN
      DEN = D2
    ELSE
      ENDIF
70  CA = 0.
    CG = -AS(J,K)/AS(K,K)
    ELSE
80  CA = (AKK)/(DEN)
    CG = - (AJJ)/(DEN)
    ENDIF

C
C    PERFORM GENERALIZED ROT TO ZERO THE PRESENT OFF DG. ELM
C

```

```

BEM19260
BEM19270
BEM19280
BEM19290
BEM19300
BEM19310
BEM19320
BEM19330
BEM19340
BEM19350
BEM19360
BEM19370
BEM19380
BEM19390
BEM19400
BEM19410
BEM19420
BEM19430
BEM19440
BEM19450
BEM19460
BEM19470
BEM19480
BEM19490
BEM19500
BEM19510
BEM19520
BEM19530
BEM19540
BEM19550
BEM19560
BEM19570
BEM19580
BEM19590
BEM19600
BEM19610
BEM19620
BEM19630
BEM19640
BEM19650
BEM19660
BEM19670
BEM19680
BEM19690
BEM19700
BEM19710
BEM19720
BEM19730
BEM19740
BEM19750
BEM19760
BEM19770
BEM19780
BEM19790
BEM19800

```

```

90 IF (NT .EQ. 2) GOTO 190
100 JP1 = J + 1
    JM1 = J - 1
    KP1 = K + 1
    KM1 = K - 1
    IF (JM1 .LT. 1) GOTO 130
110 DO 120 I = 1, JM1
    AJ = AS(I, J)
    BJ = BM(I, J)
    AK = AS(I, K)
    BK = BM(I, K)
    AS(I, J) = AJ + CG * AK
    BM(I, J) = BJ + CG * BK
    AS(I, K) = AK + CA * AJ
120 BM(I, K) = BK + CA * BJ
130 IF (KP1 .GT. NT) GOTO 160
140 DO 150 I = KP1, NT
    AJ = AS(J, I)
    BJ = BM(J, I)
    AK = AS(K, I)
    BK = BM(K, I)
    AS(J, I) = AJ + CG * AK
    BM(J, I) = BJ + CG * BK
    AS(K, I) = AK + CA * AJ
150 BM(K, I) = BK + CA * BJ
160 IF (JP1 .GT. KM1) GOTO 190
170 DO 180 I = JP1, KM1
    AJ = AS(J, I)
    BJ = BM(J, I)
    AK = AS(I, K)
    BK = BM(I, K)
    AS(J, I) = AJ + CG * AK
    BM(J, I) = BJ + CG * BK
    AS(I, K) = AK + CA * AJ
180 BM(I, K) = BK + CA * BJ
190 AK = AS(K, K)
    BK = BM(K, K)
    AS(K, K) = AK + 2.* CA * AS(J, K) + CA * CA * AS(J, J)
    BM(K, K) = BK + 2.* CA * BM(J, K) + CA * CA * BM(J, J)
    AS(J, J) = AS(J, J) + 2.* CG * AS(J, K) + CG * CG * AK
    BM(J, J) = BM(J, J) + 2.* CG * BM(J, K) + CG * CG * BK
    AS(J, K) = 0.
    BM(J, K) = 0.
C
C UPDATE EIGENVALUE MATRIX AFTER EACH ROTATION
C
    DO 200 I = 1, NT
    XJ = X(I, J)
    XK = X(I, K)
    X(I, J) = XJ + CG * XK
    X(I, K) = XK + CA * XJ
200 CONTINUE
210 CONTINUE
C
C UPDATE EIGENVALUE AFTER EACH SWEEP

```

```

BEM19810
BEM19820
BEM19830
BEM19840
BEM19850
BEM19860
BEM19870
BEM19880
BEM19890
BEM19900
BEM19910
BEM19920
BEM19930
BEM19940
BEM19950
BEM19960
BEM19970
BEM19980
BEM19990
BEM20000
BEM20010
BEM20020
BEM20030
BEM20040
BEM20050
BEM20060
BEM20070
BEM20080
BEM20090
BEM20100
BEM20110
BEM20120
BEM20130
BEM20140
BEM20150
BEM20160
BEM20170
BEM20180
BEM20190
BEM20200
BEM20210
BEM20220
BEM20230
BEM20240
BEM20250
BEM20260
BEM20270
BEM20280
BEM20290
BEM20300
BEM20310
BEM20320
BEM20330
BEM20340
BEM20350

```



```
C
DO 220 I = 1,NT
IF (AS(I,I) .GT. 0. .AND. BM(I,I) .GT. 0.) GOTO 220
WRITE (*,2020)
STOP
220 EIGV(I) = AS(I,I)/BM(I,I)
IF (IFPR .EQ. 0) GOTO 230
WRITE (*,2030)
WRITE (*,710) (EIGV(I), I = 1,NT)

C
C CHECK FOR CONVERGENCE
C
230 DO 240 I = 1,NT
TOL = RTOL * D(I)
DIF = ABS(EIGV(I) - D(I))
IF (DIF .GT. TOL) GOTO 280
240 CONTINUE

C
C CHECK ALL OFF DIAG, TO SEE IF ANOTHER SWEEP IS REQD.
C
EPS = RTOL ** 2
DO 250 J = 1,NR
JJ = J + 1
DO 253 K = JJ,NT
EPSA = (AS(J,K)*AS(J,K))/(AS(J,J)*AS(K,K))
EPSB = (BM(J,K)*BM(J,K))/(BM(J,J)*BM(K,K))
IF ((EPSA .LT. EPS) .AND. (EPSB .LT. EPS)) GOTO 253
GOTO 280
253 CONTINUE
250 CONTINUE

C
C FILL BOTTOM TRIANG OF RESLT MATRICES AND SCALE EG. VECT.
C
255 DO 260 I = 1,NT
DO 260 J = 1,NT
AS(J,I) = AS(I,J)
260 BM(J,I) = BM(I,J)
DO 270 J = 1,NT
BB = SQRT(BM(J,J))
DO 270 K = 1,NT
270 X(K,J) = X(K,J)/BB

C
C SORT EIGENVALUE AND VECTOR
C
510 KOUNT = 0
DO 520 I = 1,NT-1
IF (EIGV(I) .GT. EIGV(I+1)) THEN
KOUNT = 1
CHNG = EIGV(I)
EIGV(I) = EIGV(I+1)
EIGV(I+1) = CHNG
DO 530 J = 1,NT
CHNG = X(J,I)
X(J,I) = X(J,I+1)
X(J,I+1) = CHNG
```

```
BEM20360
BEM20370
BEM20380
BEM20390
BEM20400
BEM20410
BEM20420
BEM20430
BEM20440
BEM20450
BEM20460
BEM20470
BEM20480
BEM20490
BEM20500
BEM20510
BEM20520
BEM20530
BEM20540
BEM20550
BEM20560
BEM20570
BEM20580
BEM20590
BEM20600
BEM20610
BEM20620
BEM20630
BEM20640
BEM20650
BEM20660
BEM20670
BEM20680
BEM20690
BEM20700
BEM20710
BEM20720
BEM20730
BEM20740
BEM20750
BEM20760
BEM20770
BEM20780
BEM20790
BEM20800
BEM20810
BEM20820
BEM20830
BEM20840
BEM20850
BEM20860
BEM20870
BEM20880
BEM20890
BEM20900
```

```

530 CONTINUE                                BEM20910
ENDIF                                       BEM20920
520 CONTINUE                                BEM20930
IF (KOUNT .EQ. 1) GOTO 510                 BEM20940
IF (IFPR .EQ. 0) GOTO 276                 BEM20950
WRITE (*,960)                              BEM20960
DO 275 K = 1,NT                            BEM20970
WRITE (*,710) (X(K,J), J = 1,NT)         BEM20980
275 CONTINUE                                BEM20990
276 RETURN                                  BEM21000
C                                           BEM21010
C      UPDATE D MATRIX AND START NEW SWEEP, IF ALLOWED BEM21020
C                                           BEM21030
280 DO 290 I = 1,NT                        BEM21040
290 D(I) = EIGV(I)                         BEM21050
IF (NSWEEP .LT. NSMAX) GOTO 40            BEM21060
GOTO 255                                   BEM21070
710 FORMAT (10(E12.5,1X))                 BEM21080
                                           BEM21090
910 FORMAT (//, ' ** STIFNESS MATRIX **',/) BEM21100
920 FORMAT (9(G10.4,1X)/                  BEM21110
          5X,9(G10.4,1X))                 BEM21120
930 FORMAT (//, ' ** MASS MATRIX **',/)   BEM21130
960 FORMAT (//, '   FINAL EIGEN VECTOR OF MODAL SYSTEM',/) BEM21140
2000 FORMAT (//, ' SWEEP CYCLE IN JACOBI ' ,I4,/) BEM21150
2020 FORMAT (10X,'ERROR SOLUTION STOP',/  BEM21160
          5X,' MATRICES NOT POSITIVE DEFINITE') BEM21170
2030 FORMAT (' CURRENT EIGENVALUE IN *JACOBI* ARE',/) BEM21180
END                                         BEM21190
C                                           BEM21200
C      ----- BEM21210
C                                           BEM21220
SUBROUTINE OUTPUT(XM,YM,FID,FIB,DFI,AFR,AFI,AKR,AKI, BEM21230
* NO,N1,N2,NDOUT,FLAG,DF,NC,TPR,TPI,HXR,HXI,YR,YI) BEM21240
C      ***** BEM21250
C                                           BEM21260
IMPLICIT DOUBLE PRECISION(A-H,O-Z)        BEM21270
COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND        BEM21280
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2     BEM21290
DOUBLE PRECISION XM(1),YM(1),FID(1),FIB(1),DFI(1) BEM21300
REAL AFR(ND,ND),AFI(ND,ND),AKR(ND),AKI(ND) BEM21310
REAL UL(20,20),UB(20,20),VL(20,20),VB(20,20) BEM21320
DOUBLE PRECISION PRA(20),PIA(20),QRA(20),QIA(20) BEM21330
COMPLEX XKAP(20),PSI(20,20),TEMP1(20,20),TEMP2(20,20) BEM21340
COMPLEX SUM3,DXI(20),PSIT(20,20)          BEM21350
REAL TEMP1R(20,20),TEMP2R(20,20)          BEM21360
REAL DXR(20),DXI(20),SUM3R                BEM21370
REAL ULT(20,20),UBT(20,20),VLT(20,20),VBT(20,20) BEM21380
REAL AFRT(20,20),AFIT(20,20),TP(20,20)    BEM21390
REAL YR(20,500),YI(20,500)               BEM21400
LOGICAL FLAG                               BEM21410
C                                           BEM21420
C                                           BEM21430
RES=0.                                     BEM21440
WRITE(IMP,500)                             BEM21450

```



```
C
C      WRITE RESULTS FOR RESERVOIR SURFACE
C
      IF(NOUT.EQ.1)THEN
      WRITE(IMP,600)
      WRITE(IMP,330)
      WRITE(IMP,600)
      WRITE(IMP,100)
      WRITE(IMP,600)
      WRITE(IMP,300)
      WRITE(IMP,600)
      DO 10 I=1,N0
      WRITE(IMP,200)XM(I),YM(I),RES,DFI(I)
10  CONTINUE
      IF(FLAG)THEN
      WRITE(IMP,600)
      WRITE(IMP,320)
      WRITE(IMP,600)
      DO 20 I=1,N0
      WRITE(IMP,200)XM(I),YM(I),RES,DFI(I+N)
20  CONTINUE
      END IF
      END IF
C
C      WRITE RESULTS FOR DAM FACE
C
      WRITE(IMP,600)
      WRITE(IMP,340)
      WRITE(IMP,600)
      WRITE(IMP,100)
      WRITE(IMP,600)
      WRITE(IMP,300)
      WRITE(IMP,600)
      J1=0
      DO 30 I=N0+1,N1
      J1=J1+1
      WRITE(IMP,200)XM(I),YM(I),DFI(I),FID(I-N0)
      YR(J1,NC)=DFI(I)
30  CONTINUE
      IF(FLAG)THEN
      WRITE(IMP,600)
      WRITE(IMP,320)
      WRITE(IMP,600)
      J1=0
      DO 40 I=N0+1,N1
      J1=J1+1
      WRITE(IMP,200)XM(I),YM(I),DFI(N+I),RES
      YI(J1,NC)=DFI(N+I)
40  CONTINUE
      END IF
C
C      WRITE RESULTS FOR RESERVOIR BOTTOM
C
      IF(NOUT.EQ.1)THEN
      WRITE(IMP,600)
```

```
BEM21460
BEM21470
BEM21480
BEM21490
BEM21500
BEM21510
BEM21520
BEM21530
BEM21540
BEM21550
BEM21560
BEM21570
BEM21580
BEM21590
BEM21600
BEM21610
BEM21620
BEM21630
BEM21640
BEM21650
BEM21660
BEM21670
BEM21680
BEM21690
BEM21700
BEM21710
BEM21720
BEM21730
BEM21740
BEM21750
BEM21760
BEM21770
BEM21780
BEM21790
BEM21800
BEM21810
BEM21820
BEM21830
BEM21840
BEM21850
BEM21860
BEM21870
BEM21880
BEM21890
BEM21900
BEM21910
BEM21920
BEM21930
BEM21940
BEM21950
BEM21960
BEM21970
BEM21980
BEM21990
BEM22000
```

```

WRITE(IMP,350)
WRITE(IMP,600)
WRITE(IMP,100)
WRITE(IMP,600)
WRITE(IMP,300)
WRITE(IMP,600)
DO 50 I=N1+1,N2
  QR=FIB(I-N1)
  IF(CONST2.GT.1.0E-08)THEN
    QR=QR+CONST1*CONST2*DFI(I+N)
  END IF
  WRITE(IMP,200)XM(I),YM(I),DFI(I),QR
50 CONTINUE
  IF(FLAG)THEN
    WRITE(IMP,600)
    WRITE(IMP,320)
    WRITE(IMP,600)
    DO 60 I=N1+1,N2
      QI=0
      IF(CONST2.GT.1.0E-08)THEN
        QI=QI-CONST1*CONST2*DFI(I)
      END IF
      WRITE(IMP,200)XM(I),YM(I),DFI(I+N),QI
60 CONTINUE
    END IF
C
C   WRITE RESULTS FOR THE FAR END
C
  IF(ME.GT.0)THEN
    IF(CONST2.GT.1.0E-08)THEN
      DO 80 J=1,ME
        DO 70 I=1,ME
          UL(I,J)=AFR(I,J)*AKR(J)
          VB(I,J)=AFI(I,J)*AKI(J)
          VL(I,J)=AFI(I,J)*AKR(J)
          UB(I,J)=AFR(I,J)*AKI(J)
70 CONTINUE
80 CONTINUE
C
C   CALL TRANSP(TP)
C
  DO 81 I=1,ME
    DO 82 J=1,ME
      TEM1=0.0
      TEM2=0.0
      TEM3=0.0
      TEM4=0.0
      DO 83 K=1,ME
        TEM1=TEM1+TP(I,K)*UL(K,J)
        TEM2=TEM2+TP(I,K)*VB(K,J)
        TEM3=TEM3+TP(I,K)*VL(K,J)
        TEM4=TEM4+TP(I,K)*UB(K,J)
83 CONTINUE
      ULT(I,J)=TEM1
      VBT(I,J)=TEM2

```

```

BEM22010
BEM22020
BEM22030
BEM22040
BEM22050
BEM22060
BEM22070
BEM22080
BEM22090
BEM22100
BEM22110
BEM22120
BEM22130
BEM22140
BEM22150
BEM22160
BEM22170
BEM22180
BEM22190
BEM22200
BEM22210
BEM22220
BEM22230
BEM22240
BEM22250
BEM22260
BEM22270
BEM22280
BEM22290
BEM22300
BEM22310
BEM22320
BEM22330
BEM22340
BEM22350
BEM22360
BEM22370
BEM22380
BEM22390
BEM22400
BEM22410
BEM22420
BEM22430
BEM22440
BEM22450
BEM22460
BEM22470
BEM22480
BEM22490
BEM22500
BEM22510
BEM22520
BEM22530
BEM22540
BEM22550

```

```
      VLT(I,J)=TEM3
      UBT(I,J)=TEM4
      82 CONTINUE
      81 CONTINUE
C
      DO 105 I=1,ME
      DO 90 J=1,ME
      ULT(I,J)=ULT(I,J)-VBT(I,J)
      VLT(I,J)=VLT(I,J)+UBT(I,J)
      90 CONTINUE
      105 CONTINUE
      ELSE IF(FLAG.AND.CONST2.LE.1.0E-08)THEN
C
      CALL TRANSP(TP)
C
      DO 120 J=1,ME
      DO 110 I=1,ME
      UL(I,J)=AFR(I,J)*AKR(J)
      VL(I,J)=AFR(I,J)*AKI(J)
      110 CONTINUE
      120 CONTINUE
      DO 121 I=1,ME
      DO 122 J=1,ME
      TEM1=0.0
      TEM2=0.0
      DO 123 K=1,ME
      TEM1=TEM1+TP(I,K)*UL(K,J)
      TEM2=TEM2+TP(I,K)*VL(K,J)
      123 CONTINUE
      ULT(I,J)=TEM1
      VLT(I,J)=TEM2
      122 CONTINUE
      121 CONTINUE
      END IF
C
      IF(FLAG)THEN
      DO 125 I=1,ME
      SUM1=0
      SUM2=0
      DO 124 J=1,ME
      SUM1=SUM1+ULT(I,J)*DFI(N2+J)-VLT(I,J)*DFI(N2+J+N)
      SUM2=SUM2+VLT(I,J)*DFI(N2+J)+ULT(I,J)*DFI(N2+J+N)
      124 CONTINUE
      QRA(I)=-SUM1
      QIA(I)=-SUM2
      125 CONTINUE
      END IF
C
C
      CALL TRANSP(TP)
C
      DO 126 I=1,ME
      DO 127 J=1,ME
      TEM1=0.0
      TEM2=0.0
```

```
BEM22560
BEM22570
BEM22580
BEM22590
BEM22600
BEM22610
BEM22620
BEM22630
BEM22640
BEM22650
BEM22660
BEM22670
BEM22680
BEM22690
BEM22700
BEM22710
BEM22720
BEM22730
BEM22740
BEM22750
BEM22760
BEM22770
BEM22780
BEM22790
BEM22800
BEM22810
BEM22820
BEM22830
BEM22840
BEM22850
BEM22860
BEM22870
BEM22880
BEM22890
BEM22900
BEM22910
BEM22920
BEM22930
BEM22940
BEM22950
BEM22960
BEM22970
BEM22980
BEM22990
BEM23000
BEM23010
BEM23020
BEM23030
BEM23040
BEM23050
BEM23060
BEM23070
BEM23080
BEM23090
BEM23100
```



```
DO 128 K=1,ME                                BEM23110
TEM1=TEM1+TP(I,K)*AFRI(K,J)                  BEM23120
TEM2=TEM2+TP(I,K)*AFI(K,J)                  BEM23130
128 CONTINUE                                  BEM23140
AFRT(I,J)=TEM1                               BEM23150
AFIT(I,J)=TEM2                               BEM23160
127 CONTINUE                                  BEM23170
126 CONTINUE                                  BEM23180
C                                               BEM23190
IF(CONST2.GT.1.0E-08)THEN                    BEM23200
IF(DF.GT.0.0) THEN                          BEM23210
DO 111 I=1,ME                               BEM23220
XKAP(I)=CMPLX(AKR(I),AKI(I))                BEM23230
DO 112 J=1,ME                               BEM23240
PSIT(I,J)=CMPLX(AFRT(I,J),AFIT(I,J))        BEM23250
PSI(I,J)=CMPLX(AFR(I,J),AFI(I,J))          BEM23260
112 CONTINUE                                BEM23270
111 CONTINUE                                BEM23280
DO 113 I=1,ME                               BEM23290
DO 114 J=1,ME                               BEM23300
TEMP1(I,J)=PSIT(I,J)/(XKAP(J)*XKAP(J))      BEM23310
114 CONTINUE                                BEM23320
113 CONTINUE                                BEM23330
DO 115 I=1,ME                               BEM23340
DO 116 J=1,ME                               BEM23350
SUM3=(0.0,0.0)                              BEM23360
DO 117 K=1,ME                               BEM23370
SUM3=SUM3+TEMP1(J,K)*PSI(I,K)              BEM23380
117 CONTINUE                                BEM23390
TEMP2(J,I)=SUM3                             BEM23400
116 CONTINUE                                BEM23410
115 CONTINUE                                BEM23420
DO 118 I=1,ME                               BEM23430
DX(I)=TEMP2(I,1)*SNGL(DF)                  BEM23440
118 CONTINUE                                BEM23450
DO 119 I=1,ME                               BEM23460
DXR(I)=REAL(DX(I))                         BEM23470
DXI(I)=AIMAG(DX(I))                        BEM23480
119 CONTINUE                                BEM23490
END IF                                       BEM23500
DO 140 I=1,ME                               BEM23510
SUM1=0                                       BEM23520
SUM2=0                                       BEM23530
DO 130 J=1,ME                               BEM23540
SUM1=SUM1+AFRT(I,J)*DFI(N2+J)-AFIT(I,J)*DFI(N2+J+N) BEM23550
SUM2=SUM2+AFIT(I,J)*DFI(N2+J)+AFRT(I,J)*DFI(N2+J+N) BEM23560
130 CONTINUE                                BEM23570
IF(DF.GT.0.0) THEN                          BEM23580
PRA(I)=SUM1+DXR(I)                         BEM23590
PIA(I)=SUM2+DXI(I)                         BEM23600
ELSE                                         BEM23610
PRA(I)=SUM1                               BEM23620
PIA(I)=SUM2                               BEM23630
END IF                                       BEM23640
140 CONTINUE                                BEM23650
```

```
ELSE IF(CONST2.LE.1.0E-08.AND.FLAG)THEN
IF(DF.GT.0.0) THEN
DO 141 I=1,ME
DO 142 J=1,ME
IF(AKI(J).GT.0.0) THEN
TEMP1R(I,J)=-AFRT(I,J)/(AKI(J)*AKI(J))
ELSE
TEMP1R(I,J)=AFRT(I,J)/(AKR(J)*AKR(J))
END IF
142 CONTINUE
141 CONTINUE
DO 145 I=1,ME
DO 146 J=1,ME
SUM3R=0.0
DO 147 K=1,ME
SUM3R=SUM3R+TEMP1R(J,K)*AFR(I,K)
147 CONTINUE
TEMP2R(J,I)=SUM3R
146 CONTINUE
145 CONTINUE
DO 148 I=1,ME
DXR(I)=TEMP2R(I,1)*SNGL(DF)
148 CONTINUE
END IF
DO 160 I=1,ME
SUM1=0
SUM2=0
DO 150 J=1,ME
SUM1=SUM1+AFRT(I,J)*DFI(N2+J)
SUM2=SUM2+AFRT(I,J)*DFI(N2+J+N)
150 CONTINUE
IF(DF.GT.0.0) THEN
IF(FLAG) THEN
PRA(I)=SUM1+DXR(I)
PIA(I)=SUM2
ELSE
PRA(I)=SUM1+DXR(I)
PIA(I)=SUM2+DXI(I)
END IF
ELSE
PRA(I)=SUM1
PIA(I)=SUM2
END IF
160 CONTINUE
END IF
C
C
IF(.NOT.FLAG)THEN
IF(DF.GT.0.0) THEN
DO 311 I=1,ME
DO 312 J=1,ME
TEMP1R(I,J)=AFRT(I,J)/(AKR(J)*AKR(J))
312 CONTINUE
311 CONTINUE
DO 313 I=1,ME
```

```
BEM23660
BEM23670
BEM23680
BEM23690
BEM23700
BEM23710
BEM23720
BEM23730
BEM23740
BEM23750
BEM23760
BEM23770
BEM23780
BEM23790
BEM23800
BEM23810
BEM23820
BEM23830
BEM23840
BEM23850
BEM23860
BEM23870
BEM23880
BEM23890
BEM23900
BEM23910
BEM23920
BEM23930
BEM23940
BEM23950
BEM23960
BEM23970
BEM23980
BEM23990
BEM24000
BEM24010
BEM24020
BEM24030
BEM24040
BEM24050
BEM24060
BEM24070
BEM24080
BEM24090
BEM24100
BEM24110
BEM24120
BEM24130
BEM24140
BEM24150
BEM24160
BEM24170
BEM24180
BEM24190
BEM24200
```

```

      DO 314 J=1,ME
      SUM3R=0.0
      DO 315 K=1,ME
      SUM3R=SUM3R+TEMP1R(J,K)*AFR(I,J)
315 CONTINUE
      TEMP2R(J,I)=SUM3R
314 CONTINUE
313 CONTINUE
      DO 316 I=1,ME
      DXR(I)=TEMP2R(I,1)*SNGL(DF)
316 CONTINUE
      END IF
      DO 180 I=1,ME
      SUM1=0.
      DO 170 J=1,ME
      SUM1=SUM1+AFRT(I,J)*DFI(N2+J)
170 CONTINUE
      IF(DF.GT.0.0) THEN
      PRA(I)=SUM1+DXR(I)
      ELSE
      PRA(I)=SUM1
      END IF
180 CONTINUE
C
C
      DO 205 J=1,ME
      DO 190 I=1,ME
      UL(I,J)=AFRT(I,J)*AKR(J)
190 CONTINUE
205 CONTINUE
      DO 220 I=1,ME
      SUM1=0
      DO 210 J=1,ME
      SUM1=SUM1+UL(I,J)*DFI(N2+J)
210 CONTINUE
      QRA(I)=-SUM1
220 CONTINUE
      END IF
C
C
      WRITE(IMP,600)
      WRITE(IMP,360)
      WRITE(IMP,600)
      WRITE(IMP,100)
      WRITE(IMP,600)
      WRITE(IMP,300)
      WRITE(IMP,600)
      DO 230 I=1,ME
      K1=N2+I
      WRITE(IMP,200)XM(K1),YM(K1),PRA(I),QRA(I)
230 CONTINUE
      IF(FLAG)THEN
      WRITE(IMP,600)
      WRITE(IMP,320)
      WRITE(IMP,600)

```

```

BEM24210
BEM24220
BEM24230
BEM24240
BEM24250
BEM24260
BEM24270
BEM24280
BEM24290
BEM24300
BEM24310
BEM24320
BEM24330
BEM24340
BEM24350
BEM24360
BEM24370
BEM24380
BEM24390
BEM24400
BEM24410
BEM24420
BEM24430
BEM24440
BEM24450
BEM24460
BEM24470
BEM24480
BEM24490
BEM24500
BEM24510
BEM24520
BEM24530
BEM24540
BEM24550
BEM24560
BEM24570
BEM24580
BEM24590
BEM24600
BEM24610
BEM24620
BEM24630
BEM24640
BEM24650
BEM24660
BEM24670
BEM24680
BEM24690
BEM24700
BEM24710
BEM24720
BEM24730
BEM24740
BEM24750

```

```
      DO 240 I=1,ME
      K1=N2+I
      WRITE(IMP,200)XM(K1),YM(K1),PIA(I),QIA(I)
240  CONTINUE
      END IF
      END IF
      END IF

C
C      COMPUTE TOTAL PRESSURE ON DAM
C
      NE=N1-N0
      DY=HY/NE
      TPR=DFI(N1)
      DO 250 I=N0+1,N1-1
      TPR=TPR+DFI(I)
250  CONTINUE
      NX=0
      TPRR=DFI(N1)*0.5
      DO 251 I=N0+1,N1-1
      NX=NX+1
      RR=REAL(NX-1)
      TPRR=TPRR+DFI(I)*((N1-N0)-RR-0.5)
251  CONTINUE
      TPR=DY*TPR
      TPRR=DY*TPRR*DY
      HXR=TPRR/TPR
      WRITE(IMP,600)
      WRITE(IMP,370)TPR
      WRITE(IMP,371)HXR

C
C
      IF(FLAG)THEN
      TPI=DFI(N1+N)
      DO 260 I=N0+1,N1-1
      TPI=TPI+DFI(I+N)
260  CONTINUE
      NX=0
      TPRI=DFI(N1+N)*0.5
      DO 261 I=N0+1,N1-1
      NX=NX+1
      RR=REAL(NX-1)
      TPRI=TPRI+DFI(I+N)*((N1-N0)-RR-0.5)
261  CONTINUE
      TPI=TPI*DY
      TPRI=TPRI*DY*DY
      HXI=TPRI/TPI
      WRITE(IMP,600)
      WRITE(IMP,380)TPI
      WRITE(IMP,381)HXI
      TPRT=DSQRT(TPR**2+TPI**2)
      WRITE(IMP,600)
      WRITE(IMP,390)TPRT
      WRITE(IMP,600)
      END IF

C
```

```
BEM24760
BEM24770
BEM24780
BEM24790
BEM24800
BEM24810
BEM24820
BEM24830
BEM24840
BEM24850
BEM24860
BEM24870
BEM24880
BEM24890
BEM24900
BEM24910
BEM24920
BEM24930
BEM24940
BEM24950
BEM24960
BEM24970
BEM24980
BEM24990
BEM25000
BEM25010
BEM25020
BEM25030
BEM25040
BEM25050
BEM25060
BEM25070
BEM25080
BEM25090
BEM25100
BEM25110
BEM25120
BEM25130
BEM25140
BEM25150
BEM25160
BEM25170
BEM25180
BEM25190
BEM25200
BEM25210
BEM25220
BEM25230
BEM25240
BEM25250
BEM25260
BEM25270
BEM25280
BEM25290
BEM25300
```

```

C
320 FORMAT(1X,'IMAGINARY PART')
200 FORMAT(4(4X,E14.5))
100 FORMAT(1X,'BOUNDARY ELEMENTS',//
*,10X,'X',17X,'Y',15X,'PRESSURE',5X,'PRESSURE DERIVATIVE')
300 FORMAT(1X,'REAL PART')
330 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON RESERVOIR SURFACE'
*)
340 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON DAM FACE')
350 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON BOTTOM')
360 FORMAT(1X,'PRESSURE AND PRESSURE DERIVATIVE ON FAR END')
370 FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, REAL PART',E14.5)
371 FORMAT(1X,'HXR =',E14.5)
380 FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, IMAGINARY PART',E14.5)
381 FORMAT(1X,'HXI =',E14.5)
390 FORMAT(1X,'TOTAL PRESSURE ON DAM FACE, ABSOLUTE VALUE',E14.5)
500 FORMAT(1X,70('*'))
600 FORMAT(1X,' ')
      RETURN
      END
C
C -----
C
SUBROUTINE ADDUND(H,G,HOLD,AFR,AKR,AKI,NO,N1,N2,DFI,FID,FIB,
*FLAG,DF)
*****
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND
COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2
REAL AFR(ND,ND),AKR(ND),AKI(ND)
DOUBLE PRECISION H(NU,NU),G(NU,NU),T(100,20)
DOUBLE PRECISION FID(NU),FIB(NU),DFI(NU),HOLD(NU,NU)
REAL U(20,20),V(20,20)
REAL TEMP1(20,20),TEMP2(20,20)
REAL SUM3,DXR(20)
REAL UT(20,20),VT(20,20),AFRT(20,20),TP(20,20)
LOGICAL FLAG
C
C REARRANGES H AND G MATRICES TO TAKE THE BASE AND FAR BOUNDARY
C CONDITIONS INTO ACCOUNT
C
C ARRANGE H MATRIX
C REAL PART
C
      IF(ME.GT.0)THEN
        DO 20 J=1,ME
          DO 10 I=1,ME
            U(I,J)=AFR(I,J)*AKR(J)
          10 CONTINUE
        20 CONTINUE
C
      CALL TRANSP(TP)
C
      DO 21 I=1,ME

```

```

BEM25310
BEM25320
BEM25330
BEM25340
BEM25350
BEM25360
BEM25370
BEM25380
BEM25390
BEM25400
BEM25410
BEM25420
BEM25430
BEM25440
BEM25450
BEM25460
BEM25470
BEM25480
BEM25490
BEM25500
BEM25510
BEM25520
BEM25530
BEM25540
BEM25550
BEM25560
BEM25570
BEM25580
BEM25590
BEM25600
BEM25610
BEM25620
BEM25630
BEM25640
BEM25650
BEM25660
BEM25670
BEM25680
BEM25690
BEM25700
BEM25710
BEM25720
BEM25730
BEM25740
BEM25750
BEM25760
BEM25770
BEM25780
BEM25790
BEM25800
BEM25810
BEM25820
BEM25830
BEM25840
BEM25850

```

DO 22 J=1,ME	BEM25860
TEM1=0.0	BEM25870
DO 23 K=1,ME	BEM25880
TEM1=TEM1+TP(I,K)*U(K,J)	BEM25890
23 CONTINUE	BEM25900
UT(I,J)=TEM1	BEM25910
22 CONTINUE	BEM25920
21 CONTINUE	BEM25930
C	BEM25940
IF(FLAG)THEN	BEM25950
DO 40 J=1,ME	BEM25960
DO 30 I=1,ME	BEM25970
V(I,J)=AFR(I,J)*AKI(J)	BEM25980
30 CONTINUE	BEM25990
40 CONTINUE	BEM26000
C	BEM26010
DO 41 I=1,ME	BEM26020
DO 42 J=1,ME	BEM26030
TEM1=0.0	BEM26040
DO 43 K=1,ME	BEM26050
TEM1=TEM1+TP(I,K)*V(K,J)	BEM26060
43 CONTINUE	BEM26070
VT(I,J)=TEM1	BEM26080
42 CONTINUE	BEM26090
41 CONTINUE	BEM26100
END IF	BEM26110
C	BEM26120
C	BEM26130
DO 51 I=1,ME	BEM26140
DO 52 J=1,ME	BEM26150
TEM1=0.0	BEM26160
DO 53 K=1,ME	BEM26170
TEM1=TEM1+TP(I,K)*AFR(K,J)	BEM26180
53 CONTINUE	BEM26190
AFRT(I,J)=TEM1	BEM26200
52 CONTINUE	BEM26210
51 CONTINUE	BEM26220
C	BEM26230
DO 70 I=1,N	BEM26240
DO 60 J=1,ME	BEM26250
SUM1=0	BEM26260
SUM2=0	BEM26270
DO 50 K=1,ME	BEM26280
KK=K+N2	BEM26290
SUM1=SUM1+H(I,KK)*DBLE(AFRT(K,J))	BEM26300
SUM2=SUM2+G(I,KK)*DBLE(UT(K,J))	BEM26310
50 CONTINUE	BEM26320
T(I,J)=SUM1+SUM2	BEM26330
60 CONTINUE	BEM26340
70 CONTINUE	BEM26350
C	BEM26360
DO 75 I=1,N	BEM26370
DO 72 J=1,ME	BEM26380
JJ=J+N2	BEM26390
H(I,JJ)=T(I,J)	BEM26400

```
72 CONTINUE
75 CONTINUE
C
C IMAGINARY PART
C
  IF(FLAG)THEN
    DO 120 I=1,N
      DO 110 J=1,ME
        SUM1=0
        DO 100 K=1,ME
          KK=K+N2
          SUM1=SUM1+G(I,KK)*DBLE(VT(K,J))
100 CONTINUE
        T(I,J)=SUM1
110 CONTINUE
120 CONTINUE
C
    DO 140 I=1,N
      DO 130 J=1,ME
        H(N+I,J+N2)=T(I,J)
130 CONTINUE
140 CONTINUE
C
C FILL IN THE REST OF H MATRIX
C
C
  DO 160 I=1,N
    DO 150 J=1,N
      H(I,J+N)=-H(I+N,J)
      H(I+N,J+N)=H(I,J)
150 CONTINUE
160 CONTINUE
    END IF
    END IF
C
C FORM RIGHT HAND VECTOR
C
  DO 180 I=1,N
    DO 170 J=N0+1,N1
      DFI(I)=DFI(I)+G(I,J)*FID(J-N0)
170 CONTINUE
180 CONTINUE
C
  DO 200 I=1,N
    DO 190 J=N1+1,N2
      DFI(I)=DFI(I)+G(I,J)*FIB(J-N1)
190 CONTINUE
200 CONTINUE
C
  IF(DF.GT.0.0) THEN
    DO 210 I=1,ME
      DO 220 J=1,ME
        IF(AKI(J).GT.0.0) THEN
          TEMP1(I,J)=-AFRT(I,J)/(AKI(J)*AKI(J))
        ELSE
          BEM26410
          BEM26420
          BEM26430
          BEM26440
          BEM26450
          BEM26460
          BEM26470
          BEM26480
          BEM26490
          BEM26500
          BEM26510
          BEM26520
          BEM26530
          BEM26540
          BEM26550
          BEM26560
          BEM26570
          BEM26580
          BEM26590
          BEM26600
          BEM26610
          BEM26620
          BEM26630
          BEM26640
          BEM26650
          BEM26660
          BEM26670
          BEM26680
          BEM26690
          BEM26700
          BEM26710
          BEM26720
          BEM26730
          BEM26740
          BEM26750
          BEM26760
          BEM26770
          BEM26780
          BEM26790
          BEM26800
          BEM26810
          BEM26820
          BEM26830
          BEM26840
          BEM26850
          BEM26860
          BEM26870
          BEM26880
          BEM26890
          BEM26900
          BEM26910
          BEM26920
          BEM26930
          BEM26940
          BEM26950
```

```

      TEMP1(I,J)=AFRT(I,J)/(AKR(J)*AKR(J))      BEM26960
      END IF      BEM26970
220 CONTINUE      BEM26980
210 CONTINUE      BEM26990
      DO 230 I=1,ME      BEM27000
      DO 240 J=1,ME      BEM27010
      SUM3=0.0      BEM27020
      DO 250 K=1,ME      BEM27030
      SUM3=SUM3+TEMP1(J,K)*AFR(I,K)      BEM27040
250 CONTINUE      BEM27050
      TEMP2(J,I)=SUM3      BEM27060
240 CONTINUE      BEM27070
230 CONTINUE      BEM27080
      DO 260 I=1,ME      BEM27090
      DXR(I)=-TEMP2(I,1)*SNGL(DF)      BEM27100
260 CONTINUE      BEM27110
      DO 290 I=1,N      BEM27120
      DO 300 J=1,ME      BEM27130
      K1=J-ME+N      BEM27140
      DFI(I)=DFI(I)+HOLD(I,K1)*DBLE(DXR(J))      BEM27150
300 CONTINUE      BEM27160
290 CONTINUE      BEM27170
      END IF      BEM27180
      RETURN      BEM27190
      END      BEM27200
C      BEM27210
C      -----BEM27220
C      BEM27230
      SUBROUTINE TRANSP(TP)      BEM27240
C      *****BEM27250
C      BEM27260
C TRANSP COMPUTES THE MATRIX OF TRANSFORMATION TP      BEM27270
C FOR ADJUSTMENT OF LINEARLY FORMED FAR END CONDITION      BEM27280
C INTO CONSTANT ELEMENT FORMULATION      BEM27290
C      BEM27300
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)      BEM27310
      COMMON/MAIN/ CONST,N,LEC,IMP,NU,ND      BEM27320
      COMMON/ADD/ HY,PI,ME,MF,CONST1,CONST2      BEM27330
      REAL TP(20,20)      BEM27340
C      BEM27350
      DO 10 I=1,ME      BEM27360
      DO 20 J=1,ME      BEM27370
      TP(I,J)=0.0      BEM27380
20 CONTINUE      BEM27390
10 CONTINUE      BEM27400
C      BEM27410
      DO 30 I=1,ME-1      BEM27420
      N1=I+1      BEM27430
      DO 40 J=I,N1      BEM27440
      TP(I,J)=0.5      BEM27450
40 CONTINUE      BEM27460
30 CONTINUE      BEM27470
      TP(ME,ME)=0.5      BEM27480
*      DO 50 I=1,ME      BEM27490
*      WRITE(6,*)(TP(I,J),J=1,ME)      BEM27500

```


* 50 CONTINUE
RETURN
END

C

C

C***** THIS IS THE LAST CARD OF THE CONSTANT BEM PROGRAM*****
BEM27510
BEM27520
BEM27530
BEM27540
BEM27550
BEM27560
BEM27570
BEM27580
BEM27590
BEM27600
BEM27610
BEM27620
BEM27630
BEM27640

B.3 Program FOURTN FORTRAN A1

```

RRRRRRRRRR MM MM
RRRRRRRRRR MMM MMM
RR RR MMM MMMM
RR RR MM MM MM MM
RR RR MM MMMM MM
RRRRRRRRRR MM MM MM
RRRRRRRRRR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM

```

```

11 5555555555 11
111 5555555555 111
1111 55 1111
11 55 11
11 55 11
11 55555555 11
11 5555555555 11
11 55 11
11 55 11
11 55 11
1111111111 5555555555 1111111111
1111111111 5555555555 1111111111

```

```

MM MM
MMM MMM
MMMM MMMM
MM MM MM MM
MM MMMM MM
MM MM MM
MM MM MM
MM MM
MM MM
MM MM
MM MM
MM MM
MM MM
MM MM

```

```

222222222 00000000
22222222222 0000000000
22 22 00 0000
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 0000 00
22 000 00
22222222222 0000000000
22222222222 00000000

```

```

JJJJJJJJJ AAAAAAAAAA BBBB BBBB LL
JJJJJJJJJ AAAAAAAAAA BBBB BBBB LL
JJ AA AA BB BB LL
JJ AA AA BB BB LL
JJ AA AA BB BB LL
JJ AAAAAAAAAA BBBB BBBB LL
JJ AAAAAAAAAA BBBB BBBB LL
JJ AA AA BB BB LL
JJ JJ AA AA BB BB LL
JJ JJ AA AA BB BB LL
JJJJJJJ AA AA BBBB BBBB LLLLLLLLLL
JJJJJ AA AA BBBB BBBB LLLLLLLLLL

```

```

00000000000 NN NN SSSSSSSSS KK KK
00000000000 NNN NN SSSSSSSSSSS KK KK
00 00 NNNN NN SS SS KK KK
00 00 NN NN NN SS KK KK
00 00 NN NN NN SSSSSSSS KKKKKK
00 00 NN NN NN SSSSSSSS KKKKKK
00 00 NN NN NN SSS KK KK
00 00 NN NNNN SS KK KK
00 00 NN NNN SS SS KK KK
00000000000 NN NN SSSSSSSSSSS KK KK
00000000000 NN N SSSSSSSSS KK KK

```

```

RRRRRRRRRR SSSSSSSSS CCCCCCCCC SSSSSSSSS
RRRRRRRRRR SSSSSSSSSSS CCCCCCCCCC SSSSSSSSSSS
RR RR SS SS CC CC SS SS
RR RR SS CC SS
RR RR SSS CC SSS
RRRRRRRRRR SSSSSSSS CC SSSSSSSS
RRRRRRRRRR SSSSSSSS CC SSSSSSSS
RR RR SSS CC SSS
RR RR SS CC SS
RR RR SS CC CC SS SS
RR RR SSSSSSSSS CCCCCCCCCC SSSSSSSSSSS
RR RR SSSSSSSSS CCCCCCCCC SSSSSSSSS

```

```

11 00000000 11 11
111 0000000000 111 111
1111 00 0000 1111 1111
11 00 00 00 11 11
11 00 00 00 11 11
11 00 00 00 11 11
11 00 00 00 11 11
11 00 00 00 11 11
11 0000 00 11 11
11 000 00 11 11
1111111111 0000000000 1111111111 1111111111
1111111111 00000000 1111111111 1111111111

```

DATE/TIME= SEP 18, 1990 16:15:46
 USER ACCOUNT= ALEXNMP
 NODES D:\E\C=NRCVM01 \NRCVM01
 JOB = J1011 START

| PRINT: CLASS= E | DR. A.JABLONSKI
 | FORMS= S1PT | RM. 151
 | FCB= STD8 | M-20
 | UCS= GT15 |

```

C PROGRAM FOURTN FOU00010
C FOR CALCULATION OF THE RESPONSE OF THE LINEAR FOU00020
C DAM-RESERVOIR-FOUNDATION SYSTEM FOU00030
C TO THE EXCITATION HISTORY GIVEN IN THE DISCRETE FOU00040
C FORM A(NT*DT) FOU00050
C BY MEANS OF THE FFT TECHNIQUE FOU00060
C FOU00070
C PROGRAMMER: ALEXANDER M. JABLONSKI FOU00080
C FOU00090
C DATE: APRIL 25, 1989 FOU00100
C SUBSTANTIALLY MODIFIED JULY 11, 1989 FOU00110
C FOU00120
C CENTER: STRUCTURES SECTION FOU00130
C INSTITUTE FOR RESEARCH IN CONSTRUCTION FOU00140
C NATIONAL RESEARCH COUNCIL, CANADA FOU00150
C FOU00160
C NOTE: THIS PROGRAM IS BASED ON THE THREE PROGRAMS: FOU00170
C FOUR21 - INITIAL VERSION OF THE FFT PROGRAM; FOU00180
C ACCEL - READING A PROCESSED EARTHQUAKE DATA AND CONVERTING FOU00190
C IT INTO INPUT DATA (ACCELERATION VECTOR) FOU00200
C INPUTF - READING AN OUTPUT DATA FILE FROM THE CONSTANT FOU00210
C BOUNDARY ELEMENT PROGRAM BEMC2DN AND CONVERTING FOU00220
C IT INTO INPUT DATA (RESPONSE TO THE HARMONIC FOU00230
C EXCITATION) FOU00240
C PROGRAM FOURTN EMPLOYS THE FFT ROUTINES ORIGINALLY DEVELOPED FOU00250
C BY PAUL SWARZTRAUBER, NATIONAL CENTER FOR ATMOSPHERIC RESEARCH, FOU00260
C DENVER, COLORADO, USA, AND IN THE FORM PROVIDED BY THE NEW VERSION FOU00270
C 1.0 OF THE MATH LIBRARY, IMSL. FOU00280
C FOU00290
C PROGRAM FOURTN FOU00300
C PARAMETER (NT=4096) FOU00310
C PARAMETER (NC=200) FOU00320
C PARAMETER (ND=10) FOU00330
C NT = NUMBER OF STEPS TO BE EMPLOYED IN THE FFT FOU00340
C NC = INITIAL NUMBER OF FREQUENCY STEPS EMPLOYED TO CALCULATE FOU00350
C HARMONIC RESPONSE FOU00360
C ND = NUMBER OF NODES AT DAM FACE INTRODUCED IN BEMC2DN PROGRAM FOU00370
C AC(I) = PROCESSED ACCELERATION VECTOR FROM EARTHQUAKE RECORDS FOU00380
C A(I) = MODIFIED ACCELERATION VECTOR - INPUT TO THE FORWARD FFT FOU00390
C ROUTINE FOU00400
C TX(I) = VECTOR OF TOTAL HYDRODYNAMIC FORCE (I=1,NC) FOU00410
C Y1(I) = MODIFIED VECTOR OF TOTAL HYDRODYNAMIC FORCE (I=1,NT) FOU00420
C YC(I,J) = HYDRODYNAMIC PRESSURE MATRIX (I=1,ND, J=1,NC) FOU00430
C YC1(I,J)= MODIFIED PRESSURE MATRIX (I=1,ND, J=1,NT) FOU00440
C FOU00450
C COMPLEX AC(NT), YC(ND,NC), TX(NC), HX(NC) FOU00460
C COMPLEX YC1(ND,NT), Y1(NT) FOU00470
C COMPLEX Y2(NT), Y2N(NT), AN(NT) FOU00480
C REAL A(NT), RWSKP(24611) FOU00490
C REAL Y1R(NT),Y1I(NT) FOU00500
C FOU00510
C FOU00520
C CHARACTER*22 INPS1,INFT1,OUTFFT FOU00530
C FOU00540
C COMMON/WORKSP/ RWSKP FOU00550

```

DATA INPS1/'/INPS1 INPUT A1'/	FOU00560
DATA INFT1/'/INFT1 INPUT A1'/	FOU00570
DATA OUTFFT/'/OUTFFT OUTPUT A1'/	FOU00580
C	FOU00590
OPEN(2,FORM='FORMATTED',STATUS='OLD',FILE=INPS1)	FOU00600
OPEN(3,FORM='FORMATTED',STATUS='OLD',FILE=INFT1)	FOU00610
OPEN(5,FORM='FORMATTED',STATUS='NEW',FILE=OUTFFT)	FOU00620
C	FOU00630
C READ INPUT DATA FILE CREATED BY PROGRAM ACCEL	FOU00640
C	FOU00650
N=1+(NT/6)	FOU00660
DO 10 J=1,N	FOU00670
NX=(J-1)*6	FOU00680
READ(2,100)(A(I+NX),I=1,6)	FOU00690
10 CONTINUE	FOU00700
100 FORMAT(6F12.3)	FOU00710
C	FOU00720
C DIVIDE THE ACCELERATION VECTOR BY 100 TO HAVE IT IN M/SEC**2	FOU00730
C	FOU00740
DO 20 I=1,NT	FOU00750
A(I)=A(I)/100	FOU00760
20 CONTINUE	FOU00770
C	FOU00780
DO 30 I=1,NT	FOU00790
AC(I)=CMPLX(A(I),0.0)	FOU00800
30 CONTINUE	FOU00810
C	FOU00820
C CARRY OUT FORWARD FFT	FOU00830
C	FOU00840
CALL IWKIN(24611)	FOU00850
CALL FFTCF(NT,AC,AN)	FOU00860
C	FOU00870
C	FOU00880
C READ REQUIRED HARMONIC RESPONSES TX, HX, YC	FOU00890
C	FOU00900
C	FOU00910
DO 31 I=1,NT	FOU00920
WRITE(5,*)I,AN(I)	FOU00930
31 CONTINUE	FOU00940
WRITE(5,300)	FOU00950
300 FORMAT(//,1X,70('*'),//)	FOU00960
C	FOU00970
DO 40 I=1,NC	FOU00980
READ(3,*)I, TX(I), HX(I)	FOU00990
READ(3,300)	FOU01000
DO 50 J=1,ND	FOU01010
READ(3,*)YC(J,I)	FOU01020
50 CONTINUE	FOU01030
40 CONTINUE	FOU01040
C	FOU01050
C MODIFY HARMONIC RESPONSE TO HAVE IT IN THE FORM READY	FOU01060
C FOR FFT ROUTINE	FOU01070
C	FOU01080
DO 60 I=1,NT	FOU01090
IF(I.GT.NC) THEN	FOU01100

```
      Y1(I)=(0.0,0.0)
      GO TO 60
      END IF
C
C   NORMALIZATION OF THE VECTOR Y1(I) W.R.T. HYDROSTATIC FORCE
C   IN THIS CASE FOR H=100 M (HEIGHT OG THE DAM)
C
      Y1(I)=TX(I)/5000
60  CONTINUE
C
      DO 61 I=1,NT
      Y1R(I)=REAL(Y1(I))
      Y1I(I)=AIMAG(Y1(I))
61  CONTINUE
C
      DO 62 I=1,NT/2
      Y1R(I+NT/2)=Y1R(NT/2-I+1)
62  CONTINUE
C
      DO 63 I=1,NT/2
      Y1I(I+NT/2)=-Y1I(NT/2-I+1)
63  CONTINUE
C
      DO 64 I=1,NT
      Y1(I)=CMPLX(Y1R(I),Y1I(I))
64  CONTINUE
C
      DO 65 I=1,NT
      WRITE(5,*)I,Y1(I)
65  CONTINUE
      WRITE(5,300)
C
      DO 70 I=1,NT
      DO 80 J=1,ND
      IF(I.GT.NC) THEN
      YC1(J,I)=(0.0,0.0)
      GO TO 80
      END IF
      YC1(J,I)=YC(J,I)/100
80  CONTINUE
70  CONTINUE
C
C   CARRY OUT INVERSE FFT FOR A CHOSEN HARMONIC RESPONSE VECTOR
C
      DO 90 I=1,NT
      Y2(I)=AN(I)*Y1(I)
90  CONTINUE
C
C   CARRY INVERSE FFT
C
      CALL FFTCB(NT,Y2,Y2N)
C
      DO 101 I=1,NT
      Y2N(I)=Y2N(I)/NT
101 CONTINUE
```

```
FOU01110
FOU01120
FOU01130
FOU01140
FOU01150
FOU01160
FOU01170
FOU01180
FOU01190
FOU01200
FOU01210
FOU01220
FOU01230
FOU01240
FOU01250
FOU01260
FOU01270
FOU01280
FOU01290
FOU01300
FOU01310
FOU01320
FOU01330
FOU01340
FOU01350
FOU01360
FOU01370
FOU01380
FOU01390
FOU01400
FOU01410
FOU01420
FOU01430
FOU01440
FOU01450
FOU01460
FOU01470
FOU01480
FOU01490
FOU01500
FOU01510
FOU01520
FOU01530
FOU01540
FOU01550
FOU01560
FOU01570
FOU01580
FOU01590
FOU01600
FOU01610
FOU01620
FOU01630
FOU01640
FOU01650
```

C		FOU01660
	DO 110 I=1,NT	FOU01670
	WRITE(5,*)Y2N(I)	FOU01680
110	CONTINUE	FOU01690
C		FOU01700
	STOP	FOU01710
	END	FOU01720
C		FOU01730
C	END OF THE FOURTN PROGRAM	FOU01740
		FOU01750

RRRRRRRRRR	SSSSSSSSSS	CCCCCCCCCC	SSSSSSSSSS	11	00000000	11	11
RRRRRRRRRR	SSSSSSSSSS	CCCCCCCCCC	SSSSSSSSSS	111	000000000	111	111
RR RR	SS SS	CC CC	SS SS	1111	00 0000	1111	1111
RR RR	SS	CC	SS	11	00 00 00	11	11
RR RR	SSS	CC	SSS	11	00 00 00	11	11
RRRRRRRRRR	SSSSSSSS	CC	SSSSSSSS	11	00 00 00	11	11
RRRRRRRRRR	SSSSSSSS	CC	SSSSSSSS	11	00 00 00	11	11
RR RR	SSS	CC	SSS	11	00 00 00	11	11
RR RR	SS	CC	SS	11	0000 00	11	11
RR RR	SS SS	CC CC	SS SS	11	000 00	11	11
RR RR	SSSSSSSSSS	CCCCCCCCCC	SSSSSSSSSS	111111111	000000000	111111111	111111111
RR RR	SSSSSSSS	CCCCCCCC	SSSSSSSS	111111111	00000000	111111111	111111111

EEEEEEEEEE	NN	NN	DDDDDDDD	11	00000000	11	11
EEEEEEEEEE	NNN	NN	DDDDDDDD	111	000000000	111	111
EE	NNNN	NN	DD DD	1111	00 0000	1111	1111
EE	NN NN	NN	DD DD	11	00 00 00	11	11
EE	NN NN	NN	DD DD	11	00 00 00	11	11
EEEEEEEE	NN NN	NN	DD DD	11	00 00 00	11	11
EEEEEEEE	NN NN	NN	DD DD	11	00 00 00	11	11
EE	NN NN NN	NN	DD DD	11	00 00 00	11	11
EE	NN NN NN	NN	DD DD	11	0000 00	11	11
EE	NN NN NN	NN	DD DD	11	000 00	11	11
EEEEEEEEEE	NN	NN	DDDDDDDD	111111111	000000000	111111111	111111111
EEEEEEEEEE	NN	N	DDDDDDDD	111111111	00000000	111111111	111111111

B.4 Program SPLOTN2 FORTRA A1

```

RRRRRRRRRR MM MM
RRRRRRRRRR MMM MMM
RR RR MMM MMM
RR RR MM MM MM MM
RR RR MM MMM MM
RRRRRRRRRR MM MM
RRRRRRRRRR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM

```

```

11 5555555555 11
111 5555555555 111
1111 55 1111
11 55 11
11 55 11
11 55555555 11
11 5555555555 11
11 55 11
11 55 11
11 55 11
1111111111 5555555555 1111111111
1111111111 5555555555 1111111111

```

```

MM MM
MMM MMM
MMMM MMMM
MM MM MM MM
MM MMMM MM
MM MM MM
MM MM
MM MM
MM MM
MM MM
MM MM
MM MM

```

```

222222222 00000000
22222222222 0000000000
22 22 00 0000
22 00 00 00
22 00 00 00
22 00 00 00
22 00 00 00
22 00 00 00
22 0000 00
22 000 00
22222222222 0000000000
22222222222 00000000

```

```

JJJJJJJJJ AAAAAAAAAA B3B3B3B3B3 LL 0000000000 NN NN SSSSSSSSS KK KK
JJJJJJJJJ AAAAAAAAAA B3B3B3B3B3 LL 00000000000 NN NN SSSSSSSSSSS KK KK
JJ AA AA BB LL 00 00 NN NN NN SS KK KK
JJ AA AA BB LL 00 00 NN NN NN SS KK KK
JJ AA AA B3 BB LL 00 00 NN NN NN SSS KK KK
JJ AAAAAAAAAA B3B3B3B3 LL 00 00 NN NN NN SSSSSSSSS KKKKKKK
JJ AAAAAAAAAA B3B3B3B3 LL 00 00 NN NN NN SSSSSSSSS KKKKKKK
JJ AA BB BB LL 00 00 NN NN NN SSS KK KK
JJ JJ AA AA BB BB LL 00 00 NN NN NN SS KK KK
JJ JJ AA AA BB BB LL 00 00 NN NN NN SS KK KK
JJJJJJJ AA AA B3B3B3B3B3 LLLLLLLLLLLL 0000000000 NN NN SSSSSSSSSSS KK KK
JJJJJ AA AA B3B3B3B3B3 LLLLLLLLLLLL 0000000000 NN N SSSSSSSSS KK KK

```

```

RRRRRRRRRR SSSSSSSSS CCCCCCCCC SSSSSSSSS 7777777777 00000000 7777777777 999999999
RRRRRRRRRR SSSSSSSSS CCCCCCCCC SSSSSSSSS 7777777777 0000000000 7777777777 9999999999
RR RR SS SS CC CC SS SS 77 77 00 0000 77 77 99 99
RR RR SS CC SS 77 00 00 00 77 99 99
RR RR SSS CC SSS 77 00 00 00 77 99 99
RRRRRRRRRR SSSSSSSS CC SSSSSSSS 77 00 00 00 77 9999999999
RRRRRRRRRR SSSSSSSS CC SSSSSSSS 77 00 00 00 77 9999999999
RR RR SSS CC SSS 77 00 00 00 77 99
RR RR SS CC SS 77 0000 00 77 99
RR RR SS CC CC SS SS 77 000 00 77 99 99
RR RR SSSSSSSSS CCCCCCCCC SSSSSSSSS 77 0000000000 77 9999999999
RR RR SSSSSSSSS CCCCCCCCC SSSSSSSSS 77 00000000 77 999999999

```

```

DATE/TIME= AUG 09, 1990 16:20:04 | PRINT: CLASS= E | DR. A.JABLONSKI
USER ACCOUNT= ALEXNMP | FORMS= SIPT | RM. 151
NODES O\EXC=NRCVM01 \NRCVM01 \NRCVM01 | FCB= STD8 | M-20
JOB = J7079 START | UCS= GT15 |

```

```

C   PROGRAM SPLOTN                                SPL00010
C   CALLS SET OF DISSPLA ROUTINES TO CREATE SET OF PLOTS    SPL00020
C                                           SPL00030
C   PROGRAMMER: ALEXANDER M. JABLONSKI              SPL00040
C                                           SPL00050
C   DATE: JULY 12, 1989                            SPL00060
C   MODIFIED: JANUARY 31, 1990                      SPL00070
C   MODIFIED: APRIL 18, 1990                      SPL00080
C                                           SPL00090
C   CENTER: STRUCTURES SECTION                    SPL00100
C           INSTITUTE FOR RESEARCH IN CONSTRUCTION    SPL00110
C           NATIONAL RESEARCH COUNCIL, CANADA        SPL00120
C                                           SPL00130
C   PROGRAM SPLOTN                                SPL00140
C                                           SPL00150
C   IT PRODUCES THE FOLLOWING PLOTS:              SPL00160
C   1. ACCELERATION RECORD                        SPL00170
C   2. REAL PART OF DFT OF ACCEL.                 SPL00180
C   3. IMAG PART OF DFT OF ACCEL.                 SPL00190
C   4. REAL PART OF HARMONIC RESPONSE              SPL00200
C   5. IMAG PART OF HARMONIC RESPONSE              SPL00210
C   6. FINAL EARTHQUAKE RESPONSE                  SPL00220
C                                           SPL00230
C   PARAMETER (NT=4096)                          SPL00240
C   PARAMETER (NE=4096)                          SPL00250
C   COMPLEX AN(NE),Y2N(NE),Y1(NE)                 SPL00260
C   REAL X(NE),Y(NE),A(NE),AA(NE)                 SPL00270
C   REAL X1(NE),X2(NE),X3(NE)                     SPL00280
C   REAL ANR(NE),ANI(NE),Y2NR(NE),Y2NI(NE)        SPL00290
C   REAL Y1R(NE),Y1I(NE),Z(NE)                   SPL00300
C   CHARACTER*22 INPS1,OUTFFT,SUM1                SPL00310
C                                           SPL00320
C   DATA INPS1/'/INPS1 INPUT A1'/                SPL00330
C   DATA OUTFFT/'/OUTFFT OUTPUT A1'/              SPL00340
C   DATA SUM1/'/SUM1 OUTPUT A1'/                  SPL00350
C   OPEN(2,FORM='FORMATTED',STATUS='OLD',FILE=INPS1) SPL00360
C   OPEN(3,FORM='FORMATTED',STATUS='OLD',FILE=OUTFFT) SPL00370
C   OPEN(4,FORM='FORMATTED',STATUS='NEW',FILE=SUM1) SPL00380
C                                           SPL00390
C   CALL NOMDEV(1)                                SPL00400
C                                           SPL00410
C   PI=3.141592653                                SPL00420
C   DO 10 I=1,NE                                  SPL00430
C     X(I)=REAL(I)                                SPL00440
C   10 CONTINUE                                    SPL00450
C                                           SPL00460
C   N=1+(NT/6)                                     SPL00470
C   DO 20 J=1,N                                    SPL00480
C     NX=(J-1)*6                                   SPL00490
C     READ(2,100)(A(I+NX),I=1,6)                  SPL00500
C   20 CONTINUE                                    SPL00510
C   100 FORMAT(6F12.3)                             SPL00520
C                                           SPL00530
C   DO 30 I=1,NE                                  SPL00540
C     IF(I.GT.NT) THEN                             SPL00550

```

```
      Y(I)=0.0
      END IF
      Y(I)=(REAL(A(I)))/100.
30  CONTINUE
C
      DO 399 I=1,NE
      X1(I)=0.02*X(I)
399  CONTINUE
C
      CALL PLOT1(NE,X1,Y)
C
      DO 31 I=1,NE
      READ(3,*)I,AN(I)
31  CONTINUE
      READ(3,300)
300  FORMAT(//,1X,70(' '),//)
C
      DO 32 I=1,NE
      ANR(I)=REAL(AN(I))
      ANI(I)=IMAG(AN(I))
32  CONTINUE
C
C
      CALL PLOT2(NE,X,ANR)
C
      CALL PLOT3(NE,X,ANI)
C
C
      DO 40 I=1,NE
      READ(3,*)I,Y1(I)
40  CONTINUE
      READ(3,300)
C
      DO 41 I=1,NE
      Y1R(I)=REAL(Y1(I))
      Y1I(I)=IMAG(Y1(I))
      Z(I)=SQRT(Y1I(I)**2+Y1R(I)**2)
41  CONTINUE
C
      NX=200
      DO 410 I=1,NX
      X2(I)=(0.3068*X(I))/(2*PI)
410  CONTINUE
C
      CALL PLOT4(NX,X2,Y1R)
C
      CALL PLOT5(NX,X2,Y1I)
C
      CALL PLOT51(NX,X2,Z)
C
C
      DO 33 I=1,NE
      AA(I)=SQRT(ANR(I)**2+ANI(I)**2)
33  CONTINUE
C
```

```
SPL00560
SPL00570
SPL00580
SPL00590
SPL00600
SPL00610
SPL00620
SPL00630
SPL00640
SPL00650
SPL00660
SPL00670
SPL00680
SPL00690
SPL00700
SPL00710
SPL00720
SPL00730
SPL00740
SPL00750
SPL00760
SPL00770
SPL00780
SPL00790
SPL00800
SPL00810
SPL00820
SPL00830
SPL00840
SPL00850
SPL00860
SPL00870
SPL00880
SPL00890
SPL00900
SPL00910
SPL00920
SPL00930
SPL00940
SPL00950
SPL00960
SPL00970
SPL00980
SPL00990
SPL01000
SPL01010
SPL01020
SPL01030
SPL01040
SPL01050
SPL01060
SPL01070
SPL01080
SPL01090
SPL01100
```

CALL PLOT31(NE,X,AA)	SPL01110
C	SPL01120
DO 50 I=1,NE	SPL01130
READ(3,*)Y2N(I)	SPL01140
50 CONTINUE	SPL01150
C	SPL01160
DO 51 I=1,NE	SPL01170
Y2NR(I)=REAL(Y2N(I))	SPL01180
Y2NI(I)=IMAG(Y2N(I))	SPL01190
WRITE(4,*)Y2NR(I)	SPL01200
51 CONTINUE	SPL01210
C	SPL01220
DO 510 I=1,NE	SPL01230
X3(I)=0.02*X(I)	SPL01240
510 CONTINUE	SPL01250
C	SPL01260
CALL PLOT6(NE,X3,Y2NR)	SPL01270
C	SPL01280
CALL DONEPL	SPL01290
STOP	SPL01300
END	SPL01310
C	SPL01320
SUBROUTINE PLOT1(N,X,Y)	SPL01330
REAL X(N),Y(N)	SPL01340
RM=0.0	SPL01350
RH=0.0	SPL01360
DO 5 I=1,N	SPL01370
RMAX=MAX(RMAX,Y(I))	SPL01380
RMIN=MIN(RMIN,Y(I))	SPL01390
5 CONTINUE	SPL01400
DO 6 I=1,N	SPL01410
IF(RMAX.EQ.Y(I)) THEN	SPL01420
R1=I*0.02	SPL01430
ELSE IF(RMIN.EQ.Y(I)) THEN	SPL01440
R2=I*0.02	SPL01450
END IF	SPL01460
6 CONTINUE	SPL01470
YMAX=+2.5	SPL01480
YMIN=-2.5	SPL01490
XMIN=0	SPL01500
XMAX=90.0	SPL01510
XSTP=10.0	SPL01520
DO 10 I=1,N	SPL01530
IF(Y(I).GT.YMAX) YMAX=Y(I)	SPL01540
IF(Y(I).LT.YMIN) YMIN=Y(I)	SPL01550
10 CONTINUE	SPL01560
YSTP=0.5	SPL01570
CALL PAGE(11.0,8.5)	SPL01580
CALL NOBRDR	SPL01590
CALL PHYSOR(1.0,1.0)	SPL01600
CALL TRIPLX	SPL01610
CALL TITLE('TAFT SCHOOL HOR.',16,'TIME SEC',8,	SPL01620
*'ACCELERATION M/SEC/SEC',22,9.0,6.5)	SPL01630
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)	SPL01640
CALL MESSAGE('MAX = ',6,4.0,6)	SPL01650

CALL MSGAG('MIN = ',6,4.0,5.5)	SPL01660
CALL REALND(RMAX,+2,4.8,6)	SPL01670
CALL REALND(RMIN,+2,4.8,5.5)	SPL01680
CALL MSGAG('AT X = ',7,5.8,6)	SPL01690
CALL MSGAG('AT X = ',7,5.8,5.5)	SPL01700
CALL REALND(R1,+2,6.6,6)	SPL01710
CALL REALND(R2,+2,6.6,5.5)	SPL01720
CALL STRTPT(0.0,3.25)	SPL01730
CALL CONNPT(9.0,3.25)	SPL01740
CALL CURVE(X,Y,N,0)	SPL01750
CALL ENDPL(0)	SPL01760
RETURN	SPL01770
END	SPL01780
C	SPL01790
SUBROUTINE PLOT2(N,X,Y)	SPL01800
REAL X(N),Y(N)	SPL01810
YMAX=+200.0	SPL01820
YMIN=-200.0	SPL01830
XMIN=0	SPL01840
XMAX=4500	SPL01850
XSTP=500	SPL01860
DO 10 I=1,N	SPL01870
IF(Y(I).GT.YMAX) YMAX=Y(I)	SPL01880
IF(Y(I).LT.YMIN) YMIN=Y(I)	SPL01890
10 CONTINUE	SPL01900
YSTP=50.0	SPL01910
CALL PAGE(11.0,8.5)	SPL01920
CALL NOBRDR	SPL01930
CALL PHYSOR(1.0,1.0)	SPL01940
CALL TITLE('DFT OF ACCEL. - REAL PART',25,	SPL01950
*'STEPS, 1 STEP=0.3068 RAD/S',26,'AMPLITUDE OF ACCELERATION',25,	SPL01960
*9.0,6.5)	SPL01970
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)	SPL01980
CALL STRTPT(0.0,3.25)	SPL01990
CALL CONNPT(9.0,3.25)	SPL02000
CALL CURVE(X,Y,N,0)	SPL02010
CALL ENDPL(0)	SPL02020
RETURN	SPL02030
END	SPL02040
SUBROUTINE PLOT3(N,X,Y)	SPL02050
REAL X(N),Y(N)	SPL02060
YMAX=+200.0	SPL02070
YMIN=-200.0	SPL02080
XMIN=0	SPL02090
XMAX=4500	SPL02100
XSTP=500	SPL02110
DO 10 I=1,N	SPL02120
IF(Y(I).GT.YMAX) YMAX=Y(I)	SPL02130
IF(Y(I).LT.YMIN) YMIN=Y(I)	SPL02140
10 CONTINUE	SPL02150
YSTP=50.0	SPL02160
CALL PAGE(11.0,8.5)	SPL02170
CALL NOBRDR	SPL02180
CALL PHYSOR(1.0,1.0)	SPL02190
CALL TITLE('DFT OF ACCEL. - IMAG. PART',26,	SPL02200

```

*'STEPS, 1 STEP=0.3068 RAD/S',26,'AMPLITUDE OF ACCELERATION',25,
*9.0,6.5)
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)
CALL STRTPT(0.0,3.25)
CALL CONNPT(9.0,3.25)
CALL CURVE(X,Y,N,0)
CALL ENDPL(0)
RETURN
END
SUBROUTINE PLOT4(N,X,Y)
REAL X(N),Y(N)
PI=3.141592653
YMAX=+0.2
YMIN=-0.2
XMIN=0
XMAX=10
XSTP=1
DO 5 I=1,N
  RMAX=MAX(RMAX,Y(I))
  RMIN=MIN(RMIN,Y(I))
5 CONTINUE
DO 6 I=1,N
  IF(RMAX.EQ.Y(I)) THEN
    R1=(I*0.3068)/(2*PI)
  ELSE IF(RMIN.EQ.Y(I)) THEN
    R2=(I*0.3068)/(2*PI)
  END IF
6 CONTINUE
DO 10 I=1,N
  IF(Y(I).GT.YMAX) YMAX=Y(I)
  IF(Y(I).LT.YMIN) YMIN=Y(I)
10 CONTINUE
DO 11 I=2,N
  IF(Y(I).GE.Y(I-1)) GO TO 11
  YM=Y(I)
11 CONTINUE
YSTP=0.05
CALL PAGE(11.0,8.5)
CALL NCBDNR
CALL PHYSOR(1.0,1.0)
CALL TITLE('HARM. RESPONSE - REAL PART',26,
*'FREQUENCY HZ',12,'TOTAL HYDR. FORCE/STAT. FORCE',
*29,9.0,6.5)
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)
CALL MESSAGE('MAX = ',6,4.0,6)
CALL MESSAGE('MIN = ',6,4.0,5.5)
CALL REALND(RMAX,+2,4.8,6)
CALL REALND(RMIN,+2,4.8,5.5)
CALL MESSAGE('AT X = ',7,5.8,6)
CALL MESSAGE('AT X = ',7,5.8,5.5)
CALL REALND(R1,+2,6.6,6)
CALL REALND(R2,+2,6.6,5.5)
CALL STRTPT(0.0,3.25)
CALL CONNPT(9.0,3.25)
CALL CURVE(X,Y,N,0)

```

```

SPL02210
SPL02220
SPL02230
SPL02240
SPL02250
SPL02260
SPL02270
SPL02280
SPL02290
SPL02300
SPL02310
SPL02320
SPL02330
SPL02340
SPL02350
SPL02360
SPL02370
SPL02380
SPL02390
SPL02400
SPL02410
SPL02420
SPL02430
SPL02440
SPL02450
SPL02460
SPL02470
SPL02480
SPL02490
SPL02500
SPL02510
SPL02520
SPL02530
SPL02540
SPL02550
SPL02560
SPL02570
SPL02580
SPL02590
SPL02600
SPL02610
SPL02620
SPL02630
SPL02640
SPL02650
SPL02660
SPL02670
SPL02680
SPL02690
SPL02700
SPL02710
SPL02720
SPL02730
SPL02740
SPL02750

```



```
CALL ENDPL(0)
RETURN
END
C
SUBROUTINE PLOT5(N,X,Y)
REAL X(N),Y(N)
PI=3.141592653
YMAX=+0.2
YMIN=-0.2
XMIN=0
XMAX=10
DO 5 I=1,N
  RMAX=MAX(RMAX,Y(I))
  RMIN=MIN(RMIN,Y(I))
5 CONTINUE
DO 6 I=1,N
  IF(RMAX.EQ.Y(I)) THEN
    R1=(I*0.3068)/(2*PI)
  ELSE IF(RMIN.EQ.Y(I)) THEN
    R2=(I*0.3068)/(2*PI)
  END IF
6 CONTINUE
XSTP=1
DO 10 I=1,N
  IF(Y(I).GT.YMAX) YMAX=Y(I)
  IF(Y(I).LT.YMIN) YMIN=Y(I)
10 CONTINUE
YSTP=0.05
CALL PAGE(11.0,8.5)
CALL NOBRDR
CALL PHYSOR(1.0,1.0)
CALL TITLE('HARM. RESPONSE - IMAG. PART',27,
*'FREQUENCY HZ',12,'TOTAL HYDR. FORCE/STAT. FORCE',
*29,9.0,6.5)
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)
CALL MESSAG('MAX = ',6,4.0,6)
CALL MESSAG('MIN = ',6,4.0,5.5)
CALL REALND(RMAX,+2,4.8,6)
CALL REALND(RMIN,+2,4.8,5.5)
CALL MESSAG('AT X = ',7,5.8,6)
CALL MESSAG('AT X = ',7,5.8,5.5)
CALL REALND(R1,+2,6.6,6)
CALL REALND(R2,+2,6.6,5.5)
CALL STRTPT(0.0,3.25)
CALL CONNPT(9.0,3.25)
CALL CURVE(X,Y,N,0)
CALL ENDPL(0)
RETURN
END
SUBROUTINE PLOT6(N,X,Y)
REAL X(N),Y(N)
YMAX=+0.1
YMIN=-0.1
XMIN=0
XMAX=90.0
```

SPL02760
SPL02770
SPL02780
SPL02790
SPL02800
SPL02810
SPL02820
SPL02830
SPL02840
SPL02850
SPL02860
SPL02870
SPL02880
SPL02890
SPL02900
SPL02910
SPL02920
SPL02930
SPL02940
SPL02950
SPL02960
SPL02970
SPL02980
SPL02990
SPL03000
SPL03010
SPL03020
SPL03030
SPL03040
SPL03050
SPL03060
SPL03070
SPL03080
SPL03090
SPL03100
SPL03110
SPL03120
SPL03130
SPL03140
SPL03150
SPL03160
SPL03170
SPL03180
SPL03190
SPL03200
SPL03210
SPL03220
SPL03230
SPL03240
SPL03250
SPL03260
SPL03270
SPL03280
SPL03290
SPL03300

```
XSTP=10.0
DO 5 I=1,N
  RMAX=MAX(RMAX,Y(I))
  RMIN=MIN(RMIN,Y(I))
5 CONTINUE
DO 6 I=1,N
  IF(RMAX.EQ.Y(I)) THEN
    R1=I*0.02
  ELSE IF(RMIN.EQ.Y(I)) THEN
    R2=I*0.02
  END IF
6 CONTINUE
DO 10 I=1,N
  IF(Y(I).GT.YMAX) YMAX=Y(I)
  IF(Y(I).LT.YMIN) YMIN=Y(I)
10 CONTINUE
YSTP=0.025
CALL PAGE(11.0,8.5)
CALL NC3RDR
CALL PHYSOR(1.0,1.0)
CALL TITLE('TAFT SCHOOL RESPONSE',20,'TIME SEC'
*,8,'TOTAL HYDR. FORCE/STAT. FORCE',29,9.0,6.5)
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)
CALL MESSAGE('MAX = ',6,4.0,6)
CALL MESSAGE('MIN = ',6,4.0,5.5)
CALL REALND(RMAX,+3,4.8,6)
CALL REALND(RMIN,+3,4.8,5.5)
CALL MESSAGE('AT X = ',7,5.8,6)
CALL MESSAGE('AT X = ',7,5.8,5.5)
CALL REALND(R1,+2,6.6,6)
CALL REALND(R2,+2,6.6,5.5)
CALL STRTPT(0.0,3.25)
CALL CONNPT(9.0,3.25)
CALL CURVE(X,Y,N,0)
CALL ENDPL(0)
RETURN
END
SUBROUTINE PLOT31(N,X,Y)
REAL X(N),Y(N)
YMAX=+200.0
YMIN=-200.0
XMIN=0
XMAX=4500
XSTP=500
DO 10 I=1,N
  IF(Y(I).GT.YMAX) YMAX=Y(I)
  IF(Y(I).LT.YMIN) YMIN=Y(I)
10 CONTINUE
YSTP=50.0
CALL PAGE(11.0,8.5)
CALL NC3RDR
CALL PHYSOR(1.0,1.0)
CALL TITLE('DFT OF ACCEL. - ABS. VALUE',26,
*'STEPS, 1 STEP=0.3068 RAD/S',26,'AMPLITUDE OF ACCELERATION',
*25,9.0,6.5)
```

SPL03310
SPL03320
SPL03330
SPL03340
SPL03350
SPL03360
SPL03370
SPL03380
SPL03390
SPL03400
SPL03410
SPL03420
SPL03430
SPL03440
SPL03450
SPL03460
SPL03470
SPL03480
SPL03490
SPL03500
SPL03510
SPL03520
SPL03530
SPL03540
SPL03550
SPL03560
SPL03570
SPL03580
SPL03590
SPL03600
SPL03610
SPL03620
SPL03630
SPL03640
SPL03650
SPL03660
SPL03670
SPL03680
SPL03690
SPL03700
SPL03710
SPL03720
SPL03730
SPL03740
SPL03750
SPL03760
SPL03770
SPL03780
SPL03790
SPL03800
SPL03810
SPL03820
SPL03830
SPL03840
SPL03850

CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)	SPL03860
CALL STRTPT(0.0,3.25)	SPL03870
CALL CONNPT(9.0,3.25)	SPL03880
CALL CURVE(X,Y,N,0)	SPL03890
CALL ENDPL(0)	SPL03900
RETURN	SPL03910
END	SPL03920
SUBROUTINE PLOT51(N,X,Y)	SPL03930
REAL X(N),Y(N)	SPL03940
PI=3.141592653	SPL03950
YMAX=+0.2	SPL03960
YMIN=-0.2	SPL03970
XMIN=0	SPL03980
XMAX=10	SPL03990
XSTP=1	SPL04000
DO 5 I=1,N	SPL04010
RMAX=MAX(RMAX,Y(I))	SPL04020
RMIN=MIN(RMIN,Y(I))	SPL04030
5 CONTINUE	SPL04040
DO 6 I=1,N	SPL04050
IF(RMAX.EQ.Y(I)) THEN	SPL04060
R1=(I*0.3068)/(2*PI)	SPL04070
ELSE IF(RMIN.EQ.Y(I)) THEN	SPL04080
R2=(I*0.3068)/(2*PI)	SPL04090
END IF	SPL04100
6 CONTINUE	SPL04110
DO 10 I=1,N	SPL04120
IF(Y(I).GT.YMAX) YMAX=Y(I)	SPL04130
IF(Y(I).LT.YMIN) YMIN=Y(I)	SPL04140
10 CONTINUE	SPL04150
YSTP=0.05	SPL04160
CALL PAGE(11.0,8.5)	SPL04170
CALL NOBRDR	SPL04180
CALL PHYSOR(1.0,1.0)	SPL04190
CALL TITLE('HARM. RESPONSE - ABS. VALUE',27,	SPL04200
*'FREQUENCY HZ',12,'TOTAL HYDR. FORCE/STAT. FORCE',	SPL04210
*29,9.0,6.5)	SPL04220
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)	SPL04230
CALL MESSAGE('MAX = ',6,4.0,6)	SPL04240
CALL MESSAGE('MIN = ',6,4.0,5.5)	SPL04250
CALL REALND(RMAX,+2,4.8,6)	SPL04260
CALL REALND(RMIN,+2,4.8,5.5)	SPL04270
CALL MESSAGE('AT X = ',7,5.8,6)	SPL04280
CALL MESSAGE('AT X = ',7,5.8,5.5)	SPL04290
CALL REALND(R1,+2,6.6,6)	SPL04300
CALL REALND(R2,+2,6.6,5.5)	SPL04310
CALL STRTPT(0.0,3.25)	SPL04320
CALL CONNPT(9.0,3.25)	SPL04330
CALL CURVE(X,Y,N,0)	SPL04340
CALL ENDPL(0)	SPL04350
RETURN	SPL04360
END	SPL04370

RRRRRRRRRR	SSSSSSSSSS	CCCCCCCC	SSSSSSSSSS	7777777777	00000000	7777777777	9999999999
RRRRRRRRRR	SSSSSSSSSS	CCCCCCCC	SSSSSSSSSS	7777777777	0000000000	7777777777	999999999999
RR RR	SS SS	CC CC	SS SS	77 77	00 0000	77 77	99 99
RR RR	SS	CC	SS	77	00 00 00	77	99 99
RR RR	SSS	CC	SSS	77	00 00 00	77	99 99
RRRRRRRRRR	SSSSSSSSSS	CC	SSSSSSSSSS	77	00 00 00	77	999999999999
RRRRRRRRRR	SSSSSSSSSS	CC	SSSSSSSSSS	77	00 00 00	77	999999999999
RR RR	SSS	CC	SSS	77	00 00 00	77	99 99
RR RR	SS	CC	SS	77	0000 00	77	99 99
RR RR	SS SS	CC CC	SS SS	77	000 00	77	99 99
RR RR	SSSSSSSSSS	CCCCCCCC	SSSSSSSSSS	77	0000000000	77	999999999999
RR RR	SSSSSSSSSS	CCCCCCCC	SSSSSSSSSS	77	00000000	77	9999999999

EEEEEEEEEEEE	NN	NN	DDDDDDDD	7777777777	00000000	7777777777	9999999999
EEEEEEEEEEEE	NNN	NN	DDDDDDDDDD	7777777777	0000000000	7777777777	999999999999
EE	NNN	NN	DD DD	77 77	00 0000	77 77	99 99
EE	NN NN	NN	DD DD	77	00 00 00	77	99 99
EE	NN NN	NN	DD DD	77	00 00 00	77	99 99
EEEEEEEE	NN NN	NN	DD DD	77	00 00 00	77	999999999999
EEEEEEEE	NN NN	NN	DD DD	77	00 00 00	77	999999999999
EE	NN NN	NN	DD DD	77	00 00 00	77	99
EE	NN NN	NN	DD DD	77	0000 00	77	99
EE	NN NN	NN	DD DD	77	000 00	77	99 99
EEEEEEEEEEEE	NN	NN	DDDDDDDDDD	77	0000000000	77	999999999999
EEEEEEEEEEEE	NN	N	DDDDDDDD	77	00000000	77	9999999999

B.5 Program SUM FORTRA A1

```

RRRRRRRRRR MM MM
RRRRRRRRRRR MIM MMM
RR RR MM MM MM MM
RR RR MM MM MM MM
RRRRRRRRRRR MM MM
RRRRRRRRRRR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM
RR RR MM MM

```

```

11 5555555555 11
111 5555555555 111
1111 55 1111
11 55 11
11 55 11
11 55555555 11
11 555555555 11
11 55 11
11 55 11
11 55 11
1111111111 5555555555 1111111111
1111111111 5555555555 1111111111

```

```

MM MM
MMM MMM
MMMM MMMM
MM MM MM MM
MM MIM MM
MM MM MM
MM MM
MM MM
MM MM
MM MM
MM MM
MM MM

```

```

2222222222 00000000
22222222222 0000000000
22 22 00 0000
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 00 00 00
22 22 0000 00
22 22 000 00
222222222222 0000000000
222222222222 00000000

```

```

JJJJJJJJJ AAAAAAAAAA BBBB BBBB LL 000000000000 NN NN SSSSSSSSSS KK KK
JJJJJJJJJ AAAAAAAAAA BBBB BBBB LL 000000000000 NNN NN SSSSSSSSSSSS KK KK
JJ AA AA BB BB LL 00 00 NNNN NN SS SS KK KK
JJ AA AA BB BB LL 00 00 NN NN NN SS KK KK
JJ AA AA BB BB LL 00 00 NN NN NN SSSS KK KK
JJ AAAAAAAAAA BBBB BBBB LL 00 00 NN NN NN SSSSSSSSSS KKKKKKK
JJ AAAAAAAAAA BBBB BBBB LL 00 00 NN NN NN SSSSSSSSSS KKKKKKK
JJ AA AA BB BB LL 00 00 NN NN NN SSS KK KK
JJ JJ AA AA BB BB LL 00 00 NN NNNN SS KK KK
JJ JJ AA AA BB BB LL 00 00 NN NNN SS SS KK KK
JJJJJJJ AA AA BBBB BBBB LLLLLLLLLLLL 000000000000 NN NN SSSSSSSSSSSS KK KK
JJJJJ AA AA BBBB BBBB LLLLLLLLLLLL 000000000000 NN N SSSSSSSSSS KK KK

```

```

RRRRRRRRRRR SSSSSSSSSS CCCCCCCCCC SSSSSSSSSS 77777777777 3333333333 11 00000000
RRRRRRRRRRR SSSSSSSSSSSS CCCCCCCCCC SSSSSSSSSSSS 7777777777 33333333333 111 0000000000
RR RR SS SS CC CC SS SS 77 77 33 1111 00 0000
RR RR SS CC SS 77 33 11 00 00 00
RR RR SSS CC SSS 77 33 11 00 00 00
RRRRRRRRRRR SSSSSSSSSS CC SSSSSSSSSS 77 3333 11 00 00 00
RRRRRRRRRRR SSSSSSSSSS CC SSSSSSSSSS 77 3333 11 00 00 00
RR RR SSS CC SSS 77 33 11 00 00 00
RR RR SS SS CC SS SS 77 33 11 0000 00
RR RR SSSSSSSSSSSS CCCCCCCCCC SSSSSSSSSSSS 77 33333333333 1111111111 0000000000
RR RR SSSSSSSSSS CCCCCCCCCC SSSSSSSSSS 77 3333333333 1111111111 00000000

```

```

DATE/TIME= AUG 13, 1990 12:10:54 | PRINT: CLASS= E | DR. A.JABLONSKI
USER ACCOUNT= ALEXNMP | FORMS= S1PT | RM. 151
NODES O\EXC=NRCVM01 \NRCVM01 \NRCVM01 | FCB= STD8 | M-20
JOB = J7310 START | UCS= GT15 |

```

```
C
C      PROGRAM SUM
C      CALCULATES THE SUM OF EARTHQUAKE RESPONSES
C      TO VERTICAL AND HORIZONTAL COMPONENTS
C      OF GROUND MOTION
C
C      PROGRAMMER: ALEXANDER M. JABLONSKI
C
C      DATE: AUGUST 2, 1989
C      MODIFIED: FEBRUARY 1, 1990
C
C      CENTER: STRUCTURAL SECTION
C              INSTITUTE FOR RESEARCH IN CONSTRUCTION
C              NATIONAL RESEARCH COUNCIL, CANADA
C
C      PROGRAM SUM
C
C      PARAMETER(NT=4096)
C      REAL EH(NT), EV(NT), E(NT), X(NT)
C
C      CHARACTER*22 SUM1H,SUM1V
C      DATA SUM1H/'/SUM1H OUTPUT A1'/
C      DATA SUM1V/'/SUM1V OUTPUT A1'/
C      OPEN(2,FORM='FORMATTED',STATUS='OLD',FILE=SUM1H)
C      OPEN(3,FORM='FORMATTED',STATUS='OLD',FILE=SUM1V)
C
C      CALL NOMDEV(1)
C
C      READ(2,*)(EH(I),I=1,NT)
C      READ(3,*)(EV(I),I=1,NT)
C
C      DO 10 I=1,NT
C      E(I)=EH(I)+EV(I)
C 10 CONTINUE
C
C      DO 20 I=1,NT
C      X(I)= 0.005*REAL(I)
C 20 CONTINUE
C      CALL PLOT1(NT,X,E)
C      CALL DONEPL
C      STOP
C      END
C
C      SUBROUTINE PLOT1(N,X,Y)
C      REAL X(N),Y(N)
C      RMAX=0.0
C      RMIN=0.0
C      DO 15 I=1,N
C      RMAX=MAX(RMAX,Y(I))
C      RMIN=MIN(RMIN,Y(I))
C 15 CONTINUE
C      DO 16 I=1,N
C      IF(RMAX.EQ.Y(I)) THEN
C      I1=I*0.005
```

```
SUM00010
SUM00020
SUM00030
SUM00040
SUM00050
SUM00060
SUM00070
SUM00080
SUM00090
SUM00100
SUM00110
SUM00120
SUM00130
SUM00140
SUM00150
SUM00160
SUM00170
SUM00180
SUM00190
SUM00200
SUM00210
SUM00220
SUM00230
SUM00240
SUM00250
SUM00260
SUM00270
SUM00280
SUM00290
SUM00300
SUM00310
SUM00320
SUM00330
SUM00340
SUM00350
SUM00360
SUM00370
SUM00380
SUM00390
SUM00400
SUM00410
SUM00420
SUM00430
SUM00440
SUM00450
SUM00460
SUM00470
SUM00480
SUM00490
SUM00500
SUM00510
SUM00520
SUM00530
SUM00540
SUM00550
```

ELSE IF(RMIN.EQ.Y(I)) THEN	SUM00560
I2=I*0.005	SUM00570
END IF	SUM00580
16 CONTINUE	SUM00590
YMAX=+0.5	SUM00600
YMIN=-0.5	SUM00610
XMIN=0	SUM00620
XMAX=25.0	SUM00630
XSTP=2.5	SUM00640
DO 10 I=1,N	SUM00650
IF(Y(I).GT.YMAX) YMAX=Y(I)	SUM00660
IF(Y(I).LT.YMIN) YMIN=Y(I)	SUM00670
10 CONTINUE	SUM00680
YSTP=0.10	SUM00690
CALL PAGE(11.0,8.5)	SUM00700
CALL NOBRDR	SUM00710
CALL PHYSOR(1.0,1.0)	SUM00720
CALL TRIPLX	SUM00730
CALL TITLE('MANIC3.1 COMB. SAGUENAY CHICOUTIMI RESPONSE',43,'	SUM00740
&TIME SEC',8,'TOTAL HYDR. FORCE/STAT. FORCE',29,9.0,6.5)	SUM00750
CALL GRAF(XMIN,XSTP,XMAX,YMIN,YSTP,YMAX)	SUM00760
CALL MESSAG('MAX = ',6,4.0,6)	SUM00770
CALL MESSAG('MIN = ',6,4.0,5.5)	SUM00780
CALL REALND(RMAX,+3,4.8,6)	SUM00790
CALL REALND(RMIN,+3,4.8,5.5)	SUM00800
CALL MESSAG('AT X = ',7,5.8,6)	SUM00810
CALL MESSAG('AT X = ',7,5.8,5.5)	SUM00820
CALL INTNO(I1,6.6,6)	SUM00830
CALL INTNO(I2,6.6,5.5)	SUM00840
CALL STRTPT(0.0,3.25)	SUM00850
CALL CONNPT(9.0,3.25)	SUM00860
CALL CURVE(X,Y,N,0)	SUM00870
CALL ENDPL(0)	SUM00880
RETURN	SUM00890
END	SUM00900
C	SUM00910
C THIS IS THE END OF PROGRAM SUM	SUM00920

EEEEEEEEEEEE	NN	NN	DDDDDDDD	7777777777	333333333	11	00000000
EEEEEEEEEEEE	NNN	NN	DDDDDDDDDD	7777777777	3333333333	111	0000000000
EE	NNNN	NN	DD DD	77 77	33 33	1111	00 0000
EE	NN NN	NN	DD DD	77	33	11	00 00 00
EE	NN NN	NN	DD DD	77	33	11	00 00 00
EEEEEEEE	NN NN	NN	DD DD	77	3333	11	00 00 00
EEEEEEEE	NN NN	NN	DD DD	77	3333	11	00 00 00
EE	NN NN	NN	DD DD	77	33	11	00 00 00
EE	NN NNN	NN	DD DD	77	33	11	0000 00
EE	NN NNN	NN	DD DD	77	33 33	11	000 00
EEEEEEEEEEEE	NN NN	NN	DDDDDDDDDD	77	3333333333	1111111111	0000000000
EEEEEEEEEEEE	NN N	NN	DDDDDDDD	77	3333333333	1111111111	00000000

- 106 -