

A11106 053726

NBSIR 85-3234

Dynamic Green's Functions of an Infinite Plate - A Computer Program

ReferenceNBS
PUBLICATIONS

Nelson N. Hsu

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Manufacturing Engineering
Gaithersburg, MD 20899

August 1985

Issued November 1985

Prepared for:

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Gaithersburg, MD 20899

QC
100
.U56
85-3234
1985

NBSIR 85-3234

**DYNAMIC GREEN'S FUNCTIONS OF AN
INFINITE PLATE - A COMPUTER
PROGRAM**

Nelson N. Hsu

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Manufacturing Engineering
Gaithersburg, MD 20899

August 1985

Issued November 1985

Prepared for:
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Gaithersburg, MD 20899



U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Dynamic Green's Functions of an Infinite Plate - a Computer Program

Nelson N. Hsu
National Bureau of Standards
Gaithersburg, MD 20899 USA

Introduction

This report is a FORTRAN program to compute the Green's functions of an infinite plate. The Green's function, $G_{ij}(\underline{\xi}, \underline{x}, t)$, is defined as the i th component of the displacement at $\underline{x}$ due to a point force of step-function time dependency acting at $\underline{\xi}$ in the j th direction initiated at $t=0$. The Green's function is the fundamental solution of the transient elastic wave propagation problem. In general, the displacement field $\underline{u}(\underline{\xi}, \underline{x}, t)$ at $\underline{x}$ due to a point force of arbitrary time dependence acting at $\underline{\xi}$ can be computed by a convolution integration; i.e.,

$$u_i(\underline{\xi}, \underline{x}, t) = \int_0^{\infty} \dot{G}_{ij}(\underline{\xi}, \underline{x}, t) f_j(\tau - t) d\tau. \quad (1)$$

Here, $\dot{G}_{ij}$ is the time derivative of G_{ij} and $f_j(t)$ is the point force component of arbitrary time dependence acting in the j th direction (summation over repeated indices is used). Displacement due to point dipoles or couple forces can be represented by the spatial derivatives of G_{ij} . Displacement produced by a dynamic force distributed over a finite area can also be computed by numerical integration using the Green's function as the kernel of the integral over the finite area.

The basic formulation of the problem and derivation of the solution formulas were reported in (1). The method used is called "ray theory" in the seismological literature. Our derivation was based upon John Willis' new Fourier Inversion method (2). Similar computation results can be found in Reference (3), (4), and (5).

The program was originally written for the analysis of acoustic emission signals. Acoustic emission is a nondestructive testing and monitoring technique in which the detection of the transient stress wave produced by localized deformation or cracks are used to locate, and to assess the criticality of the defects. The theoretical computation is an important link to predict how the acoustic emission waveform evolves through the structure. How the theoretical computations are used in the study of acoustic emission can be found in References (1), (6), (7), and (8).

This computer program is made available mainly for its application to calibrate acoustic emission systems and sensors. By making the present program available, duplication of efforts can be avoided, errors in theory can be checked, and experimental results can be reproduced and verified.

How to Use the Program

I. The easiest way to run this program is by setting up nondimensional parameters and calling the subroutine GREENFCT. First, the x-y-z coordinates

are selected by choosing an x-y plane parallel to the plate and the origin at the center of the plate directly underneath the source, i.e., the source is always located at $\underline{\xi}=(0,0,0.5)$; the x axis is aligned in the direction from the source pointing to the detector. The required nondimensionalized input parameters for the subroutine GREENFCT are:

1. ALPHA = shear wave speed/longitudinal wave speed;
2. XD = x-coordinate of the detector; actual length/thickness of the plate;
3. ZD = z-coordinate of the detector; actual length/thickness of the plate.
 ZD = 0.5 if the source and the detector are on the same side of the plate;
 ZD = -0.5 if they are on opposite sides of the plate.

(Note: the y-coordinate of the detector is always zero because of the way the coordinate system is chosen.)

4. INDEX = ij; subscript of Green's function, integer number 11, 12, 13, 21, 22, 23, 31, 32, or 33. (111, 112, 113, etc. for force dipoles.)
5. TDELTA = sampling time interval in terms of nondimensionalized time unit
 = actual sampling time interval * shear wave speed/thickness of the plate; and
6. NPOINT = Total number of sampling points to be computed; must be an integer.

The subroutine will return a double precision array DISPL of dimension (NPOINT) which corresponds to the desired $G_{ij}(\underline{\xi}, \underline{x}, t)$ or $G_{ijk}(\underline{\xi}, \underline{x}, t)$ sampled at equal time intervals. However, the displacements corresponding to dipoles or couples (i.e., G_{ijk}) are for linear ramp time dependency input. Differentiating the returned DISPL with respect to time once will give the proper Green's function due to step time dependency input.

A simple program calls GREENFCT for given nondimensionalized input parameters together with the output results is included in Appendix A; a program that prompts the user for input parameters of arbitrary physical units and returns displacement in physical units is included in Appendix B. The complete listing of the subroutines is in Appendix C.

II. Another way to use this program is by calling GREENSUB which is basically a simple function: for given one time T, it returns a displacement DISPL.

However, there are two preparatory steps that must be taken in the main program:

(1) RAYTIME (NRAY, 3), the time of arrival table, TA(NRAY) and CN(NRAY, 3), two working arrays, should be dimensioned.

(2) The subroutines INIT and TIMEARRI must be called first.

The subroutine GREENFCT may serve as an example of how to use the subroutine GREENSUB.

III. In addition to the two subroutines mentioned above, there are three subroutines which may be called independently and the user may find them useful in checking experimental or theoretical results.

(1) EPIDIS computes the vertical displacement at the epicenter for given ALPHA and T.

(2) SFWAVE computes the surface wave, which corresponds to the Green's function of a semi-infinite space.

(3) TIMEARRI computes the time of arrival for various ray paths.

Comments in these three subroutines serve as instructions about how to use these subroutines.

Some Remarks

I. There are two machine dependent constants which are used in the two numerical integration routines DGLQI and GLI5T. EPMACH is the largest relative spacing. UFLOW is the smallest positive magnitude. See Reference (9) for details about how to set these constants for different computers.

II. The program uses double precision complex arithmetic. In order to run on those FORTRAN compilers lacking double precision complex function libraries, the user may have to modify the program to single precision arithmetic.

III. The ray method is an exact solution for the Green's function in the time domain produced by summing the contribution from successive arrivals of reflected rays. The numerical computation is accurate if the source and detector are near each other (say less than ten plate thicknesses) and if the maximum time is less than ten dimensionless units. If the sensor and detector are far apart or the maximum time is large, the number of rays arriving at approximately the same time may be too large to compute in a reasonable time and the accumulated error may grow.

IV. The current program is limited to the test configuration where the sensor and detector are on the surface of the plate. The displacement computed for force dipole (i.e., INDEX = 111, 112, etc.) are for point dipole of linear ramp time dependency. Differentiation with respect to time once of the displacement will produce proper Green's functions due to step time dependency (See Reference 10).

Acknowledgement

The author would like to thank Dr. David Kahaner of the Scientific Computing Division, NBS, for supplying the integration subroutines DGLQ and GLIST.

References:

1. D.G. Eitzen, et. al., "Fundamental Developments for Quantitative Acoustic Emission Measurements," EPRI NP-2089, Project 608-1, Palo Alto, CA: Electric Power Research Institute, Oct. 1981.
2. J.R. Willis, "Self-Similar Problems in Elastodynamics," Phil. Trans. of Roy. Soc. London, 274, pp. 435-491, May 1973.

3. Y. Pao, Ralph R. Gajewski and Ahmet N. Ceranoglu, "Acoustic Emission and Transient Waves in an Elastic Plate," J. Acoust. Soc. Am., 65(1), Jan. 1979.
4. A.N. Cernoglu and Y.H. Pao, "Propagation of Elastic Pulses and Acoustic Emission in a Plate," ASME Journal of Applied Mechanics, 48, pp. 125-147, Mar. 1981.
5. T.M. Proctor, F.R. Breckenridge and Y.H. Pao, "Transient Waves in an Elastic Plate: Theory and Experiment Compared," J. Acoust. Soc. Am., (6) Dec. 1983.
6. N.N. Hsu, J.A. Simmons and S.C. Hardy, "An Approach to Acoustic Emission Signal Analysis -- Theory and Experiment," Materials Evaluation, 35, pp. 100-106, Oct. 1977.
7. N.N. Hsu and S.C. Hardy, "Experiments in Acoustic Emission Waveform Analysis for Characterization of AE Sources, Sensors and Structures; In Elastic Waves and Non-Destructive Testing of Materials," AMD-Vol. 29, Y.H. Pao, Editor, The American Society of Mechanical Engineers, New York, pp. 85-106, Dec. 1978.
8. C. Chang and W. Sachse, "Analysis of Elastic Wave Signals from An Extended Source in A Plate," J. Acoust. Soc. Am., 77(4), April 1985.
9. P.A. Fox, A.D. Hall and N.L. Schryer, Bell Laboratories Computer Science Technical Report No. 47. Or: The PORT Mathematical Subroutine Library, ACM Transactions on Mathematical Software, Vol. 4, No. 2, pp 104-106.
10. J.E. Sinclair, "Epicentre Solutions for Point Multipole Sources in An Elastic Half-Space," J. Phys. D: Appl. Phys., Vol. 12, 1979, pp 1309-1315.

Appendix A

Sample Program and Results


```

C*****
C
C      PROGRAM TSGREEN
C
C      TEST PROGRAM TO COMPUTE GREEN'S FUNCTION OF A PLATE
C
C      DOUBLE PRECISION DISPLC(256)
C
C      WRITE(5,*) 'TYPE INDEX, ALPHA, XD, AND ZD'
C      READ(5,*) INDEX, ALPHA, XD, ZD
C      INDEX=33
C      ALPHA=0.59902
C      XD=1.0
C      ZD=-0.5
C
C      WRITE(5,*) 'TYPE DELTAT AND NUMBER OF POINTS'
C      READ(5,*) TDELTA, NPOINT
C      TDELTA=0.01
C      NPOINT=256
C
C      CALL GREENFCT(ALPHA, XD, ZD, INDEX, TDELTA, NPOINT, DISPLC)
C
C      DO 1000 I=1, 256
C          TIME = 0.01*REAL(I)
C          WRITE(6, 1001) TIME, DISPLC(I)
1000 CONTINUE
1001 FORMAT(2X, 'TIME = ', F8.3, ',   DISPLACEMENT = ', D21.14)
C      STOP
C      END

```

[illegible]

TIME =	0.550,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.560,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.570,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.580,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.590,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.600,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.610,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.620,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.630,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.640,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.650,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.660,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.670,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.680,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.690,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.700,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.710,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.720,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.730,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.740,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.750,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.760,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.770,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.780,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.790,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.800,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.810,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.820,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.830,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.840,	DISPLACEMENT =	0.00000000000000D+00
TIME =	0.850,	DISPLACEMENT =	0.11075067698712D+00
TIME =	0.860,	DISPLACEMENT =	0.11098606156782D+00
TIME =	0.870,	DISPLACEMENT =	0.11118024981467D+00
TIME =	0.880,	DISPLACEMENT =	0.11133277774482D+00
TIME =	0.890,	DISPLACEMENT =	0.11144318482884D+00
TIME =	0.900,	DISPLACEMENT =	0.11151102652121D+00
TIME =	0.910,	DISPLACEMENT =	0.11153587694415D+00
TIME =	0.920,	DISPLACEMENT =	0.11151733605821D+00
TIME =	0.930,	DISPLACEMENT =	0.11145503541696D+00
TIME =	0.940,	DISPLACEMENT =	0.11134864281672D+00
TIME =	0.950,	DISPLACEMENT =	0.11119787020707D+00
TIME =	0.960,	DISPLACEMENT =	0.11100247472178D+00
TIME =	0.970,	DISPLACEMENT =	0.11076226552545D+00
TIME =	0.980,	DISPLACEMENT =	0.11047710764219D+00
TIME =	0.990,	DISPLACEMENT =	0.11014692394960D+00
TIME =	1.000,	DISPLACEMENT =	0.10977171429770D+00
TIME =	1.010,	DISPLACEMENT =	0.10935150914128D+00
TIME =	1.020,	DISPLACEMENT =	0.10888645033515D+00
TIME =	1.030,	DISPLACEMENT =	0.10837678979646D+00
TIME =	1.040,	DISPLACEMENT =	0.10782270845288D+00
TIME =	1.050,	DISPLACEMENT =	0.10722461062043D+00
TIME =	1.060,	DISPLACEMENT =	0.10658293623297D+00
TIME =	1.070,	DISPLACEMENT =	0.10589820949205D+00
TIME =	1.080,	DISPLACEMENT =	0.10517111210831D+00
TIME =	1.090,	DISPLACEMENT =	0.10440220346099D+00

TIME =	1.100,	DISPLACEMENT =	0.10359233809357D+00
TIME =	1.110,	DISPLACEMENT =	0.10274239332029D+00
TIME =	1.120,	DISPLACEMENT =	0.10185342388500D+00
TIME =	1.130,	DISPLACEMENT =	0.10092631946377D+00
TIME =	1.140,	DISPLACEMENT =	0.99962313600975D-01
TIME =	1.150,	DISPLACEMENT =	0.98962647939981D-01
TIME =	1.160,	DISPLACEMENT =	0.97928754785951D-01
TIME =	1.170,	DISPLACEMENT =	0.96861858478749D-01
TIME =	1.180,	DISPLACEMENT =	0.95763566016467D-01
TIME =	1.190,	DISPLACEMENT =	0.94635474207730D-01
TIME =	1.200,	DISPLACEMENT =	0.93479376354866D-01
TIME =	1.210,	DISPLACEMENT =	0.92296816327325D-01
TIME =	1.220,	DISPLACEMENT =	0.91089747924766D-01
TIME =	1.230,	DISPLACEMENT =	0.89860093569871D-01
TIME =	1.240,	DISPLACEMENT =	0.88609970379555D-01
TIME =	1.250,	DISPLACEMENT =	0.87341207961754D-01
TIME =	1.260,	DISPLACEMENT =	0.86056060561906D-01
TIME =	1.270,	DISPLACEMENT =	0.84756728127925D-01
TIME =	1.280,	DISPLACEMENT =	0.83445596172436D-01
TIME =	1.290,	DISPLACEMENT =	0.82124730252424D-01
TIME =	1.300,	DISPLACEMENT =	0.80796622274542D-01
TIME =	1.310,	DISPLACEMENT =	0.79463686723194D-01
TIME =	1.320,	DISPLACEMENT =	0.78128380592089D-01
TIME =	1.330,	DISPLACEMENT =	0.76793325943265D-01
TIME =	1.340,	DISPLACEMENT =	0.75460795794622D-01
TIME =	1.350,	DISPLACEMENT =	0.74133473644589D-01
TIME =	1.360,	DISPLACEMENT =	0.72813939012166D-01
TIME =	1.370,	DISPLACEMENT =	0.71504914223321D-01
TIME =	1.380,	DISPLACEMENT =	0.70208761123235D-01
TIME =	1.390,	DISPLACEMENT =	0.68928225721454D-01
TIME =	1.400,	DISPLACEMENT =	0.63777644085523D-01
TIME =	1.410,	DISPLACEMENT =	-0.31205499826477D-01
TIME =	1.420,	DISPLACEMENT =	0.36764444097372D+00
TIME =	1.430,	DISPLACEMENT =	0.39105863151275D+00
TIME =	1.440,	DISPLACEMENT =	0.40949206785671D+00
TIME =	1.450,	DISPLACEMENT =	0.42584966367431D+00
TIME =	1.460,	DISPLACEMENT =	0.44102467848492D+00
TIME =	1.470,	DISPLACEMENT =	0.45540362930489D+00
TIME =	1.480,	DISPLACEMENT =	0.46918701309117D+00
TIME =	1.490,	DISPLACEMENT =	0.48248810354954D+00
TIME =	1.500,	DISPLACEMENT =	0.49537929192411D+00
TIME =	1.510,	DISPLACEMENT =	0.50790390548020D+00
TIME =	1.520,	DISPLACEMENT =	0.52009143324477D+00
TIME =	1.530,	DISPLACEMENT =	0.53196191904305D+00
TIME =	1.540,	DISPLACEMENT =	0.54352825516419D+00
TIME =	1.550,	DISPLACEMENT =	0.55480267134178D+00
TIME =	1.560,	DISPLACEMENT =	0.56579147374873D+00
TIME =	1.570,	DISPLACEMENT =	0.57650038247907D+00
TIME =	1.580,	DISPLACEMENT =	0.58693303751993D+00
TIME =	1.590,	DISPLACEMENT =	0.59709536636170D+00
TIME =	1.600,	DISPLACEMENT =	0.60698989951039D+00
TIME =	1.610,	DISPLACEMENT =	0.61661997278485D+00
TIME =	1.620,	DISPLACEMENT =	0.62598797275653D+00
TIME =	1.630,	DISPLACEMENT =	0.63509897646406D+00
TIME =	1.640,	DISPLACEMENT =	0.64395545056472D+00

TIME =	1.650,	DISPLACEMENT =	0.65256088669347D+00
TIME =	1.660,	DISPLACEMENT =	0.66091813088975D+00
TIME =	1.670,	DISPLACEMENT =	0.66903255717167D+00
TIME =	1.680,	DISPLACEMENT =	0.67690728626371D+00
TIME =	1.690,	DISPLACEMENT =	0.68454639170679D+00
TIME =	1.700,	DISPLACEMENT =	0.69195338607382D+00
TIME =	1.710,	DISPLACEMENT =	0.69913401239547D+00
TIME =	1.720,	DISPLACEMENT =	0.70609199225996D+00
TIME =	1.730,	DISPLACEMENT =	0.71283185830127D+00
TIME =	1.740,	DISPLACEMENT =	0.71935822154785D+00
TIME =	1.750,	DISPLACEMENT =	0.72567516353530D+00
TIME =	1.760,	DISPLACEMENT =	0.73178861132189D+00
TIME =	1.770,	DISPLACEMENT =	0.73770270898197D+00
TIME =	1.780,	DISPLACEMENT =	0.74342223326031D+00
TIME =	1.790,	DISPLACEMENT =	0.74895144993679D+00
TIME =	1.800,	DISPLACEMENT =	0.75429619306931D+00
TIME =	1.810,	DISPLACEMENT =	0.75946069053969D+00
TIME =	1.820,	DISPLACEMENT =	0.76444968084343D+00
TIME =	1.830,	DISPLACEMENT =	0.76926741601862D+00
TIME =	1.840,	DISPLACEMENT =	0.77391947485854D+00
TIME =	1.850,	DISPLACEMENT =	0.77840999911627D+00
TIME =	1.860,	DISPLACEMENT =	0.78274354050484D+00
TIME =	1.870,	DISPLACEMENT =	0.78692419559247D+00
TIME =	1.880,	DISPLACEMENT =	0.79095718088098D+00
TIME =	1.890,	DISPLACEMENT =	0.79484643653490D+00
TIME =	1.900,	DISPLACEMENT =	0.86806930229072D+00
TIME =	1.910,	DISPLACEMENT =	0.87369508356078D+00
TIME =	1.920,	DISPLACEMENT =	0.87923314459302D+00
TIME =	1.930,	DISPLACEMENT =	0.88468749838846D+00
TIME =	1.940,	DISPLACEMENT =	0.89006261168853D+00
TIME =	1.950,	DISPLACEMENT =	0.89536286267026D+00
TIME =	1.960,	DISPLACEMENT =	0.90059204444514D+00
TIME =	1.970,	DISPLACEMENT =	0.90575535254114D+00
TIME =	1.980,	DISPLACEMENT =	0.91085639280247D+00
TIME =	1.990,	DISPLACEMENT =	0.91589918253638D+00
TIME =	2.000,	DISPLACEMENT =	0.92088717546076D+00
TIME =	2.010,	DISPLACEMENT =	0.92582515984505D+00
TIME =	2.020,	DISPLACEMENT =	0.93071640538221D+00
TIME =	2.030,	DISPLACEMENT =	0.93556457171401D+00
TIME =	2.040,	DISPLACEMENT =	0.94037277440757D+00
TIME =	2.050,	DISPLACEMENT =	0.94514541664089D+00
TIME =	2.060,	DISPLACEMENT =	0.94988543954263D+00
TIME =	2.070,	DISPLACEMENT =	0.95459616131288D+00
TIME =	2.080,	DISPLACEMENT =	0.95928037273261D+00
TIME =	2.090,	DISPLACEMENT =	0.96394212326014D+00
TIME =	2.100,	DISPLACEMENT =	0.96858404264596D+00
TIME =	2.110,	DISPLACEMENT =	0.97320913182266D+00
TIME =	2.120,	DISPLACEMENT =	0.97781987758092D+00
TIME =	2.130,	DISPLACEMENT =	0.98242001202845D+00
TIME =	2.140,	DISPLACEMENT =	0.98701187638821D+00
TIME =	2.150,	DISPLACEMENT =	0.99159818288714D+00
TIME =	2.160,	DISPLACEMENT =	0.99618157630830D+00
TIME =	2.170,	DISPLACEMENT =	0.10007641987912D+01
TIME =	2.180,	DISPLACEMENT =	0.10053494399595D+01
TIME =	2.190,	DISPLACEMENT =	0.10099393168333D+01

TIME =	2.200,	DISPLACEMENT =	0.10145362237750D+01
TIME =	2.210,	DISPLACEMENT =	0.10191420575858D+01
TIME =	2.220,	DISPLACEMENT =	0.10237599774067D+01
TIME =	2.230,	DISPLACEMENT =	0.10283917712906D+01
TIME =	2.240,	DISPLACEMENT =	0.10330396140693D+01
TIME =	2.250,	DISPLACEMENT =	0.10377051845379D+01
TIME =	2.260,	DISPLACEMENT =	0.10423914489334D+01
TIME =	2.270,	DISPLACEMENT =	0.10470999932314D+01
TIME =	2.280,	DISPLACEMENT =	0.10518328029171D+01
TIME =	2.290,	DISPLACEMENT =	0.10565913655695D+01
TIME =	2.300,	DISPLACEMENT =	0.10613784905069D+01
TIME =	2.310,	DISPLACEMENT =	0.11476631005725D+01
TIME =	2.320,	DISPLACEMENT =	0.11533185695767D+01
TIME =	2.330,	DISPLACEMENT =	0.11590500263745D+01
TIME =	2.340,	DISPLACEMENT =	0.11648631801910D+01
TIME =	2.350,	DISPLACEMENT =	0.11707621604698D+01
TIME =	2.360,	DISPLACEMENT =	0.11767516974758D+01
TIME =	2.370,	DISPLACEMENT =	0.11828365811738D+01
TIME =	2.380,	DISPLACEMENT =	0.11890210681560D+01
TIME =	2.390,	DISPLACEMENT =	0.11953112520959D+01
TIME =	2.400,	DISPLACEMENT =	0.12017115269290D+01
TIME =	2.410,	DISPLACEMENT =	0.12082269375580D+01
TIME =	2.420,	DISPLACEMENT =	0.12148619575064D+01
TIME =	2.430,	DISPLACEMENT =	0.12216230328019D+01
TIME =	2.440,	DISPLACEMENT =	0.12285147870224D+01
TIME =	2.450,	DISPLACEMENT =	0.12355425415795D+01
TIME =	2.460,	DISPLACEMENT =	0.12427109991126D+01
TIME =	2.470,	DISPLACEMENT =	0.12500269925297D+01
TIME =	2.480,	DISPLACEMENT =	0.12574953844797D+01
TIME =	2.490,	DISPLACEMENT =	0.12651217887352D+01
TIME =	2.500,	DISPLACEMENT =	0.12729111430751D+01
TIME =	2.510,	DISPLACEMENT =	0.12808706969625D+01
TIME =	2.520,	DISPLACEMENT =	0.12890055553986D+01
TIME =	2.530,	DISPLACEMENT =	0.12973216346622D+01
TIME =	2.540,	DISPLACEMENT =	0.13058241075455D+01
TIME =	2.550,	DISPLACEMENT =	0.13145206649727D+01
TIME =	2.560,	DISPLACEMENT =	0.13234166514682D+01
TIME =	2.080,	DISPLACEMENT =	0.95928037273261D+00
TIME =	2.090,	DISPLACEMENT =	0.96394212326014D+00
TIME =	2.100,	DISPLACEMENT =	0.96858404264596D+00
TIME =	2.110,	DISPLACEMENT =	0.97320913182266D+00
TIME =	2.120,	DISPLACEMENT =	0.97781987758092D+00
TIME =	2.130,	DISPLACEMENT =	0.98242001202845D+00
TIME =	2.140,	DISPLACEMENT =	0.98701187638821D+00
TIME =	2.150,	DISPLACEMENT =	0.99159818288714D+00
TIME =	2.160,	DISPLACEMENT =	0.99618157630830D+00
TIME =	2.170,	DISPLACEMENT =	0.10007641987912D+01
TIME =	2.180,	DISPLACEMENT =	0.10053494399595D+01
TIME =	2.190,	DISPLACEMENT =	0.10099393168333D+01
TIME =	2.200,	DISPLACEMENT =	0.10145362237750D+01
TIME =	2.210,	DISPLACEMENT =	0.10191420575858D+01
TIME =	2.220,	DISPLACEMENT =	0.10237599774067D+01
TIME =	2.230,	DISPLACEMENT =	0.10283917712906D+01
TIME =	2.240,	DISPLACEMENT =	0.10330396140693D+01
TIME =	2.250,	DISPLACEMENT =	0.10377051845379D+01

TIME =	2.260,	DISPLACEMENT =	0.10423914489334D+01
TIME =	2.270,	DISPLACEMENT =	0.10470999932314D+01
TIME =	2.280,	DISPLACEMENT =	0.10518328029171D+01
TIME =	2.290,	DISPLACEMENT =	0.10565913655695D+01
TIME =	2.300,	DISPLACEMENT =	0.10613784905069D+01
TIME =	2.310,	DISPLACEMENT =	0.11476631005725D+01
TIME =	2.320,	DISPLACEMENT =	0.11533185695767D+01
TIME =	2.330,	DISPLACEMENT =	0.11590500263745D+01
TIME =	2.340,	DISPLACEMENT =	0.11648631801910D+01
TIME =	2.350,	DISPLACEMENT =	0.11707621604698D+01
TIME =	2.360,	DISPLACEMENT =	0.11767516974758D+01
TIME =	2.370,	DISPLACEMENT =	0.11828365811738D+01
TIME =	2.380,	DISPLACEMENT =	0.11890210681560D+01
TIME =	2.390,	DISPLACEMENT =	0.11953112520959D+01
TIME =	2.400,	DISPLACEMENT =	0.12017115269290D+01
TIME =	2.410,	DISPLACEMENT =	0.12082269375580D+01
TIME =	2.420,	DISPLACEMENT =	0.12148619575064D+01
TIME =	2.430,	DISPLACEMENT =	0.12216230328019D+01
TIME =	2.440,	DISPLACEMENT =	0.12285147870224D+01
TIME =	2.450,	DISPLACEMENT =	0.12355425415795D+01
TIME =	2.460,	DISPLACEMENT =	0.12427109991126D+01
TIME =	2.470,	DISPLACEMENT =	0.12500269925297D+01
TIME =	2.480,	DISPLACEMENT =	0.12574953844797D+01
TIME =	2.490,	DISPLACEMENT =	0.12651217887352D+01
TIME =	2.500,	DISPLACEMENT =	0.12729111430751D+01
TIME =	2.510,	DISPLACEMENT =	0.12808706969625D+01
TIME =	2.520,	DISPLACEMENT =	0.12890055553986D+01
TIME =	2.530,	DISPLACEMENT =	0.12973216346622D+01
TIME =	2.540,	DISPLACEMENT =	0.13058241075455D+01
TIME =	2.550,	DISPLACEMENT =	0.13145206649727D+01
TIME =	2.560,	DISPLACEMENT =	0.13234166514682D+01

Appendix B

Another sample program - use physical units rather nondimensionalized numbers.


```

PROGRAM GPLATE
DOUBLE PRECISION DISPV(1024)
DIMENSION TIME(1024)
INTEGER YORNO

```

C

```

WRITE(1,*)'THIS PROGRAM WILL COMPUTE THE DISPLACEMENT U SUB I AT'
WRITE(1,*)' A POINT ON THE SURFACE OF AN INFINITE PLATE DUE TO A'
WRITE(1,*)' POINT STEP IMPULSIVE FORCE F SUB J OR A POINT LINEAR'
WRITE(1,*)' RAMP FORCE DIPOLE F SUB J,K ACTING AT ANOTHER POINT'
WRITE(1,*)' ON THE SURFACE OF THE PLATE.'
WRITE(1,*)' INPUTS REQUIRED ARE PROMPTED AND TYPED IN ON THE '
WRITE(1,*)' KEYBOARD.'
WRITE(1,*)' '
10 WRITE(1,*)' SUBSCRIPTS OF G = ? (i. e. 11,13,131,322, etc.)'
  READ(1,*) INDEX
  WRITE(1,*)' SHEAR WAVE SPEED = ?'
  READ(1,*) SHSPD
  WRITE(1,*)' LONGITUDINAL WAVE SPEED = ?'
    READ(1,*) PSPD
100 WRITE(1,*)'ARE SOURCE AND DETECTOR ON THE SAME SIDE OF THE PLATE'
  WRITE(1,*)' PLEASE TYPE 1 FOR YES'
  READ (1,*) YORNO
  IF (YORNO.EQ.1) THEN
    ZD = 0.5
  ELSE
    ZD = -0.5
  END IF
  WRITE(1,*)' WHAT IS THE THICKNESS OF THE PLATE ?'
  READ(1,*) HTHICK
  WRITE(1,*)' WHAT IS THE DISTANCE BETWEEN DETECTOR AND SOURCE'
  WRITE(1,*)' OR EPICENTER ?'
  READ(1,*) X
  WRITE(1,*)' SAMPLING TIME INTERVAL = ?'
  READ(1,*) DT
  WRITE(1,*)' NUMBER OF SAMPLING POINTS = ? (MAX = 1024)'
  READ(1,*) NPT
  WRITE(1,*)' SHEAR MODULUS OF THE PLATE = ?'
  READ(1,*) SHMDUL
  WRITE(1,*)' THE OUTPUT WILL BE TIME VERSUS DISPLACEMENT.'
  WRITE(1,*)' '

```

C

```

ALPHA = SHSPD/PSPD
XD = X/HTHICK
TSCALE = HTHICK/SHSPD
DELTAT = DT/TSCALE
DSCALE = 1./(2.*ASIN(1.0)*SHMDUL*HTHICK)

```

C

```

WRITE(2,*)' '
WRITE(2,*)' THE FOLLOWINGS ARE INPUT PARAMETERS:'
WRITE(2,*)'           INDEX = ',INDEX
WRITE(2,*)'           SHEAR WAVE SPEED = ',SHSPD
WRITE(2,*)'LONGITUDINAL WAVE SPEED = ',PSPD
WRITE(2,*)'THICKNESS OF THE PLATE = ',HTHICK
WRITE(2,*)'SAMPLING TIME INTERVAL = ',DT

```

```

WRITE(2,*) '          SHEAR MODULUS = ', SHMDUL
WRITE(2,*) 'NUMBER OF POINTS TO BE COMPUTED = ', NPT
IF (YORNO.EQ.1) THEN
  WRITE(2,*) ' THE DETECTER AND THE SOURCE ARE ON THE SAME SIDE'
ELSE
  WRITE(2,*) 'THE DETECTER AND THE SOURCE ARE ON OPPOSITE SIDES'
END IF

```

C

```

WRITE(2,*) ' '
CALL GREENFCT(ALPHA,XD,ZD,INDEX,DELTAT,NPT,DISPV)
DO 200 I=1,NPT
  TIME(I)=DELTAT*REAL(I)*TSCALE
  DISPV(I)=DISPV(I)*DSCALE
  WRITE(2,*) ' TIME = ', TIME(I), '      DISPLACEMENT = ', DISPV(I)
200 CONTINUE
STOP
END

```

Appendix C

Listing of Subroutines for Computing Green's function of an infinite plate.


```

C*****
C
SUBROUTINE GREENFCT(ALPHA,XD,ZD,INDEX,TDELTA,NPOINT,DISPL)
C
C*****
C
C   TO COMPUT GREEN'S FUNCTION - DISPLACEMENT AS A FUNCTION OF TIME
C   - FOR GIVEN TEST CONFIGURATION
C
C   INPUTS:
C     ALPHA = SHEAR WAVE SPEED / LONGITUDINAL WAVE SPEED
C     XD = X COORDINATE OF THE DETECTOR
C     ZD = Z COORDINATE OF DETECTOR
C     ZD = 0.5 => DETECTOR & SOURCE ON THE SAME SIDE
C     ZD =-0.5 => DETECTOR & SOURCE ON OPPOSIT SIDES
C     INDEX = SUBSCRIPT OF GREEN'S FUNCTION, i.e. 33, 322
C     TDELTA = TIME INCREMENT
C     NPOINT = TOTAL NUMBER OF POINTS TO BE COMPUTED
C
C   OUTPUT:
C     DISPL = DISPLACEMENT ** DOUBLE PRECISION, MUST BE
C             DIMENSIONED (NPOINT)
C-----
C
C   NOTE 1:
C     ALL INPUTS AND OUTPUTS ARE NONDIMENSIONALIZED PARAMETERS
C     DISPL = NORMALIZED DISPLACEMENT = ACTUAL DISPLACEMENT * PI
C             * SHEAR MODULUS * PLATE THICKNESS / FORCE
C     T = NORMALIZED TIME = ACTUAL TIME *SHEAR WAVE SPEED / PLATE
C         THICKNESS
C     DISPL MUST BE DIMENSIONED IN THE CALLING PROGRAM (NPOINT)
C-----
C
C   NOTE 2:
C     THE ARRIVAL TIMES ARE STORED IN RAYTIME(N,I)
C     N = THE Nth RAY
C     RAYTIME(N,3) = THE ARRIVAL TIME
C     RAYTIME(N,1) = N1, NUMBER OF P TRIPS
C     RAYTIME(N,2) = N2, NUMBER OF S TRIPS
C     IF N2<0, IT IS A HEAD WAVE
C     IF N1=-1 , IT IS SURFACE WAVE
C     IF N1=0 AND N2=0, IT IS A RAYLEIGH WAVE
C
C   THE TIME OF ARRIVAL IMFORMATION IS ALSO PRINTED ON LU 6
C-----
C
C     DOUBLE PRECISION DISPL(NPOINT)
C     DIMENSION TA(101),CN(101,3),RAYTIME(101,3)
C     NRAY = NUMBER OF MAXIMUM RAYS EXPECTED
C     DATA NRAY/101/
C
C     CALL INIT(ALPHA,XD,ZD,INDEX)

```

```

GR000010
GR000020
GR000030
GR000040
GR000050
GR000060
GR000070
GR000080
GR000090
GR000100
GR000110
GR000120
GR000130
GR000140
GR000150
GR000160
GR000170
GR000180
GR000190
GR000200
GR000210
GR000220
GR000230
GR000240
GR000250
GR000260
GR000270
GR000280
GR000290
GR000300
GR000310
GR000320
GR000330
GR000340
GR000350
GR000360
GR000370
GR000380
GR000390
GR000400
GR000410
GR000420
GR000430
GR000440
GR000450
GR000460
GR000470
GR000480
GR000490
GR000500
GR000510
GR000520
GR000530
GR000540

```

CALL TIMEARRI(ALPHA,XD,ZD,NRAY,RAYTIME,TA,CN)	GR000550
WRITE(6,*) ' I N1 N2 TIME ARRIVAL '	GR000560
DO 500 I=1,NRAY	GR000570
WRITE (6,499) I,RAYTIME(I,1),RAYTIME(I,2),RAYTIME(I,3)	GR000580
499 FORMAT(2X,I5,3X,3F10.5)	GR000590
500 CONTINUE	GR000600
DO 1000 I=1,NPOINT	GR000610
TIMEI = FLOAT(I)*TDELTA	GR000620
CALL GREENSUB(ALPHA,INDEX,XD,ZD,NRAY,TIMEI,RAYTIME,DISPL(I))	GR000630
WRITE(6,*) 'TIME = ',TIMEI,' DISPL = ',DISPL(I)	GR000640
1000 CONTINUE	GR000650
RETURN	GR000660
END	GR000670
C*****	GR000680
C	GR000690
SUBROUTINE INIT(ALPHA,XD,ZD,INDEX)	GR000700
C	GR000710
C*****	GR000720
IMPLICIT DOUBLE COMPLEX (Y)	GR000730
IMPLICIT DOUBLE PRECISION (D)	GR000740
COMMON /BLK0/DXD,DT,DZ,DA	GR000750
COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR000760
COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR000770
COMMON /BLK5/YR,Y1,Y2	GR000780
DXD = DBLE(XD)	GR000790
DZ = DBLE(ZD)	GR000800
DA = DBLE(ALPHA)	GR000810
IF (ZD.EQ.0.5) KASE = 1	GR000820
IF (ZD.EQ.-0.5) KASE = 2	GR000830
IF (INDEX.LT.100) THEN	GR000840
K = 0	GR000850
M = INDEX/10	GR000860
L = INDEX - M*10	GR000870
ELSE IF ((INDEX.GT.100).AND.(INDEX.LT.334)) THEN	GR000880
M = INDEX/100	GR000890
L = (INDEX - M*100)/10	GR000900
K = (INDEX - M*100 - L*10)	GR000910
ELSE	GR000920
WRITE(6,*) ' WRONG INDEX '	GR000930
PAUSE	GR000940
END IF	GR000950
YI = DCMLPX(0.D0,1.D0)	GR000960
YONE = DCMLPX(1.D0,0.D0)	GR000970
YTWO = DCMLPX(2.D0,0.D0)	GR000980
YTHREE = DCMLPX(3.D0,0.D0)	GR000990
YFOUR = DCMLPX(4.D0,0.D0)	GR001000
YEIGHT = DCMLPX(8.D0,0.D0)	GR001010
YZERO = DCMLPX(0.D0,0.D0)	GR001020
CALL RAYRT(YR,Y1,Y2,ALPHA)	GR001030
RETURN	GR001040
END	GR001050
C*****	GR001060
C	GR001070
SUBROUTINE GREENSUB(ALPHA,INDEX,XD,ZD,NRAY,T,RAYTIME,DISPL)	GR001080
C	GR001090

```

C*****
C  SUBROUTINE TO COMPUTE THE GREENS FUNCTION OF A PLATE AT T
C
C  INPUTS:
C    CONFIGURATION PARAMETERS:
C      DETECTOR LOCATION:
C        X = XD
C        Y = 0
C        Z = ZD, ZD=0.5 => TOP; ZD=-0.5 => BOTTOM
C      SOURCE LOCATION:
C        XO = 0
C        YO = 0
C        ZO = +0.5 (ON TOP OF THE PLATE)
C
C    MATERIALS PARAMETERS:
C      ALPHA = RATIO OF SHEAR VELOCITY TO LONGITUDINAL VELOCITY
C
C    OTHER PARAMETERS:
C      INDEX: SUBSCRIPTS OF GREENS FUNCTION i.e. 33, or 113
C      NRAY: MAXIMUM NUMBER OF RAYS EXPECTED. ~100
C      RAYTIME: TIME OF ARRIVAL TABLE OF DIMENSION (NRAY,3)
C
C      T = NORMALIZED TIME = ACTUAL TIME *SHEAR WAVE SPEED / PLATE
C          THICKNESS
C
C    DISPLACEMENT WILL BE CORRECT ONLY IF T < RAYTIME(NRAY,3)
C
C  OUTPUT:
C    DISPL = NORMALIZED DISPLACEMENT = ACTUAL DISPLACEMENT * PI
C          * SHEAR MODULUS * PLATE THICKNESS / FORCE
C
C    NOTE: SUBROUTINE INIT(ALPHA,XD,ZD,INDEX) AND
C          SUBROUTINE TIMEARRI(.....) SHOULD BE CALLED FIRST IN
C          THE MAIN PROGRAM.
C
C    IMPLICIT DOUBLE COMPLEX (Y)
C    IMPLICIT DOUBLE PRECISION (D)
C    DOUBLE PRECISION DINTG
C    DIMENSION RAYTIME(NRAY,3)
C    COMMON /BLK0/DXD,DT,DZ,DA
C    COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA
C    COMMON /BLK2/YD,Y,YP,YQ,YRO,YRP,YRM,YDELTA,YPSQP1,YQSQP1,
1  YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2
2  ,YDPHI
C    COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO
C    COMMON /BLK5/YR,Y1,Y2
C
C    EXTERNAL DINTG
C
C    IF ( (INDEX.EQ.12).OR.(INDEX.EQ.21).OR.
+      (INDEX.EQ.23).OR.(INDEX.EQ.32).OR.
+      (INDEX.EQ.121).OR.(INDEX.EQ.211).OR.
+      (INDEX.EQ.231).OR.(INDEX.EQ.321).OR.
+      (INDEX.EQ.123).OR.(INDEX.EQ.213).OR.

```

```

GR001100
GR001110
GR001120
GR001130
GR001140
GR001150
GR001160
GR001170
GR001180
GR001190
GR001200
GR001210
GR001220
GR001230
GR001240
GR001250
GR001260
GR001270
GR001280
GR001290
GR001300
GR001310
GR001320
GR001330
GR001340
GR001350
GR001360
GR001370
GR001380
GR001390
GR001400
GR001410
GR001420
GR001430
GR001440
GR001450
GR001460
GR001470
GR001480
GR001490
GR001500
GR001510
GR001520
GR001530
GR001540
GR001550
GR001560
GR001570
GR001580
GR001590
GR001600
GR001610
GR001620
GR001630
GR001640

```

+	(INDEX.EQ.233).OR.(INDEX.EQ.323).OR.	GR001650
+	(INDEX.EQ.132).OR.(INDEX.EQ.312).OR.	GR001660
+	(INDEX.EQ.112).OR.(INDEX.EQ.222).OR.(INDEX.EQ.332)) THEN	GR001670
	DISPL=0.0D0	GR001680
	RETURN	GR001690
	ELSE	GR001700
C		GR001710
	DZERO=0.D0	GR001720
C	NMAX = NRAY; Maximum number of rays. The same as DIM RAYTIME	GR001730
	NMAX = NRAY	GR001740
	IF(T.GT.(RAYTIME(NRAY,3))) THEN	GR001750
	WRITE(6,*)'*****'	GR001760
	WRITE(6,*)'ERROR IN SUBROUTINE GREEN'	GR001770
	WRITE(6,*)'T > TARRIVAL(NRAY), INCREASE NRAY.'	GR001780
	WRITE(6,*)'PROGRAM IS PAUSED'	GR001790
	WRITE(6,*)'*****'	GR001800
	PAUSE 'GREENSUB'	GR001810
	ENDIF	GR001820
C		GR001830
	DT=DBLE(T)	GR001840
C		GR001850
C		GR001860
	DSURF = 0.D0	GR001870
	IF (KASE.EQ.1) THEN	GR001880
	CALL SFWAVE(ALPHA,XD,T,INDEX,DSURF)	GR001890
	ELSE IF (KASE.EQ.2)THEN	GR001900
	DSURF = 0.D0	GR001910
	END IF	GR001920
	DISPL=0.D0	GR001930
C		GR001940
	DO 8000 N=1,NMAX	GR001950
	IF (T.LE.RAYTIME(N,3)) GO TO 8001	GR001960
	N1 = RAYTIME(N,1)	GR001970
	N2 = RAYTIME(N,2)	GR001980
C		GR001990
	IF((N1.EQ.0).AND.(N2.EQ.0)) GOTO 8000	GR002000
	IF((N1.LT.0).AND.(N2.EQ.0)) GOTO 8000	GR002010
C***	IT IS RAYLEIGH WAVE ARRIVAL WHICH HAS BEEN TAKEN	GR002020
C	CARE OF BY SUBROUTINE SFWAVE	GR002030
	IF(N2.LT.0) THEN	GR002040
C***	IT IS A HEAD WAVE	GR002050
	N2=-N2	GR002060
	DK2=DBLE(REAL(N2))	GR002070
	DTSARR = DSQRT(DXD*DXD + DK2*DK2)	GR002080
	IF (DT.GE.DTSARR) GO TO 8000	GR002090
	DHEAD = (DT/DA)-DK2*DSQRT(1.0D0-DA*DA)/DA	GR002100
	DRAYI=DINTG(-DXD,-DHEAD)	GR002110
	ELSE	GR002120
C***	IT IS A REGULAR RAY	GR002130
	DRAYI=DINTG(DZERO,-DXD)	GR002140
	END IF	GR002150
C		GR002160
	DISPL = DISPL + DRAYI	GR002170
C8000	Y=YZERO ;set try root eq. 0	GR002180
8000	Y=YZERO	GR002190

8001	DISPL=DISPL/(6.2831853071795900)	GR002200
	DISPL=DISPL+DSURF	GR002210
	RETURN	GR002220
	END IF	GR002230
	END	GR002240
C*****		GR002250
C		GR002260
	DOUBLE PRECISION FUNCTION DINTG(DINIT,DEND)	GR002270
C		GR002280
C*****		GR002290
C		GR002300
	IMPLICIT DOUBLE PRECISION (D)	GR002310
C		GR002320
C		GR002330
	DOUBLE PRECISION EPS,R,E,W(50,6),FMIN,FMAX,F	GR002340
	INTEGER NINT,NMAX,KF,IFLAG	GR002350
	LOGICAL RST	GR002360
C		GR002370
C		GR002380
	EXTERNAL F	GR002390
C		GR002400
	EPS = 1.D-6	GR002410
	RST = .FALSE.	GR002420
	NMAX = 50	GR002430
	NINT = 1	GR002440
	W(1,1) = DINIT	GR002450
	W(2,1) = DEND	GR002460
C		GR002470
	CALL DGLQ1(F,DINIT,DEND,EPS,R,E,NINT,RST,W,NMAX,FMIN,FMAX,KF	GR002480
1	,IFLAG)	GR002490
C		GR002500
C		GR002510
	DINTG=R	GR002520
	RETURN	GR002530
	END	GR002540
C*****		GR002550
C		GR002560
	DOUBLE PRECISION FUNCTION F(DS)	GR002570
C		GR002580
C*****		GR002590
C		GR002600
	IMPLICIT DOUBLE COMPLEX (Y)	GR002610
	IMPLICIT DOUBLE PRECISION (D)	GR002620
	DOUBLE PRECISION ETA1,ETA2SQ,DD,AA	GR002630
	DIMENSION YQD(3,3),YRSTAR(3,3),YQS(3,3)	GR002640
	COMMON /BLK0/DXD,DT,DZ,DA	GR002650
	COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHEA	GR002660
	COMMON /BLK2/YD,Y,YP,YQ,YR,YRP,YRM,YDELTA,YPSQP1,YQSQP1,	GR002670
1	YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2	GR002680
2	,YDPHI	GR002690
	COMMON /BLK3/Y1,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR002700
C		GR002710
	DK1=DBLE(REAL(N1))	GR002720
	DK2=DBLE(REAL(N2))	GR002730
	D=DS	GR002740

C		GR002750
	CALL PHIEQ0(DK1,DK2,DA,DT,DXD,DS,Y,DERR,NROOT)	GR002760
C	WRITE(6,*)DK1,DK2,DS,Y,DERR,NROOT	GR002770
	IF (NROOT.EQ.0) THEN	GR002780
	F=0.D0	GR002790
C		GR002800
	ELSE	GR002810
	CALL COMMON(DS)	GR002820
	CALL RSTAR(YRSTAR,IDUMMY)	GR002830
	CALL QD(YQD,IDUMMY)	GR002840
	CALL QSU(YQS,IDUMMY)	GR002850
C		GR002860
	YINTG = YZERO	GR002870
	DO 200 I=1,2	GR002880
	DO 100 J=1,2	GR002890
	YINTG=YINTG+YQD(M,I)*YRSTAR(I,J)*YQS(L,J)	GR002900
100	CONTINUE	GR002910
200	CONTINUE	GR002920
C		GR002930
	IF ((N1.EQ.0).AND.(N2.GT.0))	GR002940
1	YINTG=YINTG+YQD(M,3)*YQS(L,3)	GR002950
C		GR002960
	YINTG=YINTG*YDPHI/CDSQRT(DCMPLX(DXD*DXD-DS*DS))	GR002970
	F=DBLE(YINTG)	GR002980
C		GR002990
	END IF	GR003000
	RETURN	GR003010
	END	GR003020
C*****		GR003030
C		GR003040
	SUBROUTINE PHIEQ0(DK1,DK2,DA,DT,DXD,D,YROOT,DERR,NROOT)	GR003050
C		GR003060
C*****		GR003070
C		GR003080
C	GIVEN DK1,DK2,DA,DT,D TO SOLVE PHI(YROOT)=0	GR003090
	IMPLICIT DOUBLE PRECISION (D)	GR003100
	IMPLICIT DOUBLE COMPLEX (Y)	GR003110
	DOUBLE PRECISION T,R,A,C1,C2	GR003120
	COMMON /PARM1/T,R,A,C1,C2	GR003130
C		GR003140
	YOLD=YROOT	GR003150
	IER=0	GR003160
	DYREAL=0.D0	GR003170
	YPHI=DCMPLX(0.D0,0.D0)	GR003180
	T = DT	GR003190
	R = D	GR003200
	A = DA	GR003210
	C1 = DK1	GR003220
	C2 = DK2	GR003230
	EPS=1.E-12	GR003240
	IEND=100	GR003250
C		GR003260
	NROOT=1	GR003270
	IF (DK2.EQ.0.D0) THEN	GR003280
C		GR003290

	DSTSQ = (DT/DA)*(DT/DA)-DK1*DK1	GR003300
	DSTAR = DSQRT(DSTSQ)	GR003310
	IF (D.LT.0.D0)DSTAR=-DSTAR	GR003320
C		GR003330
	IF (D.EQ.0.D0) THEN	GR003340
	DYIMAG=-DK1/(DSTAR*DA)	GR003350
	YROOT = DCMPLX(0.D0,DYIMAG)	GR003360
	ELSE IF (D.EQ.DSTAR) THEN	GR003370
	DYREAL=DT/(DA*DA*DSTAR)	GR003380
	YROOT = DCMPLX(DYREAL,0.0D0)	GR003390
	ELSE	GR003400
	DYREAL = D*DT/(DA*DA*DSTSQ)	GR003410
	DSMD=DSTSQ-D*D	GR003420
	IF(DSMD.LT.0.D0) WRITE(6,*) DT,D,DSMD,DSTSQ	GR003430
	DYIMAG = -(DK1/(DSTSQ*DA))*DSQRT(DSMD)	GR003440
	YROOT = DCMPLX(DYREAL,DYIMAG)	GR003450
	END IF	GR003460
	YOLD = YROOT	GR003470
	CALL DCRTNI(YROOT,YPHI,YDPHI,YOLD,EPS,IEND,IER)	GR003480
C		GR003490
	ELSE IF (DK1.EQ.0.D0) THEN	GR003500
	DTHARR = DA*DABS(DXD)+DK2*DSQRT(1.0D0-DA*DA)	GR003510
	DTSARR = DSQRT(DXD*DXD+DK2*DK2)	GR003520
	DSTSQ = (DT)*(DT)-DK2*DK2	GR003530
	DSTAR = DSQRT(DSTSQ)	GR003540
C		GR003550
	IF (DT.GT.DTSARR) THEN	GR003560
	IF (D.LT.0.D0) DSTAR=-DSTAR	GR003570
C		GR003580
	IF (D.EQ.0.D0) THEN	GR003590
	DYIMAG=-DK2/DABS(DSTAR)	GR003600
	YROOT = DCMPLX(0.D0,DYIMAG)	GR003610
	ELSE IF (D.EQ.DSTAR) THEN	GR003620
	DYREAL=DT/(DSTAR)	GR003630
	YROOT = DCMPLX(DYREAL,0.0D0)	GR003640
	ELSE	GR003650
	DYREAL = D*DT/(DSTSQ)	GR003660
	DYIMAG = -(DK2/DSTSQ)*DSQRT(DSTSQ-D*D)	GR003670
	YROOT = DCMPLX(DYREAL,DYIMAG)	GR003680
	END IF	GR003690
	ELSE IF ((DT.LT.DTSARR).AND.(DT.GT.DTHARR)) THEN	GR003700
	DHEAD = (DT/DA)-DK2*DSQRT(1.0D0-DA*DA)/DA	GR003710
	IF (D.LT.0.D0)DHEAD=-DHEAD	GR003720
	IF (D.LT.0.D0)DSTAR=-DSTAR	GR003730
		GR003740
C		GR003750
	IF (D.EQ.0.D0) THEN	GR003760
	DYIMAG=-DK2/DSTAR	GR003770
	YROOT = DCMPLX(0.D0,DYIMAG)	GR003780
	ELSE IF (D.EQ.DSTAR) THEN	GR003790
	DYREAL=DT/(DSTAR)	GR003800
	YROOT = DCMPLX(DYREAL,0.0D0)	GR003810
	ELSE IF (DABS(D).GT.DABS(DSTAR)) THEN	GR003820
	IF (D.LT.0.D0) THEN	GR003830
	DYREAL = (D*DT-DK2*DSQRT(D*D-DSTSQ))/(DSTSQ)	GR003840
	ELSE IF (D.GT.0.D0) THEN	

DYREAL = (D*DT+DK2*DSQRT(D*D-DSTSQ))/(DSTSQ)	GR003850
ENDIF	GR003860
DYIMAG = 0.D0	GR003870
YROOT = DCMPLX(DYREAL,DYIMAG)	GR003880
END IF	GR003890
END IF	GR003900
YOLD = YROOT	GR003910
CALL DCRTNI(YROOT,YPHI,YDPHI,YOLD,EPS,IEND,IER)	GR003920
C	GR003930
ELSE IF ((DK1.NE.0.D0).AND.(DK2.NE.0.D0)) THEN	GR003940
IF((D.EQ.0.D0).OR.(YOLD.EQ.DCMPLX(0.D0,0.D0))) THEN	GR003950
CALL DEQO(DT,DA,DK1,DK2,YROOT,NOROOT)	GR003960
ENDIF	GR003970
YOLD = YROOT	GR003980
CALL DCRTNI(YROOT,YPHI,YDPHI,YOLD,EPS,IEND,IER)	GR003990
END IF	GR004000
IF(IER.NE.0) NROOT=0	GR004010
C	GR004020
C TEST ROOT TO FIND ERROR	GR004030
C	GR004040
DERR=CDABS(YPHI)	GR004050
C	GR004060
RETURN	GR004070
END	GR004080
C*****	GR004090
C	GR004100
SUBROUTINE DEQO(DT,DA,DK1,DK2,Y,NOROOT)	GR004110
C	GR004120
C*****	GR004130
C	GR004140
C COMPUTE THE ROOT BY SOLVING QUARTIC EQ.	GR004150
C	GR004160
IMPLICIT DOUBLE PRECISION (D)	GR004170
IMPLICIT DOUBLE COMPLEX (Y)	GR004180
DOUBLE PRECISION B	GR004190
DIMENSION DB(5),B(3),DROOT1(2),DROOT2(2)	GR004200
C	GR004210
YF(Y)=-DCMPLX(DT,0.D0)*Y-DCMPLX(DK1,0.D0)*	GR004220
1 CDSQRT(Y*Y*DCMPLX(DA*DA,0.D0)-DCMPLX(1.D0,0.D0))	GR004230
2 -DCMPLX(DK2,0.D0)*CDSQRT(Y*Y-DCMPLX(1.D0,0.D0))	GR004240
DT2 = DT*DT	GR004250
DK12 =DK1*DK1	GR004260
DK22 = DK2*DK2	GR004270
DAA=DA*DA	GR004280
C	GR004290
DB(1)=((DT2-DK12*DAA-DK22)**2 - 4.D0*DK12*DK22*DAA)	GR004300
DB(3)=2.D0*((DT2-DK12*DAA-DK22)*(DK12+DK22)	GR004310
1 +2.D0*DK12*DK22*(DAA+1.D0))	GR004320
DB(5)=(DK12+DK22)**2-4.D0*DK12*DK22	GR004330
C	GR004340
B(1) = DB(1)	GR004350
B(2) = DB(3)	GR004360
B(3) = DB(5)	GR004370
TOL=1.E-20	GR004380
C CALL QUADRTIC EQ. TO FIND Y**2	GR004390

CALL QUADRT(B,DROOT1,DROOT2,TOL,NOROOT)	GR004400
C	GR004410
Y1=CDSQRT(DCMPLX(DROOT1(1),DROOT1(2)))	GR004420
Y2=CDSQRT(DCMPLX(DROOT2(1),DROOT2(2)))	GR004430
IF (DIMAG(Y1).GT.0.D0) Y1=-Y1	GR004440
IF (DIMAG(Y2).GT.0.D0) Y2=-Y2	GR004450
YF1=YF(Y1)	GR004460
YF2=YF(Y2)	GR004470
DERR1=CDABS(YF(Y1))	GR004480
DERR2=CDABS(YF(Y2))	GR004490
IF (DERR1.LT.DERR2) Y=Y1	GR004500
IF (DERR1.GE.DERR2) Y=Y2	GR004510
C	GR004520
IF (DIMAG(Y).GE.0.D0) Y=DCONJG(Y)	GR004530
C	GR004540
RETURN	GR004550
END	GR004560
C*****	GR004570
C	GR004580
DOUBLE COMPLEX FUNCTION PHIO(Z)	GR004590
C	GR004600
C*****	GR004610
C	GR004620
C FUNCTION PHIO	GR004630
C	GR004640
DOUBLE PRECISION T,R,A,C1,C2	GR004650
DOUBLE COMPLEX Z,YAA,YONE	GR004660
COMMON /PARM1/T,R,A,C1,C2	GR004670
C	GR004680
YONE=DCMPLX(1.D0,0.D0)	GR004690
YAA=DCMPLX(A*A,0.D0)	GR004700
C	GR004710
PHIO=DCMPLX(R,0.D0)-DCMPLX(T,0.D0)*Z	GR004720
1 -DCMPLX(C1,0.D0)*CDSQRT(YAA*Z*Z-YONE)	GR004730
2 -DCMPLX(C2,0.D0)*CDSQRT(Z*Z-YONE)	GR004740
C	GR004750
RETURN	GR004760
END	GR004770
C*****	GR004780
C	GR004790
DOUBLE COMPLEX FUNCTION DPHIO(Z)	GR004800
C	GR004810
C*****	GR004820
C	GR004830
C FUNCTION DPHIO	GR004840
C	GR004850
C	GR004860
DOUBLE PRECISION T,R,A,C1,C2	GR004870
DOUBLE COMPLEX Z,YAA,YONE	GR004880
COMMON /PARM1/T,R,A,C1,C2	GR004890
C	GR004900
YONE=DCMPLX(1.D0,0.D0)	GR004910
YAA=DCMPLX(A*A,0.D0)	GR004920
C	GR004930
DPHIO=-DCMPLX(T,0.D0)	GR004940

1	-DCMPLX(C1,0.D0)*YAA*Z/CDSQRT(YAA*Z*Z-YONE)	GR004950
2	-DCMPLX(C2,0.D0)*Z/CDSQRT(Z*Z-YONE)	GR004960
C		GR004970
	RETURN	GR004980
	END	GR004990
C	*****	GR005000
C		GR005010
	SUBROUTINE COMMON(DS)	GR005020
C		GR005030
C	*****	GR005040
C		GR005050
C	TO COMPUTE PARAMETERS DEPENDING ON Y AND TO PUT IN	GR005060
C	A COMMON BLK2	GR005070
C		GR005080
	IMPLICIT DOUBLE COMPLEX (Y)	GR005090
	IMPLICIT DOUBLE PRECISION (D)	GR005100
	DOUBLE PRECISION ETA1,ETA2SQ,DD,AA	GR005110
	COMMON /BLK0/DXD,DT,DZ,DA	GR005120
	COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR005130
	COMMON /BLK2/YD,Y,YP,YQ,YR0,YRP,YRM,YDELTA,YPSQP1,YQSQP1,	GR005140
1	YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2	GR005150
2	,YDPHI	GR005160
	COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR005170
C		GR005180
	AA=DA*DA	GR005190
	DD=DS	GR005200
	D=DS	GR005210
	DN1=DBLE(REAL(N1))	GR005220
	DN2=DBLE(REAL(N2))	GR005230
	YZERO=DCMPLX(0.D0,0.D0)	GR005240
	YI=DCMPLX(0.D0,1.0D0)	GR005250
	YAA=DCMPLX(AA,0.D0)	GR005260
	YA=DCMPLX(DA,0.D0)	GR005270
	YONE=DCMPLX(1.0D0,0.0D0)	GR005280
	YTWO=DCMPLX(2.D0,0.D0)	GR005290
	YFOUR=DCMPLX(4.0D0,0.D0)	GR005300
C		GR005310
	ETA1=DS/DXD	GR005320
	ETA2SQ=1.D0-ETA1*ETA1	GR005330
	YETA1 = DCMPLX(ETA1,0.D0)	GR005340
	YETA2S= DCMPLX(ETA2SQ,0.D0)	GR005350
	YETA2=CDSQRT(YETA2S)	GR005360
C		GR005370
	YD = DCMPLX(DD,0.D0)	GR005380
	YY=Y*Y	GR005390
	YP=CDSQRT(YAA*YY-YONE)	GR005400
	YQ=CDSQRT(YY-YONE)	GR005410
	YPSQP1=YAA*YY	GR005420
	YQSQP1=YY	GR005430
	YQSQM1=YQ*YQ-YONE	GR005440
	YPSQM1=YP*YP-YONE	GR005450
C		GR005460
	YSQRP1=Y*YA	GR005470
	YSQRQ1=Y	GR005480
	YDELTA=YQSQM1*YQSQM1+YFOUR*YP*YQ	GR005490

	YM=(YQSQM1*YQSQM1-YFOUR*YP*YQ)	GR005500
	YR0=YM/YDELTA	GR005510
	YRP=-(YFOUR*YP*YQSQM1)/(YA*YDELTA)	GR005520
	YRM=(YR0*YR0-YONE)/YRP	GR005530
C		GR005540
	YDPHI=YONE/(YD+(DCMPLX(DN1,0.D0)/YP)+(DCMPLX(DN2,0.D0)/YQ))	GR005550
	RETURN	GR005560
	END	GR005570
C	*****	GR005580
C		GR005590
	SUBROUTINE RSTAR(YRSTAR,IDUMMY)	GR005600
C		GR005610
C	*****	GR005620
C		GR005630
C	TO COMPUTE RSTAR (JS,J2) Eq.6.39	GR005640
C		GR005650
	IMPLICIT DOUBLE COMPLEX (Y)	GR005660
	IMPLICIT DOUBLE PRECISION (D)	GR005670
	DIMENSION YRSTAR(3,3),KSUM(20)	GR005680
	COMMON /BLK0/DXD,DT,DZ,DA	GR005690
	COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR005700
	COMMON /BLK2/YD,Y,YP,YQ,YR0,YRP,YRM,YDELTA,YPSQP1,YQSQP1,	GR005710
1	YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2	GR005720
2	,YDPHI	GR005730
	COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR005740
C		GR005750
	EXTERNAL KDELTA	GR005760
C		GR005770
C		GR005780
	IDUMMY=0	GR005790
	YRSTAR(3,3) = YONE	GR005800
	YRSTAR(1,3) = YZERO	GR005810
	YRSTAR(2,3) = YZERO	GR005820
	YRSTAR(3,1) = YZERO	GR005830
	YRSTAR(3,2) = YZERO	GR005840
C		GR005850
	DO 2000 JS=1,2	GR005860
	DO 1000 J2=1,2	GR005870
	IF (J2.EQ.1) N1=N1-1	GR005880
	IF (J2.EQ.2) N2=N2-1	GR005890
	ISIGN = (-1)**(J2*(N1+N2)+N2)	GR005900
	CALL COEFF(N1,N2,JS,J2,KD,KU,KL,KSUM)	GR005910
C	WRITE(7,*) N1,N2,JS,J2,KD,KU,KL,KSUM	GR005920
	YSUM=YZERO	GR005930
	DO 100 NM=KL,KU	GR005940
	YSUM=YSUM+DCMPLX(DBLE(REAL(KSUM(NM+1))),0.D0)*	GR005950
1	((YR0*YR0-YONE)/(YR0*YR0))**NM	GR005960
100	CONTINUE	GR005970
	KD2=KDELTA(1,JS)-KDELTA(1,J2)	GR005980
C		GR005990
	YSUM = ((YR0/YRP)**(KD2)) * YSUM	GR006000
	RSIGN=REAL(ISIGN)	GR006010
	DRSIGN=RSIGN	GR006020
	DKD=DBLE(REAL(KD))	GR006030
	YRSTAR(JS,J2)=DCMPLX(DRSIGN,0.D0)*(YR0**(N1+N2))	GR006040

1	*(DCMLX(DKD,0.D0)+YSUM)	GR006050
	IF (J2.EQ.1) N1=N1+1	GR006060
	IF (J2.EQ.2) N2=N2+1	GR006070
C		GR006080
1000	CONTINUE	GR006090
2000	CONTINUE	GR006100
	RETURN	GR006110
	END	GR006120
C	*****	GR006130
C		GR006140
	SUBROUTINE QD(YQD,IDUMMY)	GR006150
C		GR006160
C	*****	GR006170
C		GR006180
C	TO COMPUTE QD(I,J) BASED ON EQ. 6.21	GR006190
C		GR006200
	IMPLICIT DOUBLE COMPLEX (Y)	GR006210
	IMPLICIT DOUBLE PRECISION (D)	GR006220
	COMMON /BLK0/DXD,DT,DZ,DA	GR006230
	COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR006240
	COMMON /BLK2/YD,Y,YP,YQ,YR0,YRP,YRM,YDELTA,YPSQP1,YQSQP1,	GR006250
1	YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2	GR006260
2	,YDPHI	GR006270
	COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR006280
C		GR006290
	DIMENSION YQD(3,3)	GR006300
C		GR006310
	IDUMMY=0	GR006320
	DO 100 I=1,3	GR006330
	DO 90 J=1,3	GR006340
	YQD(I,J) = YZERO	GR006350
90	CONTINUE	GR006360
100	CONTINUE	GR006370
C		GR006380
	IF(M.EQ.1) THEN	GR006390
	YQD(1,1)=YETA1*YFOUR*YP*YQ*YQSQP1	GR006400
1	/(YSQRP1*YDELTA)	GR006410
	YQD(1,2)=-YTWO*YETA1*YQ*YQSQM1*YSQRQ1	GR006420
1	/(YDELTA)	GR006430
	YQD(1,3)=-YTWO*YETA2	GR006440
C		GR006450
	ELSE IF (M.EQ.2) THEN	GR006460
	YQD(2,1)=YETA2*YFOUR*YP*YQ*YQSQP1	GR006470
1	/(YSQRP1*YDELTA)	GR006480
	YQD(2,2)=-YTWO*YETA2*YQ*YQSQM1*YSQRQ1	GR006490
1	/(YDELTA)	GR006500
	YQD(2,3)=YTWO*YETA1	GR006510
C		GR006520
	ELSE IF (M.EQ.3) THEN	GR006530
	YQD(3,1)=-YTWO*YP*YQSQM1*YQSQP1	GR006540
1	/(YSQRP1*YDELTA)	GR006550
	YQD(3,2)=-YFOUR*YP*YQ*YSQRQ1	GR006560
1	/(YDELTA)	GR006570
	YQD(3,3)=YZERO	GR006580
C		GR006590

END IF	GR006600
RETURN	GR006610
END	GR006620
C*****	GR006630
C	GR006640
INTEGER FUNCTION KDELTA(I,J)	GR006650
C	GR006660
C*****	GR006670
C	GR006680
IF (I.EQ.J) THEN	GR006690
KDELTA=1	GR006700
ELSE	GR006710
KDELTA=0	GR006720
END IF	GR006730
RETURN	GR006740
END	GR006750
C*****	GR006760
C	GR006770
INTEGER FUNCTION IBINO(M,N)	GR006780
C	GR006790
C*****	GR006800
C	GR006810
IF (M.EQ.N)GO TO 11	GR006820
IF ((M.LT.0).OR.(N.LT.0)) GO TO 10	GR006830
K=M-N	GR006840
IF (K) 10,11,12	GR006850
10 IBINO = 0	GR006860
RETURN	GR006870
11 IBINO = 1	GR006880
RETURN	GR006890
12 IF (N.EQ.0)GO TO 11	GR006900
IN = 1	GR006910
IP = M	GR006920
IQ = 1	GR006930
DO 20 I=1,K	GR006940
IN = IN*IP/IQ	GR006950
IP = IP-1	GR006960
IQ = IQ +1	GR006970
20 CONTINUE	GR006980
IBINO = IN	GR006990
RETURN	GR007000
END	GR007010
C*****	GR007020
C	GR007030
INTEGER FUNCTION KD1(N1,N2,JS,J2)	GR007040
C	GR007050
C*****	GR007060
C	GR007070
EXTERNAL KDELTA	GR007080
KD1=KDELTA(0,N1)*KDELTA(2,J2)+KDELTA(0,N2)*KDELTA(1,J2)	GR007090
KD1=KD1*KDELTA(JS,J2)	GR007100
RETURN	GR007110
END	GR007120
C*****	GR007130
C	GR007140

INTEGER FUNCTION KUPPER(N1,N2,JS,J2)	GR007150
C	GR007160
C*****	GR007170
C	GR007180
EXTERNAL KDELTA	GR007190
K=N2+KDELTA(2,J2)*KDELTA(1,JS)-KDELTA(1,J2)*KDELTA(2,JS)	GR007200
IF (N1.LE.K) THEN	GR007210
KUPPER=N1	GR007220
ELSE	GR007230
KUPPER=K	GR007240
END IF	GR007250
RETURN	GR007260
END	GR007270
C*****	GR007280
C	GR007290
INTEGER FUNCTION KLOWER(N1,N2,JS,J2)	GR007300
C	GR007310
C*****	GR007320
C	GR007330
EXTERNAL KDELTA	GR007340
KLOWER=KDELTA(2,J2)+KDELTA(1,J2)*KDELTA(1,JS)	GR007350
RETURN	GR007360
END	GR007370
C*****	GR007380
C	GR007390
SUBROUTINE COEFF(N1,N2,JS,J2,KD,KU,KL,KSUM)	GR007400
C	GR007410
C*****	GR007420
C	GR007430
TO COMPUTE THE COEFFICIENTS FOR THE RSTAR TURM	GR007440
C	GR007450
INPUTS:	GR007460
C N1 = NUMBER OF P TRIPS	GR007470
C N2 = NUMBER OF S TRIPS	GR007480
C JS = FINAL TRIP TYPE, 1 OR 2	GR007490
C J2 = INITIAL TRIP TYPE, 1 OR 2	GR007500
C	GR007510
OUTPUT:	GR007520
C KD = FIRST COEFF	GR007530
C KU = UPPER LIMIT IN SUMMATION	GR007540
C KL = LOWER LIMIT IN SUMMATION	GR007550
C KSUM = COEFF. IN SUMMATION, MUST BE DIMENSIONED IN CALLING	GR007560
C PROGRAM TO BE MAX(N1,N2)	GR007570
C	GR007580
C-----	GR007590
C	GR007600
C PROGRAM TESTCOEF	GR007610
C DIMENSION KSUM(20)	GR007620
CC***	GR007630
C 1 WRITE(5,10)	GR007640
C 10 FORMAT (2X,'TYPE K1,K2,JS,J2')	GR007650
C READ(5,*)I,J,JS,J2	GR007660
C WRITE(6,30)I,J,JS,J2	GR007670
C 30 FORMAT(2X,' I =',I5,' J=',I5,' JS=',I2,' J2=',I2)	GR007680
CC***	GR007690

END IF	GR006600
RETURN	GR006610
END	GR006620
C*****	GR006630
C	GR006640
INTEGER FUNCTION KDELTA(I,J)	GR006650
C	GR006660
C*****	GR006670
C	GR006680
IF (I.EQ.J) THEN	GR006690
KDELTA=1	GR006700
ELSE	GR006710
KDELTA=0	GR006720
END IF	GR006730
RETURN	GR006740
END	GR006750
C*****	GR006760
C	GR006770
INTEGER FUNCTION IBINO(M,N)	GR006780
C	GR006790
C*****	GR006800
C	GR006810
IF (M.EQ.N)GO TO 11	GR006820
IF ((M.LT.0).OR.(N.LT.0)) GO TO 10	GR006830
K=M-N	GR006840
IF (K) 10,11,12	GR006850
10 IBINO = 0	GR006860
RETURN	GR006870
11 IBINO = 1	GR006880
RETURN	GR006890
12 IF (N.EQ.0)GO TO 11	GR006900
IN = 1	GR006910
IP = M	GR006920
IQ = 1	GR006930
DO 20 I=1,K	GR006940
IN = IN*IP/IQ	GR006950
IP = IP-1	GR006960
IQ = IQ +1	GR006970
20 CONTINUE	GR006980
IBINO = IN	GR006990
RETURN	GR007000
END	GR007010
C*****	GR007020
C	GR007030
INTEGER FUNCTION KD1(N1,N2,JS,J2)	GR007040
C	GR007050
C*****	GR007060
C	GR007070
EXTERNAL KDELTA	GR007080
KD1=KDELTA(0,N1)*KDELTA(2,J2)+KDELTA(0,N2)*KDELTA(1,J2)	GR007090
KD1=KD1*KDELTA(JS,J2)	GR007100
RETURN	GR007110
END	GR007120
C*****	GR007130
C	GR007140

INTEGER FUNCTION KUPPER(N1,N2,JS,J2)	GR007150
C	GR007160
C*****	GR007170
C	GR007180
EXTERNAL KDELTA	GR007190
K=N2+KDELTA(2,J2)*KDELTA(1,JS)-KDELTA(1,J2)*KDELTA(2,JS)	GR007200
IF (N1.LE.K) THEN	GR007210
KUPPER=N1	GR007220
ELSE	GR007230
KUPPER=K	GR007240
END IF	GR007250
RETURN	GR007260
END	GR007270
C*****	GR007280
C	GR007290
INTEGER FUNCTION KLOWER(N1,N2,JS,J2)	GR007300
C	GR007310
C*****	GR007320
C	GR007330
EXTERNAL KDELTA	GR007340
KLOWER=KDELTA(2,J2)+KDELTA(1,J2)*KDELTA(1,JS)	GR007350
RETURN	GR007360
END	GR007370
C*****	GR007380
C	GR007390
SUBROUTINE COEFF(N1,N2,JS,J2,KD,KU,KL,KSUM)	GR007400
C	GR007410
C*****	GR007420
C	GR007430
TO COMPUTE THE COEFFICIENTS FOR THE RSTAR TURM	GR007440
C	GR007450
INPUTS:	GR007460
C N1 = NUMBER OF P TRIPS	GR007470
C N2 = NUMBER OF S TRIPS	GR007480
C JS = FINAL TRIP TYPE, 1 OR 2	GR007490
C J2 = INITIAL TRIP TYPE, 1 OR 2	GR007500
C	GR007510
OUTPUT:	GR007520
C KD = FIRST COEFF	GR007530
C KU = UPPER LIMIT IN SUMMATION	GR007540
C KL = LOWER LIMIT IN SUMMATION	GR007550
C KSUM = COEFF. IN SUMMATION, MUST BE DIMENSIONED IN CALLING	GR007560
C PROGRAM TO BE MAX(N1,N2)	GR007570
C	GR007580
C-----	GR007590
C	GR007600
C PROGRAM TESTCOEF	GR007610
C DIMENSION KSUM(20)	GR007620
CC***	GR007630
C 1 WRITE(5,10)	GR007640
C 10 FORMAT (2X,'TYPE K1,K2,JS,J2')	GR007650
C READ(5,*)I,J,JS,J2	GR007660
C WRITE(6,30)I,J,JS,J2	GR007670
C 30 FORMAT(2X,' I =',I5,' J=',I5,' JS=',I2,' J2=',I2)	GR007680
CC***	GR007690

IF (KASE.EQ.1) THEN	GR008800
C SOURCE AT TOP SURFACE	GR008810
C	GR008820
DO 200 I=1,3	GR008830
YQS(I,1)=YRRP(1,1)*YUP(I,1)*YP1+YRRP(1,2)*YUP(I,2)*YP2	GR008840
1 - YUM(I,1)*YP1	GR008850
YQS(I,2)=YRRP(2,1)*YUP(I,1)*YP1	GR008860
1 +YRRP(2,2)*YUP(I,2)*YP2 - YUM(I,2)*YP2	GR008870
YQS(I,3)=(YRRP(3,3)*YUP(I,3) - YUM(I,3))*YP2	GR008880
200 CONTINUE	GR008890
C	GR008900
C	GR008910
ELSE IF (KASE.EQ.2) THEN	GR008920
C SOURCE AT BOTTOM SURFACE	GR008930
C	GR008940
DO 300 I=1,3	GR008950
YQS(I,1)=YRRM(1,1)*YUM(I,1)*YP1+YRRM(1,2)*YUM(I,2)*YP2	GR008960
1 -YUP(I,1)*YP1	GR008970
YQS(I,2)=YRRM(2,1)*YUM(I,1)*YP1	GR008980
1 +YRRM(2,2)*YUM(I,2)*YP2 - YUP(I,2)*YP2	GR008990
YQS(I,3)=(YRRM(3,3)*YUM(I,3) - YUP(I,3))*YP2	GR009000
300 CONTINUE	GR009010
C	GR009020
ENDIF	GR009030
RETURN	GR009040
END	GR009050
C *****	GR009060
C	GR009070
SUBROUTINE EPIDIS(A,T,U)	GR009080
C	GR009090
C *****	GR009100
C	GR009110
C	GR009120
C THIS PROGRAM CALCULATES THE VERTICAL DISPLACEMENT	GR009130
C AT THE EPICENTER OF A LARGE PLATE DUE TO AN UNIT	GR009140
C STEP FUNCTION VERTICAL FORCE.	GR009150
C DOUBLE PRECISION IS USED.	GR009160
C	GR009170
C INPUT PARAMETERS :	GR009180
C	GR009190
C A = SHEAR WAVE SPEED / LONGITUDINAL WAVE SPEED	GR009200
C e.g. ALUMINUM : A=0.495	GR009210
C T = NORMALIZED TIME = ACTUAL TIME *SHEAR WAVE SPEED / PLATE	GR009220
C THICKNESS	GR009230
C e.g. SHEAR WAVE SPEED ~ 3 mm /micro second	GR009240
C	GR009250
C OUTPUT PARAMETERS :	GR009260
C	GR009270
C U = NORMALIZED DISPLACEMENT = ACTUAL DISPLACEMENT * 3.14159	GR009280
C * SHEAR MODULUS * PLATE THICKNESS / FORCE	GR009290
C	GR009300
C SUBROUTINES REQUIRED:	GR009310
C	GR009320
C QPARA(T,A), AND FUNCTION BINO	GR009330
C	GR009340

C-----	GR009350
C	GR009360
C THE PROGRAM MAY BE SET UP AS THE FOLLOWINGS:	GR009370
C	GR009380
C- PROGRAM EPICT	GR009390
C- DOUBLE PRECISION A,DT,DELTAT,DU(1024)	GR009400
C- A = 0.584D0	GR009410
C- NPT=1024	GR009420
C- DELTAT=0.00277D0	GR009430
C- DO 100 I=1,NPT	GR009440
C- DT=DELTAT*DBLE(REAL(I))	GR009450
C- CALL EPIDIS(A,DT,DU(I))	GR009460
C-100 CONTINUE	GR009470
C- WRITE(7) DU	GR009480
C- STOP	GR009490
C- END	GR009500
C	GR009510
C-----	GR009520
C	GR009530
IMPLICIT DOUBLE PRECISION (Q)	GR009540
DOUBLE PRECISION U,T,A,TA,DXM	GR009550
INTEGER BINO	GR009560
COMMON /BLOCK/ QN1,QN2,QO,QP,QM,QPP,QPS,QSP,QSS,QPHI	GR009570
C	GR009580
C NRAY = MAX. POSSIBLE NUMBER OF P RAYS	GR009590
C NRAY2 = MAX. POSSIBLE NUMBER OF S RAYS	GR009600
NRAY = 20	GR009610
NRAY2 = 12	GR009620
U = 0.0	GR009630
DO 300 NA = 1,NRAY	GR009640
DO 200 NB = 1,NRAY2	GR009650
N1 = NA-1	GR009660
N2 = NB-1	GR009670
C N1+N2 MUST BE AN ODD NUMBER	GR009680
IF ((FLOAT(N1+N2)/2.).EQ.FLOAT((N1+N2)/2))GO TO 200	GR009690
TA = N2 + N1*A	GR009700
IF (T.LE.TA) GO TO 200	GR009710
QN1 = N1	GR009720
QN2 = N2	GR009730
C	GR009740
CALL QPARA(T,A)	GR009750
C	GR009760
NTERM = N2	GR009770
IF (N1.LT.N2) NTERM=N1	GR009780
NTERM=NTERM+1	GR009790
QSUM = 0.0	GR009800
DO 100 MT = 1,NTERM	GR009810
MM = MT-1	GR009820
QFPP = BINO(N1+1,MM+2)*BINO(N2+1,MM+1)	GR009830
QFPS = BINO(N2+1,MM+1)*BINO(N1+1,MM+1)	GR009840
QFSS = BINO(N2+1,MM+2)*BINO(N1+1,MM+1)	GR009850
MO = N1 + N2 - 1 - 2*MM	GR009860
C	GR009870
QSUM = ((-1.)**N2)*	GR009880
1 ((QFPP*QPP-QFSS*QSS)*(DXM(QO,MO)*DXM((QM*QP*(-1.)),MM))	GR009890

2	+QFPS*(QPS*DXM(QO,(MO+1))*DXM(((-1.)*QP),MM)*DXM(QM,MM)	GR009900
3	-QSP*DXM(QO,(MO+1))*DXM(((-1.)*QM),MM)*DXM(QP,MM))	GR009910
4	+QSUM	GR009920
C		GR009930
100	CONTINUE	GR009940
	U = U + QSUM/QPHI	GR009950
200	CONTINUE	GR009960
300	CONTINUE	GR009970
	RETURN	GR009980
	END	GR009990
C		GR010000
C	*****	GR010010
C		GR010020
	SUBROUTINE QPARA(T,A)	GR010030
C		GR010040
C	*****	GR010050
C		GR010060
	IMPLICIT DOUBLE PRECISION (Q)	GR010070
	DOUBLE PRECISION T,A,B,AC,BAC,XX,X,SX,SAX	GR010080
	COMMON /BLOCK/ QN1,QN2,QO,QP,QM,QPP,QPS,QSP,QSS,QPHI	GR010090
C		GR010100
C	CALCULATE X	GR010110
C		GR010120
	Q1=QN1*QN1	GR010130
	Q2=QN2*QN2	GR010140
	IF(QN1.EQ.0.D0) THEN	GR010150
	XX=(T*T-Q2)/Q2	GR010160
	ELSE IF (QN2.EQ.0.D0) THEN	GR010170
	XX=(T*T-Q1*A*A)/Q1	GR010180
	ELSE	GR010190
	B = ((Q1+Q2)*T*T -(Q1*A*A-Q2)*(Q1-Q2))/((Q1-Q2)**2)	GR010200
	AC= ((Q1*A*A-Q2)**2+T*T*(T*T-2.*Q1*A*A-2.*Q2))	GR010210
1	/((Q1-Q2)**2)	GR010220
	BAC =B*B -AC	GR010230
	IF (BAC.LE.0.0) BAC = 0.0	GR010240
	XX = B-DSQRT(BAC)	GR010250
	END IF	GR010260
	X = DSQRT(XX)	GR010270
	SX = DSQRT(1.+XX)	GR010280
	SAX = DSQRT(A*A+XX)	GR010290
C		GR010300
C	CALCULATE R'S, Q'S, AND QPHI	GR010310
C		GR010320
	QA= (1.+2.*XX)**2	GR010330
	QB= 4.*SAX*SX*XX	GR010340
	QO= (QA+QB)/(QA-QB)	GR010350
	QP= -4.*X*A*SX*(1.+2.*XX)/(QA-QB)	GR010360
	QM= 4.*X*SAX*(1.+2.*XX)/((QA-QB)*A)	GR010370
	QPP= 2.*SAX*QA/((QA-QB)**2)	GR010380
	QPS= SAX/(QA-QB)	GR010390
	QSP=-SAX/(QA-QB)	GR010400
	QSS= -2.*SAX*QB/((QA-QB)**2)	GR010410
	QPHI= 2.*(QN1/SAX + QN2/SX)	GR010420
	RETURN	GR010430
	END	GR010440

C		GR010450
C	*****	GR010460
C		GR010470
	INTEGER FUNCTION BINO(J1,J2)	GR010480
C		GR010490
C	*****	GR010500
C		GR010510
	M = J1-2	GR010520
	N = J2-2	GR010530
	IF ((M.EQ.N).OR.(N.EQ.0)) THEN	GR010540
	BINO=1	GR010550
	RETURN	GR010560
	ELSE IF ((M.LT.0).OR.(N.LT.0).OR.(M.LT.N)) THEN	GR010570
	BINO = 0	GR010580
	RETURN	GR010590
	ELSE	GR010600
	K=M-N	GR010610
	IN = 1	GR010620
	IP = M	GR010630
	IQ = 1	GR010640
	DO 20 I=1,K	GR010650
	IN = IN*IP/IQ	GR010660
	IP = IP-1	GR010670
	IQ = IQ +1	GR010680
20	CONTINUE	GR010690
	BINO = IN	GR010700
	RETURN	GR010710
	ENDIF	GR010720
	END	GR010730
C	*****	GR010740
C		GR010750
	FUNCTION DXM(D,M)	GR010760
C		GR010770
C	*****	GR010780
C		GR010790
C	THE FUNCTION DXM(D,M) IS	GR010800
C	THE SAME AS D**M. BOTH D AND DXM MUST BE DECLARED IN	GR010810
C	THE MAIN PROGRAM AS DOUBLE PRECISION VARIABLES.	GR010820
C		GR010830
	DOUBLE PRECISION DXM,D	GR010840
	IF (M.EQ.0) DXM=1.0D0	GR010850
	IF (M.GT.0) GO TO 11	GR010860
	IF (M.LT.0) GO TO 22	GR010870
	RETURN	GR010880
C		GR010890
11	DXM=1.0D0	GR010900
	DO 10 I=1,M	GR010910
	DXM=DXM*D	GR010920
10	CONTINUE	GR010930
	RETURN	GR010940
C		GR010950
22	DXM=1.0D0	GR010960
	N=-M	GR010970
	DO 20 I=1,N	GR010980
	DXM=DXM/D	GR010990

20	CONTINUE	GR011000
	RETURN	GR011010
C		GR011020
	END	GR011030
C	*****	GR011040
C		GR011050
	SUBROUTINE SFWAVE(ALPHA,XR,TTIME,INDEX,DISPL)	GR011060
C		GR011070
C	*****	GR011080
C	SFWAVE	GR011090
C	SUBROUTINE TO COMPUTE THE SURFACE DISPLACEMENT	GR011100
C	INPUTS:	GR011110
C	ALPHA=RATIO OF SHEAR WAVE SPEED TO LONG. WAVE SPEED	GR011120
C	XR =DISTANCE BETWEEN SOURCE AND DETECTOR	GR011130
C	=ACTUAL DIST./H	GR011140
C	TTIME =NONDIMENSIONALIZED TTIME	GR011150
C	=ACTUAL TIME*SHEAR WAVE SPEED / H	GR011160
C	INDEX=SUBSCRIPT OF THE GREEN'S FUNCTION	GR011170
C	e.g. 11,22,33,12,13 or 131,113, etc.	GR011180
C		GR011190
C	OUTPUT:	GR011200
C	DISPL=NONDIMENSIONALIZED DISPLACEMENT	GR011210
C	=ACTUAL DISPL.* PI * SHEAR MODULUS * H / FORCE	GR011220
C	NOTE:	GR011230
C	THE PARAMETERS IN COMMON BLKS 0,1, & 5 SHOULD BE ENTERED	GR011240
C	BEFORE THE FIRST CALL OF THIS SUBROUTINE,	GR011250
C	I. E. CALL INIT(ALPHA,XR,ZD,INDEX) FIRST.	GR011260
C		GR011270
C	SUBROUTINES REQUIRED:	GR011280
C		GR011290
C	DSFINT,DQR,QU,COMMON,RAYRT & INTEGRATION ROUTINES	GR011300
C		GR011310
C	REMARK: THE COMPUTATION IS DONE IN Y-PLANE, 4th QUARD. ON A	GR011320
C	THREE LEGGED U-PATH TO REPLACE THE INTEGRATION FROM	GR011330
C	-V TO V.	GR011340
C		GR011350
C		GR011360
C		GR011370
	IMPLICIT DOUBLE PRECISION (D)	GR011380
	IMPLICIT DOUBLE COMPLEX (Y)	GR011390
	COMMON /BLK0/DXD,DT,DZ,DA	GR011400
	COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR011410
	COMMON /BLK5/YR,Y1,Y2	GR011420
C		GR011430
	N1=0	GR011440
	N2=0	GR011450
	DYR=DBLE(YR)	GR011460
	DT=DBLE(TTIME)	GR011470
	DZERO=0.000	GR011480
	DMONEH=-0.500	GR011490
	DSQRYR=DSQRT(DYR)	GR011500
	RWSPD=DSQRYR	GR011510
C		GR011520
	DPF=1.200/DBLE(ALPHA)	GR011530
	DV=DXD/DT	GR011540

IF (TTIME.LE.(ALPHA*XR)) THEN	GR011550
DISPL=0.000	GR011560
RETURN	GR011570
ELSE	GR011580
DISPL=0.00	GR011590
DISPL=DISPL+DSFING(DZERO,DMONEH,1)	GR011600
2 +DSFING(DV,DPF,2)	GR011610
3 +DSFING(DMONEH,DZERO,3)	GR011620
END IF	GR011630
END	GR011640
C*****	GR011650
C	GR011660
DOUBLE PRECISION FUNCTION DSFING(DINIT,DEND,IPATH)	GR011670
C	GR011680
C*****	GR011690
C	GR011700
IMPLICIT DOUBLE PRECISION (D)	GR011710
C	GR011720
DOUBLE PRECISION EPS,R,E,W(50,6),FMIN,FMAX,F	GR011730
INTEGER NINT,NMAX,KF,IFLAG	GR011740
LOGICAL RST	GR011750
C	GR011760
C	GR011770
EXTERNAL DQR1,DQR2,DQR3	GR011780
C	GR011790
EPS = 1.D-8	GR011800
RST = .FALSE.	GR011810
NMAX = 50	GR011820
NINT = 1	GR011830
W(1,1) = DINIT	GR011840
W(2,1) = DEND	GR011850
IF (IPATH.EQ.1) THEN	GR011860
CALL DGLQ1(DQR1,DINIT,DEND,EPS,R,E,NINT,RST,W,NMAX,FMIN,FMAX,	GR011870
+ KF,IFLAG)	GR011880
ELSE IF (IPATH.EQ.2) THEN	GR011890
CALL DGLQ1(DQR2,DINIT,DEND,EPS,R,E,NINT,RST,W,NMAX,FMIN,FMAX,	GR011900
+ KF,IFLAG)	GR011910
ELSE IF (IPATH.EQ.3) THEN	GR011920
CALL DGLQ1(DQR3,DINIT,DEND,EPS,R,E,NINT,RST,W,NMAX,FMIN,FMAX,	GR011930
+ KF,IFLAG)	GR011940
ELSE	GR011950
PAUSE 'DSFING: IPATH .NE. 1,2, OR 3'	GR011960
END IF	GR011970
C	GR011980
D2PEI=8.00*DASIN(1.00)	GR011990
DSFING=-R/D2PEI	GR012000
RETURN	GR012010
END	GR012020
C*****	GR012030
C	GR012040
DOUBLE PRECISION FUNCTION DQR1(DL)	GR012050
C	GR012060
C*****	GR012070
IMPLICIT DOUBLE PRECISION (D)	GR012080
EXTERNAL DQR	GR012090

DQR1=DQR(DL,1)	GR012100
RETURN	GR012110
END	GR012120
C*****	GR012130
C	GR012140
DOUBLE PRECISION FUNCTION DQR2(DL)	GR012150
C	GR012160
C*****	GR012170
IMPLICIT DOUBLE PRECISION (D)	GR012180
EXTERNAL DQR	GR012190
DQR2=DQR(DL,2)	GR012200
RETURN	GR012210
END	GR012220
C*****	GR012230
C	GR012240
DOUBLE PRECISION FUNCTION DQR3(DL)	GR012250
C	GR012260
C*****	GR012270
IMPLICIT DOUBLE PRECISION (D)	GR012280
EXTERNAL DQR	GR012290
DQR3=DQR(DL,3)	GR012300
RETURN	GR012310
END	GR012320
C*****	GR012330
C	GR012340
DOUBLE PRECISION FUNCTION DQR(DL,IPATH)	GR012350
C	GR012360
C*****	GR012370
C	GR012380
IMPLICIT DOUBLE COMPLEX (Y)	GR012390
IMPLICIT DOUBLE PRECISION (D)	GR012400
DIMENSION YQD(3,3),YUP(3,3),YQK(3),YFEE(3)	GR012410
COMMON /BLK0/DXD,DT,DZ,DA	GR012420
COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR012430
COMMON /BLK2/YD,Y,YP,YQ,YR0,YRP,YRM,YDELTA,YPSQP1,YQSQP1,	GR012440
1 YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2	GR012450
2 ,YDPHI	GR012460
COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR012470
C	GR012480
C	GR012490
DV=DXD/DT	GR012500
IF (IPATH.EQ.1) THEN	GR012510
Y= DCMPLX(DV,DL)	GR012520
ELSE IF (IPATH.EQ.2) THEN	GR012530
Y= DCMPLX(DL,-0.5D0)	GR012540
ELSE IF (IPATH.EQ.3) THEN	GR012550
DPF=1.2D0/DA	GR012560
Y= DCMPLX(DPF,DL)	GR012570
ELSE	GR012580
PAUSE 'DRQ: IPATH ?'	GR012590
ENDIF	GR012600
CALL SFCOM(DL)	GR012610
CALL QD(YQD,IDUMMY)	GR012620
CALL QU(YUP,IDUMMY)	GR012630
C	GR012640

	YFEE(1)=YAA/(YP*YD)	GR012650
	YFEE(2)=YONE/(YQ*YD)	GR012660
	YFEE(3)=YONE/(YQ*YD)	GR012670
C		GR012680
	YQR = YZERO	GR012690
C		GR012700
	IF (K.EQ.0) THEN	GR012710
	YQK(1)=YONE	GR012720
	YQK(2)=YONE	GR012730
	YQK(3)=YONE	GR012740
	ELSE IF (K.EQ.1) THEN	GR012750
	YQK(1)=YETA1/Y	GR012760
	YQK(2)=YETA1/Y	GR012770
	YQK(3)=YETA1/Y	GR012780
	ELSE IF (K.EQ.2) THEN	GR012790
	YQK(1)=YETA2/Y	GR012800
	YQK(2)=YETA2/Y	GR012810
	YQK(3)=YETA2/Y	GR012820
	ELSE IF (K.EQ.3) THEN	GR012830
	YQK(1)=-YP/Y	GR012840
	YQK(2)=-YQ/Y	GR012850
	YQK(3)=-YQ/Y	GR012860
	ELSE	GR012870
	PAUSE ' FUUNCTION DQR: WRONG INDEX, K .NE. 0,1,2, OR 3'	GR012880
	END IF	GR012890
	DO 100 I=1,3	GR012900
	YQR=YQR+YQD(M,I)*YUP(L,I)*YQK(I)*YFEE(I)	GR012910
100	CONTINUE	GR012920
C		GR012930
	IF ((IPATH.EQ.1).OR.(IPATH.EQ.3)) THEN	GR012940
	YQR=YQR*YI/CDSQRT(DCMPLX(DV*DV,0.D0)-YY)	GR012950
	ELSE	GR012960
	YQR=YQR/CDSQRT(DCMPLX(DV*DV,0.D0)-YY)	GR012970
	END IF	GR012980
	DQR=DBLE(YQR)	GR012990
C		GR013000
	RETURN	GR013010
	END	GR013020
C*****		GR013030
C		GR013040
	SUBROUTINE SFCOM(DL)	GR013050
C		GR013060
C*****		GR013070
C		GR013080
C	TO COMPUTE PARAMETERS DEPENDING ON Y AND TO PUT IN	GR013090
C	A COMMON BLK2	GR013100
C		GR013110
	IMPLICIT DOUBLE COMPLEX (Y)	GR013120
	IMPLICIT DOUBLE PRECISION (D)	GR013130
	DOUBLE PRECISION ETA1,ETA2SQ,DD,AA	GR013140
	COMMON /BLK0/DXD,DT,DZ,DA	GR013150
	COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR013160
	COMMON /BLK2/YD,Y,YP,YQ,YRO,YRP,YRM,YDELTA,YPSQP1,YQSQP1,	GR013170
1	YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2	GR013180
2	,YDPHI	GR013190

	COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR013200
C		GR013210
	AA=DA*DA	GR013220
	DV=DXD/DT	GR013230
	DN1=DBLE(REAL(N1))	GR013240
	DN2=DBLE(REAL(N2))	GR013250
	YZERO=DCMLX(0.00,0.00)	GR013260
	YI=DCMLX(0.00,1.000)	GR013270
	YAA=DCMLX(AA,0.00)	GR013280
	YA=DCMLX(DA,0.00)	GR013290
	YONE=DCMLX(1.000,0.000)	GR013300
	YTWO=DCMLX(2.00,0.00)	GR013310
	YFOUR=DCMLX(4.000,0.00)	GR013320
C		GR013330
	YD=-Y*DCMLX(DT,0.00)	GR013340
	YETA1 = -Y/DCMLX(DV,0.00)	GR013350
	YETA2=CDSQRT(YONE-YETA1*YETA1)	GR013360
	IF (DBLE(Y).LT.0.00) YETA2=-YETA2	GR013370
	YY=Y*Y	GR013380
	YP=CDSQRT(YAA*YY-YONE)	GR013390
	YQ=CDSQRT(YY-YONE)	GR013400
	YPSQP1=YAA*YY	GR013410
	YQSQP1=YY	GR013420
	YQSQM1=YQ*YQ-YONE	GR013430
	YPSQM1=YP*YP-YONE	GR013440
C		GR013450
	YSQRP1=Y*YA	GR013460
	YSQRQ1=Y	GR013470
	YDELTA=YQSQM1*YQSQM1+YFOUR*YP*YQ	GR013480
	YM=(YQSQM1*YQSQM1-YFOUR*YP*YQ)	GR013490
	YR0=YM/YDELTA	GR013500
	YRP=-(YFOUR*YP*YQSQM1)/(YA*YDELTA)	GR013510
	YRM=(YR0*YR0-YONE)/YRP	GR013520
C		GR013530
	RETURN	GR013540
	END	GR013550
C*****		GR013560
C		GR013570
	SUBROUTINE QU(YUP,IDUMMY)	GR013580
C		GR013590
C*****		GR013600
C		GR013610
C	TO COMPUTE QU(I,J) BASED ON EQ. 6.4	GR013620
C		GR013630
	IMPLICIT DOUBLE COMPLEX (Y)	GR013640
	COMMON /BLK0/DXD,DT,DZ,DA	GR013650
	COMMON /BLK1/KASE,M,L,K,N1,N2,D,DTHETA	GR013660
	COMMON /BLK2/YD,Y,YP,YQ,YR0,YRP,YRM,YDELTA,YPSQP1,YQSQP1,	GR013670
1	YPSQM1,YQSQM1,YSQRP1,YSQRQ1,YY,YA,YAA,YETA1,YETA2	GR013680
2	,YDPHI	GR013690
	COMMON /BLK3/YI,YONE,YTWO,YTHREE,YFOUR,YEIGHT,YZERO	GR013700
C		GR013710
	DIMENSION YUP(3,3)	GR013720
C		GR013730
	IDUMMY=0	GR013740

YUP(1,1)=YETA1/YSQRP1	GR013750
YUP(1,2)=-YETA1*YQ/YSQRQ1	GR013760
YUP(1,3)=-YETA2	GR013770
YUP(2,1)=YETA2/YSQRP1	GR013780
YUP(2,2)=-YETA2*YQ/YSQRQ1	GR013790
YUP(2,3)=YETA1	GR013800
YUP(3,1)=-YP/YSQRP1	GR013810
YUP(3,2)=-YONE/YSQRQ1	GR013820
YUP(3,3)=YZERO	GR013830
C	GR013840
RETURN	GR013850
END	GR013860
C*****	GR013870
C	GR013880
SUBROUTINE RAYRT(YR,YI1,YI2,ALPHA)	GR013890
C	GR013900
C*****	GR013910
C	GR013920
C TO SOLVE RAYLEIGH EQUITION	GR013930
C	GR013940
C INPUT: ALPHA = SHEAR WAVE SPEED / LONG. WAVE SPEED	GR013950
C	GR013960
C OUTPUT: YR,YI1,YI2 ARE THE ROOTS TO	GR013970
C	GR013980
C $Y*Y*Y-8*Y*Y+8*(3-2*AA)*Y-16*(1-AA) = 0$	GR013990
C	GR014000
DOUBLE COMPLEX YR,YI1,YI2	GR014010
DOUBLE PRECISION A,AA,RT1,RT2,RT3	GR014020
DIMENSION A(4),RT1(2),RT2(2),RT3(2)	GR014030
C	GR014040
IF((ALPHA.LE.0.).OR.(ALPHA.GE.1.0)) THEN	GR014050
WRITE(6,*)'RAYRT: WRONG VALUE OF ALPHA'	GR014060
WRITE(6,*) ALPHA	GR014070
WRITE(6,*)'RAYRT: COMPUTATION PAUSED'	GR014080
PAUSE	GR014090
ENDIF	GR014100
C	GR014110
AA=ALPHA*ALPHA	GR014120
TOL= 1.E-37	GR014130
A(1) = 1.0	GR014140
A(2) = -8.000	GR014150
A(3) = 8.*(3.-2.*AA)	GR014160
A(4) = -16.*(1.-AA)	GR014170
C	GR014180
CALL CUBIC(A,RT1,RT2,RT3,TOL,NOROOT)	GR014190
C	GR014200
C IF (ALPHA.LT.0.567) THEN	GR014210
C	GR014220
C ONLY ONE REAL ROOT	GR014230
C	GR014240
C ELSE	GR014250
C	GR014260
C ALL THREE ROOTS ARE REAL	GR014270
C	GR014280
C IF ((RT1(1).LT.RT2(1)).AND.(RT1(1).LT.RT3(1)))THEN	GR014290

YR=DCMLX(RT1(1),RT1(2))	GR014300
YI1=DCMLX(RT2(1),RT2(2))	GR014310
YI2=DCMLX(RT3(1),RT3(2))	GR014320
ELSEIF ((RT2(1).LT.RT1(1)).AND.(RT2(1).LT.RT3(1))) THEN	GR014330
YI1=DCMLX(RT1(1),RT1(2))	GR014340
YR =DCMLX(RT2(1),RT2(2))	GR014350
YI2=DCMLX(RT3(1),RT3(2))	GR014360
ELSEIF ((RT3(1).LT.RT1(1)).AND.(RT3(1).LT.RT2(1)))THEN	GR014370
YI2=DCMLX(RT1(1),RT1(2))	GR014380
YI1=DCMLX(RT2(1),RT2(2))	GR014390
YR =DCMLX(RT3(1),RT3(2))	GR014400
ENDIF	GR014410
RETURN	GR014420
END	GR014430
C*****	GR014440
C	GR014450
SUBROUTINE TIMEARRI(A,R,ZD,NRAY,RCN,TA,CN)	GR014460
C	GR014470
C*****	GR014480
C	GR014490
C 9/27/82	GR014500
C	GR014510
C	GR014520
DIMENSION Z1(4),Z2(4), TA(NRAY),CN(NRAY,3),RCN(NRAY,3)	GR014530
DOUBLE COMPLEX YR,YI1,YI2	GR014540
ITER=100	GR014550
NRAY=NRAY	GR014560
I=0	GR014570
LM=1	GR014580
C Z = ZD+0.5 ; ZD refers to origin at the center of the plate	GR014590
Z = ZD+0.5	GR014600
Z1(1)=0.0	GR014610
Z2(1)=Z	GR014620
Z1(2)=Z	GR014630
Z2(2)=0.0	GR014640
Z1(3)=-Z	GR014650
Z2(3)=0.0	GR014660
Z1(4)=0.0	GR014670
Z2(4)=-Z	GR014680
C	GR014690
IF (Z.EQ.1.0) Z2(1)=0.0	GR014700
OREVEN=FLOAT(INT(Z))*0.5	GR014710
DO 100 J=1,21	GR014720
DO 200 K=1,12	GR014730
JJ=J-1	GR014740
KK=K-1	GR014750
IF ((K.EQ.1).AND.(J.EQ.1)) THEN	GR014760
IF(Z.EQ.1.0) THEN	GR014770
C THREE SURFACE WAVE ARRIVALS:	GR014780
I=I+1	GR014790
CN(I,1)=-1.	GR014800
CN(I,2)=0.	GR014810
CN(I,3)=R*A	GR014820
TA(I)=CN(I,3)	GR014830
I=I+1	GR014840

CN(I,1)=-1.	GR014850
CN(I,2)=-1.	GR014860
CN(I,3)=R	GR014870
TA(I)=CN(I,3)	GR014880
CALL RAYRT(YR,YI1,YI2,A)	GR014890
I=I+1	GR014900
CN(I,1)=0.	GR014910
CN(I,2)=0.	GR014920
CN(I,3)=R/DSQRT(DBLE(YR))	GR014930
TA(I)=CN(I,3)	GR014940
ENDIF	GR014950
GOTO 200	GR014960
ENDIF	GR014970
IF ((Z.EQ.0.0).OR.(Z.EQ.1.0)) LM=4	GR014980
IF (((FLOAT(JJ+KK))/2.0-OREVEN).EQ.FLOAT((JJ+KK)/2))	GR014990
1 GO TO 200	GR015000
DO 300 LL=1,4,LM	GR015010
C1=JJ+Z1(LL)	GR015020
C2=KK+Z2(LL)	GR015030
IF ((C1.LT.0.0) .OR. (C2.LT.0.0)) GO TO 300	GR015040
I=I+1	GR015050
IF (I.GT.NRAY) GO TO 99	GR015060
CALL TARRV (A,C1,C2,R,ITER,ERROR,T)	GR015070
IF (ERROR.GT.1.E-6) WRITE (6,603) I,C1,C2,ERROR	GR015080
CN(I,1)=C1	GR015090
CN(I,2)=C2	GR015100
CN(I,3)=T	GR015110
TA(I)=T	GR015120
IF (C1.GT.0.)GO TO 300	GR015130
APJ=A*C2/SQRT(1.-A*A)	GR015140
IF(R.LE.APJ) GO TO 300	GR015150
C2=-C2	GR015160
CALL TARRV (A,C1,C2,R,ITER,ERROR,T)	GR015170
I=I+1	GR015180
IF (I.GT.NRAY) GO TO 99	GR015190
CN(I,1)=C1	GR015200
CN(I,2)=C2	GR015210
CN(I,3)=T	GR015220
TA(I)=T	GR015230
300 CONTINUE	GR015240
200 CONTINUE	GR015250
100 CONTINUE	GR015260
C	GR015270
99 CONTINUE	GR015280
CALL RSORT(CN,TA,RCN,NRAY,3,0)	GR015290
C	GR015300
603 FORMAT(2X,'ERROR IS LARGER THAN 1.E-6 ',I3,3F10.5)	GR015310
RETURN	GR015320
END	GR015330
C*****	GR015340
C	GR015350
SUBROUTINE TARRV(A,C1,C2,R,ITER,ERR,TARR)	GR015360
C	GR015370
C*****	GR015380
C	GR015390

	IMPLICIT DOUBLE PRECISION (D)	GR015400
C		GR015410
	NUM = 0	GR015420
	DT1=0.1570796326794901	GR015430
	DT2=0.000	GR015440
	DA=DBLE(A)	GR015450
	DAA=DA*DA	GR015460
	DN1=DBLE(C1)	GR015470
	DN2=DBLE(C2)	GR015480
	DR=DBLE(R)	GR015490
	IF (C1.EQ.0.0) GO TO 2	GR015500
	IF (C2.EQ.0.0) GO TO 3	GR015510
C		GR015520
	1 CONTINUE	GR015530
	NUM = NUM + 1	GR015540
	DT=(DT1+DT2)/2.000	GR015550
	DAASS=DAA*DSIN(DT)*DSIN(DT)	GR015560
	DAASS=1.000-DAASS	GR015570
	DCS2=DSQRT(DAASS)	GR015580
	DRT = DN1*DTAN(DT) + DN2*DA*DSIN(DT)/DCS2	GR015590
C	WRITE (6,20) NUM,DT,DRT,C1,C2	GR015600
	IF (NUM.GE.ITER) GO TO 10	GR015610
	IF (DABS(DRT-DR).LE.5.D-11) GO TO 10	GR015620
	IF (DRT.LE.DR) DT2=DT	GR015630
	IF (DRT.GT.DR) DT1=DT	GR015640
	GO TO 1	GR015650
C		GR015660
C 20	FORMAT ('NUM= ',I3,' T= ',D17.10,' RT= ',D17.10,2(2X,F4.0))	GR015670
10	ITER=NUM	GR015680
	DTARR=DA*DN1/DCOS(DT) + DN2/DCS2	GR015690
	DERR=DABS(DR-DRT)	GR015700
	ERR=DERR	GR015710
	TARR=DTARR	GR015720
	RETURN	GR015730
C		GR015740
	2 IF (C2.LT.0.0) GO TO 4	GR015750
	DTARR=DSQRT(DR*DR + DN2*DN2)	GR015760
	TARR=DTARR	GR015770
	ERR=0.0	GR015780
	RETURN	GR015790
C		GR015800
	3 DTARR=DSQRT(DR*DR+DN1*DN1)*DA	GR015810
	TARR=DTARR	GR015820
	ERR=0.0	GR015830
	RETURN	GR015840
C		GR015850
	4 DTARR=DR*DA - DN2*DSQRT(1.00-DAA)	GR015860
C	NOTE HERE DN2 < 0	GR015870
	TARR=DTARR	GR015880
	ERR=0.0	GR015890
	RETURN	GR015900
C		GR015910
	END	GR015920
C	*****	GR015930
C		GR015940

SUBROUTINE CUBIC(A,RT1,RT2,RT3,TOL,NOROOT)	GR015950
C	GR015960
C*****	GR015970
C	GR015980
C THIS SUBROUTINE FINDS THE ROOTS OF A CUBIC EQUATION	GR015990
C	GR016000
C INPUTS:	GR016010
C A COEFFICIENTS, A(4)	GR016020
C	GR016030
C OUTPUTS:	GR016040
C RT1,RT2,RT3 THREE ROOTS	GR016050
C RT1(1)=REAL PART OF RT1, ETC.	GR016060
C	GR016070
C THE ROUTINE CALLS LINCNG,AND QUADRT	GR016080
C	GR016090
DOUBLE PRECISION A,B,C,RT1,RT2,RT3,ZERO,X,Y,ZZ,SQT3,RC27,ANG	GR016100
DIMENSION A(4),RT1(2),RT2(2),RT3(2),B(2),C(4)	GR016110
C	GR016120
ZERO=TOL/10.	GR016130
SQT3=DSQRT(3.0D0)	GR016140
RC27=1.0/27.0	GR016150
C	GR016160
IF (DABS(A(1))-ZERO) 7,7,12	GR016170
7 CALL QUADRT (A(2),RT1,RT2,TOL,NOROOT)	GR016180
RETURN	GR016190
C	GR016200
12 NOROOT=3	GR016210
IT=0	GR016220
C	GR016230
DO 1 I=2,4	GR016240
A(I)=A(I)/A(1)	GR016250
1 CONTINUE	GR016260
A(1)=1.0	GR016270
C	GR016280
IF (DABS(A(2)).LE.ZERO) GOTO 2	GR016290
NDA=3	GR016300
B(1)=1.0	GR016310
B(2)=-(A(2)/3.0)	GR016320
CALL LINCNG(A,NDA,B,C)	GR016330
IT=1	GR016340
GO TO 4	GR016350
2 DO 3 I=1,4	GR016360
C(I)=A(I)	GR016370
3 CONTINUE	GR016380
C	GR016390
4 X = C(4)*C(4)*0.25+(C(3)**3)*RC27	GR016400
IF (X.LT.0.0) GO TO 3	GR016410
X = DSQRT(X)	GR016420
Y = -(C(4)*0.5)	GR016430
I=1	GR016440
RT1(I) =Y+X	GR016450
C	GR016460
5 N = 0	GR016470
IF(RT1(I).LT.0.0) N=1	GR016480
IF (DABS(RT1(I)).LE.ZERO) GO TO 6	GR016490

RT1(I) = DABS(RT1(I))**(1./3.)	GR016500
IF (N.EQ.1) RT1(I) = -RT1(I)	GR016510
6 IF (I.EQ.2) GOTO 8	GR016520
I = 2	GR016530
RT1(I) = Y-X	GR016540
GO TO 5	GR016550
C	GR016560
8 RT2(2) = ((RT1(1) - RT1(2))*0.5)*SQRT3	GR016570
RT1(1) = RT1(1) + RT1(2)	GR016580
RT1(2) = 0.0	GR016590
RT2(1) = -RT1(1)*0.5	GR016600
RT3(1) = RT2(1)	GR016610
RT3(2) = -RT2(2)	GR016620
GO TO 11	GR016630
C	GR016640
9 ZZ = DABS(C(3))	GR016650
X = -(C(4)*0.5)/DSQRT((ZZ**3)*RC27)	GR016660
ANG = DACOS(X)	GR016670
Y = 2.0*(DSQRT(ZZ/3.0))	GR016680
ANG=ANG/3.0	GR016690
RT1(1) = Y*DCOS(ANG)	GR016700
RT2(1) = Y*DCOS(ANG + 2.0943951024D0)	GR016710
RT3(1) = Y*DCOS(ANG + 4.1887902047D0)	GR016720
RT1(2) = 0.0	GR016730
RT2(2) = 0.0	GR016740
RT3(2) = 0.0	GR016750
C	GR016760
11 IF (IT.EQ.1) THEN	GR016770
RT1(1) = RT1(1) + B(2)	GR016780
RT2(1) = RT2(1) + B(2)	GR016790
RT3(1) = RT3(1) + B(2)	GR016800
ELSE	GR016810
ENDIF	GR016820
RETURN	GR016830
END	GR016840
C*****	GR016850
C	GR016860
SUBROUTINE LINCNG(A,NDA,B,C)	GR016870
C	GR016880
C*****	GR016890
C	GR016900
C TO MAKE A LINEAR CHANGE OF VARIABLES	GR016910
C IN A GIVEN POLYNOMIAL	GR016920
C	GR016930
DOUBLE PRECISION A,B,C,BIPWR	GR016940
DIMENSION A(1),B(2),C(1)	GR016950
C	GR016960
BIPWR = 1.0D0	GR016970
NZ = NDA +1	GR016980
C	GR016990
DO 4 I=1,NZ	GR017000
C(I) = A(I)	GR017010
4 CONTINUE	GR017020
C	GR017030
6 DO 8 I=2,NZ	GR017040

C(I) = C(I) + C(I-1)*B(2)	GR017050
8 CONTINUE	GR017060
C	GR017070
C(NZ) = C(NZ)*BIPWR	GR017080
NZ = NZ - 1	GR017090
BIPWR = BIPWR*B(1)	GR017100
IF (NZ.GT.1) GO TO 6	GR017110
C(1) = C(1)*BIPWR	GR017120
RETURN	GR017130
END	GR017140
C*****	GR017150
C	GR017160
SUBROUTINE DGLQ1(F,A,B,EPS,R,E,NINT,RST,W,NMAX,FMIN,FMAX,KF,IFLAG)	GR017170
C	GR017180
C*****	GR017190
C	GR017200
C	GR017210
*** ALL REAL VARIABLES ARE TYPED DOUBLE PRECISION	GR017220
*** 9/20/82 NH	GR017230
C	GR017240
C	GR017250
DGLQ1 IS A SUBROUTINE FOR THE AUTOMATIC EVALUATION	GR017260
OF DEFINITE INTEGRALS OF A USER DEFINED FUNCTION	GR017270
OF ONE VARIABLE PROVIDING FLEXIBLE USAGE.	GR017280
C	GR017290
FOR AN EASY TO USE VERSION SEE SUBROUTINE DGLQ.	GR017300
C	GR017310
CAPABILITIES OF DGLQ1 (IN ADDITION TO THOSE OF DGLQ)	GR017320
INCLUDE:	GR017330
ABILITY TO RESTART A CALCULATION TO GREATER	GR017340
ACCURACY WITHOUT PENALTY...	GR017350
ABILITY TO SPECIFY AN INITIAL PARTITION OF	GR017360
THE INTEGRATION INTERVAL...	GR017370
ABILITY TO INCREASE THE WORK SPACE TO HANDLE	GR017380
MORE DIFFICULT PROBLEMS...	GR017390
OUTPUT OF LARGEST/smallest INTEGRAND VALUE FOR	GR017400
APPLICATIONS SUCH AS GRAPHING...	GR017410
C	GR017420
ARGUMENTS IN THE CALL SEQUENCE	GR017430
C	GR017440
F (INPUT) THE NAME OF YOUR INTEGRAND FUNCTION.	GR017450
THIS NAME MUST APPEAR IN AN EXTERNAL STATEMENT	GR017460
IN ANY PROGRAM WHICH CALLS DGLQ1.	GR017470
YOU MUST WRITE F IN THE FORM	GR017480
FUNCTION F(X)	GR017490
F=(EVALUATE INTEGRAND AT THE POINT X)	GR017500
RETURN	GR017510
C	GR017520
END	GR017530
A	GR017540
B (INPUT) ENDPOINTS OF INTEGRATION INTERVAL	GR017550
EPS (INPUT) ACCURACY TO WHICH THE INTEGRAL IS TO BE CALCULATED	GR017560
DGLQ1 WILL TRY TO ACHIEVE RELATIVE ACCURACY,	GR017570
E.G. SET EPS=.01 FOR 2 DIGITS, .001 FOR 3, ETC	GR017580
R (OUTPUT) THE ESTIMATE OF THE INTEGRAL	GR017590
E (OUTPUT) THE ESTIMATE OF THE ABSOLUTE ERROR IN R.	

C	NINT	(INPUT	GR017600
C		OUTPUT)	GR017610
C		AS AN OUTPUT QUANTITY, NINT GIVES THE	GR017620
C		NUMBER OF SUBINTERVALS IN THE FINAL	GR017630
C		PARTITION OF [A,B].	GR017640
C		AS AN INPUT QUANTITY, NINT MUST BE SET TO	GR017650
C		THE NUMBER OF SUBINTERVALS IN THE INITIAL	GR017660
C		PARTITION OF [A,B]. FOR MOST PROBLEMS	GR017670
C		THIS IS JUST 1, THE INTERVAL [A,B] ITSELF.	GR017680
C		NINT IS USEFUL IF YOU WOULD LIKE TO HELP	GR017690
C		DGLQ1 LOCATE A DIFFICULT SPOT ON [A,B].	GR017700
C		IN THIS REGARD NINT IS USED ALONG	GR017710
C		WITH THE ARRAY W (SEE BELOW). IF YOU SET	GR017720
C		NINT=1 IT IS NOT NECESSARY TO BE CONCERNED	GR017730
C		WITH W, EXCEPT THAT IT MUST BE DIMENSIONED...	GR017740
C		AS AN EXAMPLE OF MORE GENERAL APPLICATIONS,	GR017750
C		IF [A,B]=[0,1] BUT THE INTEGRAND JUMPS AT 0.3	GR017760
C		IT WOULD BE WISE TO SET NINT=2 AND THEN SET	GR017770
C		W(1,1)=0.0 (LEFT ENDPOINT)	GR017780
C		W(2,1)=0.3 (SINGULAR POINT)	GR017790
C		W(3,1)=1.0 (RIGHT ENDPOINT)	GR017800
C		IF YOU SET NINT GREATER THAN 1, BE SURE TO	GR017810
C		CHECK THAT YOU HAVE ALSO SET	GR017820
C		W(1,1)=A AND W(NINT+1,1)=B	GR017830
C	RST	(INPUT) A LOGICAL VARIABLE (E.G. TRUE OR FALSE)	GR017840
C		SET RST=.FALSE. FOR INITIAL CALL TO DGLQ1	GR017850
C		SET RST=.TRUE. FOR A SUBSEQUENT CALL,	GR017860
C		E.G. ONE FOR WHICH MORE ACCURACY IS	GR017870
C		DESIRED (SMALLER EPS). A RESTART ONLY	GR017880
C		MAKES SENSE IF THE PRECEDING CALL RETURNED	GR017890
C		WITH A VALUE OF IFLAG (SEE BELOW) LESS THAN	GR017900
C		ON A RESTART YOU MAY NOT CHANGE THE VALUES	GR017910
C		ANY ARGUMENTS IN THE CALL SEQUENCE, EXCEPT	GR017920
C	W(NMAX,6)	W IS AN ARRAY USED FOR SCRATCH STORAGE BY DGLQ1.	GR017930
C		YOU MUST INCLUDE A DIMENSION STATEMENT IN	GR017940
C		YOUR CALLING PROGRAM TO ALLOCATE THIS STORAGE.	GR017950
C		THIS SHOULD BE OF THE FORM	GR017960
C		DIMENSION W(NMAX,6)	GR017970
C		WHERE NMAX IS AN INTEGER. AN ADEQUATE VALUE OF	GR017980
C		NMAX IS 50.	GR017990
C	NMAX	(INPUT) AN INTEGER EQUAL TO THE FIRST SUBSCRIPT IN THE	GR018000
C		DIMENSION STATEMENT FOR THE ARRAY W. THIS IS	GR018010
C		ALSO EQUAL TO THE MAXIMUM NUMBER OF SUBINTERVAL	GR018020
C		PERMITTED IN THE INTERNAL PARTITION OF [A,B].	GR018030
C		A VALUE OF 50 IS AMPLE FOR MOST PROBLEMS.	GR018040
C			GR018050
C	FMIN	(OUTPUT) THE SMALLEST AND LARGEST VALUES OF THE INTEGRAND	GR018060
C		WHICH OCCURRED DURING THE CALCULATION. THE	GR018070
C		ACTUAL INTEGRAND RANGE ON [A,B] MAY, OF COURSE	GR018080
C		BE GREATER BUT PROBABLY NOT BY MORE THAN 10%.	GR018090
C	KF	(OUTPUT) THE ACTUAL NUMBER OF INTEGRAND EVALUATIONS USED	GR018100
C		BY DGLQ1 TO APPROXIMATE THIS INTEGRAL. KF	GR018110
C		WILL ALWAYS BE AT LEAST 30.	GR018120
C	IFLAG	(OUTPUT) TERMINATION FLAG...POSSIBLE VALUES ARE	GR018130
C		0 NORMAL COMPLETION, E SATISFIES	GR018140

C	E<EPS AND E<EPS*DABS(R)	GR018150
C	1 NORMAL COMPLETION, E SATISFIES	GR018160
C	E<EPS, BUT NOT RELATIVE ERROR REQUEST	GR018170
C	2 NORMAL COMPLETION, E SATISFIES	GR018180
C	E<EPS*DABS(R)	GR018190
C	3 NORMAL COMPLETION BUT EPS WAS TOO SMALL TO	GR018200
C	SATISFY ABSOLUTE OR RELATIVE ERROR REQUEST.	GR018210
C	4 ABORTED CALCULATION BECAUSE OF SERIOUS ROUNDING	GR018220
C	ERROR. PROBABLY E AND R ARE CONSISTENT.	GR018230
C	5 ABORTED CALCULATION BECAUSE OF INSUFFICIENT STORAG	GR018240
C	R AND E ARE CONSISTENT. PERHAPS INCREASING NMAG	GR018250
C	WILL PRODUCE BETTER RESULTS.	GR018260
C	6 ABORTED CALCULATION BECAUSE OF SERIOUS DIFFICULTY	GR018270
C	MEETING YOUR ERROR REQUEST.	GR018280
C	7 ABORTED CALCULATION BECAUSE EITHER EPS, NINT OR NGR	GR018290
C	HAS BEEN SET TO AN ILLEGAL VALUE.	GR018300
C	8 ABORTED CALCULATION BECAUSE YOU SET NINT>1 BUT FOG	GR018310
C	TO SET W(1,1)=A AND W(NINT+1,1)=B	GR018320
C		GR018330
C	T Y P I C A L P R O B L E M S E T U P	GR018340
C		GR018350
C	DIMENSION W(50,6)	GR018360
C	LOGICAL RST	GR018370
C	EXTERNAL F	GR018380
C	A=0.0	GR018390
C	B=1.0	GR018400
C	W(1,1)=A	GR018410
C	W(2,1)=.3 [SET INTERNAL PARTITION POINT AT .3]	GR018420
C	W(3,1)=B	GR018430
C	NINT=2 [INITIAL PARTITION HAS 2 INTERVALS]	GR018440
C	RST=.FALSE.	GR018450
C	EPS=.001	GR018460
C	NMAX=50	GR018470
C		GR018480
C	CALL DGLQ1(F,A,B,EPS,R,E,NINT,RST,W,NMAX,FMIN,FMAX,KF,IFLAG)	GR018490
C		GR018500
C	1 A,B,EPS,R,E,NINT,FMIN,FMAX,KF,IFLAG	GR018510
C	IF(EPS.EQ. .0001 .OR. IFLAG.GE.3)STOP	GR018520
C	RST=.TRUE	GR018530
C	EPS=.0001 [ASK FOR ANOTHER DIGIT]	GR018540
C	GO TO 1	GR018550
C	END	GR018560
C	FUNCTION F(X)	GR018570
C	IF(X.LT. .3)	GR018580
C	1 THEN	GR018590
C	F=X**(0.2D0)*ALOG(X)	GR018600
C	ELSE	GR018610
C	F=DSIN(X)	GR018620
C	ENDIF	GR018630
C	RETURN	GR018640
C	END	GR018650
C		GR018660
C	E N D O F D O C U M E N T A T I O N	GR018670
C		GR018680
C	INTEGER C	GR018690

	DOUBLE PRECISION A,B,E,EB,EPMACH,EPS,FMAX,FMAXL,FMAXR,FMIN,FMINL	GR018700
1	,FMINR,FMN,FMX,R,RAB,RABS,RAV,SIGN,T,TE	GR018710
2	,TE1,TE2,TR,TR1,TR2,UFLOW,W,XM,F	GR018720
	DIMENSION W(NMAX,6)	GR018730
	EXTERNAL F	GR018740
	LOGICAL RST,DEBUG	GR018750
	DATA SIGN /-1.DO/	GR018760
C	THE FOLLOWING DATA ARE FOR PERKIN-ELMER COMPUTERS	GR018770
	DATA EPMACH,UFLOW/Z341000000000000,Z001000000000000/	GR018780
C	THE FOLLOWING DATA ARE FOR VAX COMPUTERS	GR018790
C	DATA EPMACH,UFLOW/Z0000250000000000,Z0000008000000000/	GR018800
C		GR018810
C	EPMACH = Y'341000000000000'	GR018820
C	UFLOW = Y'00100000'	GR018830
C	EPMACH = DBLE(16**(-13))	GR018840
C	UFLOW = DBLE(16**(-65))	GR018850
	IF(A.EQ.B) THEN	GR018860
	R=0.	GR018870
	E=0.	GR018880
	NINT=0	GR018890
	IFLAG=0	GR018900
	KF=1	GR018910
	FMIN=F(A)	GR018920
	FMAX=FMIN	GR018930
	GO TO 20	GR018940
	ENDIF	GR018950
	IF(RST) THEN	GR018960
	IF(IFLAG.LT.3) GO TO 15	GR018970
	GO TO 20	GR018980
	ENDIF	GR018990
	KF=0	GR019000
	IF(EPS .LE. 0. .OR. NMAX .LE. 1 .OR. NINT .LE. 0) THEN	GR019010
	IFLAG=7	GR019020
	GO TO 20	GR019030
	ENDIF	GR019040
	IF(NINT.EQ.1) THEN	GR019050
C	1 THEN	GR019060
	W(1,1)=A	GR019070
	W(2,2)=B	GR019080
	W(1,5)=A	GR019090
	W(1,6)=B	GR019100
	W(2,5)=A	GR019110
	W(2,6)=B	GR019120
	W(1,2)=A+(B-A)/2.DO	GR019130
	W(2,1)=A+(B-A)/2.DO	GR019140
	NINT=2	GR019150
	ELSE	GR019160
	IF(W(1,1).NE.A .AND. W(NINT+1,1).NE.B) THEN	GR019170
	IFLAG=8	GR019180
	GO TO 20	GR019190
	ENDIF	GR019200
	W(1,5)=A	GR019210
	DO 89 I=1,NINT	GR019220
	W(I,2)=W(I+1,1)	GR019230
	W(I,5)=W(I,1)	GR019240

	W(I,6)=W(I,2)	GR019250
89	CONTINUE	GR019260
	ENDIF	GR019270
C		GR019280
	DEBUG=.FALSE.	GR019290
	IFLAG = 0	GR019300
	IROFF=0	GR019310
	RABS=0.0	GR019320
	DO 3 I=1,NINT	GR019330
	CALL GL15T(F,W(I,1),W(I,2),W(I,5),W(I,6),	GR019340
1	W(I,4),W(I,3),RAB,RAV,FMN,FMX)	GR019350
	IF(DEBUG) WRITE(7,*)'INITIALIZE',(W(I,J),J=1,6)	GR019360
	KF=KF+15	GR019370
	IF(I.EQ.1)	GR019380
1	THEN	GR019390
	R=W(I,4)	GR019400
	E=W(I,3)	GR019410
	RABS=RABS+RAB	GR019420
	FMIN=FMN	GR019430
	FMAX=FMX	GR019440
	ELSE	GR019450
	R=R+W(I,4)	GR019460
	E=E+W(I,3)	GR019470
	RABS=RABS+RAB	GR019480
	FMAX=DMAX1(FMAX,FMX)	GR019490
	FMIN=DMIN1(FMIN,FMN)	GR019500
	ENDIF	GR019510
3	CONTINUE	GR019520
	DO 10 I=NINT+1,NMAX	GR019530
	W(I,3) = 0.	GR019540
10	CONTINUE	GR019550
15	CONTINUE	GR019560
C		GR019570
C	MAIN SUBPROGRAM LOOP	GR019580
C		GR019590
	IF(100.DO*EPMACH*RABS.GE.DABS(R) .AND. E.LT.EPS)GO TO 20	GR019600
	EB=DMAX1(100.DO*UFLOW,DMAX1(EPS,50.DO*EPMACH)*DABS(R))	GR019610
	IF(E.LE.EB) GO TO 20	GR019620
	IF (NINT.LT.NMAX)	GR019630
1	THEN	GR019640
	NINT = NINT+1	GR019650
	C = NINT	GR019660
	ELSE	GR019670
	C=0	GR019680
16	IF(C.EQ.NMAX) THEN	GR019690
	IFLAG=5	GR019700
	GO TO 20	GR019710
	ENDIF	GR019720
	C=C+1	GR019730
	IF(W(C,3).GT.0.DO) GO TO 16	GR019740
	END IF	GR019750
C	LOC=ISAMAX(NINT,W(1,3),1)	GR019760
	CALL LSE(W,NMAX,NINT,LOC)	GR019770
	IF(DEBUG)WRITE(7,200)LOC,(W(LOC,I),I=1,6),R,E	GR019780
	XM = W(LOC,1)+(W(LOC,2)-W(LOC,1))/2.	GR019790

IF ((DMAX1(DABS(W(LOC,1)),DABS(W(LOC,2))))).GT.	GR019800
1 ((1.D0+100.D0*EPMACH)*(DABS(XM)+0.1D+04*UFLOW)))	GR019810
2 THEN	GR019820
CALL GL15T(F,W(LOC,1),XM,W(LOC,5),W(LOC,6),	GR019830
1 TR1,TE1,RAB,RAV,FMINL,FMAXL)	GR019840
KF=KF+15	GR019850
IF (TE1.LT.(EB*(XM-W(LOC,1))/(B-A)))	GR019860
A TE1 = TE1*SIGN	GR019870
CALL GL15T(F,XM,W(LOC,2),W(LOC,5),W(LOC,6),	GR019880
1 TR2,TE2,RAB,RAV,FMINR,FMAXR)	GR019890
KF=KF+15	GR019900
FMIN=DMIN1(FMIN,FMINL,FMINR)	GR019910
FMAX=DMAX1(FMAX,FMAXL,FMAXR)	GR019920
IF (TE2.LT.(EB*(W(LOC,2)-XM)/(B-A))) TE2=TE2*SIGN	GR019930
TE = DABS(W(LOC,3))	GR019940
TR = W(LOC,4)	GR019950
W(C,3) = TE2	GR019960
W(C,4) = TR2	GR019970
W(C,1) = XM	GR019980
W(C,2) = W(LOC,2)	GR019990
W(C,5) = W(LOC,5)	GR020000
W(C,6) = W(LOC,6)	GR020010
W(LOC,3) = TE1	GR020020
W(LOC,4) = TR1	GR020030
W(LOC,2) = XM	GR020040
IF(DEBUG)WRITE(7,200)C,(W(C,K),K=1,6)	GR020050
IF(DEBUG)WRITE(7,200)LOC,(W(LOC,K),K=1,6)	GR020060
E = E-TE+(DABS(TE1)+DABS(TE2))	GR020070
R = R-TR+(TR1+TR2)	GR020080
IF(DEBUG)WRITE(7,*)NINT,R,E	GR020090
IF(DABS(DABS(TE1)+DABS(TE2)-TE).LT..001D0*TE) THEN	GR020100
IROFF=IROFF+1	GR020110
IF(IROFF.GE.10) THEN	GR020120
IFLAG=4	GR020130
GO TO 20	GR020140
ENDIF	GR020150
ENDIF	GR020160
ELSE	GR020170
IF (EB.GT.W(LOC,3))	GR020180
1 THEN	GR020190
W(LOC,3) = 0.	GR020200
ELSE	GR020210
IFLAG=6	GR020220
GO TO 20	GR020230
END IF	GR020240
END IF	GR020250
GO TO 15	GR020260
C	GR020270
C ALL EXITS FROM HERE	GR020280
C	GR020290
20 CONTINUE	GR020300
IF(IFLAG.GE.4)RETURN	GR020310
IFLAG=3	GR020320
T=EPS*DABS(R)	GR020330
IF(E.GT.EPS .AND. E.GT.T)RETURN	GR020340

IFLAG=2	GR020350
IF(E.GT.EPS .AND. E.LT.T)RETURN	GR020360
IFLAG=1	GR020370
IF(E.LT.EPS .AND. E.GT.T)RETURN	GR020380
IFLAG=0	GR020390
RETURN	GR020400
200 FORMAT (I4,8(E11.3))	GR020410
END	GR020420
C*****	GR020430
C	GR020440
SUBROUTINE LSE (WORK,NMAX,NI,LOC)	GR020450
C	GR020460
C*****	GR020470
C	GR020480
C THIS SUBPROGRAM FINDS THE CELL IN THE WORK AREA	GR020490
C THAT HAS THE LARGEST ABSOLUTE ERROR AND	GR020500
C RETURNS THE LOCATION OF THAT CELL.	GR020510
C	GR020520
DOUBLE PRECISION ERROR,WORK	GR020530
DIMENSION WORK(NMAX,6)	GR020540
C	GR020550
C INITIALIZE VARIABLES	GR020560
C	GR020570
ERROR = DABS(WORK(1,3))	GR020580
LOC = 1	GR020590
C	GR020600
C MAIN SUBPROGRAM LOOP	GR020610
C	GR020620
DO 20 I=1,NI	GR020630
IF (DABS(WORK(I,3)).GT.ERROR) THEN	GR020640
ERROR = DABS(WORK(I,3))	GR020650
LOC = I	GR020660
END IF	GR020670
20 CONTINUE	GR020680
RETURN	GR020690
END	GR020700
C*****	GR020710
C	GR020720
SUBROUTINE GL15T(F,A,B,XL,XR,R,AE,RA,	GR020730
1 RASC,FMIN,FMAX)	GR020740
C	GR020750
C*****	GR020760
C	GR020770
C***AUTHORS ROBERT PIESSENS AND ELISE DE DONCKER	GR020780
C APPL. MATH. AND PROGR. DIV.- K.U.LEUVEN	GR020790
C DAVID KAHANER, NBS WASHINGTON	GR020800
C 	GR020810
C PURPOSE	GR020820
C TO COMPUTE I = INTEGRAL OF G(X) OVER (A,B),	GR020830
C WITH ERROR ESTIMATE	GR020840
C J = INTEGRAL OF DABS(G) OVER (A,B)	GR020850
C	GR020860
C PARAMETERS	GR020870
C ON ENTRY	GR020880
C F - DOUBLE PRECISION	GR020890

C	FUNCTION SUBPROGRAM DEFINING THE INTEGRAND	GR020900
C	FUNCTION F(X). THE ACTUAL NAME FOR F NEEDS	GR020910
C	TO BE DECLARED E X T E R N A L IN THE	GR020920
C	CALLING PROGRAM.	GR020930
C	THE FUNCTION G(X) IS DEFINED TO BE	GR020940
C	$G(X)=F(\text{PHI}(X))*\text{PHIP}(X)$	GR020950
C	WHERE $\text{PHI}(X)$ IS THE CUBIC GIVEN BY	GR020960
C	THE ARITHMETIC STATEMENT FUNCTION BELOW.	GR020970
C	$\text{PHIP}(X)$ IS ITS DERIVATIVE. THE VARIABLES	GR020980
C	XL AND XR ARE THE LEFT AND RIGHT ENDPOINTS	GR020990
C	OF A PARENT INTERVAL OF WHICH (A,B) IS A PART.	GR021000
C		GR021010
C	A - DOUBLE PRECISION	GR021020
C	LOWER LIMIT OF INTEGRATION	GR021030
C		GR021040
C	B - DOUBLE PRECISION	GR021050
C	UPINTG LIMIT OF INTEGRATION	GR021060
C		GR021070
C	XL - DOUBLE PRECISION	GR021080
C	LEFT ENDPOINT OF PARENT INTERVAL	GR021090
C		GR021100
C	XR - DOUBLE PRECISION	GR021110
C	RIGHT ENDPOINT OF PARENT INTERVAL	GR021120
C		GR021130
C	ON RETURN	GR021140
C	R - DOUBLE PRECISION	GR021150
C	APPROXIMATION TO THE INTEGRAL I	GR021160
C	R IS COMPUTED BY APPLYING THE 15-POINT	GR021170
C	KRONROD RULE (RESK) OBTAINED BY OPTIMAL	GR021180
C	ADDITION OF ABSCISSAE TO THE 7-POINT GAUSS	GR021190
C	RULE (RESG).	GR021200
C		GR021210
C	AE - DOUBLE PRECISION	GR021220
C	ESTIMATE OF THE MODULUS OF THE ABSOLUTE ERROR,	GR021230
C	WHICH SHOULD NOT EXCEED DABS(I-R)	GR021240
C		GR021250
C	RA - DOUBLE PRECISION	GR021260
C	APPROXIMATION TO THE INTEGRAL J	GR021270
C		GR021280
C	RASC - DOUBLE PRECISION	GR021290
C	APPROXIMATION TO THE INTEGRAL OF $DABS(G-I/(B-A))$	GR021300
C	OVER (A,B)	GR021310
C		GR021320
C	FMAX, FMIN - DOUBLE PRECISION	GR021330
C	MAX AND MIN VALUES OF THE FUNCTION F ON (A,B)	GR021340
C	SUBROUTINES OR FUNCTIONS NEEDED	GR021350
C	- F (USER-PROVIDED FUNCTION)	GR021360
C	- R1MACH	GR021370
C	- FORTRAN DABS, DMAX1, DMIN1	GR021380
C		GR021390
C		GR021400
C	C***END PROLOGUE	GR021410
C		GR021420
	DOUBLE PRECISION A,AE,B,DHLGTH,EPMACH,F,FC,FSUM,FVAL1,FVAL2,	GR021430
	* FV1,FV2,HLGTH,RA,RASC,RESG,RESK,RESKH,R,R1MACH,UFLOW,	GR021440

	* WG,WGK,XGK	GR021450
	DOUBLE PRECISION XL,XR,CENTR,ABSC,U,FMAX,FMIN,PHI,PHIP	GR021460
	INTEGER J,JTW,JTWM1	GR021470
C		GR021480
	DIMENSION FV1(7),FV2(7),WG(4),WGK(8),XGK(8)	GR021490
C		GR021500
C	THE ABSCISSAE AND WEIGHTS ARE GIVEN FOR THE INTERVAL (-1,1)	GR021510
C	BECAUSE OF SYMMETRY ONLY THE POSITIVE ABSCISSAE AND THEIR	GR021520
C	CORRESPONDING WEIGHTS ARE GIVEN.	GR021530
C		GR021540
C	XGK - ABSCISSAE OF THE 15-POINT KRONROD RULE	GR021550
C	XGK(2), XGK(4), ... ABSCISSAE OF THE 7-POINT	GR021560
C	GAUSS RULE	GR021570
C	XGK(1), XGK(3), ... ABSCISSAE WHICH ARE OPTIMALLY	GR021580
C	ADDED TO THE 7-POINT GAUSS RULE	GR021590
C		GR021600
C	WGK - WEIGHTS OF THE 15-POINT KRONROD RULE	GR021610
C		GR021620
C	WG - WEIGHTS OF THE 7-POINT GAUSS RULE	GR021630
C		GR021640
	DATA XGK(1),XGK(2),XGK(3),XGK(4),XGK(5),XGK(6),XGK(7),XGK(8)/	GR021650
	* 0.9914553711208126D+00, 0.9491079123427585D+00,	GR021660
	* 0.8648644233597691D+00, 0.7415311855993944D+00,	GR021670
	* 0.5860872354676911D+00, 0.4058451513773972D+00,	GR021680
	* 0.2077849550078985D+00, 0.0D+00 /	GR021690
	DATA WGK(1),WGK(2),WGK(3),WGK(4),WGK(5),WGK(6),WGK(7),WGK(8)/	GR021700
	* 0.2293532201052922D-01, 0.6309209262997855D-01,	GR021710
	* 0.1047900103222502D+00, 0.1406532597155259D+00,	GR021720
	* 0.1690047266392679D+00, 0.1903505780647854D+00,	GR021730
	* 0.2044329400752989D+00, 0.2094821410847278D+00/	GR021740
	DATA WG(1),WG(2),WG(3),WG(4)/	GR021750
	* 0.1294849661688697D+00, 0.2797053914892767D+00,	GR021760
	* 0.3818300505051189D+00, 0.4179591836734694D+00/	GR021770
	DATA EPMACH,UFLOW/Z341000000000000,Z0010000000000000/	GR021780
C		GR021790
	PHI(U)=XR-(XR-XL)*U*(2.D0*U+3.D0)	GR021800
	PHIP(U)=-6.D0*U*(U+1.D0)	GR021810
C		GR021820
C	LIST OF MAJOR VARIABLES	GR021830
C	-----	GR021840
C		GR021850
C	CENTR - MID POINT OF THE INTERVAL	GR021860
C	HLGTH - HALF-LENGTH OF THE INTERVAL	GR021870
C	ABSC - ABSCISSA	GR021880
C	FVAL* - FUNCTION VALUE	GR021890
C	RESG - R OF THE 7-POINT GAUSS FORMULA	GR021900
C	RESK - R OF THE 15-POINT KRONROD FORMULA	GR021910
C	RESKH - APPROXIMATION TO THE MEAN VALUE OF F OVER (A,B),	GR021920
C	I.E. TO 1/(B-A) *	GR021930
C		GR021940
C	MACHINE DEPENDENT CONSTANTS	GR021950
C	-----	GR021960
C		GR021970
C	EPMACH IS THE LARGEST RELATIVE SPACING.	GR021980
C	UFLOW IS THE SMALLEST POSITIVE MAGNITUDE.	GR021990

C		GR022000
C***FIRST EXECUTABLE STATEMENT		GR022010
C	EPMACH = DBLE(16**(-13))	GR022020
C	UFLOW = DBLE(16**(-65))	GR022030
C		GR022040
	HLGTH = 0.5D+00*(B-A)	GR022050
	CENTR = A+HLGTH	GR022060
	DHLGTH = DABS(HLGTH)	GR022070
C		GR022080
C	COMPUTE THE 15-POINT KRONROD APPROXIMATION TO	GR022090
C	THE INTEGRAL, AND ESTIMATE THE ABSOLUTE ERROR.	GR022100
C		GR022110
	U=(CENTR-XR)/(XR-XL)	GR022120
	FMIN=F(PHI(U))	GR022130
	FMAX=FMIN	GR022140
	FC=FMIN*PHIP(U)	GR022150
	RESG = FC*WG(4)	GR022160
	RESK = FC*WGK(8)	GR022170
	RA = DABS(RESK)	GR022180
	DO 10 J=1,3	GR022190
	JTW = J*2	GR022200
	ABSC = HLGTH*WGK(JTW)	GR022210
	U=(CENTR-ABSC-XR)/(XR-XL)	GR022220
	FVAL1 = F(PHI(U))	GR022230
	FMAX=DMAX1(FMAX,FVAL1)	GR022240
	FMIN=DMIN1(FMIN,FVAL1)	GR022250
	FVAL1=FVAL1*PHIP(U)	GR022260
	U=(CENTR+ABSC-XR)/(XR-XL)	GR022270
	FVAL2=F(PHI(U))	GR022280
	FMAX=DMAX1(FMAX,FVAL2)	GR022290
	FMIN=DMIN1(FMIN,FVAL2)	GR022300
	FVAL2=FVAL2*PHIP(U)	GR022310
	FV1(JTW) = FVAL1	GR022320
	FV2(JTW) = FVAL2	GR022330
	FSUM = FVAL1+FVAL2	GR022340
	RESG = RESG+WG(J)*FSUM	GR022350
	RESK = RESK+WGK(JTW)*FSUM	GR022360
	RA = RA+WGK(JTW)*(DABS(FVAL1)+DABS(FVAL2))	GR022370
10	CONTINUE	GR022380
	DO 15 J = 1,4	GR022390
	JTW1 = J*2-1	GR022400
	ABSC = HLGTH*WGK(JTW1)	GR022410
	U=(CENTR-ABSC-XR)/(XR-XL)	GR022420
	FVAL1=F(PHI(U))	GR022430
	FMAX=DMAX1(FMAX,FVAL1)	GR022440
	FMIN=DMIN1(FMIN,FVAL1)	GR022450
	FVAL1=FVAL1*PHIP(U)	GR022460
	U=(CENTR+ABSC-XR)/(XR-XL)	GR022470
	FVAL2=F(PHI(U))	GR022480
	FMAX=DMAX1(FMAX,FVAL2)	GR022490
	FMIN=DMIN1(FMIN,FVAL2)	GR022500
	FVAL2=FVAL2*PHIP(U)	GR022510
	FV1(JTW1) = FVAL1	GR022520
	FV2(JTW1) = FVAL2	GR022530
	FSUM = FVAL1+FVAL2	GR022540

RESK = RESK+WGK(JTWM1)*FSUM	GR022550
RA = RA+WGK(JTWM1)*(DABS(FVAL1)+DABS(FVAL2))	GR022560
15 CONTINUE	GR022570
RESKH = RESK*0.5D+00	GR022580
RASC = WGK(8)*DABS(FC-RESKH)	GR022590
DO 20 J=1,7	GR022600
RASC = RASC+WGK(J)*(DABS(FV1(J)-RESKH)+DABS(FV2(J)-RESKH))	GR022610
20 CONTINUE	GR022620
R = RESK*HLGTH	GR022630
RA = RA*DHLGTH	GR022640
RASC = RASC*DHLGTH	GR022650
AE = DABS((RESK-RESG)*HLGTH)	GR022660
IF(RASC.NE.0.0D+00.AND.AE.NE.0.0D+00)	GR022670
* AE = RASC*DMIN1(0.1D+01,	GR022680
* (0.2D+03*AE/RASC)**1.5D+00)	GR022690
IF(RA.GT.UFLOW/(0.5D+02*EPMACH)) AE = DMAX1	GR022700
* ((EPMACH*0.5D+02)*RA,AE)	GR022710
RETURN	GR022720
END	GR022730
C*****	GR022740
C	GR022750
SUBROUTINE QUADRT(A,ROOT1,ROOT2,TOL,NOROOT)	GR022760
C	GR022770
C*****	GR022780
C	GR022790
C TO SOLVE ANY QUADRTIC EQUATION OF THE FORM	GR022800
C	GR022810
C A(1)*X**2+A(2)*X+A(3) = 0	GR022820
C	GR022830
C DOUBLE PRECISION A,ROOT1,ROOT2,X,Y,Z,W,ZERO	GR022840
C DIMENSION A(3),ROOT1(2),ROOT2(2)	GR022850
C	GR022860
C ZERO = TOL/10.0	GR022870
C	GR022880
C IF (DABS(A(1))-ZERO)2,2,1	GR022890
2 IF (DABS(A(2))-ZERO) 3,3,4	GR022900
3 NOROOT =0	GR022910
C RETURN	GR022920
C	GR022930
1 NOROOT= 2	GR022940
X = A(2)*A(2) - 4.0*A(1)*A(3)	GR022950
Y = A(1) + A(1)	GR022960
Z = DSQRT(DABS(X))/Y	GR022970
W = -A(2)/Y	GR022980
C	GR022990
IF (X.LT.0.0) GO TO 7	GR023000
ROOT1(1)=W+Z	GR023010
ROOT1(2)=0.0	GR023020
ROOT2(1)=W-Z	GR023030
ROOT2(2)=0.0	GR023040
C RETURN	GR023050
C	GR023060
7 ROOT1(1)=W	GR023070
ROOT1(2)=Z	GR023080
ROOT2(1)=W	GR023090

ROOT2(2)=-Z	GR023100
RETURN	GR023110
C	GR023120
4 NOROOT=1	GR023130
ROOT1(1) = -A(3)/A(2)	GR023140
ROOT1(2) = 0.0	GR023150
RETURN	GR023160
END	GR023170
C*****	GR023180
C	GR023190
SUBROUTINE DCRTNI(X,F,DERF,XST,EPS,IEND,IER)	GR023200
C	GR023210
C*****	GR023220
C	GR023230
C PURPOSE	GR023240
C TO SOLVE PHI = 0	GR023250
C BY MEANS OF NEWTON-S ITERATION METHOD.	GR023260
C	GR023270
C USAGE	GR023280
C CALL DCRTNI (X,F,DERF,XST,EPS,IEND,IER)	GR023290
C	GR023300
C DESCRIPTION OF PARAMETERS	GR023310
C X - DOUBLE COMPLEX RESULTANT ROOT OF EQUATION F(X)=0.	GR023320
C F - DOUBLE COMPLEX RESULTANT FUNCTION VALUE AT	GR023330
C ROOT X.	GR023340
C DERF - DOUBLE COMPLEX RESULTANT VALUE OF DERIVATIVE	GR023350
C AT ROOT X.	GR023360
C XST - DOUBLE COMPLEX INPUT VALUE WHICH SPECIFIES THE	GR023370
C INITIAL GUESS OF THE ROOT X.	GR023380
C EPS - SINGLE PRECISION INPUT VALUE WHICH SPECIFIES THE	GR023390
C UPPER BOUND OF THE ERROR OF RESULT X.	GR023400
C IEND - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.	GR023410
C IER - RESULTANT ERROR PARAMETER CODED AS FOLLOWS	GR023420
C IER=0 - NO ERROR,	GR023430
C IER=1 - NO CONVERGENCE AFTER IEND ITERATION STEPS,	GR023440
C IER=2 - AT ANY ITERATION STEP DERIVATIVE DERF WAS	GR023450
C EQUAL TO ZERO.	GR023460
C	GR023470
C	GR023480
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED	GR023490
C PHIO,AND DPHIO	GR023500
C	GR023510
C METHOD	GR023520
C SOLUTION OF EQUATION F(X)=0 IS DONE BY MEANS OF NEWTON-S	GR023530
C ITERATION METHOD, WHICH STARTS AT THE INITIAL GUESS XST OF	GR023540
C A ROOT X. CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF	GR023550
C F(X) AT ROOT X IS NOT EQUAL TO ZERO. ONE ITERATION STEP	GR023560
C REQUIRES ONE EVALUATION OF F(X) AND ONE EVALUATION OF THE	GR023570
C DERIVATIVE OF F(X).	GR023580
C FOR REFERENCE, SEE R. ZURMUEHL, PRAKTISCHE MATHEMATIK FUER	GR023590
C INGENIEURE UND PHYSIKER, SPRINGER, BERLIN/GOETTINGEN/	GR023600
C HEIDELBERG, 1963, PP.12-17.	GR023610
C	GR023620
C	GR023630
DOUBLE COMPLEX X,F,DERF,XST,TOL,DX,PHIO,DPHIO	GR023640

	DOUBLE PRECISION A,DTOL,DTOLF	GR023650
	EXTERNAL PHIO,DPHIO	GR023660
C	PREPARE ITERATION	GR023670
	IER=0	GR023680
	X=XST	GR023690
	TOL=X	GR023700
	F = PHIO(TOL)	GR023710
	DERF = DPHIO(TOL)	GR023720
	DTOLF=100.00*EPS	GR023730
C		GR023740
C		GR023750
C	START ITERATION LOOP	GR023760
	DO 6 I=1,IEND	GR023770
	IF(CDABS(F))1,7,1	GR023780
C		GR023790
C	EQUATION IS NOT SATISFIED BY X	GR023800
	1 IF(CDABS(DERF))2,8,2	GR023810
C		GR023820
C	ITERATION IS POSSIBLE	GR023830
	2 DX=F/DERF	GR023840
	X=X-DX	GR023850
	TOL=X	GR023860
	F = PHIO(TOL)	GR023870
	DERF = DPHIO(TOL)	GR023880
C		GR023890
C	TEST ON SATISFACTORY ACCURACY	GR023900
	DTOL=DBLE(EPS)	GR023910
	A=CDABS(X)	GR023920
	IF(A-1.00)4,4,3	GR023930
	3 DTOL=DTOL*A	GR023940
	4 IF(CDABS(DX)-DTOL)5,5,6	GR023950
	5 IF(CDABS(F)-DTOLF)7,7,6	GR023960
	6 CONTINUE	GR023970
C	END OF ITERATION LOOP	GR023980
C		GR023990
C		GR024000
C	NO CONVERGENCE AFTER IEND ITERATION STEPS. ERROR RETURN.	GR024010
	IER=1	GR024020
	7 RETURN	GR024030
C		GR024040
C	ERROR RETURN IN CASE OF ZERO DIVISOR	GR024050
	8 IER=2	GR024060
	RETURN	GR024070
	END	GR024080
C	*****	GR024090
C		GR024100
	SUBROUTINE RSORT(A,B,R,N,M,MS)	GR024110
C		GR024120
C	*****	GR024130
	DIMENSION A(1),B(1),R(1)	GR024140
C		GR024150
C	MOVE SORTING KEY VECTOR TO FIRST COLUMN OF OUTPUT MATRIX	GR024160
C	AND BUILD ORIGINAL SEQUENCE LIST IN SECOND COLUMN	GR024170
C		GR024180
	DO 10 I=1,N	GR024190

R(I)=B(I)	GR024200
I2=I+N	GR024210
10 R(I2)=I	GR024220
C	GR024230
C SORT ELEMENTS IN SORTING KEY VECTOR (ORIGINAL SEQUENCE LIST	GR024240
C IS RESEQUENCED ACCORDINGLY)	GR024250
C	GR024260
L=N+1	GR024270
20 ISORT=0	GR024280
L=L-1	GR024290
IF(L .LT. 2) GO TO 50	GR024300
DO 40 I=2,L	GR024310
IF(R(I)-R(I-1)) 30,40,40	GR024320
30 ISORT=1	GR024330
RSAVE=R(I)	GR024340
R(I)=R(I-1)	GR024350
R(I-1)=RSAVE	GR024360
I2=I+N	GR024370
SAVER=R(I2)	GR024380
R(I2)=R(I2-1)	GR024390
R(I2-1)=SAVER	GR024400
40 CONTINUE	GR024410
IF(ISORT) 20,50,20	GR024420
C	GR024430
C MOVE ROWS FROM MATRIX A TO MATRIX R (NUMBER IN SECOND COLUMN	GR024440
C OF R REPRESENTS ROW NUMBER OF MATRIX A TO BE MOVED)	GR024450
C	GR024460
50 DO 80 I=1,N	GR024470
C	GR024480
C GET ROW NUMBER IN MATRIX A	GR024490
C	GR024500
I2=I+N	GR024510
IN=R(I2)	GR024520
C	GR024530
IR=I-N	GR024540
DO 80 J=1,M	GR024550
C	GR024560
C LOCATE ELEMENT IN OUTPUT MATRIX	GR024570
C	GR024580
IR=IR+N	GR024590
C	GR024600
C LOCATE ELEMENT IN INPUT MATRIX	GR024610
C	GR024620
CALL LOC(IN,J,IA,N,M,MS)	GR024630
C	GR024640
C TEST FOR ZERO ELEMENT IN DIAGONAL MATRIX	GR024650
C	GR024660
IF(IA) 60,70,60	GR024670
C	GR024680
C MOVE ELEMENT TO OUTPUT MATRIX	GR024690
C	GR024700
60 R(IR)=A(IA)	GR024710
GO TO 80	GR024720
70 R(IR)=0	GR024730
80 CONTINUE	GR024740

RETURN	GR024750
END	GR024760
C*****	GR024770
C	GR024780
SUBROUTINE LOC(I,J,IR,N,M,MS)	GR024790
C	GR024800
C*****	GR024810
C	GR024820
C PURPOSE	GR024830
C COMPUTE A VECTOR SUBSCRIPT FOR AN ELEMENT IN A MATRIX OF	GR024840
C SPECIFIED STORAGE MODE	GR024850
C	GR024860
C USAGE	GR024870
C CALL LOC (I,J,IR,N,M,MS)	GR024880
C	GR024890
C DESCRIPTION OF PARAMETERS	GR024900
C I - ROW NUMBER OF ELEMENT	GR024910
C J - COLUMN NUMBER OF ELEMENT	GR024920
C IR - RESULTANT VECTOR SUBSCRIPT	GR024930
C N - NUMBER OF ROWS IN MATRIX	GR024940
C M - NUMBER OF COLUMNS IN MATRIX	GR024950
C MS - ONE DIGIT NUMBER FOR STORAGE MODE OF MATRIX	GR024960
C 0 - GENERAL	GR024970
C 1 - SYMMETRIC	GR024980
C 2 - DIAGONAL	GR024990
C	GR025000
C REMARKS	GR025010
C NONE	GR025020
C M IS UNUSED BUT IS REQUIRED TO BE PRESENT	GR025030
C	GR025040
C SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED	GR025050
C NONE	GR025060
C	GR025070
C METHOD	GR025080
C MS=0 SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*M ELEMENTS	GR025090
C IN STORAGE (GENERAL MATRIX)	GR025100
C MS=1 SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N*(N+1)/2 IN	GR025110
C STORAGE (UPPER TRIANGLE OF SYMMETRIC MATRIX). IF	GR025120
C ELEMENT IS IN LOWER TRIANGULAR PORTION, SUBSCRIPT IS	GR025130
C CORRESPONDING ELEMENT IN UPPER TRIANGLE.	GR025140
C MS=2 SUBSCRIPT IS COMPUTED FOR A MATRIX WITH N ELEMENTS	GR025150
C IN STORAGE (DIAGONAL ELEMENTS OF DIAGONAL MATRIX).	GR025160
C IF ELEMENT IS NOT ON DIAGONAL (AND THEREFORE NOT IN	GR025170
C STORAGE), IR IS SET TO ZERO.	GR025180
C	GR025190
C	GR025200
C	GR025210
C IX=I	GR025220
C JX=J	GR025230
C IF(MS-1) 10,20,30	GR025240
10 IRX=N*(JX-1)+IX	GR025250
GO TO 36	GR025260
20 IF(IX-JX) 22,24,24	GR025270
22 IRX=IX+(JX-JX)/2	GR025280
GO TO 36	GR025290

```
24 IRX=JX+(IX*IX-IX)/2
   GO TO 36
30 IRX=0
   IF(IX-JX) 36,32,36
32 IRX=IX
36 IR=IRX
   RETURN
   END
```

```
GR025300
GR025310
GR025320
GR025330
GR025340
GR025350
GR025360
GR025370
```


U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)		1. PUBLICATION OR REPORT NO. NBSIR/85-3234	2. Performing Organ. Report No.	3. Publication Date NOVEMBER 1985
4. TITLE AND SUBTITLE Dynamic Green's Functions of An Infinite Plate - A Computer Program				
5. AUTHOR(S) Nelson N. Hsu				
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			7. Contract/Grant No.	8. Type of Report & Period Covered
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)				
10. SUPPLEMENTARY NOTES ☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.				
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here) This report is a FORTRAN program to compute the Green's functions of an infinite plate. The Green's function, $G_{ij}(\underline{\xi}, \underline{x}, t)$, is defined as the i th component of the displacement at $\underline{x}$ due to a point force of step-function time dependency acting at $\underline{\xi}$ in the j th direction initiated at $t=0$. The Green's function is the fundamental solution of the transient elastic wave propagation problem. In general, the displacement field $\underline{u}(\underline{\xi}, \underline{x}, t)$ at $\underline{x}$ due to a point force of arbitrary time dependence acting at $\underline{\xi}$ can be computed by a convolution integration; i.e., $u_i(\underline{\xi}, \underline{x}, t) = \int_0^\infty G_{ij}(\underline{\xi}, \underline{x}, t) f_j(\tau-t) d\tau. \quad (1)$ Here, G_{ij} is the time derivative of G_{ij} and $f_j(t)$ is the point force component of arbitrary time dependence acting in the j th direction (summation over repeated indices is used). Displacement due to point dipoles or couple forces can be represented by the spatial derivatives of G_{ij} . Displacement produced by a dynamic force distributed over a finite area can also be computed by numerical integration using the Green's function as the kernel of the integral over the finite area.				
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) Acoustic emission; dynamic displacement due to point impact; Green's function; ray theory; transient wave propagation; waveform analysis				
13. AVAILABILITY ☒ Unlimited ☐ For Official Distribution. Do Not Release to NTIS ☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. ☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 66 15. Price \$11.95	

