

U.S. Department of the Interior
Geological Survey

Mail Stop 964
Box 25046, Federal Center
Denver, Colorado 80225

Programs EMCUPL and SCHCOPL:
Computation of electromagnetic coupling
on a layered halfspace with complex conductivities

by

James Kauahikaua
and
Walter L. Anderson

Open-File Report 79-1430

1979

CONTENTS

DISCLAIMER	3
ABSTRACT	4
INTRODUCTION	4
ALGORITHM	6
Theory	6
Computation of the frequency response	9
Computation of the transient response	12
RESULTS	14
REFERENCES	16
FIGURES	18
APPENDIX	22
Description of Programs and Input Parameters	22
Example Calculations using EMCUPL	25
Main Programs and Subprograms Source Listing	32
Source availability	33

DISCLAIMER

These programs were written in Fortran IV for a Honeywell Multics 68/80 system*. Although program tests have been made, no guarantee (expressed or implied) is made by the authors regarding accuracy or proper functioning of these programs on all computer systems.

* Brand or manufacturers' names used in this report are for descriptive purposes only and do not constitute endorsement by the U.S. Geological Survey.

ABSTRACT

A number of efficient numerical computer algorithms are incorporated into a general program called EMCUPL, which calculates the electromagnetic (EM) coupling between two straight wires on the surface of a multilayered halfspace. Each layer has an isotropic conductivity which may be either real or complex. A second computer program, called SCHCOPL, is described which calculates the coupling for the special case of a Schlumberger or Wenner array also on a multilayered halfspace. Comparison with other programs shows that EMCUPL is at least as accurate, more generally applicable, and computationally more efficient. FORTRAN listings of all subprograms and example calculations are given in the Appendix.

INTRODUCTION

This work describes the algorithms used in programs EMCUPL and SCHCOPL, which compute the electromagnetic (EM) coupling, or mutual impedance, between two straight, grounded wires on the surface of a horizontally layered halfspace having either real isotropic or complex isotropic conductivities. These algorithms offer several advantages over previously employed algorithms. First, the electromagnetic coupling for a Schlumberger-type array can be calculated as easily as the coupling for any other

collinear array. Second, digital filters are used instead of direct integration between Bessel function zeroes for considerably quicker computation of the necessary Hankel transforms. Third, automatic Gaussian quadrature integration routines replace summation of a small fixed number of integrand evaluations for more accurate integration over the wire lengths. The automatic integration routines also allow completely general wire orientations without requiring separate integration or summation routines. Finally, the integrand for integration along the wire lengths is replaced by a quintic spline based on a set of nodes which are calculated with a very fast lagged digital filter algorithm.

The inclusion of complex conductivities in the layered earth model is essential for modelling realistic earth materials. Recent papers (Zonge and Wynn, 1975; Pelton et al., 1978) demonstrate that the differences between induced polarization spectra can be exploited for mineral discrimination. These papers contain excellent discussions and references on the use of complex conductivities in mineral prospecting problems.

ALGORITHM

Theory

The equation describing electromagnetic coupling between two straight grounded wires of arbitrary length and orientation on the surface of a halfspace having horizontal layers with real isotropic or complex isotropic conductivities is (see Figure 1)

$$Z = Q(r_{\alpha\alpha}) - Q(r_{\alpha\beta}) + Q(r_{\beta\beta}) - Q(r_{\beta\alpha}) + \cos(\theta) \int_a^b \int_{\alpha}^{\beta} P(r) dS ds, \quad (1)$$

$$\text{where } P(r) = -\frac{I}{2\pi\sigma_1} \left\{ \frac{2i}{\delta_1^3} \int_0^{\infty} f_3(g) J_0(gB) dg + \frac{1}{r^3} [1 - (1 + \gamma_1 r) \exp(-\gamma_1 r)] \right\} \quad (2)$$

$$Q(r) = -\frac{I}{2\pi\sigma_1} \left\{ \frac{1}{\delta_1} \int_0^{\infty} f_7(g) J_0(gB) dg - 1/r \right\} \quad (3)$$

Z EM coupling or mutual impedance,

i $= \sqrt{-1}$,

g Hankel transform variable,

α, β are current electrode positions,

a, b are potential electrode positions,

dS is an infinitesimal element of source wire,

ds is an infinitesimal element of receiver wire,

θ angle between the wires ($=0$ for parallel wires),

I current in source wire,

J_0 Bessel function of first kind, order zero,

$$\mu_0 = 4 * 10^{**}(-7) ,$$

$\sigma_j(w)$ complex conductivity of jth layer at frequency w,
 $= \sigma_j'(w) + i \sigma_j''(w)$, $\sigma_j'(w)$, $\sigma_j''(w)$ real,

$\sigma_j(0) = \sigma_j(w)$ at $w = 0$,
 $= \sigma_j'(0)$, $\sigma_j''(0) = 0$ ($\sigma_j(0)$ must be real and constant),

d_j thickness of jth layer,

m number of layers,

$\delta_1 = (2/\mu_0 \omega \sigma_1(0))^{1/2}$, skin depth in first layer,

$\gamma_1^2 = i \mu_0 \omega \sigma_1$ (quasistatic assumption),

B = r/δ_1 ,

r = $(x^2 + y^2)^{1/2}$,

$r_{\alpha a}$ distance between α and a
(similar definitions for $r_{\alpha b}$, $r_{\beta a}$, and $r_{\beta b}$),

$V_j(g) = (g^2 + 2i \sigma_j(w) / \sigma_1(0))^{1/2}$

$E_j(g) = (1 - \exp(-2d_j V_j(g) / \delta_1)) / (1 + \exp(-2d_j V_j(g) / \delta_1))$

$$F_j(g) = \frac{V_{j+1}(g) F_{j+1}(g) + V_j(g) E_j(g)}{V_j(g) + V_{j+1}(g) F_{j+1}(g) E_j(g)}$$

$$L_j(g) = \frac{\sigma_j(w) V_{j+1}(g) L_{j+1}(g) + \sigma_{j+1}(w) V_j(g) E_j(g)}{\sigma_{j+1}(w) V_j(g) + \sigma_j(w) V_{j+1}(g) L_{j+1}(g) E_j(g)}$$

$F_m(g) = L_m(g) = 1.0$ for m-layered earth,

$f_3(g) = g V_1(g) (1 - F_1(g)) / [(g + V_1(g) F_1(g)) (g + V_1(g))]$, and

$f_7(g) = \frac{1 V_1(g) (L_1(g) - 1) / g +}{2 V_1(g) (1 - F_1(g)) / [(g + V_1(g) F_1(g)) (g + V_1(g))]} .$

The halfspace terms have been separated from the integrals in equations (2) and (3). Equation (1) is completely general; however, the authors have encountered severe numerical problems when using equation (1) to compute the coupling for the special case of two closely spaced, parallel wires as in a Schlumberger or Wenner array. Therefore equation (1) was rewritten for this case in a mathematically equivalent, but numerically advantageous way. Assuming $r_{\alpha a} = r_{\beta b}$, $r_{\alpha b} = r_{\beta a}$, and $\cos(\theta) = 1$, equation (1) becomes

$$Z_S = 2 \left\{ Q(r_{\alpha a}) - Q(r_{\alpha b}) \right\} + (r_{ab}) R(r) - 2 \int_a^b \int_{\beta}^{\infty} P(r) dS ds, \quad (4)$$

where Z_S is the mutual impedance for this configuration,

$$R(r) = - \frac{2iI}{\pi \sigma_1 \delta_1^2} \int_0^{\infty} f_2(g) \cos(gB) dg, \text{ and} \quad (5)$$

$$f_2(g) = (1 - F_1(g)) V_1(g) / [(g + V_1(g) F_1(g)) (g + V_1(g))] .$$

$R(r)$, defined by equation (5), is the electric field of an infinitely long wire. The double integral is now a correction term which becomes less significant as the wires are moved closer together. Note that the $P(r)$, $Q(r)$, and $R(r)$ integrals have been normalized by dividing all lengths by the skin depth in the first layer (Anderson, 1974).

Computation of the Frequency Response

The numerical evaluation of equation (1) and (4) presents two computational problems: 1) calculation of the Hankel and Fourier transform integrals for multilayered models, and 2) calculation of the double integral over the wire lengths. In previous studies (Hohmann, 1973; Dey and Morrison, 1973; Wynn and Zonge, 1975, 1977), the $P(r)$ and $Q(r)$ integrals were computed as a series of integrals between the zeroes of the J_0 Bessel function. The double integral was computed as a double series of a predetermined and constant number of $P(r)$ evaluations. Anderson (1974) took a different approach by computing EM coupling directly as the double integral of the electric field of an electric dipole. The finite integrations were computed with an automatic Gaussian quadrature routine while the Hankel transforms were computed with digital filters. The resulting program produced EM coupling values which compared very closely with those published by Hohmann (1973) and Dey and Morrison (1973); however, the routine was somewhat time-consuming as each evaluation of the integrand required evaluation of a J_0 and J_1 Hankel transform integral. In the present study, the authors use Anderson's (1974, 1975, 1979) automatic integration and digital filter numerical routines to evaluate the more efficiently formulated equations (1) and (4) which have a specified relative accuracy (usually 10^{-3} to 10^{-4}).

The theory of computing Hankel and Fourier transform integrals as digital filter convolutions is fundamental to the understanding of how the double integral of $P(r)$ is calculated and will, therefore, bear a quick review (after Koefoed, Ghosh, and Polman, 1972). The J_0 Hankel transform integral in equations (2) and (3) and the cosine integral in equation (5) can be rewritten as convolution integrals by the change of variables

$$u = \ln(B) \text{ and } v = \ln(1/g) . \quad (6)$$

The relation between the transform integral and the convolution integral becomes

$$\int_0^{\infty} f(g)h(gB)dg = \int_{-\infty}^{\infty} f(v)\exp(-v)h(uv)dv, \quad (7)$$

where $f(g)$ is the kernel function, and
 $h(gB)$ is the transform function (either cosine or J_0 Bessel function).

The convolution integral can then be discretized and evaluated as a convolution using an N-point digital filter,

$$(1/B) \int_{-\infty}^{\infty} f(v) \left\{ \exp(u-v)h(u-v) \right\} dv \approx \sum_{i=1}^N f(v_i) h_a(u-v_i), \quad (8)$$

where $h_a(u-v_i)$ are the digital filter weights. The actual filter weights that are used were developed by Anderson (1975) and were

previously used with excellent results for the calculation of the electromagnetic fields about electric (Anderson, 1974) and magnetic sources, for the calculation of the Green's functions used in the integral equation formulation of a 2-D plane-wave modelling program (Anderson, Hohmann, and Smith, 1976), and for the calculation of a variety of DC sounding models. We are satisfied that the digital filters can be used to rapidly calculate the Hankel and Fourier integrals commonly encountered in electromagnetic problems with a relative accuracy of at least 10^{-4} .

The double integral in equation (1) and (4) is calculated in a straightforward manner using two separately named and coded, but otherwise identical automatic Gaussian quadrature integration routines (Patterson, 1973). This approach is superior to the double summation of a fixed number of integrand evaluations used by previous authors because of its ability to use more finely spaced integrand values within integration limits if the complexity of the integrand requires it. The biggest drawback to using an automatic integration routine is, of course, that it requires many more integrand evaluations. To compensate for this, the $P(r)$ integrand is not evaluated directly but is represented by a quintic spline (Herriot and Reinsch, 1976). The spline nodes are calculated throughout the range of interest (the closest and farthest distance between the wires) at the same interval as the digital filter allowing equation (8) to be used as a convolution at different lags while saving previously

computed $f_3(g)$ values (Anderson, 1975). This procedure is exactly analogous to time-series convolution and yields the maximum number of $P(r)$ evaluations for a minimum number of $f_3(g)$ evaluations. For example, if one evaluation of $P(r)$ normally requires 25 $f_3(g)$ evaluations, then 26 evaluations of $P(r)$ would normally require 650 $f_3(g)$ evaluations. However, the 26 $P(r)$ values may be obtained with a total of only 50 evaluations if the r values are spaced at the same interval as the digital filter values. At Anderson's filter interval of $0.2 \cdot \log(e)$ (roughly 12 per decade), 26 values of $P(r)$ will span a two-decade range of separations. As an example, this span of $P(r)$ values would be enough to calculate the double integral accurately for a collinear dipole-dipole model for $n=1$ to 100 (where the separation between dipoles is n times the dipole length; see Figure 1).

Computation of the Transient Response

The EM coupling, as computed by either equation (1) or (4), is stated as a function of frequency. Because the transient response must be zero prior to energizing the source, it may easily be calculated using either the sine or cosine transform of $Z(w)S(w)$. For example,

$$z(t) = - \frac{2}{\pi} \int_0^{\infty} \text{Im}[Z(w)S(w)] \sin(wt) dw, \quad (9)$$

where $z(t)$ is the transient response,
 $w = 2\pi f$,
 $Z(w)$ is the frequency response,
 $S(w)$ is the source or energizing function
frequency response, and
 Im signifies taking the imaginary part

(see Bracewell, 1965, p. 271, for example). The coupling programs in the appendix are coded to compute the theoretical step response using $S(w) = 1/iw$. The integrand is then interpolated using a cubic spline, and the sine integral is calculated with a digital filter also developed by Anderson (1975).

Other source functions may be used in place of a theoretical step function by replacing the appropriate statements in program EMCUPL (line 00003960) and SCHCOPL (line 00013110). However, certain difficulties should be expected by the user. First, use of the digital filter approximation to the sine integral requires that $Z(w)S(w)$ be a continuous function of frequency. Discrete frequency functions, such as square waveforms, triangular waveforms, and IP waveforms (in other words, any periodic waveform) may be used, but only by replacing the digital filter transform with a common discrete Fourier transform. Second, the digital filter approximation will converge only if the product $Z(w)S(w)$ is a nonincreasing function either as w gets very small or very large. In order to use a theoretical step source

function, for example, the transform kernel had to be changed to $(Z(w)-Z(0))/iw$ to avoid an infinite value at zero frequency. A step function of magnitude $Z(0)$ was later added to the resulting transient response.

RESULTS

Two main interactive programs to compute EM coupling and the associated subprograms were written in FORTRAN IV for use on a Honeywell Multics 68/80 system. Program EMCUPL calculates the coupling between two arbitrarily oriented wires using equation (1), and program SCHCOPL calculates coupling between closely-spaced, parallel, equatorially centered wires using equation (4). To determine how these programs compare with those used by other authors, several sets of previously published results were recomputed using program EMCUPL and SCHCOPL. A multitude of halfspace, two-, and three-layer model plots were available for comparison in the papers of Millett (1967), Dey and Morrison (1973), Hohmann (1973), and Wynn and Zonge (1975, 1977). With the exception of two models, the EMCUPL and SCHCOPL results virtually duplicated these previous results within the accuracy attainable by visually matching data plots. The first discrepancy noted involved the magnitude and phase plots for a dipole-dipole configuration over a homogeneous earth shown in Figures 4 and 5 of Dey and Morrison (1973). The EMCUPL results

(shown here in Figures 2 and 3) do agree closely for the magnitudes and the low frequency phases; however, there are marked differences in the phases at higher frequencies. Several checks of the EMCUPL results convinced the authors of their validity and suggested that the double-summation series approximation to the double integral using Simpson's 3-point rule as used by Dey and Morrison (1973) may not have been accurate in this frequency range. The second discrepancy involved EM coupling plots for a Wenner array over a homogeneous and two-layer earth shown in Figures 11 and 12 of Wynn and Zonge (1977). The identical EMCUPL and SCHCOPL results (shown here in Figure 4) exhibit a different functional behavior and plot in a different quadrant of the complex plane than the Wynn-Zonge results. Note that the curves in Figure 4 are similar in form and plot in the same quadrant as the curves for two equatorial, infinitesimal dipoles (Anderson, 1974).

REFERENCES

Anderson, W. L., 1974, Electromagnetic fields about a finite electric wire source: U. S. Geol. Survey Rept. USGS-GD-74-041, 209 p.; available from U. S. Dept. Commerce Natl. Tech. Inf. Service, Springfield, VA 22161, as Rept. PB-238-199.

_____, 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: U.S. Geol. Survey Rept. USGS-GD-75-012, 119 p.; available from U. S. Dept. Commerce Natl. Tech. Inf. Service, Springfield, VA 22161, as Rept. PB-242-800.

_____, 1979, Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics, vol. 44, no. 7, p. 1287-1305.

Anderson, W. L., Hohmann, G. W., and Smith, B. D., 1976, EM scattering by multiple conductors in the earth due to a plane wave source, U. S. Geol. Survey Rept. USGS-GD-76-019, 75 p., available from U. S. Dept. Commerce Natl. Tech. Inf. Service, Springfield, VA 22161, as Rept. PB-261-183.

Bracewell, R., 1965, The Fourier transform and its application, McGraw-Hill Book Co., New York, 381 p.

Dey, A. and Morrison, H. F., 1973, Electromagnetic coupling in frequency and time-domain induced polarization surveys over a multilayered earth: Geophysics, v. 38, p. 380-405.

Herriot, J. G., and Reinsch, C. H., 1976, Algorithm 507, Procedures for quintic natural spline interpolation: ACM Transactions on Math. Software, v. 2, no. 3, p. 281-289.

Hohmann, G. W., 1973, Electromagnetic coupling between grounded wires at the surface of a two-layered earth: Geophysics, v. 38, p. 854-863.

Koefoed, O., Ghosh, D. P., and Polman, G. J., 1972, Computation of type curves for electromagnetic depth sounding with a horizontal transmitting coil by means of a digital filter: Geophysical Prospecting, v. 20, p. 406-420.

Millett, F., 1967, Electromagnetic Coupling of collinear dipoles on a uniform half-space, in Mining Geophysics, vol. II, SEG, Tulsa.

Patterson, T. N. L., 1973, Algorithm for automatic numerical integration over a finite interval [D1]: Assoc. for computing machinery Comm., v. 16, no. 11, p. 694-699.

Pelton, W. H., Ward, S. H., Hallof, P. G., Sill, W. R., and Nelson, P. H., 1978, Mineral Discrimination and removal of Inductive coupling with multifrequency IP: Geophysics, v. 43, p. 588-609.

Wynn, J. C. and Zonge, K. L., 1975, EM coupling, its intrinsic value, its removal and the cultural coupling problem: Geophysics, v. 40, p. 831-850.

_____, 1977, Electromagnetic Coupling: Geophysical Prospecting, v. 25, p. 29-51.

Zonge, K. L. and Wynn, J. C., 1975, Recent Advances in complex resistivity measurements: Geophysics, v. 40, p. 851-864.

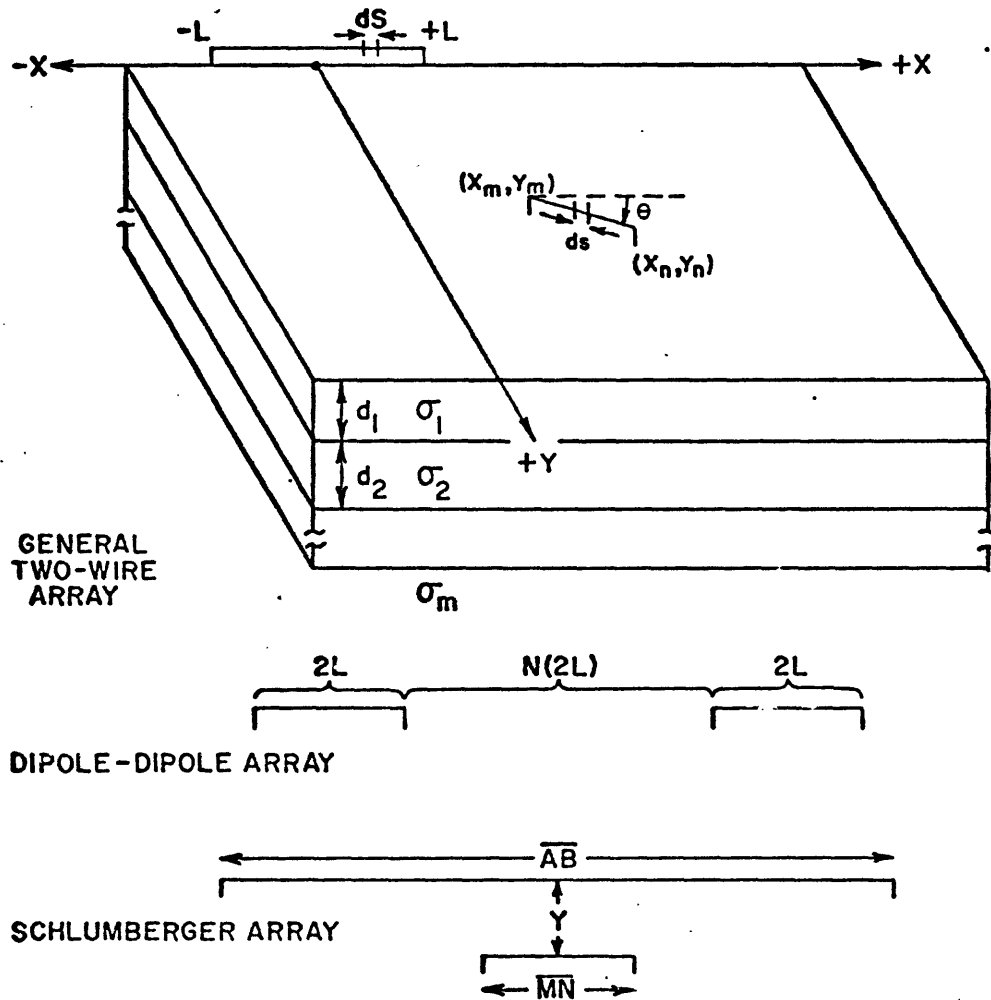


Fig. 1. Diagram defining general, dipole-dipole, and Schlumberger wire geometries and earth model parameters.

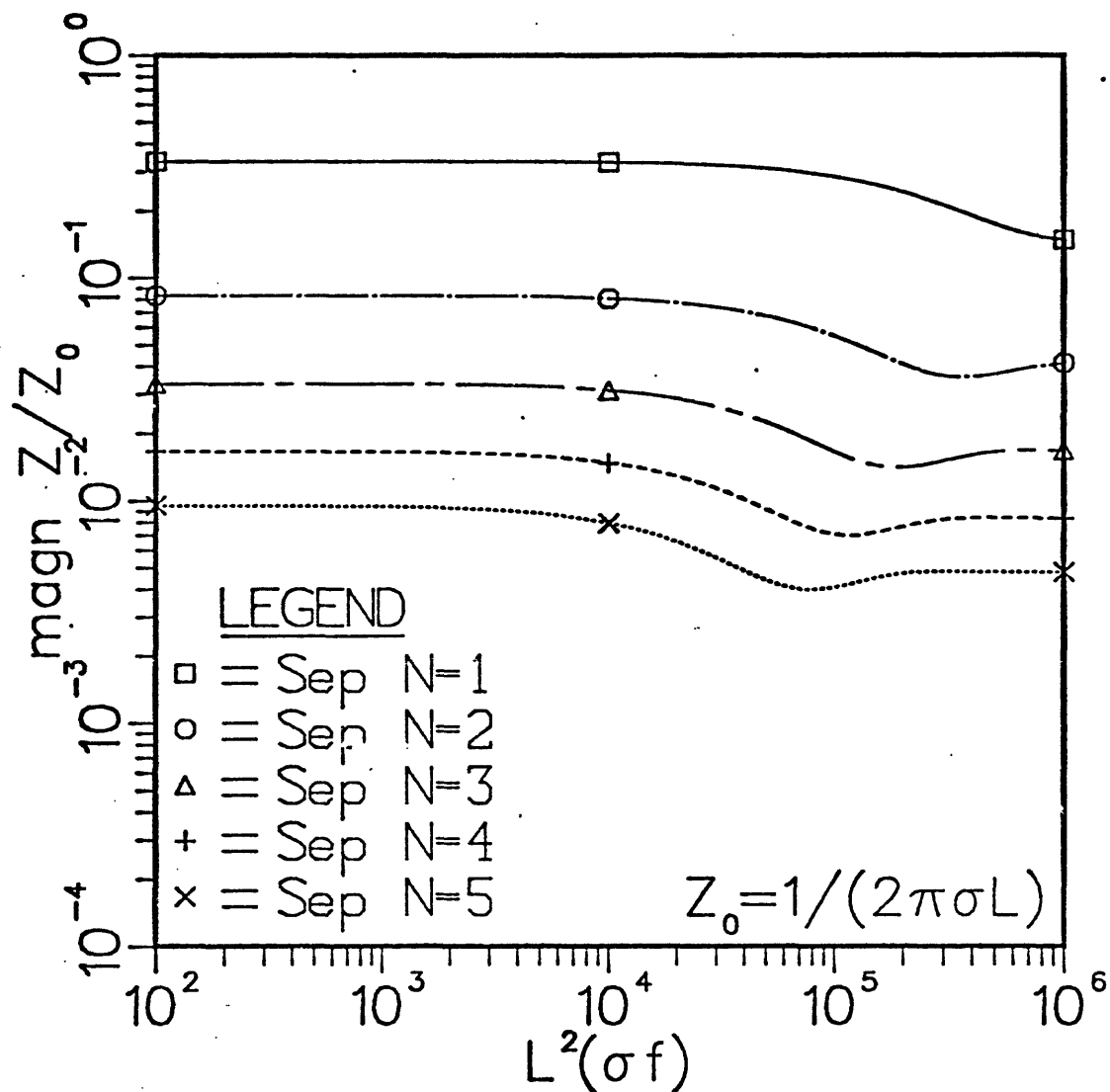


Fig. 2. Amplitude of normalized mutual impedance for a dipole-dipole configuration over a homogeneous halfspace (compare with Dey and Morrison, 1973, Figure 4).

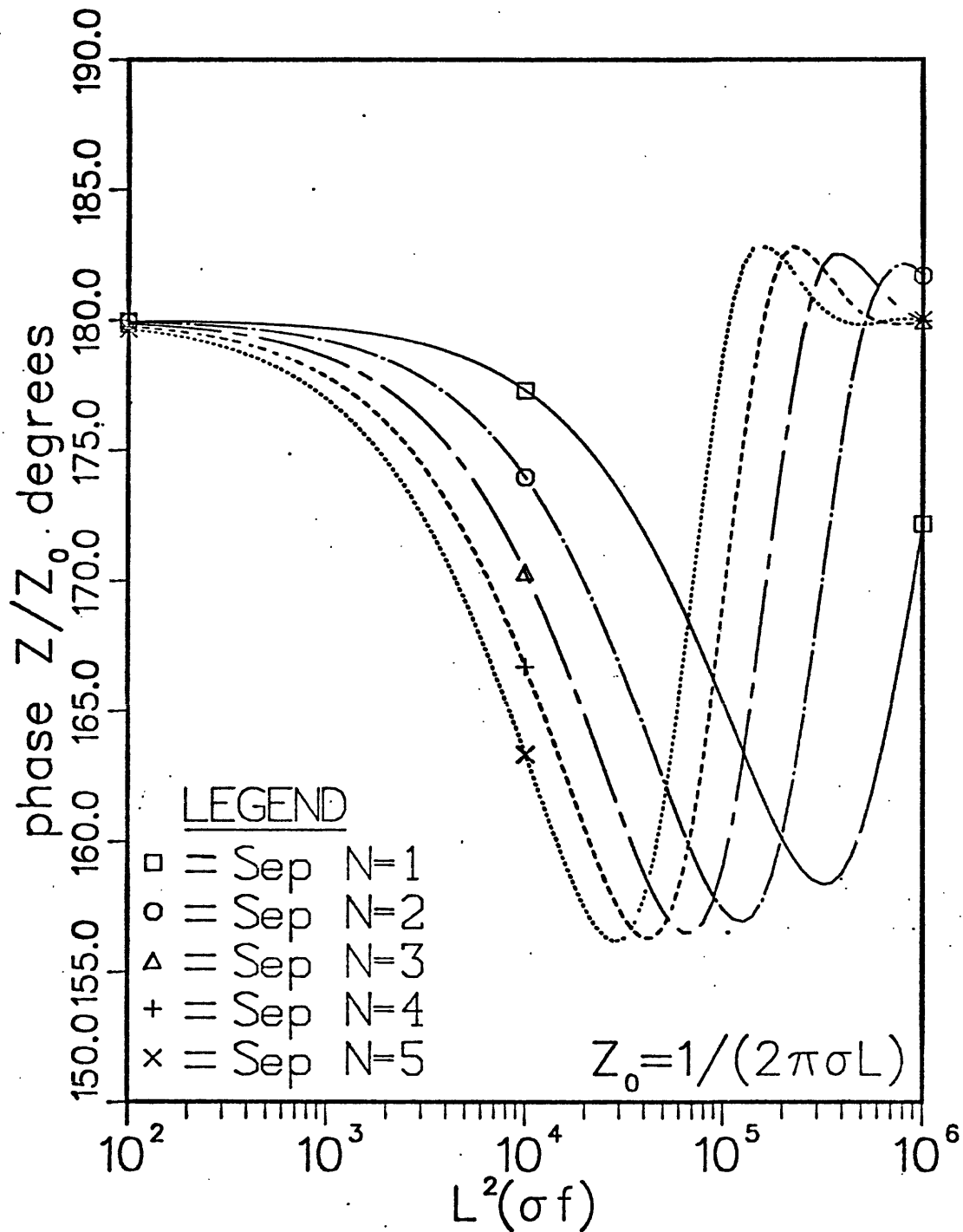


Fig. 3. Phase of normalized mutual impedance for a dipole-dipole configuration over a homogeneous halfspace (compare with Dey and Morrison, 1973, Figure 5).

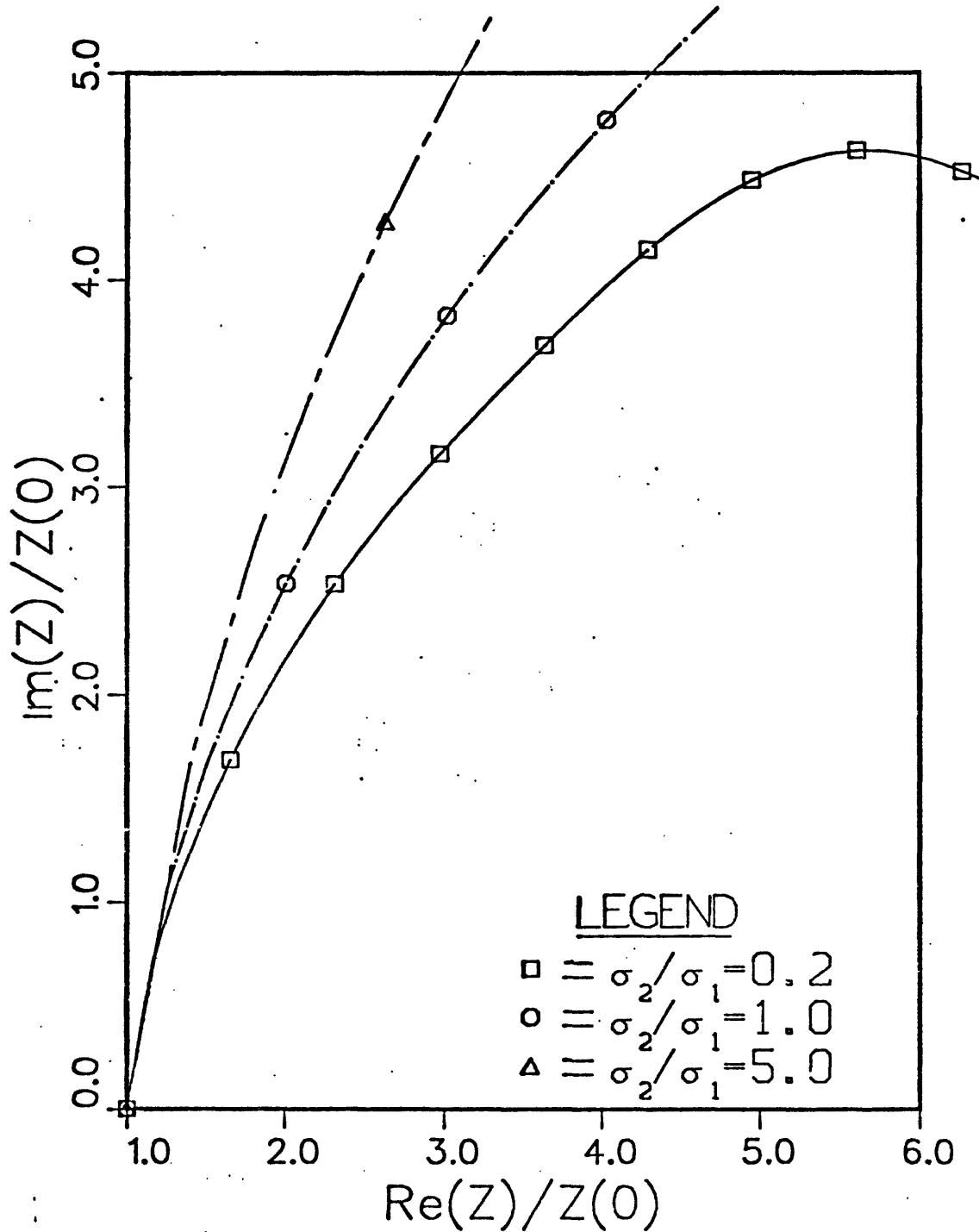


Fig. 4. EM coupling for the Wenner array as a function of resistivity of a two-layer earth with all other parameters held constant (compare with Wynn and Zonge, 1977, Figure 2). Source wire is 450 m long and receiver wire is 150 m long and 75 m away from the source wire. Conductivities for the first and second layers are 0.1 and 0.5 or 0.02 mhos/m, respectively. The first layer is 150 m thick.

APPENDIX

Description of Programs and Input Parameters

The input of model and other control parameters to programs EMCUPL and SCHCOPL is done through a single FORTRAN namelist called PARMS. Each different model requires only one namelist specification. The input parameter lists for the programs are identical except for slight differences in the wire geometry parameters which are noted in the input parameter descriptions.

Besides choosing appropriate parameter values, the user must also code the complex function SIGMA(J,CF) prior to calculation of a model with complex conductivities. SIGMA(J,CF) must return the complex conductivity of the J-th layer at the normalized frequency CF ($=1/\delta_1^2$). In this way, either a mathematical expression or interpolated data values may be used. SIGMA(J,CF) will be required to produce all conductivities if any of them are complex. An example function SIGMA which computes a Cole-Cole (Pelton et al, 1978) conductivity with a frequency dependence of 0.25 and a chargeability of 30 percent is listed in this appendix.

The parameters are identical for both programs except where noted. Parameter defaults are given and those parameters which have no defaults and must therefore be specified are indicated with an asterisk (*) to the left of the parameter name. MKS units are assumed throughout.

MODEL PARAMETERS:

- * SIG(J) real conductivity of the Jth layer,
 (used only if ICMPLX=0),
- * D(J) thickness of the Jth layer,
- M number of layers (default=1, maximum=10),
- ICMPLX switch for complex or real conductivities,
 =0 use real conductivities in SIG(J) (default),
 =1 use complex conductivities computed by user-
 defined function SIGMA(J,CF).
- * NF >0 number of frequencies desired per decade
 between f0 and fm,
 <0 number of frequencies in FNF array,
- * F0,FM minimum and maximum frequencies desired,
- * FNF array of up to 50 particular frequencies.

WIRE GEOMETRY PARAMETERS (source wire centered on x-axis):

for EMCUPL

- * L source wire halflength,
- * XM,YM coordinates of one end of the receiver wire,
- * XN,YN coordinates of other end of receiver wire,
- DX,XMAX increment for stepping receiver wire position in
 x direction, and maximum x value considered,
 (default DX=XMAX=0.0),
 example: for a collinear dipole-dipole setup with
 100 m long wires, dx=100 and xmax=1300 will yield
 coupling values for each frequency chosen for the
 equivalent of n=1 to 10,
- DY,YMAX see DX,XMAX (default DY=YMAX=0.0),

for SCHCOPL

- AB(J) source wire halflengths (maximum=30),
- MN(J) receiver wire halflengths (receiver wire assumed parallel
 to source wire and equatorially centered, maximum=30),
- * NSP number of AB and MN pairs to be calculated (maximum=30),
- Y separation of source and receiver wires (default=.01),

RMAX upper limit of double integral which is used to correct the values obtained with an infinite wire electric field routine (default=1000*MAX(AB(J))).

DOUBLE AND HANKEL INTEGRATION CONTROL PARAMETERS:

TOL tolerance for adaptive Hankel transform calculations, (default=1.e-8), see Anderson (1975) for details,

FINTL1 tolerance for integration along source wire (default=1.e-6),

FINTL2 tolerance for integration along receiver wire (default=1.e-4) [Note: TOL<FINTL1<FINTL2 is recommended],

IN1,IN2 -1 for adaptive quadrature integration,
=2 for non-adaptive quadrature integration,
(IN1 is for integral across receiver wire and
IN2 is for integral across source wire, defaults=1),
see Patterson (1973) for details,

NFIN interval in log-space with which the quintic spline nodes for double integration are calculated, interval=0.2/nfin (default=1),

MEV1,MEV2 maximum number of function evaluations for respective integration routines (default=300). These values may need to be increased if FINTL1 or FINTL2 are decreased, respectively.

TRANSIENT CALCULATION PARAMETERS:

* TMAX maximum time (seconds) desired,

* TMIN minimum time desired,

TFLAG =0 computes frequency response alone (default),
=1 computes frequency and transient response,
=2 computes transient response alone,
=3 computes transient response for a frequency response previously computed in order to conserve calculations;
note: the transient response computed is the step response,

RC time constant of single pole low-pass filter to be convolved with frequency response prior to transient calculation (default=0.0). No convolution is done if RC=0.0.

Program output will be the EM coupling (real and imaginary parts, or amplitude in volts per amp and phase in degrees) and Percent Frequency Effect (PFE) using the the formula

$$PFE(J) = 100. * (1. - CABS(Z(J)) / CABS(Z(1))) ,$$

where Z(J) is the EM coupling at the J-th frequency and CABS signifies taking the absolute value of a complex number. No attempt is made to normalize the coupling values. As an example, the coupling between two collinear wires 100 m in length whose closest points are 100 m apart (dipole-dipole configuration, N=1) on a 0.1 mho/meter halfspace is computed below using EMCUPL. The lines beginning and ending with a dollar sign (normal namelist delimiters) are input lines. Remember that the program is interactive so each input line is followed immediately by the corresponding output.

Example Calculations Using EMCUPL:

\$PARMS M=1,SIG=.1,L=50,YM=0,YN=0,XM=150,XN=250,FM=100,FO=.01,NF=4\$

SOURCE LENGTH = 0.100E+03

1 LAYER MODEL
SIG= 0.1000E+00

RECEIVER ELECTRODES AT (0.1500E+03,0.0000E+00)
(0.2500E+03,0.0000E+00)

FREQUENCY	REAL	IMAG	MAGN	PFE	PHASE
0.1000E-01	0.5305E-02	-0.3261E-06	0.5305E-02	0.0000E+00	-0.3522E-02
0.1778E-01	0.5305E-02	-0.5784E-06	0.5305E-02	0.6583E-04	-0.6247E-02
0.3162E-01	0.5305E-02	-0.1025E-05	0.5305E-02	0.2227E-03	-0.1107E-01
0.5623E-01	0.5305E-02	-0.1814E-05	0.5305E-02	0.5914E-03	-0.1959E-01

0.1000E+00	0.5305E-02	-0.3204E-05	0.5305E-02	0.1455E-02	-0.3461E-01
0.1778E+00	0.5305E-02	-0.5649E-05	0.5305E-02	0.3468E-02	-0.6101E-01
0.3162E+00	0.5305E-02	-0.9929E-05	0.5305E-02	0.8140E-02	-0.1072E+00
0.5623E+00	0.5304E-02	-0.1738E-04	0.5304E-02	0.1889E-01	-0.1877E+00
0.1000E+01	0.5303E-02	-0.3025E-04	0.5303E-02	0.4342E-01	-0.3269E+00
0.1778E+01	0.5300E-02	-0.5226E-04	0.5300E-02	0.9864E-01	-0.5649E+00
0.3162E+01	0.5293E-02	-0.8930E-04	0.5293E-02	0.2209E+00	-0.9667E+00
0.5623E+01	0.5277E-02	-0.1503E-03	0.5279E-02	0.4854E+00	-0.1632E+01.
0.1000E+02	0.5244E-02	-0.2478E-03	0.5250E-02	0.1042E+01	-0.2706E+01
0.1778E+02	0.5175E-02	-0.3963E-03	0.5190E-02	0.2169E+01	-0.4380E+01
0.3162E+02	0.5038E-02	-0.6069E-03	0.5075E-02	0.4347E+01	-0.6869E+01
0.5623E+02	0.4786E-02	-0.8723E-03	0.4865E-02	0.8303E+01	-0.1033E+02
0.1000E+03	0.4365E-02	-0.1143E-02	0.4512E-02	0.1495E+02	-0.1467E+02

ENTER \$PARMS CHANGES ONLY\$

\$PARMS TFLAG=3,TMAX=1,TMIN=.001\$

RECEIVER ELECTRODES AT (0.1500E+03,0.0000E+00)
(0.2500E+03,0.0000E+00)

TIME(SEC)	OBS VOLTS/AMP
0.1000E+01	0.5305E-02
0.8187E+00	0.5305E-02
0.6703E+00	0.5305E-02
0.5488E+00	0.5305E-02
0.4493E+00	0.5305E-02
0.3679E+00	0.5305E-02
0.3012E+00	0.5305E-02
0.2466E+00	0.5305E-02
0.2019E+00	0.5304E-02
0.1653E+00	0.5304E-02
0.1353E+00	0.5304E-02
0.1108E+00	0.5303E-02
0.9072E-01	0.5303E-02
0.7427E-01	0.5302E-02
0.6081E-01	0.5301E-02
0.4979E-01	0.5299E-02
0.4076E-01	0.5297E-02
0.3337E-01	0.5294E-02
0.2732E-01	0.5291E-02
0.2237E-01	0.5286E-02
0.1832E-01	0.5279E-02
0.1500E-01	0.5271E-02
0.1228E-01	0.5260E-02
0.1005E-01	0.5244E-02
0.8230E-02	0.5223E-02
0.6738E-02	0.5195E-02
0.5517E-02	0.5160E-02
0.4517E-02	0.5118E-02
0.3698E-02	0.5068E-02
0.3028E-02	0.5008E-02

0.2479E-02	0.4935E-02
0.2029E-02	0.4848E-02
0.1662E-02	0.4742E-02
0.1360E-02	0.4613E-02
0.1114E-02	0.4456E-02

ENTER \$PARMS CHANGES ONLY\$
\$PARMS M=0\$

STOP

The programs could have been run in batch mode for this same model with the following input cards:

\$PARMS M=1,SIG=.1,L=50,YM=0,YN=0,XM=150,XN=250,FM=100,F0=.01,NF=4\$
\$PARMS TFLAG=3,TMAX=1,TMIN=.001\$
\$PARMS M=0\$

The initial model was computed for four frequencies per decade between 0.01 and 100 Hertz. The second input line caused the transient step response to be computed between 0.001 and 1 second using the previously computed frequency response. The third and final line, specifying the number of layers as zero, causes the program to stop execution. The example is a simple one, but it illustrates the basic usage of the two main EM coupling programs described in this paper. For a general

multilayered earth model with real conductivities, one would replace the "M=1,SIG=.1" with M= the number of layers, SIG= the conductivities, and D= the layer thicknesses. For complex conductivities, ICMPLX would be set to 1 and a user-defined function SIGMA would be required. Parameters M and D would still be set in this case. As an example of using complex conductivities, the coupling between the two wires in the previous example will be computed for a halfspace whose conductivity varies with frequency as described by a Cole-Cole relaxation model

$$\text{sigl}(\omega) = \text{sigl}(0) / [1 - m * (1 - 1 / (1 + (i * \omega * \tau)^c))],$$

where $\text{sigl}(0) = 0.1$ mho/meter,

$m = 0.3$, chargeability,

$i = \sqrt{-1}$,

$c = 0.25$, frequency dependence, and

$\tau = 0.4$, time constant.

For the FORTRAN subprogram, see COMPLEX FUNCTION SIGMA in this appendix.

ENTER \$PARMS PARAMETERS\$

\$PARMS M=1,ICMPLX=1,L=50,YM=0,YN=0,XM=150,XN=250,FM=100,F0=.01,NF=4\$

SOURCE LENGTH = 0.100E+03

1 LAYER MODEL
COMPLEX CONDUCTIVITIES USED

RECEIVER ELECTRODES AT (0.1500E+03,0.0000E+00)

(0.2500E+03,0.0000E+00)

FREQUENCY	REAL	IMAG	MAGN	PFE	PHASE
0.1000E-01	0.4863E-02	-0.1283E-03	0.4865E-02	0.0000E+00	-0.1512E+01
0.1778E-01	0.4814E-02	-0.1365E-03	0.4816E-02	0.1003E+01	-0.1624E+01
0.3162E-01	0.4762E-02	-0.1439E-03	0.4764E-02	0.2066E+01	-0.1731E+01
0.5623E-01	0.4707E-02	-0.1507E-03	0.4710E-02	0.3184E+01	-0.1833E+01
0.1000E+00	0.4651E-02	-0.1567E-03	0.4653E-02	0.4346E+01	-0.1930E+01
0.1778E+00	0.4592E-02	-0.1623E-03	0.4595E-02	0.5541E+01	-0.2024E+01
0.3162E+00	0.4533E-02	-0.1681E-03	0.4536E-02	0.6759E+01	-0.2123E+01
0.5623E+00	0.4473E-02	-0.1753E-03	0.4476E-02	0.7989E+01	-0.2244E+01
0.1000E+01	0.4412E-02	-0.1862E-03	0.4416E-02	0.9225E+01	-0.2416E+01
0.1778E+01	0.4350E-02	-0.2044E-03	0.4355E-02	0.1047E+02	-0.2690E+01
0.3162E+01	0.4286E-02	-0.2357E-03	0.4293E-02	0.1175E+02	-0.3147E+01
0.5623E+01	0.4215E-02	-0.2884E-03	0.4225E-02	0.1315E+02	-0.3914E+01
0.1000E+02	0.4127E-02	-0.3734E-03	0.4144E-02	0.1481E+02	-0.5170E+01
0.1778E+02	0.4004E-02	-0.5016E-03	0.4036E-02	0.1704E+02	-0.7140E+01
0.3162E+02	0.3815E-02	-0.6762E-03	0.3875E-02	0.2034E+02	-0.1005E+02
0.5623E+02	0.3520E-02	-0.8771E-03	0.3627E-02	0.2544E+02	-0.1399E+02
0.1000E+03	0.3086E-02	-0.1040E-02	0.3256E-02	0.3306E+02	-0.1863E+02

ENTER \$PARMS CHANGES ONLY\$

\$PARMS TFLAG=3,TMAX=1,TMIN=.001\$

RECEIVER ELECTRODES AT (0.1500E+03,0.0000E+00)
(0.2500E+03,0.0000E+00)

TIME(SEC)	OBS VOLTS/AMP
0.1000E+01	0.4659E-02
0.8187E+00	0.4641E-02
0.6703E+00	0.4618E-02
0.5488E+00	0.4594E-02
0.4493E+00	0.4574E-02
0.3679E+00	0.4557E-02
0.3012E+00	0.4541E-02
0.2466E+00	0.4521E-02
0.2019E+00	0.4495E-02
0.1653E+00	0.4471E-02
0.1353E+00	0.4451E-02
0.1108E+00	0.4436E-02
0.9072E-01	0.4419E-02
0.7427E-01	0.4396E-02
0.6081E-01	0.4369E-02
0.4979E-01	0.4345E-02
0.4076E-01	0.4325E-02
0.3337E-01	0.4309E-02
0.2732E-01	0.4289E-02
0.2237E-01	0.4263E-02
0.1832E-01	0.4227E-02
0.1500E-01	0.4197E-02
0.1228E-01	0.4182E-02

0.1005E-01	0.4143E-02
0.8230E-02	0.4087E-02
0.6738E-02	0.4036E-02
0.5517E-02	0.3999E-02
0.4517E-02	0.3970E-02
0.3698E-02	0.3937E-02
0.3028E-02	0.3888E-02
0.2479E-02	0.3813E-02
0.2029E-02	0.3704E-02
0.1662E-02	0.3553E-02
0.1360E-02	0.3351E-02
0.1114E-02	0.3091E-02

ENTER \$PARMS CHANGES ONLY\$
\$PARMS M=0\$

STOP

Note that the only difference in input between this example and the previous one is the replacement of SIG=.1 with ICMPLX=1. The ICMPLX parameter signals EMCUPL to use complex function SIGMA for all necessary layer conductivities and to ignore the values stored in the SIG parameter array. Comparison of the EM coupling over both halfspace models shows a decrease in impedance magnitude and an increase in PFE and impedance phase when the halfspace conductivity is complex. The theoretical step response has a longer decay time constant for the halfspace having a complex conductivity. Although these characteristics seem to be apparent in most of the models computed thus far, generalizations about the effects of complex conductivities on EM coupling are well beyond the scope of this paper.

Program EMCUPL is an entirely general routine allowing arbitrary orientations for the two wires; however, most IP and

complex resistivity work is done with a few basic arrays. For example, the collinear dipole-dipole array for 100 m dipoles would correspond to the namelist specification "..., L=50, XM=150, XN=250, YM=0, YN=0, DX=100, XMAX=1100, ..." for n=1 to 9. The collinear bipole-dipole array for a 1500 m bipole and a 100 m dipole would be "..., L=750, XM=850, XN=950, YM=0, YN=0, DX=100, XMAX=1800, ...", again for n=1 to 9. The equatorial dipole-dipole array for 100 m dipoles would be "..., L=50, XM=-50, XN=50, YM=100, YN=100, DX=0, DY=100, YMAX=950, ..." for n=1 to 9. Finally, the EM reflection, or perpendicular wire array, corresponds to "..., L=50, XM=50, XN=50, YM=100, YN=200, DX=0, DY=100, YMAX=1050, ..." for n=1 to 9. The program determines whether previous computations may be saved for further use for these symmetric arrays.

EMCUPL could be used to compute coupling for a Schlumberger or Wenner array, but it is not recommended because the adaptive integration routines become numerically unstable for small Y separations, resulting in inaccurate results and excessive run times. As stated earlier, this is precisely why equation (1) was reformulated as equation (4) and program SCHCOPL was written.

Main Programs and Subprograms Source Listing

The following is a complete alphabetical listing of the main programs and subprograms preceded by a list of names and the line numbers they begin on:

Name	Beginning Line
CORFUN	00000010
CSPLNT	00000210
CZEX	00000430
EMCUPL (Main)	00000700
F2MOD	00004310
F3	00004400
F7G	00004480
FINFUN	00004600
FINQ	00005020
FINQDF	00005180
FUNINT	00006210
INFNEX	00006370
QPOINT	00006520
QUINT *	00006710
RECUR2	00007480
RECURS	00007830
RLAGF1	00008140
SCHCOPL (Main)	00010500
SETSPL	00013450
SIGMA	00014110
SPLIN1	00014290
ZBLOCK	00015490
ZEX	00018570
ZFOUR0	00018780
ZHANK0	00020770
ZLAGH0	00022520
ZQUAD1 **	00024760
ZQUAD2 **	00025360
ZSUB1 **	00025960
ZSUB2 **	00026920
ZSUBA1 **	00027880
ZSUBA2 **	00028760

* Converted from ALGOL to FORTRAN, as published by Herriot and Reinsch (1976): Copyright 1976, Assoc. for Computing Machinery, Inc.; permission to republish, all or in part, granted by ACM.

** These are modified versions of subroutines QUAD, QSUB, and QSUBA published by Patterson (1973): Copyright 1973, Assoc. for Computing Machinery, Inc.; permission to republish, all or in part, granted by ACM.

Source Availability

The current version of the source code may be obtained by writing directly to the authors. A magnetic tape copy of the source code will be sent to requestors to be copied and returned to the authors. This method of releasing the program was selected in order to satisfy requests for the latest updated version. The magnetic tape will be recorded in the following mode (unless otherwise requested):

Industry compatible: 9-track, unlabeled, EBCDIC mode, odd-parity, 800 bpi density, 80-character records (unblocked card images), and contained on one file.

COMPLEX FUNCTION CORFUN(X)	00000010
C--COMPUTES FINITE INTEGRAL OVER (RMIN,RMAX) OF	00000020
C COMPLEX FUNCTION FUNINT (SPLINE INTERPOLATOR)	00000030
C ASSUMES PRIOR CALL TO SETSPL	00000040
REAL L	00000050
EXTERNAL FUNINT	00000060
COMPLEX ESUM,ZSUBA2,ZSUB2	00000070
COMMON/FIN/R1,R2,R,L,SIG1,XX,YY	00000080
COMMON/CORRF/Y,RMIN,RMAX	00000090
COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW	00000100
XX=X	00000110
YY=Y	00000120
C ADAPTIVE GAUSSIAN QUADRATURE INTEGRATION	00000130
IF(INTYPE.EQ.1) CORFUN=ZSUBA2(RMIN-X,RMAX-X,FINTOL,NEV,ICK,ESUM,	00000140
1 FUNINT,MEV)	00000150
C NON-ADAPTIVE GAUSSIAN QUADRATURE INTEGRATION	00000160
IF(INTYPE.EQ.2) CORFUN=ZSUB2(RMIN-X,RMAX-X,FINTOL,NEV,ICK,ESUM,	00000170
1 FUNINT,MEV)	00000180
RETURN	00000190
END	00000200
REAL FUNCTION CSPLNT(T)	00000210
C--CUBIC SPLINE INTERPOLATOR OF INDEPENDENT VARIABLE Y	00000220
C FOR DEPENDENT VARIABLE X. IF	00000230
C T >=X(N) CSPLNT=Y(N)	00000240
C T <=X(1) CSPLNT=Y(1)	00000250
C--ASSUMES PRIOR CALL TO SPLIN1	00000260
COMMON/CSPLINE/A(50),B(50),C(50),N,X(50),Y(50)	00000270
IF(T.LT.X(1).OR.T.GT.X(N)) GO TO 2	00000280
N1=N-1	00000290
DO 1 I=1,N1	00000300
J=I	00000310
IF(T.LT.X(I+1)) GO TO 3	00000320
1 CONTINUE	00000330
2 IF(T.GT.X(N)) CS=Y(N)	00000340
IF(T.LT.X(1)) CS=Y(1)	00000350
CSPLNT=CS	00000360
RETURN	00000370
3 Z=T-X(J)	00000380
CS=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z	00000390
CSPLNT=CS	00000400
RETURN	00000410
END	00000420
COMPLEX FUNCTION CZEX(B,NEW,R)	00000430
C--CZEX COMPUTES THE P(R) TERM WHICH IS	00000440
C DOUBLE INTEGRATED OVER FINITE LIMITS.	00000450
C IT IS PART OF THE EQUATION FOR THE	00000460
C ELECTRIC FIELD OF AN ELECTRIC DIPOLE.	00000470
C	00000480
C B INDUCTION NUMBER	00000490
C R DISTANCE	00000500

C	NEW CONTROLS ZLAGH0 INTEGRATION	00000510
C		00000520
C	--CZEX IS IDENTICAL TO ZEX EXCEPT THAT	00000530
C	IT ALLOWS FOR COMPLEX CONDUCTIVITIES	00000540
C	WHICH ARE COMPUTED BY USER-DEFINED COMPLEX ROUTINE	00000550
C	SIGMA	00000560
	COMPLEX ZLAGH0,TWODEL3,ONE,SIGMA,SIGMA1,CB,CK	00000570
	EXTERNAL F3	00000580
	COMMON/CFLAG/CK(10),ICMPLX	00000590
	COMMON/PARM/ISTEP,A1,A2,A3,SIG1,A5,M,TOL	00000600
	COMMON/CONST/DEL,DEL2,TWODEL3	00000610
	DATA ONE/(1.0,0.0)/	00000620
	CZEX=CMPLX(0.0,0.0)	00000630
	CB=CMPLX(B,B)*CSQRT(CK(1))	00000640
	IF(M.EQ.1) GO TO 2	00000650
	CZEX=ZLAGH0(ALOG(B),F3,TOL,LW,NEW)/B	00000660
2	CZEX=TWODEL3*CZEX+(ONE-(ONE+CB)*CEXP(-CB))/R**3	00000670
	RETURN	00000680
	END	00000690
C	***** PROGRAM EMCUPL *****	00000700
C	--PROGRAM EMCUPL CALCULATES THE ELECTROMAGNETIC COUP-	00000710
C	LING BETWEEN TWO STRAIGHT GROUNDED WIRES OF	00000720
C	ARBITRARY LENGTH AND ORIENTATION ON THE SURFACE OF A LAYERED	00000730
C	EARTH. ONE WIRE (REFERRED TO AS THE SOURCE WIRE) IS	00000740
C	CONSTRAINED TO LIE ALONG THE X-AXIS BETWEEN -L AND L.	00000750
C	THE OTHER WIRE (RECEIVER WIRE)	00000760
C	MUST BE SPECIFIED AS TWO PAIRS OF X,Y COORDINATES- ONE	00000770
C	PAIR FOR EACH WIRE END. THE INPUT PARAMETERS ARE:	00000780
C		00000790
C	SIG(I) CONDUCTIVITY OF THE ITH LAYER	00000800
C	D(I) THICKNESS OF THE ITH LAYER	00000810
C	M NUMBER OF LAYERS	00000820
C	XM,YM COORDINATES OF ONE RECEIVER WIRE END	00000830
C	XN,YN COORDINATES OF THE OTHER END	00000840
C	L SOURCE WIRE HALFLENGTH	00000850
C	DX INCREMENT FOR STEPPING RECEIVER POSITION	00000860
C	XMAX MAXIMUM X-VALUE CONSIDERED	00000870
C	DY INCREMENT FOR STEPPING RECEIVER POSITION	00000880
C	YMAX MAXIMUM Y-VALUE CONSIDERED	00000890
C	NF >0, NUMBER OF FREQUENCIES DESIRED PER DECADE	00000900
C	BETWEEN F0 AND FM	00000910
C	<0, NUMBER OF SPECIFIED FREQUENCIES IN FNF	00000920
C	F0,FM MINIMUM AND MAXIMUM FREQUENCIES DESIRED	00000930
C	FNF SPECIFIED FREQUENCIES	00000940
C	TOL TOLERANCE FOR HANKEL TRANSFORM CALCULATIONS	00000950
C	FINTL1 TOLERANCE FOR INTEGRATION ALONG SOURCE WIRE	00000960
C	FINTL2 TOLERANCE FOR INTEGRATION ALONG RECEIVER WIRE	00000970
C	IN1,IN2 =1 FOR ADAPTIVE QUADRATURE INTEGRATION	00000980
C	=2 FOR NON-ADAPTIVE QUADRATURE INTEGRATION	00000990
C	IN1 IS FOR THE OUTER INTEGRAL ACROSS THE RECEIVER	00001000
C	IN2 IS FOR THE INNER INTEGRAL ACROSS THE SOURCE	00001010

```

C  NFIN      INTERVAL IN LOG-SPACE WITH WHICH THE SPLINE      00001020
C            NODES FOR FINITE WIRE INTEGRATION ARE             00001030
C            CALCULATED, E. G. INTERVAL=0.2/NFIN               00001040
C            (DEFAULT=1)                                        00001050
C  MEV1,MEV2  MAXIMUM NUMBER OF FUNCTION EVALUATIONS FOR      00001060
C            RESPECTIVE INTEGRATION ROUTINES                   00001070
C  TMAX,TMIN  MAXIMUM AND MINIMUM TIME VALUES DESIRED        00001080
C  TFLAG      =0 COMPUTES FREQUENCY RESPONSE ALONE             00001090
C            =1 COMPUTES FREQUENCY AND TRANSIENT RESPONSE      00001100
C            =2 COMPUTES TRANSIENT RESPONSE ALONE              00001110
C            =3 COMPUTES TRANSIENT RESPONSE FOR A              00001120
C            FREQUENCY RESPONSE PREVIOUSLY COMPUTED            00001130
C  RC         TIME CONSTANT OF SINGLE POLE LOW-PASS            00001140
C            FILTER TO BE CONVOLVED WITH FREQUENCY             00001150
C            RESPONSE FOR TRANSIENT RESPONSE                   00001160
C            CALCULATIONS                                       00001170
C            NOTE: TRANSIENT RESPONSE IS THE STEP              00001180
C            RESPONSE                                           00001190
C  ICMPLX     =0 COMPUTES THE COUPLING USING THE REAL          00001200
C            CONDUCTIVITIES IN SIG ARRAY,                      00001210
C            =1 COMPUTES THE COUPLING USING THE COMPLEX         00001220
C            CONDUCTIVITIES COMPUTED BY THE USER-             00001230
C            DEFINED FUNCTION SIGMA(J,1./(DEL*DEL))             00001240
C  INFILE     (DEFAULT=5) INPUT FILE NUMBER                    00001250
C  OUFIL      (DEFAULT=6) OUTPUT FILE NUMBER                   00001260
C                                                    00001270
C  PROGRAM ORGANIZATION IS AS FOLLOWS:                        00001280
C                                                    00001290
C            EMCUPL                                             00001300
C            *                                                    00001310
C            *            DOUBLE            INTEGRATION          TRANSIENT CALC 00001320
C            *****                                                    00001330
C            *            *            *            *            *            * 00001340
C  SIGMA      FINQDF      SETSPL      ZSUB1  ZSUBA1      SPLIN1  RLAGF1 00001350
C            *            *            *            *            *            * 00001360
C            *            *****            *****            *            00001370
C            *            *            *            *            *            * 00001380
C            FINQ      ZEX  CZEX  QUINT  FINFUN      CSPLNT 00001390
C            *            *            *            *            *            * 00001400
C            *            *****            *****            *            00001410
C            *            *            *            *            *            * 00001420
C            ZHANKO     ZLAGHO  SIGMA  ZSUB2  ZSUBA2      00001430
C            *            *            *            *            *            * 00001440
C            *            *            *****            *            00001450
C            *            *            *            *            *            * 00001460
C            F7G      F3      FUNINT      00001470
C            *            *            *            *            *            * 00001480
C            RECUR2     RECURS      QPOINT      00001490
C                                                    00001500
C                                                    00001510
C  COMMON/CFLAG/CK(10),ICMPLX      00001520
C  COMMON/FINERR/HAKTOL,FINTL1,IN2,NFIN,NEV2,MEV2,ZERR2,LW 00001530

```

COMMON/FIN/RMAX,RMIN,R0,L,SIG1,XX,YY	00001540
COMMON/ENDS/XXM,YYM,XXN,YYN,DS	00001550
COMMON/THICK/D(9)	00001560
COMMON/MODEL/RK(10),DD(9),M	00001570
COMMON/PARM/IS,A1,A2,A3,SSIG1,A5,NLYR,TOL	00001580
COMMON/CSPLINE/D1(50),D2(50),D3(50),NN,BB(50),SPEC(50)	00001590
INTEGER OUFIL	00001600
REAL L,SIG(10),FNF(50),P(50),S(50),PS(2),T(50),V(50)	00001610
COMPLEX ZSUB1,ZSUBA1,ZERR1,ZERR2,ECOPL,FINQDF,EC(30,30),	00001620
1 CSPEC,SIGMA,SIGMA1,CK	00001630
EXTERNAL ZEX,FINFUN,CSPLNT,CZEX	00001640
NAMLIST/PARMS/SIG,M,D,XM,YM,XN,YN,L,TOL,FINTL1,FINTL2	00001650
1,IN1,IN2,NFIN,MEV1,MEV2,FM,F0,NF,FNF,DX,DY,XMAX,YMAX,TMAX,TMIN,	00001660
2 TFLAG,RC,ICMPLX,INFILE,OUFIL	00001670
DATA DEG/57.29577951/	00001680
DATA PS,TWOPI/0.0,0.0,6.283185308/	00001690
C	00001700
C--ASSIGN NAMLIST PARAMETER DEFAULTS	00001710
C	00001720
INFILE=5	00001730
OUFIL=6	00001740
MEV1=300	00001750
MEV2=300	00001760
NFIN=1	00001770
IN1=1	00001780
IN2=1	00001790
FINTL1=1.E-6	00001800
FINTL2=1.E-4	00001810
TOL=1.E-8	00001820
M=1	00001830
DX=0.0	00001840
DY=0.0	00001850
XMAX=0.0	00001860
YMAX=0.0	00001870
TFLAG=0	00001880
ICMPLX=0	00001890
RC=0.0	00001900
C	00001910
WRITE(OUFIL,501)	00001920
501 FORMAT(25H ENTER \$PARMS PARAMETERS\$)	00001930
100 READ(INFILE,PARMS)	00001940
IF(M.EQ.0) STOP	00001950
C--TFLAG = 3 ASSUMES THAT FREQUENCY VALUES HAVE ALREADY	00001960
C BEEN COMPUTED IN A PREVIOUS RUN	00001970
IF(TFLAG.EQ.3) GO TO 21	00001980
IF(DX.EQ.0.0) XMAX=0.0	00001990
IF(DY.EQ.0.0) YMAX=0.0	00002000
C	00002010
C--DEFINE EQUIVALENT COMMON PARAMETERS	00002020
C	00002030
NLYR=M	00002040
HAKTOL=TOL	00002050

C		00002060
	IF(ICMPLX.EQ.1) GO TO 300	00002070
	SIG1=SIG(1)	00002080
	DO 10 I=1,M	00002090
10	RK(I)=SIG(I)/SIG1	00002100
	GO TO 301	00002110
300	SIG1=REAL(SIGMA(1,0.0))	00002120
	RK(1)=SIG1	00002130
301	M1=M-1	00002140
	SSIG1=SIG1	00002150
C		00002160
C--CHECK THAT ICMPLX = 0 OR 1 ONLY		00002170
C		00002180
	IF(ICMPLX.EQ.0.OR.ICMPLX.EQ.1) GO TO 302	00002190
	WRITE(OUFILE,502)	00002200
502	FORMAT(34H ICMPLX MUST BE SET TO 0 OR 1 ONLY)	00002210
	GO TO 100	00002220
302	TWL=2.*L	00002230
C		00002240
C--PRINT MODEL PARAMETERS		00002250
C		00002260
	WRITE(OUFILE,503) TWL,M	00002270
503	FORMAT(/16H SOURCE LENGTH =,E15.3//5X,I3,12H LAYER MODEL)	00002280
	IF(ICMPLX.EQ.0) WRITE(OUFILE,504) (SIG(I),I=1,M)	00002290
504	FORMAT(5H SIG=,12E12.4)	00002300
505	FORMAT(5H D =,12E12.4)	00002310
	IF(ICMPLX.EQ.1) WRITE(OUFILE,506)	00002320
506	FORMAT(28H COMPLEX CONDUCTIVITIES USED)	00002330
	IF(M.GT.1) WRITE(OUFILE,505) (D(I),I=1,M1)	00002340
C--MAKE SURE XM IS LESS THAN XN		00002350
	IF(XM.LT.XN) GO TO 1	00002360
	X=XM	00002370
	XM=XN	00002380
	XN=X	00002390
	Y=YM	00002400
	YM=YN	00002410
	YN=Y	00002420
C		00002430
C--CALCULATE MAXIMUM AND MINIMUM DISTANCE BETWEEN WIRES		00002440
C	IN ORDER TO DEFINE ARGUMENT RANGE FOR ZEX OR CZEX	00002450
C	EVALUATIONS	00002460
C		00002470
1	Y2=YM*YM	00002480
	R1=SQRT((XM+L)**2+Y2)	00002490
	R2=SQRT((XM-L)**2+Y2)	00002500
	Y2=YN*YN	00002510
	R3=SQRT((XN+L)**2+Y2)	00002520
	R4=SQRT((XN-L)**2+Y2)	00002530
	RMAX=(L+AMAX1(ABS(XN),ABS(XM),ABS(XMAX)))*2	00002540
	RMAX=SQRT(RMAX+AMAX1(ABS(YN),ABS(YM),ABS(YMAX)))*2)	00002550
	RMIN=AMIN1(R1,R2,R3,R4)	00002560
	XX=XM+0.5*(XN-XM)	00002570

CXX=XX	00002580
YY=YM+0.5*(YN-YM)	00002590
RO=SQRT(XX*XX+YY*YY)	00002600
IF(XN.NE.XM) DS=(YN-YM)/(XN-XM)	00002610
CDS=COS(ATAN(DS))	00002620
3 ITRUE=0	00002630
IF(ABS(XM).LE.L) ITRUE=1	00002640
IF(ABS(XN).LE.L) ITRUE=ITRUE+2	00002650
IF(ITRUE.EQ.0) GO TO 6	00002660
IF(ITRUE.LT.3) GO TO 5	00002670
4 RMIN=AMIN1(YM,YN)	00002680
GO TO 6	00002690
5 IF((ITRUE.EQ.1.AND.YN.LT.YM).OR.(ITRUE.EQ.2.AND.YM.LT.YN)) GO TO 4	00002700
RL=L*SIGN(1.,ITRUE-1.5)	00002710
XP=(-YM/DS)+XM-RL	00002720
YP=-(XP+RL)/DS	00002730
RMIN=SQRT(XP*XP+YP*YP)	00002740
6 CONTINUE	00002750
CON=-1./(SIG1*6.283185308)	00002760
XB=SQRT(1./(SIG1*3.9478417E-6))	00002770
NX=0	00002780
NY=0	00002790
IF(DX.NE.0.0) NX=1+(XMAX-AMAX1(XM,XN))/DX	00002800
IF(DY.NE.0.0) NY=1+(YMAX-AMAX1(YM,YN))/DY	00002810
IF(NF.LT.0) GO TO 13	00002820
NN=NF*ALOG10(FM/F0)+1	00002830
F=F0	00002840
DELX=EXP(2.30258509/FLOAT(NF))	00002850
GO TO 14	00002860
13 NN=-NF	00002870
C	00002880
C--FIRST LOOP OVER FREQUENCIES	00002890
C	00002900
14 DO 20 JJ=1,NN	00002910
IF(NF.GT.0) GO TO 11	00002920
F=FNF(JJ)	00002930
GO TO 12	00002940
11 IF(JJ.GT.1) F=F*DELX	00002950
12 BB(JJ)=F	00002960
DEL=XB/SQRT(F)	00002970
C--COMPUTE COMPLEX SIGMA1	00002980
CF=1./(DEL*DEL)	00002990
IF(ICMPLX.EQ.1) SIGMA1=SIGMA(1,CF)/SIG1	00003000
DO 15 I=1,M	00003010
IF(ICMPLX.EQ.1) CK(I)=SIGMA(I,CF)/SIG1	00003020
15 DD(I)=2.*D(I)/DEL	00003030
IPOS=0	00003040
C--SET SPLINE COEFFICIENTS UNLESS WIRES ARE	00003050
C PERPENDICULAR	00003060
IF(XN.EQ.XM) GO TO 16	00003070
IF(ICMPLX.EQ.0) CALL SETSPL(ZEX,DEL,RMAX,RMIN)	00003080
IF(ICMPLX.EQ.1) CALL SETSPL(CZEX,DEL,RMAX,RMIN)	00003090

```

16      NEW=1                                00003100
C                                           00003110
C--LOOP OVER X COORDINATE FIRST, THEN Y COORDINATE 00003120
C                                           00003130
      DO 20 IX=1,NX                        00003140
        XXM=XM+DX*(IX-1)                  00003150
        XXN=XN+DX*(IX-1)                  00003160
        DO 20 IY=1,NY                      00003170
          YYM=YM+DY*(IY-1)                00003180
          YYN=YN+DY*(IY-1)                00003190
          Y2=YYM*YYM                      00003200
          R1=SQRT((XXM+L)**2+Y2)           00003210
          R2=SQRT((XXM-L)**2+Y2)           00003220
          Y2=YYN*YYN                      00003230
          R3=SQRT((XXN+L)**2+Y2)           00003240
          R4=SQRT((XXN-L)**2+Y2)           00003250
18      IPOS=IPOS+1                        00003260
          ECOPL=CMPLX(0.0,0.0)             00003270
          IF(XXN.EQ.XXM) GO TO 19           00003280
C--CALCULATE WIRE COUPLING (DOUBLE INTEGRAL) UNLESS WIRES ARE 00003290
C PERPENDICULAR                          00003300
          IF(DS.EQ.0.0.AND.CXX.EQ.0.0) GO TO 24 00003310
          IF(IN1.EQ.1) ECOPL=CDS*ZSUBA1(XXM,XXN,FINTL1,NEV1,ICK, 00003320
1      ZERR1,FINFUN,MEV1)                 00003330
          IF(IN1.EQ.2) ECOPL=CDS*ZSUB1(XXM,XXN,FINTL1,NEV1,ICK, 00003340
1      ZERR1,FINFUN,MEV1)                 00003350
          GO TO 25                          00003360
24      IF(IN1.EQ.1) ECOPL=2.*CDS*ZSUBA1(0.0,XXN,FINTL1,NEV1, 00003370
1      ICK,ZERR1,FINFUN,MEV1)             00003380
          IF(IN1.EQ.2) ECOPL=2.*CDS*ZSUB1(0.0,XXN,FINTL1,NEV1, 00003390
1      ICK,ZERR1,FINFUN,MEV1)             00003400
25      IF(MEV1.GE.NEV1-1.AND.ICK.GE.0) GO TO 19 00003410
          WRITE(OUFILE,520) NEV1,MEV1,ICK,BR(JJ),IX,IY 00003420
520      FORMAT(40H GAUSS QUADRATURE: COMPUTED INTEGRAL MAY, 00003430
1      13H BE ERRONEOUS/6H NEV1=,I4,6H MEV1=,I4,5H ICK=,I2, 00003440
2      6H FREQ=,E15.4,9H POS: IX=,I3,4H IY=,I3/) 00003450
19      EC(JJ,IPOS)=CON*(ECOPL-FINQDF(DEL,R4,R3,R2,R1, 00003460
1      FINTL1,NEW))                       00003470
          IF(ICMPLX.EQ.0) GO TO 20         00003480
          EC(JJ,IPOS)=EC(JJ,IPOS)/SIGMA1  00003490
20      NEW=0                              00003500
          IF(TFLAG.EQ.0) GO TO 22          00003510
21      NT=AIN(5.*ALOG(TMAX/TMIN))+1      00003520
          NT1=NT+1                        00003530
C--CALCULATE IN NORMALIZED TIME          00003540
C NORM TIME =TWOPI * REAL TIME          00003550
          XO=ALOG(TMAX*TWOPI)+0.2         00003560
22      IPOS=0                            00003570
C                                           00003580
C--PRINT OUTPUT                          00003590
C                                           00003600
          DO 41 IX=1,NX                    00003610

```


	CALL SPLIN1(NN,0,BB,SPEC,D1,D2,D3,0,PS,P,S)	00004060
	NEW=1	00004070
	WRITE(OUFILE,510)	00004080
510	FORMAT(/6X,9HTIME(SEC),4X,15HOBS VOLTAGE/AMP)	00004090
C		00004100
C--	COMPUTE SINE TRANSFORM (RLAGF1) OF CUBICALLY SPLINED	00004110
C	FREQUENCY FUNCTION (SPLINE INTERPOLATOR-CSPLNT)	00004120
C		00004130

```

DO 60 J=1,NT                                00004140
    I=NT1-J                                00004150
    X=X0-0.2*J                              00004160
    T(I)=EXP(X)                             00004170
    FDC=1.0                                00004180
    IF(RC.GT.0.0) FDC=1.-EXP_(-T(I)/(TWOPI*RC)) 00004190
    V(I)=0.636619772*RLAGF1(X,CSPLNT,TOL,LW,NEW)/T(I)+ 00004200
1      DC*FDC                               00004210.
    T(I)=T(I)/TWOPI                         00004220
    WRITE(OUFILE,509) T(I),V(I)             00004230
60      NEW=0                               00004240
41      WRITE(OUFILE,511)                   00004250
511  FORMAT(1H )                           00004260
    WRITE(OUFILE,512)                       00004270
512  FORMAT(27H ENTER $PARMS CHANGES ONLY$) 00004280
    GO TO 100                               00004290
    END                                     00004300

    COMPLEX FUNCTION F2MOD(G)                00004310
C--KERNEL FUNCTION FOR SINE INTEGRAL IN ROUTINE INFNEX 00004320
C--CALLS RECURS                            00004330
    COMPLEX V1,F1,C                         00004340
    CALL RECURS(G,V1,F1)                   00004350
    C=G                                    00004360
    F2MOD=CMPLX(1.0,0.0)/(C+V1*F1)          00004370
    RETURN                                00004380
    END                                    00004390

    COMPLEX FUNCTION F3(G)                  00004400
    COMPLEX V1,F1,C,ONE                     00004410
    DATA ONE/(1.0,0.0)/                   00004420
    CALL RECURS(G,V1,F1)                   00004430
    C=G                                    00004440
    F3=(V1*C*(ONE-F1))/((C+V1*F1)*(C+V1)) 00004450
    RETURN                                00004460
    END                                    00004470

    COMPLEX FUNCTION F7G(G)                 00004480
C--KERNEL OF HANKEL TRANSFORM USED BY        00004490
C ROUTINES FINQ AND SCHCOPL                 00004500
C--CALLS RECUR2                            00004510
    COMPLEX V1,F1,L1,I1,ONE,TWO,C          00004520
    DATA I1,ONE,TWO/(0.0,1.0),(1.0,0.0),(2.0,0.0)/ 00004530
    CALL RECUR2(G,V1,F1,L1)                00004540
    C=G                                    00004550
    F7G=I1*V1*(L1-ONE)+(TWO*V1*(ONE-F1))/((C+V1*F1)*(C+V1)) 00004560
    F7G=F7G/G                              00004570
    RETURN                                00004580
    END                                    00004590

    COMPLEX FUNCTION FINFUN(X)              00004600
C--COMPUTES FINITE INTEGRAL OVER INTERVAL -L,L 00004610

```

C (L PASSED IN THROUGH COMMON AREA FIN) OF	00004620
C COMPLEX FUNCTION FUNINT (SPLINE INTERPOLATOR)	00004630
C AT FIELD POINT (XX,YY).	00004640
C ASSUMES PRIOR CALL TO SETSPL	00004650
C--CALLS FUNINT	00004660
C ZSUBA2 - ADAPTIVE GAUSSIAN INTEGRATION	00004670
C ZSUB2 - NON-ADAPTIVE GAUSSIAN INTEGRATION	00004680
REAL L	00004690
EXTERNAL FUNINT	00004700
COMPLEX ESUM,ZSUBA2,ZSUB2	00004710
COMMON/FIN/R1,R2,R,L,SIG1,XX,YY	00004720
COMMON/ENDS/XM,YM,XN,YN,DS	00004730
COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW	00004740
C CHECK TO SEE THAT X IS IN THE RANGE XM -> XN	00004750
IF(X.LE.XN.AND.X.GE.XM) GO TO 1	00004760
WRITE(6,100) X	00004770
100 FORMAT(34H FINFUN: X NOT IN PROPER RANCE, X=,E12.4)	00004780
STOP	00004790
1 XX=X	00004800
YY=YM+DS*(X-XM)	00004810
IF(ABS(X).LT.L) GO TO 8	00004820
IF(INTYPE.EQ.1) FINFUN=ZSUBA2(X-L,X+L,FINTOL,NEV,ICK,ESUM,FUNINT,	00004830
1 MEV)	00004840
IF(INTYPE.EQ.2) FINFUN=ZSUB2(X-L,X+L,FINTOL,NEV,ICK,ESUM,FUNINT,	00004850
1 MEV)	00004860
GO TO 10	00004870
8 XMIN=AMIN1(ABS(X-L),ABS(X+L))	00004880
XMAX=AMAX1(ABS(X-L),ABS(X+L))	00004890
IF(INTYPE.EQ.1) FINFUN=2.*ZSUBA2(0.,XMIN,FINTOL,NEV,ICK,ESUM,	00004900
1 FUNINT,MEV)	00004910
IF(INTYPE.EQ.2) FINFUN=2.*ZSUB2(0.,XMIN,FINTOL,NEV,ICK,ESUM,	00004920
1 FUNINT,MEV)	00004930
IF(X.EQ.0.0) GO TO 10	00004940
IF(INTYPE.EQ.1) FINFUN=FINFUN+ZSUBA2(XMIN,XMAX,FINTOL,NEV1,ICK,	00004950
1 ESUM,FUNINT,MEV)	00004960
IF(INTYPE.EQ.2) FINFUN=FINFUN+ZSUB2(XMIN,XMAX,FINTOL,NEV1,ICK,	00004970
1 ESUM,FUNINT,MEV)	00004980
NEV=NEV+NEV1	00004990
10 RETURN	00005000
END	00005010
COMPLEX FUNCTION FINQ(DEL,R,TOL)	00005020
C--FINQ CALCULATES THE JO HANKEL TRANSFORM	00005030
C (USING DIGITAL FILTER ROUTINE ZHANK0)	00005040
C REQUIRED BY PROGRAM EMCUPL	00005050
C--CALLS ZHANK0, F7G	00005060
COMMON/MODEL/RK(10),D(9),M	00005070
COMPLEX ZHANK0,ES	00005080
EXTERNAL F7G	00005090
B=R/DEL	00005100
FINQ=CMPLX(0.0,0.0)	00005110
IF(M.EQ.1) GO TO 1	00005120

FINQ=ZHANKO(ALOG(B),F7G,TOL,LW)/B	00005130
FINQ=FINQ*CMPLX(0.0,1./DEL)	00005140
1 FINQ=FINQ-1./R	00005150
RETURN	00005160
END	00005170
COMPLEX FUNCTION FINQDF(DEL,R1,R2,R3,R4,TOL,NEW)	
C--COMPUTES FINQ AT FOUR SEPARATIONS USING	00005180
C PREVIOUS CALCULATED RESULTS IF NEW=0	00005190
C DEL SKIN DEPTH IN FIRST LAYER	00005200
C R1,R2,R3,R4 THE FOUR SEPARATIONS	00005210
C NEW = 1 FIRST CALL FOR A MODEL	00005220
C = 0 SUBSEQUENT CALL WHERE PREVIOUS	00005230
C RESULTS MAY BE USED	00005240
C	00005250
C	00005260
C FINQDF = FINQ(R1)-FINQ(R2) - (FINQ(R3)-FINQ(R4))	00005270
C	00005280
C--CALLS FINQ	00005290
COMPLEX FINQ,Q1,Q2,Q3,Q4,ZERO,QOLD(4)	00005300
REAL ROLD(4)	00005310
INTEGER ISET(4)	00005320
COMMON/SAVEQ/ROLD,QOLD	00005330
DATA ZERO/(0.0,0.0)/	00005340
FINQDF=ZERO	00005350
Q1=ZERO	00005360
Q2=ZERO	00005370
Q3=ZERO	00005380
Q4=ZERO	00005390
DO 1 I=1,4	00005400
1 ISET(I)=0	00005410
IF(NEW.EQ.0) GO TO 10	00005420
DO 2 I=1,4	00005430
ROLD(I)=0E0	00005440
2 QOLD(I)=ZERO	00005450
GO TO 100	00005460
C CHECK TO SEE IF Q VALUE WAS CALCULATED	00005470
C IN PREVIOUS CALL TO FINQDF	00005480
10 DO 20 I=1,4	00005490
IF(ROLD(I).NE.R1) GO TO 11	00005500
Q1=QOLD(I)	00005510
ISET(1)=I+1	00005520
GO TO 20	00005530
11 IF(ROLD(I).NE.R2) GO TO 12	00005540
Q2=QOLD(I)	00005550
ISET(2)=I+1	00005560
GO TO 20	00005570
12 IF(ROLD(I).NE.R3) GO TO 13	00005580
Q3=QOLD(I)	00005590
ISET(3)=I+1	00005600
GO TO 20	00005610
13 IF(ROLD(I).NE.R4) GO TO 20	00005620
Q4=QOLD(I)	00005630

	ISET(4)=I+1	00005640
20	CONTINUE	00005650
100	IF(R1.EQ.R2) GO TO 150	00005660
	IF(R1.EQ.R3) GO TO 400	00005670
101	IF(ISET(1).GT.0) GO TO 110	00005680
	ISET(1)=1	00005690
	Q1=FINQ(DEL,R1,TOL)	00005700
110	IF(ISET(2).GT.0) GO TO 120	00005710
	ISET(2)=1	00005720
	Q2=FINQ(DEL,R2,TOL)	00005730
120	FINQDF=Q1-Q2	00005740
150	IF(R3.EQ.R4) GO TO 300	00005750
	IF(ISET(3).GT.0) GO TO 250	00005760
	IF(R3.NE.R1.OR.ISET(1).EQ.0) GO TO 155	00005770
	Q3=Q1	00005780
	GO TO 250	00005790
155	IF(R3.NE.R2.OR.ISET(2).EQ.0) GO TO 160	00005800
	Q3=Q2	00005810
	GO TO 250	00005820
160	Q3=FINQ(DEL,R3,TOL)	00005830
	ISET(3)=1	00005840
250	IF(ISET(4).GT.0) GO TO 290	00005850
	IF(R4.NE.R1.OR.ISET(1).EQ.0) GO TO 255	00005860
	Q4=Q1	00005870
	GO TO 290	00005880
255	IF(R4.NE.R2.OR.ISET(2).EQ.0) GO TO 260	00005890
	Q4=Q2	00005900
	GO TO 290	00005910
260	Q4=FINQ(DEL,R3,TOL)	00005920
	ISET(4)=1	00005930
290	FINQDF=FINQDF-(Q3-Q4)	00005940
300	GO TO 500	00005950
400	IF(R2.EQ.R4) GO TO 500	00005960
	IF(ISET(2).GT.0) GO TO 410	00005970
	ISET(2)=1	00005980
	Q2=FINQ(DEL,R2,TOL)	00005990
410	IF(ISET(4).GT.0) GO TO 420	00006000
	ISET(4)=1	00006010
	Q4=FINQ(DEL,R4,TOL)	00006020
420	FINQDF=Q4-Q2	00006030
C		00006040
C	SAVE CALCULATED (ISET(I)=1) VALUES	00006050
C		00006060
500	IF(ISET(1).NE.1) GO TO 510	00006070
	ROLD(1)=R1	00006080
	QOLD(1)=Q1	00006090
510	IF(ISET(2).NE.1) GO TO 520	00006100
	ROLD(2)=R2	00006110
	QOLD(2)=Q2	00006120
520	IF(ISET(3).NE.1) GO TO 530	00006130
	ROLD(3)=R3	00006140
	QOLD(3)=Q3	00006150

530 IF(ISET(4).NE.1) GO TO 540	00006160
ROLD(4)=R4	00006170
QOLD(4)=Q4	00006180
540 RETURN	00006190
END	00006200
COMPLEX FUNCTION FUNINT(X)	00006210
C--COMPLEX FUNCTION INTERPOLATION BY QUINTIC SPLINE VIA	00006220
C CALL TO 'QPOINT', WHERE THE QUINTIC SPLINE	00006230
C COEFFICIENTS AR,BR,CR,DR,ER, AI,BI,CI,DI,EI WERE	00006240
C PREVIOUSLY OBTAINED BY SUBR 'QUINT'.	00006250
C	00006260
DIMENSION SR(80),AR(80),BR(80),CR(80),DR(80),ER(80),	00006270
& SI(80),AI(80),BI(80),CI(80),DI(80),EI(80)	00006280
COMMON/SPLN80/SR,AR,BR,CR,DR,ER,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,NL	00006290
COMMON/FIN/R1,R2,R0,XL,SIG1,XX,Y	00006300
R=ALOG(SQRT(X*X+Y*Y))	00006310
CALL QPOINT(NL,SR,AR,BR,CR,DR,ER,RLM1,DELRLM,R,YR)	00006320
CALL QPOINT(NL,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,R,YI)	00006330
FUNINT=CMPLX(YR,YI)	00006340
RETURN	00006350
END	00006360
COMPLEX FUNCTION INFNEX(B)	00006370
C--INFNEX COMPUTES THE ELECTRIC FIELD PARALLEL TO AN	00006380
C INFINITELY LONG WIRE SOURCE AT THE EARTH'S SURFACE	00006390
C--CALLS ZFOURO(SINE TRANSFORM) AND F2MOD	00006400
EXTERNAL F2MOD	00006410
COMPLEX ZFOURO	00006420
COMMON/PARM/IS,X,Y,R,SIG1,BNYQ,M,TOL	00006430
SB=SQRT(B)	00006440
DEL=R/SB	00006450
DEL2=DEL*DEL	00006460
INFNEX=CMPLX(0.0,0.0)	00006470
INFNEX=ZFOURO(ALOG(SB),F2MOD,TOL,LW)*CMPLX(0.0,-2./(SB*DEL2))	00006480
INFNEX=INFNEX/(3.1415927*SIG1)	00006490
RETURN	00006500
END	00006510
SUBROUTINE QPOINT(NY,Y,B,C,D,E,F,X1,DELX,XX,YY)	00006520
C GIVEN THE QUINTIC SPLINE COEFF'S B(*),C(*),D(*),E(*),F(*) AS	00006530
C OBTAINED FROM SUBR 'QUINT', AND GIVEN NY OBS. DATA Y(NY) EQUALLY	00006540
C SPACED BY DELX STARTING AT X1, THEN 'QPOINT' INTERPOLATES	00006550
C YY AT ANY XX IN (X1,X1+(NY-1)*DELX).	00006560
C	00006570
DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)	00006580
XMAX=X1+(NY-1)*DELX	00006590
IF(XX.LT.X1.OR.XX.GT.XMAX) GO TO 2	00006600
I=(XX-X1)/DELX+1	00006610
XI=X1+(I-1)*DELX	00006620
T=(XX-XI)/DELX	00006630
YY=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I)	00006640

1	RETURN	00006650
2	WRITE(6,3) XX,X1,XMAX	00006660
3	FORMAT('OQPOINT ERROR-- XX=',E16.8,' NOT IN CLOSED INTERVAL ('	00006670
	& E16.8,',',E16.8,')')	00006680
	GO TO 1	00006690
	END	00006700
	SUBROUTINE QUINT(NY,Y,B,C,D,E,F)	00006710
C--	COMPUTES COEFFICIENTS OF A QUINTIC NATURAL SPLINE S(X) GIVEN	00006720
C	THE ORDINATES Y(I) AT ASSUMED EQUIDISTANT POINTS X(I),I=1 TO NY.	00006730
C		00006740
C	TRANSLATED FROM ALGOL TO FORTRAN BY	00006750
C	W.L. ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.	00006760
C	REF: ACM TRANSACTIONS ON MATH. SOFTWARE, SEPT 1976, V.2, N. 3,	00006770
C	PP.281-289.	00006780
C		00006790
C	PARAMETERS:	00006800
C		00006810
C	NY = NUMBER OF DATA POINTS GIVEN IN Y(NY), NY.GT.2.	00006820
C	Y()= ARRAY OF NY GIVEN ORDINATES (DIM.GE.NY).	00006830
C	Y() POINTS ASSUMED EQUALLY SPACED IN X-DIRECTION.	00006840
C	B,C,D,E,F() = RESULTING ARRAYS (EACH DIM.GE.NY) OF	00006850
C	QUINTIC SPLINE COEFFICIENTS, WHERE	00006860
C	FOR ANY XX IN [X(I),X(I+1)):	00006870
C	S(XX)=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I) WITH	00006880
C	T=(XX-X(I))/DELX, DELX=(X(I+1)-X(I)) FOR ANY I.	00006890
C	NOTE: SEE PROC 'QPOINT' TO EVAL THE QUINTIC SPLINE AFTER.	00006900
C	'QUINT' IS CALLED.	00006910
C		00006920
	DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)	00006930
	IF(NY.LE.2) GO TO 4	00006940
	N=NY-3	00006950
	P=0.0	00006960
	Q=0.0	00006970
	R=0.0	00006980
	S=0.0	00006990
	T=0.0	00007000
	DO 1 I=1,N	00007010
	U=P*R	00007020
	B(I)=1.0/(66.0-U*R-Q)	00007030
	R=26.0-U	00007040
	C(I)=R	00007050
	D(I)=Y(I+3)-3.0*(Y(I+2)-Y(I+1))-Y(I)-U*S-Q*T	00007060
	Q=P	00007070
	P=B(I)	00007080
	T=S	00007090
	S=D(I)	00007100
1.	CONTINUE	00007110
	D(N+2)=0.0	00007120
	N1=N+1	00007130
	D(N1)=0.0	00007140
	DO 2 J=1,N	00007150

	I=N1-J	00007160
	D(I)=(D(I)-C(I)*D(I+1)-D(I+2))*B(I)	00007170
2	CONTINUE	00007180
	N=NY-1	00007190
	Q=0.0	00007200
	V=D(1)	00007210
	T=V	00007220
	R=V	00007230.
	DO 3 I=2,N	00007240
	P=Q	00007250
	Q=R	00007260
	R=D(I)	00007270
	S=T	00007280
	T=P-Q-Q+R	00007290
	F(I)=T	00007300
	U=5.0*(-P+Q)	00007310
	E(I)=U	00007320
	D(I)=10.0*(P+Q)	00007330
	C(I)=0.5*(Y(I+1)+Y(I-1)+S-T)-Y(I)-U	00007340
	B(I)=0.5*(Y(I+1)-Y(I-1)-S-T)-D(I)	00007350
3	CONTINUE	00007360
	F(1)=V	00007370
	E(1)=0.0	00007380
	E(NY)=0.0	00007390
	D(1)=0.0	00007400
	D(NY)=0.0	00007410
	C(1)=C(2)-10.0*V	00007420
	C(NY)=C(NY-1)+10.0*T	00007430
	B(1)=Y(2)-Y(1)-C(1)-V	00007440
	B(NY)=Y(NY)-Y(NY-1)+C(NY)-T	00007450
4	RETURN	00007460
	END	00007470
	SUBROUTINE RECUR2(G,V1,F1,L1)	00007480
C	RECUR2 RECURSIVELY COMPUTES THE ELEMENTS WHICH ARE USED	00007490
C	IN THE LAYERED-EARTH HANKEL TRANSFORM KERNELS	00007500
C	--ORIGINAL ALGORITHM FROM ANDERSON(1974) HAS BEEN	00007510
C	MODIFIED TO USE COMPLEX CONDUCTIVITIES IF ICMLPX=1	00007520
C		00007530
	COMMON/MODEL/K,D,M	00007540
	COMMON/CFLAG/CK,ICMLPX	00007550
	REAL K(10),D(9)	00007560
	COMPLEX C,VM,V1,F1,L1,E,ONE,CK(10)	00007570
	DATA ONE/(1.0,0.0)/	00007580
	F1=ONE	00007590
	L1=ONE	00007600
	G2=G*G	00007610
	IF(ICMLPX.EQ.0) VM=CSQRT(CMPLX(G2,2.*K(M)))	00007620
	IF(ICMLPX.EQ.1) VM=CSQRT(CMPLX(G2-2.*AIMAG(CK(M)),2.*REAL(CK(M))))	00007630
	IF(M.EQ.1) GO TO 2	00007640
	J=M-1	00007650
1	IF(ICMLPX.EQ.0) V1=CSQRT(CMPLX(G2,2.*K(J)))	00007660


```

IF(ICMPLX.EQ.1) V1=CSQRT(CMPLX(G2-2.*AIMAG(CK(J)),2.*REAL(CK(J))))00007670
E=CEXP(-V1*D(J))00007680
C=(ONE-E)/(ONE+E)00007690
F1=(VM*F1+V1*C)/(V1+VM*F1*C)00007700
IF(ICMPLX.EQ.0) E=K(J+1)*V1+K(J)*VM*L1*C00007710
IF(ICMPLX.EQ.1) E=CK(J+1)*V1+CK(J)*VM*L1*C00007720
IF(REAL(E).EQ.0.0.AND.AIMAG(E).EQ.0.0) E=CMPLX(1.E-30,1.E-30)00007730
IF(ICMPLX.EQ.0) L1=(K(J)*VM*L1+K(J+1)*V1*C)/E00007740
IF(ICMPLX.EQ.1) L1=(CK(J)*VM*L1+CK(J+1)*V1*C)/E00007750
IF(J.EQ.1) GO TO 300007760
J=J-100007770
VM=V100007780
GO TO 100007790
2 V1=VM00007800
3 RETURN00007810
END00007820

SUBROUTINE RECURS(G,V1,F1)00007830
C RECURS RECURSIVELY COMPUTES THE ELEMENTS WHICH ARE USED00007840
C IN THE LAYERED-EARTH HANKEL TRANSFORM KERNELS00007850
C--ORIGINAL ALGORITHM FROM ANDERSON(1974) HAS BEEN00007860
C--MODIFIED TO USE COMPLEX CONDUCTIVITIES IF ICMPLX=100007870
C00007880
COMPLEX C,VM,V1,F1,EVD,ONE,T,CK(10)00007890
REAL K(10),D(9)00007900
COMMON/MODEL/K,D,M00007910
COMMON/CFLAG/CK,ICMPLX00007920
DATA ONE/(1.0,0.0)/00007930
F1=ONE00007940
G2=G*G00007950
IF(ICMPLX.EQ.0) VM=CSQRT(CMPLX(G2,2.0*K(M)))00007960
IF(ICMPLX.EQ.1) VM=CSQRT(CMPLX(G2-2.*AIMAG(CK(M)),2.*REAL(CK(M))))00007970
IF(M.EQ.1) GO TO 300007980
J=M-100007990
10 IF(ICMPLX.EQ.0) V1=CSQRT(CMPLX(G2,2.0*K(J)))00008000
IF(ICMPLX.EQ.1) V1=CSQRT(CMPLX(G2-2.*AIMAG(CK(J)),2.*REAL(CK(J))))00008010
EVD=-V1*D(J)00008020
EVD=CEXP(EVD)00008030
20 C=(ONE-EVD)/(ONE+EVD)00008040
T=VM*F100008050
F1=(T+V1*C)/(V1+T*C)00008060
IF(J.EQ.1) GO TO 400008070
J=J-100008080
VM=V100008090
GO TO 100008100
30 V1=VM00008110
40 RETURN00008120
END00008130

REAL FUNCTION RLAGF1(X,FUN,TOL,L,NEW)00008140
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE00008150
C INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*SIN(G*B)*DG' DEFINED AS THE00008160

```

```

C  REAL FOURIER SINE TRANSFORM WITH ARGUMENT X(=ALOG(B))                00008170
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND              00008180
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....      00008190
C                                                                      00008200
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.          00008210
C                                                                      00008220
C--PARAMETERS:                                                         00008230
C                                                                      00008240
C  * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM 00008250
C             'RLAGF1' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,    00008260
C             SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,  00008270
C             THEN SUBPROGRAM 'RFOUR1' IS ADVISED FOR GENERAL USE.     00008280
C             (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).              00008290
C  FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).       00008300
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN    00008310
C             CALLING PROGRAM AND IN SUBPROGRAM FUN.                   00008320
C             THE REAL FUNCTION FUN SHOULD BE A MONOTONE               00008330
C             DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...    00008340
C  TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,       00008350
C             IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.    00008360
C             THIS IS DONE AT BOTH ENDS OF FILTER. TYPICALLY,          00008370
C             TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON          00008380
C             THE FUNCTION FUN AND PARAMETER X...IN GENERAL,            00008390
C             A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'    00008400
C             BUT WITH 'MORE WEIGHTS' BEING USED. TOL IS NOT DIRECTLY    00008410
C             RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN    00008420
C             APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,     00008430
C             ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...    00008440
C  L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE            00008450
C             CONVOLUTION (L DEPENDS ON TOL AND FUN).                   00008460
C             MIN.L=20 AND MAX.L=266--WHICH COULD                      00008470
C             OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING     00008480
C             VERY FAST...                                              00008490
C  * NEW=    1 IS NECESSARY 1ST TIME OR BRAND NEW X.                   00008500
C             0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20          00008510
C             IS ASSUMED INTERNALLY BY THIS ROUTINE.                   00008520
C             NOTE: IF THIS IS NOT TRUE, ROUTINE WILL                   00008530
C             STILL ASSUME X=(LAST X)-0.20 ANYWAY...                     00008540
C             IT IS THE USERS RESPONSIBILITY TO NORMALIZE               00008550
C             BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).    00008560
C             THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT         00008570
C             TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A               00008580
C             SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING     00008590
C             ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...             00008600
C             THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED                00008610
C             KERNELS WILL BE USED IN THE LAGGED CONVOLUTION             00008620
C             WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS        00008630
C             WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)                 00008640
C                                                                      00008650
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF1; THE FOURIER   00008660
C TRANSFORM IS THEN RLAGF1/B WHICH IS TO BE COMPUTED AFTER EXIT FROM    00008670
C THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...          00008680

```

```

C                                                    00008690
C--USAGE-- 'RLAGF1' IS CALLED AS FOLLOWS:          00008700
C    ...                                           00008710
C    EXTERNAL RF                                   00008720
C    ...                                           00008730
C    R=RLAGF1(ALOG(B),RF,TOL,L,NEW)/B             00008740
C    ...                                           00008750
C    END                                           00008760
C    REAL FUNCTION RF(G)                          00008770
C    ...USER SUPPLIED CODE...                     00008780
C    END                                           00008790
C                                                    00008800
C--NOTES:                                           00008810
C    (1). EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM 00008820
C    BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS 00008830
C    ANY & ALL EXP-UNDERFLOW'S TO 0.0....         00008840
C    (2). AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION 00008850
C    METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN      00008860
C    THAT THE ACTUAL NUMBER OF B'S IS NB=AIN(5.*ALOG(BMAX/BMIN))+1, 00008870
C    PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED' 00008880
C    BMINA=BMAX*EXP(-.2*(NB-1)). THE METHOD GENERATES THE DECREASING 00008890
C    ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA). 00008900
C    FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:           00008910
C    ...                                           00008920
C    NB=AIN(5.*ALOG(BMAX/BMIN))+1                 00008930
C    NB1=NB+1                                       00008940
C    X0=ALOG(BMAX)+.2                              00008950
C    NEW=1                                          00008960
C    DO 1 J=1,NB                                   00008970
C    I=NB1-J                                        00008980
C    X=X0-.2*J                                      00008990
C    ARG(I)=EXP(X)                                  00009000
C    ANS(I)=RLAGF1(X,RF,TOL,L,NEW)/ARG(I)          00009010
C    1 NEW=0                                        00009020
C    ...                                           00009030
C    (3). IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR 00009040
C    ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE, 00009050
C    TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)      00009060
C    SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.           00009070
C    (4). IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY      00009080
C    ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW         00009090
C    BMAX,BMIN AND BY SETTING NEW=1....           00009100
C    (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE 00009110
C                                                    00009120
C                                                    00009130
C    DIMENSION KEY(266),SAVE(266)                  00009140
C    DIMENSION WT(266),W1(76),W2(76),W3(76),W4(38) 00009150
C    EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),    00009160
C    1 (WT(229),W4(1))                                         00009170
C--SIN-EXTENDED FILTER WEIGHT ARRAYS:              00009180
C    DATA W1/                                              00009190
C    1-1.1113940E-09,-1.3237246E-12, 1.5091739E-12,-1.6240954E-12, 00009200

```

2	1.7236636E-12,-1.8227727E-12,	1.9255992E-12,-2.0335514E-12,	00009210
3	2.1473541E-12,-2.2675549E-12,	2.3946842E-12,-2.5292661E-12,	00009220
4	2.6718110E-12,-2.8227693E-12,	2.9825171E-12,-3.1514006E-12,	00009230
5	3.3297565E-12,-3.5179095E-12,	3.7163306E-12,-3.9256378E-12,	00009240
6	4.1464798E-12,-4.3794552E-12,	4.6252131E-12,-4.8845227E-12,	00009250
7	5.1582809E-12,-5.4474462E-12,	5.7530277E-12,-6.0760464E-12,	00009260
8	6.4175083E-12,-6.7783691E-12,	7.1595239E-12,-7.5618782E-12,	00009270
9	7.9864477E-12,-8.4344110E-12,	8.9072422E-12,-9.4067705E-12,	00009280
1	9.9349439E-12,-1.0493731E-11,	1.1084900E-11,-1.1709937E-11,	00009290
2	1.2370354E-11,-1.3067414E-11,	1.3802200E-11,-1.4575980E-11,	00009300
3	1.5390685E-11,-1.6249313E-11,	1.7155934E-11,-1.8115250E-11,	00009310
4	1.9131898E-11,-2.0209795E-11,	2.1352159E-11,-2.2561735E-11,	00009320
5	2.3840976E-11,-2.5192263E-11,	2.6618319E-11,-2.8122547E-11,	00009330
6	2.9709129E-11,-3.1382870E-11,	3.3149030E-11,-3.5013168E-11,	00009340
7	3.6981050E-11,-3.9058553E-11,	4.1251694E-11,-4.3566777E-11,	00009350
8	4.6010537E-11,-4.8590396E-11,	5.1314761E-11,-5.4193353E-11,	00009360
9	5.7236720E-11,-6.0455911E-11,	6.3861222E-11,-6.7461492E-11,	00009370
1	7.1265224E-11,-7.5279775E-11,	7.9512249E-11,-8.3971327E-11/	00009380
	DATA W2/		00009390
1	8.8668961E-11,-9.3621900E-11,	9.8851764E-11,-1.0438319E-10,	00009400
2	1.1024087E-10,-1.1644680E-10,	1.2301979E-10,-1.2997646E-10,	00009410
3	1.3733244E-10,-1.4510363E-10,	1.5330772E-10,-1.6196550E-10,	00009420
4	1.7110130E-10,-1.8074257E-10,	1.9091922E-10,-2.0166306E-10,	00009430
5	2.1300756E-10,-2.2498755E-10,	2.3763936E-10,-2.5100098E-10,	00009440
6	2.6511250E-10,-2.8001616E-10,	2.9575691E-10,-3.1238237E-10,	00009450
7	3.2994314E-10,-3.4849209E-10,	3.6808529E-10,-3.8878042E-10,	00009460
8	4.1063982E-10,-4.3372666E-10,	4.5811059E-10,-4.8386049E-10,	00009470
9	5.1105728E-10,-5.3977672E-10,	5.7011632E-10,-6.0215516E-10,	00009480
1	6.3601273E-10,-6.7175964E-10,	7.0955028E-10,-7.4942601E-10,	00009490
2	7.9161025E-10,-8.3606980E-10,	8.8317110E-10,-9.3270330E-10,	00009500
3	9.8533749E-10,-1.0404508E-09,	1.0993731E-09,-1.1605442E-09,	00009510
4	1.2267391E-09,-1.2942905E-09,	1.3691677E-09,-1.4429912E-09,	00009520
5	1.5288164E-09,-1.6077524E-09,	1.7085998E-09,-1.7890471E-09,	00009530
6	1.9129068E-09,-1.9857116E-09,	2.1491608E-09,-2.1926779E-09,	00009540
7	2.4312660E-09,-2.3959044E-09,	2.7872500E-09,-2.5610596E-09,	00009550
8	3.2762318E-09,-2.6082940E-09,	4.0261453E-09,-2.3560563E-09,	00009560
9	5.3176554E-09,-1.3960161E-09,	7.7708747E-09, 1.1853546E-09,	00009570
1	1.2760851E-08, 7.4264707E-09,	2.3342187E-08, 2.1869851E-08/	00009580
	DATA W3/		00009590
1	4.6306744E-08, 5.4631686E-08,	9.6763087E-08, 1.2823337E-07,	00009600
2	2.0832812E-07, 2.9280540E-07,	4.5580888E-07, 6.5992437E-07,	00009610
3	1.0056815E-06, 1.4779183E-06,	2.2284335E-06, 3.2994604E-06,	00009620
4	4.9485823E-06, 7.3545473E-06,	1.1001083E-05, 1.6380539E-05,	00009630
5	2.4469550E-05, 3.6469246E-05,	5.4441527E-05, 8.1176726E-05,	00009640
6	1.2113828E-04, 1.8066494E-04,	2.6954609E-04, 4.0202288E-04,	00009650
7	5.9969995E-04, 8.9437312E-04,	1.3338166E-03, 1.9886697E-03,	00009660
8	2.9643943E-03, 4.4168923E-03,	6.5773518E-03, 9.7855105E-03,	00009670
9	1.4539361E-02, 2.1558670E-02,	3.1871864E-02, 4.6903518E-02,	00009680
1	6.8559512E-02, 9.9170152E-02,	1.4120770E-01, 1.9610835E-01,	00009690
2	2.6192603E-01, 3.2743321E-01,	3.6407406E-01, 3.1257559E-01,	00009700
3	9.0460168E-02,-3.6051039E-01,-8.6324760E-01,-8.1178720E-01,		00009710
4	5.2205241E-01, 1.5449873E+00,-1.1817933E+00,-2.6759896E-01,		00009720

5	8.0869203E-01,-6.2757149E-01,	3.4062630E-01,-1.5885304E-01,	00009730
6	7.0472984E-02,-3.1624462E-02,	1.4894068E-02,-7.4821176E-03,	00009740
7	4.0035936E-03,-2.2543784E-03,	1.3160358E-03,-7.8636604E-04,	00009750
8	4.7658745E-04,-2.9125817E-04,	1.7885105E-04,-1.1012416E-04,	00009760
9	6.7910334E-05,-4.1914054E-05,	2.5881544E-05,-1.5985851E-05,	00009770
1	9.8751880E-06,-6.1008526E-06,	3.7692543E-06,-2.3287953E-06/	00009780
	DATA W4/		00009790
1	1.4388425E-06,-8.8899353E-07,	5.4926991E-07,-3.3937048E-07,	00009800
2	2.0968284E-07,-1.2955437E-07,	8.0046336E-08,-4.9457371E-08,	00009810
3	3.0557711E-08,-1.8880390E-08,	1.1665454E-08,-7.2076428E-09,	00009820
4	4.4533423E-09,-2.7515696E-09,	1.7001092E-09,-1.0504494E-09,	00009830
5	6.4904567E-10,-4.0102999E-10,	2.4778763E-10,-1.5310321E-10,	00009840
6	9.4600354E-11,-5.8453314E-11,	3.6119400E-11,-2.2320056E-11,	00009850
7	1.3793460E-11,-8.5242656E-12,	5.2675102E-12,-3.2543076E-12,	00009860
8	2.0097689E-12,-1.2405412E-12,	7.6530538E-13,-4.7191929E-13,	00009870
9	2.9084993E-13,-1.7923661E-13,	1.1018948E-13,-6.7885902E-14,	00009880
1	4.2025050E-14,-2.1314731E-14/		00009890
	C--\$ENDATA		00009900
	C		00009910
	IF(NEW) 10,30,10		00009920
10	LAG=-1		00009930
	X0=-X-38.30455704		00009940
	DO 20 IR=1,266		00009950
20	KEY(IR)=0		00009960
30	LAG=LAG+1		00009970
	RLAGF1=0.0		00009980
	CMAX=0.0		00009990
	L=0		00010000
	ASSIGN 110 TO M		00010010
	I=191		00010020
	GO TO 200		00010030
110	CMAX=AMAX1(ABS(C),CMAX)		00010040
	I=I+1		00010050
	IF(I.LE.208) GO TO 200		00010060
	IF(CMAX.EQ.0.0) GO TO 150		00010070
	CMAX=TOL*CMAX		00010080
	ASSIGN 120 TO M		00010090
	I=190		00010100
	GO TO 200		00010110
120	IF(ABS(C).LE.CMAX) GO TO 130		00010120
	I=I-1		00010130
	IF(I.GT.0) GO TO 200		00010140
130	ASSIGN 140 TO M		00010150
	I=209		00010160
	GO TO 200		00010170
140	IF(ABS(C).LE.CMAX) GO TO 190		00010180
	I=I+1		00010190
	IF(I.LE.266) GO TO 200		00010200
	GO TO 190		00010210
150	ASSIGN 160 TO M		00010220
	I=1		00010230
	GO TO 200		00010240

160	IF(C.EQ.0.0) GO TO 170	00010250
	I=I+1	00010260
	IF(I.LE.190) GO TO 200	00010270
170	ASSIGN 180 TO M	00010280
	I=266	00010290
	GO TO 200	00010300
180	IF(C.EQ.0.0) GO TO 190	00010310
	I=I-1	00010320
	IF(I.GE.209) GO TO 200	00010330
190	RETURN	00010340
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)		00010350
200	LOOK=I+LAG	00010360
	IQ=LOOK/267	00010370
	IR=MOD(LOOK,267)	00010380
	IF(IR.EQ.0) IR=1	00010390
	IROLL=IQ*266	00010400
	IF(KEY(IR).LE.IROLL) GO TO 220	00010410
210	C=SAVE(IR)*WT(I)	00010420
	RLAGF1=RLAGF1+C	00010430
	L=L+1	00010440
	GO TO M,(110,120,140,160,180)	00010450
220	KEY(IR)=IROLL+IR	00010460
	SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))	00010470
	GO TO 210	00010480
	END	00010490
C ***** PROGRAM SCHCOPL *****		00010500
C--PROGRAM SCHCOPL CALCULATES THE ELECTROMAGNETIC COUPLING		00010510
C BETWEEN TWO STRAIGHT PARALLEL WIRES WHICH ARE VERY		00010520
C CLOSE TOGETHER IN A SCHLUMBERGER OR WENNER TYPE ARRAY.		00010530
C THE SOURCE WIRE IS ASSUMED TO LIE ALONG THE X-AXIS		00010540
C CENTERED AT THE ORIGIN AND EXTENDED BETWEEN +-AB WHILE		00010550
C THE RECEIVER WIRE IS EXTENDED BETWEEN +-MN AND SEPARATED		00010560
C FROM THE SOURCE WIRE BY DISTANCE Y . THE INPUT PARAMETERS ARE:		00010570
C		00010580
C	SIG(I) CONDUCTIVITY OF THE ITH LAYER	00010590
C	D(I) THICKNESS OF THE ITH LAYER	00010600
C	M NUMBER OF LAYERS	00010610
C	AB ARRAY OF SOURCE HALFLENGTHS	00010620
C	MN ARRAY OF RECEIVER HALFLENGTHS	00010630
C	NSP NUMBER OF DIFFERENT VALUES IN	00010640
C	MN AND AB ARRAYS	00010650
C	Y SEPARATION BETWEEN WIRES	00010660
C	RMAX (DEFAULT=1000*(LARGEST AB USED)) IS UPPER LIMIT OF	00010670
C	DOUBLE INTEGRAL WHICH IS USED TO CORRECT INFINITE	00010680
C	LINE INTEGRAL	00010690
C	NF >0, NUMBER OF FREQUENCIES DESIRED PER DECADE	00010700
C	BETWEEN F0 AND FM	00010710
C	<0, NUMBER OF SPECIFIED FREQUENCIES IN FNF	00010720
C	F0,FM MINIMUM AND MAXIMUM FREQUENCIES DESIRED	00010730
C	FNF SPECIFIED FREQUENCIES	00010740
C	TOL TOLERANCE FOR HANKEL TRANSFORM CALCULATIONS	00010750

C	FINTL1	TOLERANCE FOR INTEGRATION ALONG SOURCE WIRE	00010760
C	FINTL2	TOLERANCE FOR INTEGRATION ALONG RECEIVER WIRE	00010770
C	IN1,IN2	=1 FOR ADAPTIVE QUADRATURE INTEGRATION	00010780
C		=2 FOR NON-ADAPTIVE QUADRATURE INTEGRATION	00010790
C		IN1 IS FOR THE OUTER INTEGRAL ACROSS THE RECEIVER	00010800
C		IN2 IS FOR THE INNER INTEGRAL ACROSS THE SOURCE	00010810
C	NFIN	INTERVAL IN LOG-SPACE WITH WHICH THE SPLINE	00010820
C		NODES FOR FINITE WIRE INTEGRATION ARE	00010830
C		CALCULATED, E. G. INTERVAL=0.2/NFIN	00010840
C		(DEFAULT=1)	00010850
C	MEV1,MEV2	MAXIMUM NUMBER OF FUNCTION EVALUATIONS FOR	00010860
C		RESPECTIVE INTEGRATION ROUTINES	00010870
C	TMAX,TMIN	MAXIMUM AND MINIMUM TIME VALUES DESIRED	00010880
C	TFLAG	=0 COMPUTES FREQUENCY RESPONSE ALONE	00010890
C		=1 COMPUTES FREQUENCY AND TRANSIENT RESPONSE	00010900
C		=2 COMPUTES TRANSIENT RESPONSE ALONE	00010910
C		=3 COMPUTES TRANSIENT RESPONSE FOR A	00010920
C		FREQUENCY RESPONSE PREVIOUSLY COMPUTED	00010930
C	RC	TIME CONSTANT OF SINGLE POLE LOW-PASS	00010940
C		FILTER TO BE CONVOLVED WITH FREQUENCY	00010950
C		RESPONSE FOR TRANSIENT RESPONSE	00010960
C		CALCULATIONS	00010970
C		NOTE: TRANSIENT RESPONSE IS THE STEP	00010980
C		RESPONSE	00010990
C	ICMPLX	=0 COMPUTES THE COUPLING USING THE REAL	00011000
C		CONDUCTIVITIES IN SIG ARRAY,	00011010
C		=1 COMPUTES THE COUPLING USING THE COMPLEX	00011020
C		CONDUCTIVITIES COMPUTED BY THE USER-	00011030
C		DEFINED FUNCTION SIGMA(J,1./(DEL*DEL))	00011040
C	INFILE	(DEFAULT=5) INPUT FILE NUMBER	00011050
C	OUFIL	(DEFAULT=6) OUTPUT FILE NUMBER	00011060

C THE SUBPROGRAMS ARE ORGANIZED AS FOLLOWS:

C	SCHCOPL							00011080
C		*						00011090
C		*	INFINITE	DOUBLE		TRANSIENT		00011100
C		*	WIRE	INTEGRAL		RESPONSE		00011110
C	*****							00011120
C	*	*	*	*	*	*	*	00011130
C	SIGMA	SETSPL	INFNEX	ZSUB1	ZSUBA1	ZHANKO	SPLIN1	RLAGF1
C		*	*	*	*	*	*	00011140
C	*****		*	*****	*	*		00011150
C	*	*	*	*	*	*	*	00011160
C	ZEX	CZEX	QUINT	ZFOURO	CORFUN	F7G	CSPLNT	00011170
C	*	*	*	*	*	*		00011180
C	*****		*	*****	*	*		00011190
C	*	*	*	*	*	*		00011200
C	ZLAGHO	SIGMA	F2MOD	ZSUB2	ZSUBA2	RECUR2		00011210
C	*	*	*	*	*	*		00011220
C	F3		*	*****				00011230
C	*		*	*				00011240
C			*	*				00011250
C			*	*				00011260
C			*	*				00011270

C	RECURS	RECURS	FUNINT***QPOINT	00011280
C				00011290
	COMPLEX INFNEX,FINQ,ZHANK0,SCH(30,30),ECOPL,CSPEC,			00011300
	1 ZERR1,ZERR2,CORECT,ZSUB1,ZSUBA1,SIGMA,SIGMA1,CK(10)			00011310
	REAL MN(30),AB(30),K(10),D(9),DD(9),SIG(10)			00011320
	REAL P(50),S(50),PS(2),FNF(50)			00011330
	INTEGER OUFIL			00011340
	EXTERNAL F7G,CSPLNT,CORFUN,ZEX,CZEX			00011350
	COMMON/CFLAG/CK,ICMPLX			00011360
	COMMON/FINERR/TL,FINTL1,IN1,NFIN,NEV1,MEV1,ZERR1,LW			00011370
	COMMON/CORRF/YY,XMIN,RMAX			00011380
	COMMON/CSPLINE/D1(50),D2(50),D3(50),NN,FREQ(50),SPEC(50)			00011390
	COMMON/MODEL/K,DD,M			00011400
	COMMON/THICK/D			00011410
	COMMON/PARM/IS,X,Y,R,SIG1,BNYQ,NLYR,TOL			00011420
	NAMLIST/PARMS/SIG,M,D,MN,AB,FM,NF,F0,NSP,Y,TMAX,TMIN,TFLAG,RC			00011430
	1,TOL,FINTL1,FINTL2,IN1,IN2,NFIN,MEV1,MEV2,RMAX,ICMPLX,FNF,			00011440
	2 INFILE,OUFIL			00011450
	DATA DEG/57.29577951/			00011460
	DATA PS,TWOPI/0.0,0.0,6.283185308/			00011470
C				00011480
C	---DEFAULT AB/2 AND MN/2 VALUES			00011490
C				00011500
	DATA AB/10,13,16,20,20,25,30,40,50,65,80,100,100,130,160,200,200,			00011510
	1 250,300,400,500,650,800,1000,1000,1300,1600,2000,2000,2500/			00011520
	DATA MN/2,2,2,2,4,4,4,4,4,4,4,4,20,20,20,20,40,40,40,40,40,			00011530
	1 40,40,200,200,200,200,400,400/			00011540
C				00011550
C	---ASSIGN NAMLIST DEFAULT PARAMETERS			00011560
C				00011570
	INFILE=5			00011580
	OUFIL=6			00011590
	X=0			00011600
	IS=0			00011610
	Y=.01			00011620
	BNYQ=1.E30			00011630
	TOL=1.E-6			00011640
	M=1			00011650
	TFLAG=0			00011660
	FINTL1=1.E-3			00011670
	FINTL2=1.E-5			00011680
	MEV1=300			00011690
	MEV2=300			00011700
	TOL=1.E-6			00011710
	IN1=1			00011720
	IN2=1			00011730
	NFIN=1			00011740
	RC=0.0			00011750
	ICMPLX=0			00011760
	RMAX=0.0			00011770
C				00011780
	WRITE(OUFIL,501)			00011790

501	FORMAT(25H ENTER \$PARMS PARAMETERS\$)	00011800
100	READ(INFILE,PARMS)	00011810
	IF(M.EQ.0) STOP	00011820
	IF(TFLAG.EQ.3) GO TO 20	00011830
C		00011840
C--	TFLAG=3 ASSUMES THAT THE FREQUENCY FUNCTION TO BE TRANSFORMED	00011850
C	HAS ALREADY BEEN COMPUTED AND RESIDES IN THE SCH ARRAY	00011860
C		00011870
C--	DEFINE EQUIVALENT COMMON PARAMETERS	00011880
	NLYR=M	00011890
	R=Y	00011900
	YY=Y	00011910
C		00011920
	SIG1=SIG(1)	00011930
	IF(ICMPLX.EQ.1) GO TO 2	00011940
	DO 10 I=1,M	00011950
10	K(I)=SIG(I)/SIG1	00011960
	GO TO 3	00011970
2	SIG1=REAL(SIGMA(1,0.0))	00011980
	K(1)=SIG1	00011990
3	Y2=Y*Y	00012000
	M1=M-1	00012010
C		00012020
C--	PRINT MODEL PARAMETERS	00012030
C		00012040
	WRITE(OUFILE,503) M	00012050
503	FORMAT(/5X,I3,12H LAYER MODEL)	00012060
	IF(ICMPLX.EQ.0) WRITE(OUFILE,504) (SIG(I),I=1,M)	00012070
504	FORMAT(5H SIG=,12E12.4)	00012080
505	FORMAT(5H D =,12E12.4)	00012090
	IF(ICMPLX.EQ.1) WRITE(OUFILE,506)	00012100
506	FORMAT(28H COMPLEX CONDUCTIVITIES USED)	00012110
	IF(M.GT.1) WRITE(OUFILE,505) (D(I),I=1,M1)	00012120
C		00012130
C--	COMPUTE MAXIMUM AND MINIMUM DISTANCE BETWEEN WIRES	00012140
C	IN ORDER TO DEFINE ARGUMENT RANGE FOR ZEX OR CZEX	00012150
C	EVALUATIONS	00012160
C		00012170
	RMIN=AB(1)-MN(1)	00012180
	IF(RMAX.EQ.0.0) RMAX=100.*AB(NSP)	00012190
	CON=1./(SIG1*6.283185308)	00012200
	XB=R*R*SIG1*39.47841763E-7	00012210
	IF(NF.LT.0) GO TO 41	00012220
	NN=NF*ALOG10(FM/F0)+1	00012230
	DX=EXP(2.30258509/FLOAT(NF))	00012240
	B=XB*F0	00012250
	FREQ(1)=F0	00012260
	GO TO 42	00012270
41	NN=-NF	00012280
C		00012290
C--	FIRST LOOP OVER NN FREQUENCIES	00012300
C		00012310

42	DO 7 JJ=1,NN	00012320
	IF(NF.LT.0) GO TO 43	00012330
	IF(JJ.GT.1) B=B*DX	00012340
	IF(JJ.GT.1) FREQ(JJ)=FREQ(JJ-1)*DX	00012350
	GO TO 44	00012360
43	B=XB*FNF(JJ)	00012370
	FREQ(JJ)=FNF(JJ)	00012380
44	SB=SQRT(B)	00012390
	DEL=R/SB	00012400
	CF=1./(DEL*DEL)	00012410
	IF(ICMPLX.EQ.1) SIGMA1=SIGMA(1,CF)/SIG1	00012420
	DO 8 I=1,M	00012430
	IF(ICMPLX.EQ.1) CK(I)=SIGMA(I,CF)/SIG1	00012440
8	DD(I)=2.*D(I)/DEL	00012450
C--	COMPUTE THE WIRE COUPLING	00012460
C	ASSUMING INFINITELY-LONG SOURCE WIRE	00012470
	ECOPL=INFNEX(B)	00012480
C		00012490
C--	COMPUTE SPLINE COEFFICIENTS	00012500
C		00012510
	IF(ICMPLX.EQ.0) CALL SETSPL(ZEX,DEL,RMAX,RMIN)	00012520
	IF(ICMPLX.EQ.1) CALL SETSPL(CZEX,DEL,RMAX,RMIN)	00012530
C		00012540
C--	NEXT LOOP OVER NSP SPACINGS	00012550
C		00012560
	DO 7 ISP=1,NSP	00012570
	R1=SQRT((AB(ISP)-MN(ISP))*2+Y2)	00012580
	R2=SQRT((AB(ISP)+MN(ISP))*2+Y2)	00012590
	FINQ=CMPLX(0.0,0.0)	00012600
	IF(M.EQ.1) GO TO 5	00012610
	B1=R1/DEL	00012620
	B2=R2/DEL	00012630
	FINQ=ZHANK0(ALOG(B1),F7G,TOL,LW)/B1	00012640
	FINQ=FINQ-ZHANK0(ALOG(B2),F7G,TOL,LW)/B2	00012650
	FINQ=FINQ*CMPLX(0.0,1./DEL)	00012660
5	FINQ=FINQ-CMPLX((1./R1-1./R2),0.0)	00012670
	XMIN=AB(ISP)	00012680
C		00012690
C--	COMPUTE DOUBLE INTEGRAL CORRECTION TERM	00012700
C		00012710
	IF(IN1.EQ.1) CORECT=2.*ZSUBA1(-MN(ISP),MN(ISP),FINTL1,	00012720
1	NEV1,ICK1,ZERR1,CORFUN,MEV1)	00012730
	IF(IN1.EQ.2) CORECT=2.*ZSUB1(-MN(ISP),MN(ISP),FINTL1,	00012740
1	NEV1,ICK1,ZERR1,CORFUN,MEV1)	00012750
C		00012760
	SCH(JJ,ISP)=2.*MN(ISP)*ECOPL+CON*CORECT+2.*CON*FINQ	00012770
7	IF(ICMPLX.EQ.1) SCH(JJ,ISP)=SCH(JJ,ISP)/SIGMA1	00012780
	IF(TFLAG.EQ.0) GO TO 21	00012790
20	NT=AIN(5.*ALOG(TMAX/TMIN))+1	00012800
	NT1=NT+1	00012810
C--	CALCULATED IN NORMALIZED TIME	00012820
C	NORM TIME = TWOPI * REAL TIME	00012830

	XO=ALOG(TMAX*TWOPI)+0.2	00012840
C		00012850
21	DO 40 IPOS=1,NSP	00012860
	WRITE(OUFILE,507) AB(IPOS),MN(IPOS)	00012870
507	FORMAT(/3X,6HAB/2 =,E12.4/3X,6HMN/2 =,E12.4/)	00012880
	IF(TFLAG.GT.1) GO TO 35	00012890
	WRITE(OUFILE,200)	00012900
200	FORMAT(4X,9HFREQUENCY,5X,4HREAL,8X,4HIMAG,8X,4HMAGN,8X,3HPFE,	00012910
1	8X,5HPHASE)	00012920
	PFE0=CABS(SCH(1,IPOS))	00012930
	DO 30 J=1,NN	00012940
	AMP=CABS(SCH(J,IPOS))	00012950
	PFE=100.*(1.-AMP/PFE0)	00012960
	PHZ=DEG*ATAN2(AIMAG(SCH(J,IPOS)),REAL(SCH(J,IPOS)))	00012970
30	WRITE(OUFILE,509) FREQ(J),SCH(J,IPOS),AMP,PFE,PHZ	00012980
509	FORMAT(1H ,6E12.4)	00012990
	IF(TFLAG.EQ.0) GO TO 40	00013000
C		00013010
C	--COMPUTE THE STEP TRANSIENT RESPONSE USING A CUBICALLY	00013020
C	SPLINED FREQUENCY RESPONSE FUNCTION	00013030
C		00013040
35	DC=REAL(SCH(1,IPOS))	00013050
	DO 50 II=1,NN	00013060
C		00013070
C	--MULTIPLY THE FREQUENCY RESPONSE (DC VALUE SUBTRACTED)	00013080
C	BY THE SOURCE FUNCTION (1/FREQ FOR STEP FUNCTION)	00013090
C		00013100
	SPEC(II)=(REAL(SCH(II,IPOS))-DC)/FREQ(II)	00013110
	IF(RC.EQ.0.0) GO TO 50	00013120
C		00013130
C	--MULTIPLY THE FREQUENCY RESPONSE BY LOW-PASS	00013140
C	FILTER TRANSFER FUNCTION IF RC GT 0	00013150
C		00013160
	TCON=FREQ(II)*RC	00013170
	CSPEC=SCH(II,IPOS)*CMPLX(1.,-TCON)	00013180
	SPEC(II)=(REAL(CSPEC)-DC)/FREQ(II)/(1.+TCON*TCON)	00013190
50	CONTINUE	00013200
	CALL SPLIN1(NN,0,FREQ,SPEC,D1,D2,D3,0,PS,P,S)	00013210
	NEW=1	00013220
	WRITE(OUFILE,510)	00013230
510	FORMAT(/6X,9HTIME(SEC),4X,15HOBS VOLTAGE/AMP)	00013240
	DO 60 J=1,NT	00013250
	I=NT1-J	00013260
	X=X0-0.2*J	00013270
	T=EXP(X)	00013280
	FDC=1.	00013290
	IF(RC.GT.0.0) FDC=1.-EXP(-T/(RC*TWOPI))	00013300
C		00013310
C	--COMPUTE SINE TRANSFORM (RLAGF1) OF CUBICALLY SPLINED	00013320
C	FREQUENCY FUNCTION (SPLINE INTERPOLATOR CSPLNT)	00013330
C		00013340
	V=0.636619772*RLAGF1(X,CSPLNT,TOL,LW,NEW)/T+DC*FDC	00013350

```

        T=T/TWOPI                                00013360
        WRITE(OUFILE,509) T,V                    00013370
60      NEW=0                                      00013380
40      WRITE(OUFILE,512)                        00013390
512    FORMAT(1H )                              00013400
        WRITE(OUFILE,513)                        00013410
513    FORMAT(27H ENTER $PARMS CHANGES ONLY$)   00013420
        GO TO 100                                00013430
        END                                       00013440

        SUBROUTINE SETSPL(FUNC,DEL,RMAX,RMIN)      00013450
C--COMPUTE THE QUINTIC SPLINE COEFFICIENTS TO    00013460
C REPRESENT FUNC(R) FOR THE RANGE RMAX TO RMIN.  00013470
C 'SETSPL' CALLS 'FUNC' (WHICH CALLS 'ZLAGH1 OR ZLAGH0') AND 'QUINT'. 00013480
C                                                00013490
C PARAMETERS:                                    00013500
C                                                00013510
C FUNC = EXTERNAL DECLARED COMPLEX FUNCTION DEFINING THE DIPOLE FIELD 00013520
C FUNCTION WITH CALLING SEQ: FUNC(B,NEW,R), WHERE 00013530
C B = ANY IND. NO.                              00013540
C NEW = 1 FIRST TIME, 0 OTHERWISE (REF: ZLAGH1 OR ZLAGH0) 00013550
C R = B*DEL FOR ANY B OR DEL (SKIN DEPTH).        00013560
C DEL = SKIN DEPTH.                             00013570
C RMAX = MAXIMUM R AT WHICH FUNC WILL NEED TO BE EVALUATED 00013580
C RMIN = MINIMUM R AT WHICH FUNC WILL NEED TO BE EVALUATED 00013590
C                                                00013600
        COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW 00013610
        COMMON/SPLN80/FDR(80),AR(30),BR(80),CR(80),DR(80),ER(80), 00013620
        & FDI(80),AI(80),BI(80),CI(80),DI(80),EI(80),RLM1,DELRLM,NB 00013630
        COMMON/CONST/DELL,DEL2,Z2DEL3            00013640
        COMPLEX FUNC,ESUM,FD,Z2DEL3              00013650
C--ISIZE IS THE MAXIMUM POSSIBLE NUMBER OF NODES IN QUINTIC SPLINE 00013660
C AND ALSO IS THE DIMENSION OF ALL ARRAYS IN COMMON AREA SPLN80. 00013670
        DATA ISIZE/80/                          00013680
        DELL=DEL                                  00013690
        DEL2=DEL*DEL                              00013700
        Z2DEL3=CMPLX(0.0,2./(DEL2*DEL))          00013710
        BMAX=RMAX/DEL                             00013720
        BMIN=RMIN/DEL                             00013730
        NB=AIN(5.*ALOG(BMAX/BMIN))+2              00013740
        NB=MAX(NB,3)                              00013750
        X0=ALOG(BMIN)+NB*0.2                      00013760
        NB=NB+3                                    00013770
C--RANGE OF RMIN,RMAX EXTENDED BY AT LEAST 2 ON EACH 00013780
C END IN ORDER TO MAKE THE END CONDITIONS CHOSEN FOR 00013790
C THE SPLINE IRRELEVANT TO THE REAL RANGE OF INTEREST. 00013800
        NRMAX=ISIZE/NB                           00013810
        IF(NFIN.LE.NRMAX) GO TO 3                 00013820
        IF(NRMAX.GT.0.0) GO TO 2                  00013830
        WRITE(6,100)                              00013840
100    FORMAT(43H ERROR IN SETSPL: INSUFFICIENT SPLINE NODES) 00013850
        STOP                                       00013860

```

```

2 NFIN=NRMAX                                00013870
  WRITE(6,110) NFIN                          00013880
110 FORMAT(43H ERROR IN SETSPL: NFIN TOO LARGE, RESET TO ,I2) 00013890
3 DELRLM=.2/FLOAT(NFIN)                     00013900
  XO=XO-DELRLM                              00013910
  DO 5 ITIME=1,NFIN                         00013920
    NEW=1                                    00013930
    XO=XO+DELRLM                            00013940
    DO 5 J=1,NB                             00013950
      I=(NB+1)-J                            00013960
      I=NFIN*(I-1)+ITIME                    00013970
      XX=XO-0.2*J                           00013980
      BM=EXP(XX)                             00013990
      RM=BM*DEL                             00014000
      IF(I.EQ.1) RLM1=ALOG(RM)              00014010
      FD=FUNC(BM,NEW,RM)                   00014020
      FDR(I)=REAL(FD)                      00014030
      FDI(I)=AIMAG(FD)                     00014040
5    NEW=0                                   00014050
  NB=NFIN*NB                                00014060
  CALL QUINT(NB,FDR,AR,BR,CR,DR,ER)        00014070
  CALL QUINT(NB,FDI,AI,BI,CI,DI,EI)        00014080
10 RETURN                                  00014090
  END                                        00014100

  COMPLEX FUNCTION SIGMA(J,CF)              00014110
C--THIS IS AN EXAMPLE FUNCTION WHICH       00014120
C COMPUTES A COMPLEX CONDUCTIVITY.         00014130
C IT WAS USED FOR A HALFSPACE MODEL AND IGNORES THE 00014140
C J ARGUMENT.                              00014150
  COMMON/PARM/IS,A1,A2,A3,SIG1,A5,M,A7     00014160
  COMPLEX ONE                              00014170
  DATA ONE/(1.0,0.0)/                     00014180
  RM=0.3                                    00014190
  TAU=0.4                                   00014200
  IF(SIG1.EQ.0.0.AND.CF.EQ.0.0) SIG1=1.0  00014210
  OMEGA=CF/(SIG1*6.28318531E-7)            00014220
  SIGMA=CSQRT(CSQRT(CMPLX(0.0,OMEGA*TAU))) 00014230
  SIGMA=ONE/(ONE+SIGMA)                    00014240
  SIGMA=ONE-RM*(ONE-SIGMA)                 00014250
  SIGMA=0.1/SIGMA                          00014260
  RETURN                                   00014270
  END                                        00014280

  SUBROUTINE SPLINI(M,H,X,Y,A,B,C,IT,D,P,S) 00014290
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION. 00014300
C                                                    00014310
C      BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO 00014320
C                                                    00014330
C PARS--- M= NUMBER OF DATA POINTS .GT. 2      00014340
C      H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE), 00014350
C      UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE) 00014360

```

```

C      X= INDEP.VAR WHEN H=0. (DIM .GE. M). 00014370
C      Y= DEPENDENT VARIABLE (DIM .GE. M). 00014380
C      A,B,C=COEFF.ARRAYS (EACH DIM .GE. M) 00014390
C      RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C. 00014400
C      ALSO USED AS WORK ARRAYS DURING EXECUTION. 00014410
C      IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE 00014420
C      IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR 00014430
C      IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS. 00014440.
C      D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY. 00014450
C      P,S= WORK ARRAYS (EACH DIM=M). 00014460
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE. 00014470
C THE RESULTING CUBIC SPLINE IS OF THE FORM: 00014480
C      Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3 00014490
C      FOR I=1,2,...,M-1 00014500
C 00014510
C 00014520
C      REAL*4 X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL 00014530
C      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999 00014540
C      N=M-1 00014550
C      IF(IT.EQ.0) GO TO 20 00014560
C--1ST DERIVATIVE BOUNDARIES GIVEN 00014570
C      NE=N-1 00014580
C      IF(H) 999,11,1 00014590
C--EQUAL SPACING H .GT. 0. AND IT=1 00014600
C      1 HH=3.0/H 00014610
C      DO 2 I=1,NE 00014620
C      B(I)=4.0 00014630
C      C(I)=1.0 00014640
C      A(I)=1.0 00014650
C      2 P(I)=HH*(Y(I+2)-Y(I)) 00014660
C      P(1)=P(1)-D(1) 00014670
C      P(NE)=P(NE)-D(2) 00014680
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE 00014690
C      3 C(1)=C(1)/B(1) 00014700
C      P(1)=P(1)/B(1) 00014710
C      DO 4 I=2,NE 00014720
C      MUL=1.0/(B(I)-A(I)*C(I-1)) 00014730
C      C(I)=MUL*C(I) 00014740
C      4 P(I)=MUL*(P(I)-A(I)*P(I-1)) 00014750
C--OBTAIN SPLINE COEFFICIENTS 00014760
C      A(NE+IT)=P(NE) 00014770
C      I=NE-1 00014780
C      5 A(I+IT)=P(I)-C(I)*A(I+IT+1) 00014790
C      I=I-1 00014800
C      IF(I.GE.1) GO TO 5 00014810
C      IF(IT.EQ.0) GO TO 6 00014820
C      A(1)=D(1) 00014830
C      A(M)=D(2) 00014840
C      6 IF(H.EQ.0.) GO TO 14 00014850
C      HH=1.0/H 00014860
C      DO 7 I=1,N 00014870
C      MUL=HH*(Y(I+1)-Y(I)) 00014880

```

B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))	00014890
7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))	00014900
RETURN	00014910
C--UNEQUAL SPACING H=0.. AND IT=1	00014920
11 DO 12 I=1,N	00014930
12 S(I+1)=X(I+1)-X(I)	00014940
DO 13 I=1,NE	00014950
B(I)=2.0*(S(I+1)+S(I+2))	00014960
C(I)=S(I+1)	00014970
A(I)=S(I+2)	00014980
13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/	00014990
\$ (S(I+1)*S(I+2))	00015000
P(1)=P(1)-S(3)*D(1)	00015010
P(NE)=P(NE)-S(N)*D(2)	00015020
GO TO 3	00015030
14 DO 15 I=1,N	00015040
HH=1.0/S(I+1)	00015050
MUL=(Y(I+1)-Y(I))*HH**2	00015060
B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH	00015070
15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2	00015080
RETURN	00015090
C--2ND DERIVATIVE BOUNDARIES GIVEN	00015100
20 NE=N+1	00015110
IF(H) 999,31,21	00015120
C--EQUAL SPACING H .GT. 0 AND IT=0	00015130
21 HH=3.0/H	00015140
DO 22 I=2,N	00015150
B(I)=4.0	00015160
C(I)=1.0	00015170
A(I)=1.0	00015180
22 P(I)=HH*(Y(I+1)-Y(I-1))	00015190
B(1)=2.0	00015200
B(NE)=2.0	00015210
C(1)=1.0	00015220
C(NE)=1.0	00015230
A(NE)=1.0	00015240
P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)	00015250
P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)	00015260
GO TO 3	00015270
C--UNEQUAL SPACING H=0 AND IT=0	00015280
31 DO 32 I=1,N	00015290
32 S(I+1)=X(I+1)-X(I)	00015300
N1=N-1	00015310
DO 33 I=1,N1	00015320
B(I+1)=2.0*(S(I+1)+S(I+2))	00015330
C(I+1)=S(I+1)	00015340
A(I+1)=S(I+2)	00015350
33 P(I+1)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/	00015360
* (S(I+1)*S(I+2))	00015370
B(1)=2.0	00015380
B(NE)=2.0	00015390
C(1)=1.0	00015400

C(NE)=1.0	00015410
A(NE)=1.0	00015420
P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)	00015430
P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)	00015440
GO TO 3	00015450
999 M=-IABS(M)	00015460
RETURN	00015470
END	00015480
C--ZQUAD PACKAGE (ZBLOCK,ZQUAD1,ZSUB1,ZSUBA1,ZQUAD2,ZSUB2,ZSUBA2)	00015490
C FOR AUTOMATIC COMPLEX GAUSSIAN DOUBLE INTEGRATION OVER A	00015500
C FINITE INTERVAL.	00015510
C	00015520
C--MODIFIED BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO	00015530
C 12/30/75.	00015540
C	00015550
C--USAGE:	00015560
C	00015570
C USE 'ZSUB1' OR 'ZSUBA1' FOR 1ST COMPLEX INTEGRATION (CALLS ZQUAD1)	00015580
C AND 'ZSUB2' OR 'ZSUBA2' FOR 2ND COMPLEX INTEGRATION (CALLS ZQUAD2)	00015590
C	00015600
C--REFERENCES:	00015610
C	00015620
C (1) PATTERSON,T.N.L, 1973, ALGORITHM FOR AUTOMATIC	00015630
C NUMERICAL INTEGRATION OVER A FINITE INTERVAL [D1]	00015640
C ACM COMM. V.16, NO.11, P.694-699.	00015650
C (2) ANDERSON,W.L., 1974, ELECTROMAGNETIC FIELDS ABOUT A	00015660
C FINITE ELECTRIC WIRE SOURCE:	00015670
C N.T.I.S REPORT PB-238199, 209P.	00015680
C	00015690
C--NOTES:	00015700
C	00015710
C (A). SEE REF(1) FOR A COMPLETE DISCUSSION OF THE BASIC	00015720
C ALGORITHM(S) AS ORIGINALLY DEVELOPED FOR	00015730
C SINGLE REAL FUNCTION AUTOMATIC GAUSSIAN INTEGRATION.	00015740
C (B). SEE REF(2) FOR A MODIFIED VERSION FOR SINGLE COMPLEX	00015750
C FUNCTION AUTOMATIC GAUSSIAN INTEGRATION.	00015760
C (C). ALL CALLING PARMS USED BELOW IN THE ZQUAD PACKAGE ARE	00015770
C IDENTICAL TO THOSE USED IN REF(2). THEREFORE, SEE	00015780
C REF(2) FOR COMMENTS ON THESE ANALOGOUS ROUTINES.	00015790
C REF(1) MAY ALSO BE USED FOR DEFINITIONS OF MOST OF	00015800
C THE PARMS...	00015810
C	00015820
C-- MULTICS VERSION USES CALL ZBLOCK TO INITILIZE COMMON/ZQUADP	00015830
C FOR OTHER SYSTEMS, CHANGE SUBROUTINE ZBLOCK TO A	00015840
C BLOCK DATA SUBPROGRAM -- AND REMOVE THE ASSIGNMENTS STATEMENTS.	00015850
C	00015860
SUBROUTINE ZBLOCK	00015870
DIMENSION P(381)	00015880
COMMON/ZQUADP/Q(381)	00015890
DATA MULTICS/0/	00015900
DATA	00015910


```

* P( 1),P( 2),P( 3),P( 4),P( 5),P( 6),P( 7),      00015920
* P( 8),P( 9),P(10),P(11),P(12),P(13),P(14),      00015930
* P(15),P(16),P(17),P(18),P(19),P(20),P(21),      00015940
* P(22),P(23),P(24),P(25),P(26),P(27),P(28)/      00015950
* 0.77459666924148337704E 00,0.5555555555555555556E 00,      00015960
* 0.88888888888888888889E 00,0.26848808986833344073E 00,      00015970
* 0.96049126870802028342E 00,0.10465622602646726519E 00,      00015980
* 0.43424374934680255800E 00,0.40139741477596222291E 00,      00015990
* 0.45091653865847414235E 00,0.13441525524378422036E 00,      00016000
* 0.51603282997079739697E-01,0.20062852937698902103E 00,      00016010
* 0.99383196321275502221E 00,0.17001719629940260339E-01,      00016020
* 0.88845923287225699889E 00,0.92927195315124537686E-01,      00016030
* 0.62110294673722640294E 00,0.17151190913639138079E 00,      00016040
* 0.22338668642896688163E 00,0.21915685840158749640E 00,      00016050
* 0.22551049979820668739E 00,0.67207754295990703540E-01,      00016060
* 0.25807598096176653565E-01,0.10031427861179557877E 00,      00016070
* 0.84345657393211062463E-02,0.46462893261757986541E-01,      00016080
* 0.85755920049990351154E-01,0.10957842105592463824E 00/      00016090
DATA      00016100
* P(29),P(30),P(31),P(32),P(33),P(34),P(35),      00016110
* P(36),P(37),P(38),P(39),P(40),P(41),P(42),      00016120
* P(43),P(44),P(45),P(46),P(47),P(48),P(49),      00016130
* P(50),P(51),P(52),P(53),P(54),P(55),P(56)/      00016140
* 0.99909812496766759766E 00,0.25447807915618744154E-02,      00016150
* 0.98153114955374010687E 00,0.16446049854387810934E-01,      00016160
* 0.92965485742974005667E 00,0.35957103307129322097E-01,      00016170
* 0.83672593816886873550E 00,0.56979509494123357412E-01,      00016180
* 0.70249620649152707861E 00,0.76879620499003531043E-01,      00016190
* 0.53131974364437562397E 00,0.93627109981264473617E-01,      00016200
* 0.33113539325797683309E 00,0.10566989358023480974E 00,      00016210
* 0.11248894313318662575E 00,0.11195687302095345688E 00,      00016220
* 0.11275525672076869161E 00,0.33603877148207730542E-01,      00016230
* 0.12903800100351265626E-01,0.50157139305899537414E-01,      00016240
* 0.42176304415588548391E-02,0.23231446639910269443E-01,      00016250
* 0.42877960025007734493E-01,0.54789210527962865032E-01,      00016260
* 0.12651565562300680114E-02,0.82230079572359296693E-02,      00016270
* 0.17978551568128270333E-01,0.28489754745833548613E-01/      00016280
DATA      00016290
* P(57),P(58),P(59),P(60),P(61),P(62),P(63),      00016300
* P(64),P(65),P(66),P(67),P(68),P(69),P(70),      00016310
* P(71),P(72),P(73),P(74),P(75),P(76),P(77),      00016320
* P(78),P(79),P(80),P(81),P(82),P(83),P(84)/      00016330
* 0.38439810249455532039E-01,0.46813554990628012403E-01,      00016340
* 0.52834946790116519862E-01,0.55978436510476319408E-01,      00016350
* 0.99987288812035761194E 00,0.36322148184553065969E-03,      00016360
* 0.99720625937222195908E 00,0.25790497946856882724E-02,      00016370
* 0.98868475754742947994E 00,0.61155068221172463397E-02,      00016380
* 0.97218287474858179658E 00,0.10498246909621321898E-01,      00016390
* 0.94634285837340290515E 00,0.15406750466559497802E-01,      00016400
* 0.91037115695700429250E 00,0.20594233915912711149E-01,      00016410
* 0.86390793819369047715E 00,0.25869679327214746911E-01,      00016420
* 0.80694053195021761186E 00,0.31073551111687964880E-01,      00016430

```

```

* 0.73975604435269475868E 00,0.36064432780782572640E-01,      00016440
* 0.66290966002478059546E 00,0.40715510116944318934E-01,      00016450
* 0.57719571005204581484E 00,0.44914531653632197414E-01,      00016460
* 0.48361802694584102756E 00,0.48564330406673198716E-01/      00016470
DATA      00016480
* P( 85),P( 86),P( 87),P( 88),P( 89),P( 90),P( 91),      00016490
* P( 92),P( 93),P( 94),P( 95),P( 96),P( 97),P( 98),      00016500
* P( 99),P(100),P(101),P(102),P(103),P(104),P(105),      00016510
* P(106),P(107),P(108),P(109),P(110),P(111),P(112)/      00016520
* 0.38335932419873034692E 00,0.51583253952048458777E-01,      00016530
* 0.27774982202182431507E 00,0.53905499335266063927E-01,      00016540
* 0.16823525155220746498E 00,0.55481404356559363988E-01,      00016550
* 0.56344313046592789972E-01,0.56277699831254301273E-01,      00016560
* 0.56377628360384717388E-01,0.16801938574103865271E-01,      00016570
* 0.64519000501757369228E-02,0.25078569652949768707E-01,      00016580
* 0.21088152457266328793E-02,0.11615723319955134727E-01,      00016590
* 0.21438980012503867246E-01,0.27394605263981432516E-01,      00016600
* 0.63260731936263354422E-03,0.41115039786546930472E-02,      00016610
* 0.89892757840641357233E-02,0.14244877372916774306E-01,      00016620
* 0.19219905124727766019E-01,0.23406777495314006201E-01,      00016630
* 0.26417473395058259931E-01,0.27989218255238159704E-01,      00016640
* 0.18073956444538835782E-03,0.12895240826104173921E-02,      00016650
* 0.30577534101755311361E-02,0.52491234548088591251E-02/      00016660
DATA      00016670
* P(113),P(114),P(115),P(116),P(117),P(118),P(119),      00016680
* P(120),P(121),P(122),P(123),P(124),P(125),P(126),      00016690
* P(127),P(128),P(129),P(130),P(131),P(132),P(133),      00016700
* P(134),P(135),P(136),P(137),P(138),P(139),P(140)/      00016710
* 0.77033752332797418482E-02,0.10297116957956355524E-01,      00016720
* 0.12934839663607373455E-01,0.15536775555843982440E-01,      00016730
* 0.18032216390391286320E-01,0.20357755058472159467E-01,      00016740
* 0.22457265826816098707E-01,0.24282165203336599358E-01,      00016750
* 0.25791626976024229388E-01,0.26952749667633031963E-01,      00016760
* 0.27740702178279681994E-01,0.28138849915627150636E-01,      00016770
* 0.99998243035489159858E 00,0.50536095207862517625E-04,      00016780
* 0.99959879967191068325E 00,0.37774664632698466027E-03,      00016790
* 0.99831663531840739253E 00,0.93836984854238150079E-03,      00016800
* 0.99572410469840718851E 00,0.16811428654214699063E-02,      00016810
* 0.99149572117810613240E 00,0.25687649437940203731E-02,      00016820
* 0.98537149959852037111E 00,0.35728927835172996494E-02,      00016830
* 0.97714151463970571416E 00,0.46710503721143217474E-02,      00016840
* 0.96663785155841656709E 00,0.58434498758356395076E-02/      00016850
DATA      00016860
* P(141),P(142),P(143),P(144),P(145),P(146),P(147),      00016870
* P(148),P(149),P(150),P(151),P(152),P(153),P(154),      00016880
* P(155),P(156),P(157),P(158),P(159),P(160),P(161),      00016890
* P(162),P(163),P(164),P(165),P(166),P(167),P(168)/      00016900
* 0.95373000642576113641E 00,0.70724899954335554680E-02,      00016910
* 0.93832039777959288365E 00,0.83428387539681577056E-02,      00016920
* 0.92034002547001242073E 00,0.96411777297025366953E-02,      00016930
* 0.89974489977694003664E 00,0.10955733387837901648E-01,      00016940
* 0.87651341448470526974E 00,0.12275830560082770087E-01,      00016950

```

```

* 0.85064449476835027976E 00,0.13591571009765546790E-01, 00016960
* 0.82215625436498040737E 00,0.14893641664815182035E-01, 00016970
* 0.79108493379984836143E 00,0.16173218729577719942E-01, 00016980
* 0.75748396638051363793E 00,0.17421930159464173747E-01, 00016990
* 0.72142308537009891548E 00,0.18631848256138790186E-01, 00017000
* 0.68298743109107922809E 00,0.19795495048097499488E-01, 00017010
* 0.64227664250975951377E 00,0.20905851445812023852E-01, 00017020
* 0.59940393024224289297E 00,0.21956366305317824939E-01, 00017030
* 0.55449513263193254887E 00,0.22940964229387748761E-01/ 00017040
DATA 00017050
* P(169),P(170),P(171),P(172),P(173),P(174),P(175), 00017060
* P(176),P(177),P(178),P(179),P(180),P(181),P(182), 00017070
* P(183),P(184),P(185),P(186),P(187),P(188),P(189), 00017080
* P(190),P(191),P(192),P(193),P(194),P(195),P(196)/ 00017090
* 0.50768775753371660215E 00,0.23854052106038540080E-01, 00017100
* 0.45913001198983233287E 00,0.24690524744487676909E-01, 00017110
* 0.40897982122988867241E 00,0.25445769965464765813E-01, 00017120
* 0.35740383783153215238E 00,0.26115673376706097680E-01, 00017130
* 0.30457644155671404334E 00,0.26696622927450359906E-01, 00017140
* 0.25067873030348317661E 00,0.27185513229624791819E-01, 00017150
* 0.19589750271110015392E 00,0.27579749566481873035E-01, 00017160
* 0.14042423315256017459E 00,0.27877251476613701609E-01, 00017170
* 0.84454040083710883710E-01,0.28076455793817246607E-01, 00017180
* 0.28184648949745694339E-01,0.28176319033016602131E-01, 00017190
* 0.28188814180192358694E-01,0.84009692870519326354E-02, 00017200
* 0.32259500250878684614E-02,0.12539284826474884353E-01, 00017210
* 0.10544076228633167722E-02,0.58078616599775673635E-02, 00017220
* 0.10719490006251933623E-01,0.13697302631990716258E-01/ 00017230
DATA 00017240
* P(197),P(198),P(199),P(200),P(201),P(202),P(203), 00017250
* P(204),P(205),P(206),P(207),P(208),P(209),P(210), 00017260
* P(211),P(212),P(213),P(214),P(215),P(216),P(217), 00017270
* P(218),P(219),P(220),P(221),P(222),P(223),P(224)/ 00017280
* 0.31630366082226447689E-03,0.20557519893273465236E-02, 00017290
* 0.44946378920320678616E-02,0.71224386864583871532E-02, 00017300
* 0.96099525623638830097E-02,0.11703388747657003101E-01, 00017310
* 0.13208736697529129966E-01,0.13994609127619079852E-01, 00017320
* 0.90372734658751149261E-04,0.64476204130572477933E-03, 00017330
* 0.15288767050877655684E-02,0.26245617274044295626E-02, 00017340
* 0.38516876166398709241E-02,0.51485584789781777618E-02, 00017350
* 0.64674198318036867274E-02,0.77683877779219912200E-02, 00017360
* 0.90161081951956431600E-02,0.10178877529236079733E-01, 00017370
* 0.11228632913408049354E-01,0.12141082601668299679E-01, 00017380
* 0.12895813488012114694E-01,0.13476374833816515982E-01, 00017390
* 0.13870351089139840997E-01,0.14069424957813575318E-01, 00017400
* 0.25157870384280661489E-04,0.18887326450650491366E-03, 00017410
* 0.46918492424785040975E-03,0.84057143271072246365E-03/ 00017420
DATA 00017430
* P(225),P(226),P(227),P(228),P(229),P(230),P(231), 00017440
* P(232),P(233),P(234),P(235),P(236),P(237),P(238), 00017450
* P(239),P(240),P(241),P(242),P(243),P(244),P(245), 00017460
* P(246),P(247),P(248),P(249),P(250),P(251),P(252)/ 00017470

```

```

* 0.12843824718970101768E-02,0.17864463917586498247E-02,      00017480
* 0.23355251860571608737E-02,0.29217249379178197538E-02,      00017490
* 0.35362449977167777340E-02,0.41714193769840788528E-02,      00017500
* 0.48205888648512683476E-02,0.54778666939189508240E-02,      00017510
* 0.61379152800413850435E-02,0.67957855048827733948E-02,      00017520
* 0.74468208324075910174E-02,0.80866093647888599710E-02,      00017530
* 0.87109650797320868736E-02,0.93159241280693950932E-02,      00017540
* 0.98977475240487497440E-02,0.10452925722906011926E-01,      00017550.
* 0.10978183152658912470E-01,0.11470482114693874380E-01,      00017560
* 0.11927026053019270040E-01,0.12345262372243838455E-01,      00017570
* 0.12722884982732382906E-01,0.13057836688353048840E-01,      00017580
* 0.13348311463725179953E-01,0.13592756614812395910E-01,      00017590
* 0.13789874783240936517E-01,0.13938625738306850804E-01,      00017600
* 0.14038227896908623303E-01,0.14088159516508301065E-01/      00017610
DATA      00017620
* P(253),P(254),P(255),P(256),P(257),P(258),P(259),      00017630
* P(260),P(261),P(262),P(263),P(264),P(265),P(266),      00017640
* P(267),P(268),P(269),P(270),P(271),P(272),P(273),      00017650
* P(274),P(275),P(276),P(277),P(278),P(279),P(280)/      00017660
* 0.99999759637974846462E 00,0.69379364324108267170E-05,      00017670
* 0.99994399620705437576E 00,0.53275293669780613125E-04,      00017680
* 0.99976049092443204733E 00,0.13575491094922871973E-03,      00017690
* 0.99938033802502358193E 00,0.24921240048299729402E-03,      00017700
* 0.99874561446809511470E 00,0.38974528447328229322E-03,      00017710
* 0.99780535449595727456E 00,0.55429531493037471492E-03,      00017720
* 0.99651414591489027385E 00,0.74028280424450333046E-03,      00017730
* 0.99483150280062100052E 00,0.94536151685852538246E-03,      00017740
* 0.99272134428278861533E 00,0.11674841174299594077E-02,      00017750
* 0.99015137040077015918E 00,0.14049079956551446427E-02,      00017760
* 0.98709252795403406719E 00,0.16561127281544526052E-02,      00017770
* 0.98351865757863272876E 00,0.19197129710138724125E-02,      00017780
* 0.97940628167086268381E 00,0.21944069253638388388E-02,      00017790
* 0.97473445975240266776E 00,0.24789582266575679307E-02/      00017800
DATA      00017810
* P(281),P(282),P(283),P(284),P(285),P(286),P(287),      00017820
* P(288),P(289),P(290),P(291),P(292),P(293),P(294),      00017830
* P(295),P(296),P(297),P(298),P(299),P(300),P(301),      00017840
* P(302),P(303),P(304),P(305),P(306),P(307),P(308)/      00017850
* 0.96948465950245923177E 00,0.27721957645934509940E-02,      00017860
* 0.96364062156981213252E 00,0.30730184347025783234E-02,      00017870
* 0.95718821610986096274E 00,0.33803979910869203823E-02,      00017880
* 0.95011529752129487656E 00,0.36933779170256508183E-02,      00017890
* 0.94241156519108305981E 00,0.40110687240750233989E-02,      00017900
* 0.93406843615772578800E 00,0.43326409680929828545E-02,      00017910
* 0.92507893290707565236E 00,0.46573172997568547773E-02,      00017920
* 0.91540758715576504064E 00,0.49843645647655386012E-02,      00017930
* 0.90514035881326159519E 00,0.53130866051870565663E-02,      00017940
* 0.89418456833555902286E 00,0.56428181013844441585E-02,      00017950
* 0.88256884024734190684E 00,0.59729195655081658049E-02,      00017960
* 0.87029305554811390585E 00,0.63027734490857587172E-02,      00017970
* 0.85735831088623215653E 00,0.66317812429018878941E-02,      00017980
* 0.84376688267270860104E 00,0.69593614093904229394E-02/      00017990

```

```

DATA
* P(309),P(310),P(311),P(312),P(313),P(314),P(315),
* P(316),P(317),P(318),P(319),P(320),P(321),P(322),
* P(323),P(324),P(325),P(326),P(327),P(328),P(329),
* P(330),P(331),P(332),P(333),P(334),P(335),P(336)/
* 0.82952219463740140018E 00,0.72849479805538070639E-02,
* 0.81462878765513741344E 00,0.76079896657190565832E-02,
* 0.79909229096084140180E 00,0.79279493342948491103E-02,
* 0.78291939411828301639E 00,0.82443037630328680306E-02,
* 0.76611781930376009072E 00,0.85565435613076896192E-02,
* 0.74869629361693660282E 00,0.88641732094824942641E-02,
* 0.73066452124218126133E 00,0.91667111635607884067E-02,
* 0.71203315536225203459E 00,0.94636899938300652943E-02,
* 0.69281376977911470289E 00,0.97546565363174114611E-02,
* 0.67301883023041847920E 00,0.10039172044056840798E-01,
* 0.65266166541001749610E 00,0.10316812330947621682E-01,
* 0.63175643771119423041E 00,0.10587167904885197931E-01,
* 0.61031811371518640016E 00,0.10849844089337314099E-01,
* 0.58836243444766254143E 00,0.11104461134006926537E-01/
DATA
* P(337),P(338),P(339),P(340),P(341),P(342),P(343),
* P(344),P(345),P(346),P(347),P(348),P(349),P(350),
* P(351),P(352),P(353),P(354),P(355),P(356),P(357),
* P(358),P(359),P(360),P(361),P(362),P(363),P(364)/
* 0.56590588542365442262E 00,0.11350654315980596602E-01,
* 0.54296566649831149049E 00,0.11588074033043952568E-01,
* 0.51955966153745702199E 00,0.11816385890830235763E-01,
* 0.49570640791876146017E 00,0.12035270785279562630E-01,
* 0.47142506587165887693E 00,0.12244424981611985899E-01,
* 0.44673538766202847374E 00,0.12443560190714035263E-01,
* 0.42165768662616330006E 00,0.12632403643542078765E-01,
* 0.39621280605761593918E 00,0.12810698163877361967E-01,
* 0.37042208795007823014E 00,0.12978202239537399286E-01,
* 0.34430734159943802278E 00,0.13134690091960152836E-01,
* 0.31789081206847668318E 00,0.13279951743930530650E-01,
* 0.29119514851824668196E 00,0.13413793085110098513E-01,
* 0.26424337241092676194E 00,0.13536035934956213614E-01,
* 0.23705884558982972721E 00,0.13646518102571291428E-01/
DATA
* P(365),P(366),P(367),P(368),P(369),P(370),P(371),
* P(372),P(373),P(374),P(375),P(376),P(377),P(378),
* P(379),P(380),P(381)/
* 0.20966523824318119477E 00,0.13745093443001896632E-01,
* 0.18208649675925219825E 00,0.13831631909506428676E-01,
* 0.15434681148137810869E 00,0.13906019601325461264E-01,
* 0.12647058437230196685E 00,0.13968158806516938516E-01,
* 0.98482396598119202090E-01,0.14017968039456608810E-01,
* 0.70406976042855179063E-01,0.14055382072649964277E-01,
* 0.42269164765363603212E-01,0.14080351962553661325E-01,
* 0.14093886410782462614E-01,0.14092845069160408355E-01,
* 0.14094407090096179347E-01/
IF(MULTICS.EQ.1) RETURN

```

```

DO 1 I=1,381                                00018520
1  Q(I)=P(I)                                00018530
    MULTICS=1                                00018540
    RETURN                                    00018550
    END                                        00018560

    COMPLEX FUNCTION ZEX(B,NEW,R)              00018570
C--ZEX COMPUTES THE P(R) TERM WHICH IS        00018580
C DOUBLE INTEGRATED OVER FINITE LIMITS.       00018590
C IT IS PART OF THE EQUATION FOR THE          00018600
C ELECTRIC FIELD OF AN ELECTRIC DIPOLE.       00018610
C                                              00018620
C      B      INDUCTION NUMBER                00018630
C      R      DISTANCE                        00018640
C      NEW CONTROLS ZLAGHO INTEGRATION        00018650
C                                              00018660
    COMPLEX ZLAGHO,TWODEL3,ONE                00018670
    EXTERNAL F3                                00018680
    COMMON/PAARM/ISTEP,A1,A2,A3,A4,A5,M,TOL    00018690
    COMMON/CONST/DEL,DEL2,TWODEL3            00018700
    DATA ONE/(1.0,0.0)/                      00018710
    ZEX=CMPLX(0.0,0.0)                        00018720
    IF(M.EQ.1) GO TO 2                        00018730
    ZEX=ZLAGHO(ALOG(B),F3,TOL,LW,NEW)/B        00018740
2  ZEX=TWODEL3*ZEX+(ONE-(ONE+CMPLX(B,B))*CEXP(-CMPLX(B,B)))/R**3 00018750
    RETURN                                    00018760
    END                                        00018770

    COMPLEX FUNCTION ZFOURO(X,FUN,TOL,L)       00018780
C--REVISED VERSION: 12-13-76 -- SEE NOTE(2) BELOW. 00018790
C--INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*COS(G*B)*DG' DEFINED AS THE 00018800
C COMPLEX FOURIER COSINE TRANSFORM WITH ARGUMENT X(=ALOG(B)) 00018810
C BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND 00018820
C USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS.... 00018830
C                                              00018840
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO. 00018850
C                                              00018860
C--PARAMETERS:                                00018870
C                                              00018880
C      X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM 00018890
C      FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED) 00018900
C              OF A REAL ARGUMENT G.          00018910
C      NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN 00018920
C              CALLING PROGRAM AND IN SUBPROGRAM FUN.          00018930
C              THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE 00018940
C              DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00018950
C              FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RFOURO' IS ADVISED;00018960
C              HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE 00018970
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G))00018980
C      TOL=    REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E., 00018990
C              IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00019000
C              THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY, 00019010

```

```

C          TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON      00019020
C          THE FUNCTION FUN AND PARAMETER X...IN GENERAL,        00019030
C          A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY' 00019040
C          BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY 00019050
C          RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00019060
C          APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,   00019070
C          ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...  00019080
C          L=      RESULTING NO. FILTER WTS. USED IN THE VARIABLE  00019090
C          CONVOLUTION (L DEPENDS ON TOL AND FUN).                00019100
C          MIN.L=24 AND MAX.L=281--WHICH COULD                    00019110
C          OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING  00019120
C          VERY FAST...                                           00019130
C                                                                00019140
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZFOURO; THE FOURIER 00019150
C TRANSFORM IS THEN ZFOURO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM 00019160
C THIS ROUTINE....                                              00019170
C                                                                00019180
C--USAGE-- 'ZFOURO' IS CALLED AS FOLLOWS:                        00019190
C      ...                                                       00019200
C      COMPLEX Z,ZFOURO,ZF                                       00019210
C      EXTERNAL ZF                                               00019220
C      ...                                                       00019230
C      Z=ZFOURO(ALOG(B),ZF,TOL,L)/B                             00019240
C      ...                                                       00019250
C      END                                                       00019260
C      COMPLEX FUNCTION ZF(G)                                     00019270
C      ...USER SUPPLIED CODE...                                  00019280
C      END                                                       00019290
C                                                                00019300
C--NOTES:                                                        00019310
C      (1). EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THIS SUBPROGRAM; 00019320
C          THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS 00019330
C          EXP-UNDERFLOW'S TO 0.0.....                          00019340
C      (2). THIS SUBPROGRAM IS AN ANSI REVISION OF THE ORIGINAL    00019350
C          PUBLISHED VERSION IN NTIS REPT. PB-242-800, P.54-58;    00019360
C          IMPROVEMENTS HAVE BEEN MADE IN OVERALL EXECUTION TIME,  00019370
C          HOWEVER, THE CALLING SEQUENCE AND FILTER WEIGHTS        00019380
C          WERE NOT CHANGED.                                       00019390
C      (3). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED        00019400
C          TO SAVE STORAGE.                                       00019410
C          COMPLEX FUN,C,CMAX                                     00019420
C          DOUBLE PRECISION A,E,ER,Y1,Y                          00019430
C          DIMENSION T(2),TMAX(2)                                00019440
C          DIMENSION WT(281),W1(76),W2(76),W3(76),W4(53)        00019450
C          EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)), 00019460
C          1 (WT(229),W4(1))                                     00019470
C          EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                  00019480
C          DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/ 00019490
C--COS-EXTENDED FILTER WEIGHT ARRAYS:                            00019500
C      DATA W1/                                                 00019510
C      1 5.1178101E-14, 2.9433849E-14, 2.5492522E-14, 1.9034819E-14, 00019520
C      2 6.4179780E-14, 1.3085746E-15, 1.1989957E-13,-1.2216234E-14, 00019530

```

```

3 1.7534103E-13, 7.9373498E-15, 2.1235658E-13, 7.9981520E-14, 00019540
4 2.3815757E-13, 1.9714260E-13, 2.8920132E-13, 3.4161340E-13, 00019550
5 4.0349917E-13, 5.2203885E-13, 5.9837223E-13, 7.8015306E-13, 00019560
6 8.8911655E-13, 1.1709731E-12, 1.3165595E-12, 1.7578463E-12, 00019570
7 1.9538564E-12, 2.6289768E-12, 2.9167697E-12, 3.9044344E-12, 00019580
8 4.3927341E-12, 5.7526904E-12, 6.6569552E-12, 8.4555678E-12, 00019590
9 1.0063229E-11, 1.2487964E-11, 1.5134682E-11, 1.8501488E-11, 00019600
1 2.2720051E-11, 2.7452598E-11, 3.4025443E-11, 4.0875985E-11, 00019610
2 5.0751668E-11, 6.1094382E-11, 7.5492982E-11, 9.1445759E-11, 00019620
3 1.1227336E-10, 1.3676464E-10, 1.6720269E-10, 2.0423244E-10, 00019630
4 2.4932743E-10, 3.0470661E-10, 3.7198526E-10, 4.5449934E-10, 00019640
5 5.5502537E-10, 6.7793669E-10, 8.2810001E-10, 1.0112626E-09, 00019650
6 1.2354800E-09, 1.5085255E-09, 1.8432253E-09, 2.2503397E-09, 00019660
7 2.7499027E-09, 3.3569525E-09, 4.1025670E-09, 5.0077487E-09, 00019670
8 6.1205950E-09, 7.4703399E-09, 9.1312760E-09, 1.1143911E-08, 00019680
9 1.3622929E-08, 1.6623917E-08, 2.0324094E-08, 2.4798610E-08, 00019690
1 3.0321709E-08, 3.6992986E-08, 4.5237482E-08, 5.5183434E-08/ 00019700
DATA W2/ 00019710
1 6.7491070E-08, 8.2317946E-08, 1.0069271E-07, 1.2279375E-07, 00019720
2 1.5022907E-07, 1.8316969E-07, 2.2413747E-07, 2.7322865E-07, 00019730
3 3.3441046E-07, 4.0756197E-07, 4.9894278E-07, 6.0793233E-07, 00019740
4 7.4443665E-07, 9.0679753E-07, 1.1107379E-06, 1.3525651E-06, 00019750
5 1.6573073E-06, 2.0174273E-06, 2.4728798E-06, 3.0090445E-06, 00019760
6 3.6898816E-06, 4.4879625E-06, 5.5059521E-06, 6.6935820E-06, 00019770
7 8.2160716E-06, 9.9828691E-06, 1.2260527E-05, 1.4888061E-05, 00019780
8 1.8296530E-05, 2.2202672E-05, 2.7305154E-05, 3.3109672E-05, 00019790
9 4.0751046E-05, 4.9372484E-05, 6.0820947E-05, 7.3619571E-05, 00019800
1 9.0780005E-05, 1.0976837E-04, 1.3550409E-04, 1.6365676E-04, 00019810
2 2.0227521E-04, 2.4398338E-04, 3.0197018E-04, 3.6370760E-04, 00019820
3 4.5083748E-04, 5.4213338E-04, 6.7315347E-04, 8.0800951E-04, 00019830
4 1.0051938E-03, 1.2041401E-03, 1.5011708E-03, 1.7942344E-03, 00019840
5 2.2421056E-03, 2.6730676E-03, 3.3490681E-03, 3.9815050E-03, 00019850
6 5.0028666E-03, 5.9285668E-03, 7.4730905E-03, 8.8233510E-03, 00019860
7 1.1160132E-02, 1.3119627E-02, 1.6653199E-02, 1.9472767E-02, 00019870
8 2.4800811E-02, 2.8793704E-02, 3.6762063E-02, 4.2228780E-02, 00019880
9 5.3905163E-02, 6.0804660E-02, 7.7081738E-02, 8.3874501E-02, 00019890
1 1.0377190E-01, 1.0377718E-01, 1.1892208E-01, 9.0437429E-02/ 00019900
DATA W3/ 00019910
1 7.1685138E-02, -3.9473064E-02, -1.5078720E-01, -4.0489859E-01, 00019920
2 -5.6018995E-01, -6.8050388E-01, -1.5094224E-01, 6.6304064E-01, 00019930
3 1.3766748E+00, -8.0373222E-01, -1.0869629E+00, 1.2812892E+00, 00019940
4 -5.0341082E-01, -4.4274455E-02, 2.0913102E-01, -1.9999661E-01, 00019950
5 1.5207664E-01, -1.0920260E-01, 7.8169956E-02, -5.6651561E-02, 00019960
6 4.1611799E-02, -3.0880012E-02, 2.3072559E-02, -1.7311631E-02, 00019970
7 1.3021442E-02, -9.8085025E-03, 7.3943529E-03, -5.5769518E-03, 00019980
8 4.2073164E-03, -3.1745026E-03, 2.3954154E-03, -1.8076122E-03, 00019990
9 1.3640816E-03, -1.0293934E-03, 7.7682952E-04, -5.8623518E-04, 00020000
1 4.4240399E-04, -3.3386183E-04, 2.5195025E-04, -1.9013541E-04, 00020010
2 1.4348659E-04, -1.0828284E-04, 8.1716174E-05, -6.1667509E-05, 00020020
3 4.6537684E-05, -3.5119887E-05, 2.6503388E-05, -2.0000904E-05, 00020030
4 1.5093768E-05, -1.1390572E-05, 8.5959318E-06, -6.4869407E-06, 00020040
5 4.8953713E-06, -3.6942830E-06, 2.7878625E-06, -2.1038241E-06, 00020050

```


6	1.5875917E-06,-1.1980090E-06,	9.0398030E-07,-6.8208296E-07,	00020060
7	5.1458650E-07,-3.8817581E-07,	2.9272267E-07,-2.2067921E-07,	00020070
8	1.6623514E-07,-1.2514102E-07,	9.4034535E-08,-7.0556837E-08,	00020080
9	5.2741581E-08,-3.9298610E-08,	2.9107255E-08,-2.1413893E-08,	00020090
1	1.5742032E-08,-1.1498608E-08,	8.7561571E-09,-7.2959446E-09/	00020100
	DATA W4/		00020110
1	6.8816619E-09,-8.9679825E-09,	1.4258275E-08,-1.9564299E-08,	00020120
2	2.0235313E-08,-1.4725545E-08,	5.4632820E-09, 3.5995580E-09,	00020130
3	-9.5287133E-09, 1.1460041E-08,-1.0250532E-08,	7.4641748E-09,	00020140
4	-4.4703465E-09, 2.0499053E-09,-4.4806353E-10,-4.0374336E-10,		00020150
5	7.0321001E-10,-6.7067960E-10, 4.9130404E-10,-2.8840747E-10,		00020160
6	1.2373144E-10,-1.5260443E-11,-4.2027559E-11, 6.1885474E-11,		00020170
7	-5.9273937E-11, 4.6588766E-11,-3.2054182E-11, 1.9831637E-11,		00020180
8	-1.1210098E-11, 5.9567021E-12,-3.2427812E-12, 2.1353868E-12,		00020190
9	-1.8476851E-12, 1.8438474E-12,-1.8362842E-12, 1.7241847E-12,		00020200
1	-1.5161479E-12, 1.2627657E-12,-1.0129176E-12, 7.9578625E-13,		00020210
2	-6.2131435E-13, 4.8745900E-13,-3.8703630E-13, 3.1172547E-13,		00020220
3	-2.5397802E-13, 2.0824130E-13,-1.7123163E-13, 1.4113344E-13,		00020230
4	-1.1687986E-13, 9.7664016E-14,-8.2977176E-14, 7.2515267E-14,		00020240
5	-5.6047478E-14/		00020250

C--\$ENDATA

C

	A=DBLE(EXP(-X-30.3025124))	00020280
	ZFOURO=(0.0,0.0)	00020290
	CMAX=(0.0,0.0)	00020300
	L=22	00020310
	Y1=A*0.7163358133446166781 D+13	00020320
	Y=Y1	00020330
	DO 110 I=149,170	00020340
	Y=Y*E	00020350
	C=FUN(SNGL(Y))*WT(I)	00020360
	ZFOURO=ZFOURO+C	00020370
	TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))	00020380
	TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))	00020390
110	CONTINUE	00020400
	IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150	00020410
	CMAX=TOL*CMAX	00020420
	DO 120 I=171,281	00020430
	Y=Y*E	00020440
	C=FUN(SNGL(Y))*WT(I)	00020450
	ZFOURO=ZFOURO+C	00020460
	L=L+1	00020470
	IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130	00020480
120	CONTINUE	00020490
130	Y=Y1*E	00020500
	DO 140 I=1,148	00020510
	Y=Y*ER	00020520
	C=FUN(SNGL(Y))*WT(149-I)	00020530
	ZFOURO=ZFOURO+C	00020540
	L=L+1	00020550
	IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190	00020560
140	CONTINUE	00020570

```

150  GO TO 190                                00020580
      Y=A                                    00020590
      DO 160 I=1,148                        00020600
      Y=Y*E                                  00020610
      C=FUN(SNGL(Y))*WT(I)                  00020620
      ZFOURO=ZFOURO+C                       00020630
      L=L+1                                 00020640
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170 00020650
160  CONTINUE                                00020660
170  Y=A*0.3120389295208079937 D+25        00020670
      DO 180 I=1,111                        00020680
      Y=Y*ER                                  00020690
      C=FUN(SNGL(Y))*WT(282-I)              00020700
      ZFOURO=ZFOURO+C                       00020710
      L=L+1                                 00020720
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190 00020730
180  CONTINUE                                00020740
190  RETURN                                  00020750
      END                                    00020760

      COMPLEX FUNCTION ZHANKO(X,FUN,TOL,L)    00020770
C--REVISED VERSION: 12-13-76 -- SEE NOTE(2) BELOW. 00020780
C--INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*JO(G*B)*DG' DEFINED AS THE 00020790
C  COMPLEX HANKEL TRANSFORM OF ORDER 0 AND ARGUMENT X(=ALOG(B)) 00020800
C  BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND 00020810
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS.... 00020820
C 00020830
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO. 00020840
C 00020850
C--PARAMETERS: 00020860
C 00020870
C      X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM 00020880
C      FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED) 00020890
C              OF A REAL ARGUMENT G. 00020900
C      NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN 00020910
C              CALLING PROGRAM AND IN SUBPROGRAM FUN. 00020920
C              THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE 00020930
C              DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE... 00020940
C              FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RHANKO' IS ADVISED; 00020950
C              HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE 00020960
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)) 00020970
C      TOL=    REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E., 00020980
C              IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED. 00020990
C              THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY, 00021000
C              TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON 00021010
C              THE FUNCTION FUN AND PARAMETER X...IN GENERAL, 00021020
C              A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY' 00021030
C              BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY 00021040
C              RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00021050
C              APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B, 00021060
C              ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE... 00021070
C      L=      RESULTING NO. FILTER WTS. USED IN THE VARIABLE . 00021080

```

```

C          CONVOLUTION (L DEPENDS ON TOL AND FUN).          00021090
C          MIN.L=20 AND MAX.L=193--WHICH COULD             00021100
C          OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING 00021110
C          VERY FAST...                                     00021120
C                                                         00021130
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZHANKO; THE HANKEL 00021140
C TRANSFORM IS THEN ZHANKO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM 00021150
C THIS ROUTINE....                                         00021160
C                                                         00021170
C--USAGE-- 'ZHANKO' IS CALLED AS FOLLOWS:                 00021180
C          ...                                             00021190
C          COMPLEX Z,ZHANKO,ZF                             00021200
C          EXTERNAL ZF                                     00021210
C          ...                                             00021220
C          Z=ZHANKO(ALOG(B),ZF,TOL,L)/B                    00021230
C          ...                                             00021240
C          END                                              00021250
C          COMPLEX FUNCTION ZF(G)                          00021260
C          ...USER SUPPLIED CODE...                       00021270
C          END                                              00021280
C                                                         00021290
C--NOTES:                                                  00021300
C          (1). EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THIS SUBPROGRAM; 00021310
C          THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS 00021320
C          EXP-UNDERFLOW'S TO 0.0.....                   00021330
C          (2). THIS SUBPROGRAM IS AN ANSI REVISION OF THE ORIGINAL 00021340
C          PUBLISHED VERSION IN NTIS REPT. PB-242-800, P.45-48; 00021350
C          IMPROVEMENTS HAVE BEEN MADE IN OVERALL EXECUTION TIME, 00021360
C          HOWEVER, THE CALLING SEQUENCE AND FILTER WEIGHTS 00021370
C          WERE NOT CHANGED.                                00021380
C          (3). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED 00021390
C          TO SAVE STORAGE.                                00021400
C          COMPLEX FUN,C,CMAX                               00021410
C          DOUBLE PRECISION A,E,ER,Y1,Y                    00021420
C          DIMENSION T(2),TMAX(2)                          00021430
C          DIMENSION WT(193),W1(76),W2(76),W3(41)          00021440
C          EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)) 00021450
C          EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))             00021460
C          DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/ 00021470
C--JO-EXTENDED FILTER WEIGHT ARRAYS:                      00021480
C          DATA W1/                                        00021490
C          1 5.8565723E-08, 7.1143477E-11,-7.8395565E-11, 8.7489547E-11, 00021500
C          2-8.9007811E-11, 9.8790055E-11,-9.8675347E-11, 1.1118797E-10, 00021510
C          3-1.0893474E-10, 1.2543400E-10,-1.1979399E-10, 1.4200767E-10, 00021520
C          4-1.3106341E-10, 1.6153229E-10,-1.4238602E-10, 1.8486236E-10, 00021530
C          5-1.5315381E-10, 2.1319755E-10,-1.6238115E-10, 2.4824144E-10, 00021540
C          6-1.6850378E-10, 2.9243813E-10,-1.6909302E-10, 3.4934366E-10, 00021550
C          7-1.6043759E-10, 4.2417082E-10,-1.3690001E-10, 5.2458440E-10, 00021560
C          8-8.9946096E-11, 6.6188220E-10,-6.6964033E-12, 8.5276151E-10, 00021570
C          9 1.3222770E-10, 1.1219600E-09, 3.5591442E-10, 1.5061956E-09, 00021580
C          1 7.0795382E-10, 2.0600379E-09, 1.2535947E-09, 2.8646623E-09, 00021590
C          2 2.0904225E-09, 4.0409101E-09, 3.3642886E-09, 5.7687700E-09, 00021600

```

```

3 5.2930786E-09, 8.3164338E-09, 8.2021809E-09, 1.2083635E-08, 00021610
4 1.2577400E-08, 1.7666303E-08, 1.9143895E-08, 2.5953011E-08, 00021620
5 2.8983953E-08, 3.8268851E-08, 4.3712685E-08, 5.6590075E-08, 00021630
6 6.5740136E-08, 8.3864288E-08, 9.8662323E-08, 1.2448811E-07, 00021640
7 1.4784461E-07, 1.8501974E-07, 2.2129198E-07, 2.7524203E-07, 00021650
8 3.3094739E-07, 4.0974828E-07, 4.9462868E-07, 6.1030809E-07, 00021660
9 7.3891802E-07, 9.0939667E-07, 1.1034727E-06, 1.3554600E-06, 00021670
1 1.6474556E-06, 2.0207696E-06, 2.4591294E-06, 3.0131400E-06/ 00021680
DATA W2/ 00021690
1 3.6701680E-06, 4.4934101E-06, 5.4770076E-06, 6.7015208E-06, 00021700
2 8.1726989E-06, 9.9954201E-06, 1.2194425E-05, 1.4909101E-05, 00021710
3 1.8194388E-05, 2.2239184E-05, 2.7145562E-05, 3.3174088E-05, 00021720
4 4.0499452E-05, 4.9486730E-05, 6.0421440E-05, 7.3822001E-05, 00021730
5 9.0141902E-05, 1.1012552E-04, 1.3448017E-04, 1.6428337E-04, 00021740
6 2.0062570E-04, 2.4507680E-04, 2.9930366E-04, 3.6560582E-04, 00021750
7 4.4651421E-04, 5.4541300E-04, 6.6612648E-04, 8.1365181E-04, 00021760
8 9.9374786E-04, 1.2138120E-03, 1.4824945E-03, 1.8107657E-03, 00021770
9 2.2115938E-03, 2.7012675E-03, 3.2991969E-03, 4.0295817E-03, 00021780
1 4.9214244E-03, 6.0106700E-03, 7.3405529E-03, 8.9643708E-03, 00021790
2 1.0946310E-02, 1.3365017E-02, 1.6314985E-02, 1.9910907E-02, 00021800
3 2.4289325E-02, 2.9612896E-02, 3.6070402E-02, 4.3876936E-02, 00021810
4 5.3264829E-02, 6.4465091E-02, 7.7664144E-02, 9.2918324E-02, 00021820
5 1.1000121E-01, 1.2811102E-01, 1.4543025E-01, 1.5832248E-01, 00021830
6 1.6049224E-01, 1.4170064E-01, 8.8788108E-02, -1.1330934E-02, 00021840
7 -1.5331864E-01, -2.9094670E-01, -2.9084655E-01, -2.9708834E-02, 00021850
8 3.9009601E-01, 1.7999785E-01, -4.1858139E-01, 1.5317216E-01, 00021860
9 6.5184953E-02, -1.0751806E-01, 7.8429567E-02, -4.6019124E-02, 00021870
1 2.5309571E-02, -1.3904823E-02, 7.8187120E-03, -4.5190369E-03/ 00021880
DATA W3/ 00021890
1 2.6729062E-03, -1.6073718E-03, 9.7715622E-04, -5.9804407E-04, 00021900
2 3.6749320E-04, -2.2635296E-04, 1.3960805E-04, -8.6172618E-05, 00021910
3 5.3212947E-05, -3.2867888E-05, 2.0304203E-05, -1.2543926E-05, 00021920
4 7.7499633E-06, -4.7882430E-06, 2.9584108E-06, -1.8278645E-06, 00021930
5 1.1293571E-06, -6.9778174E-07, 4.3113019E-07, -2.6637753E-07, 00021940
6 1.6458373E-07, -1.0168954E-07, 6.2829807E-08, -3.8819969E-08, 00021950
7 2.3985272E-08, -1.4819520E-08, 9.1563774E-09, -5.6573541E-09, 00021960
8 3.4954514E-09, -2.1597005E-09, 1.3343946E-09, -8.2447148E-10, 00021970
9 5.0941033E-10, -3.1474631E-10, 1.9447072E-10, -1.2015685E-10, 00021980
1 7.4241055E-11, -4.5871468E-11, 2.8343095E-11, -1.7513137E-11, 00021990
2 6.9049613E-12/ 00022000
C--$$ENDATA 00022010
C 00022020
A=DBLE(EXP(-X-26.3045570)) 00022030
ZHANK0=(0.0,0.0) 00022040
CMAX=(0.0,0.0) 00022050
L=18 00022060
Y1=A*0.1312014808028768988 D+12 00022070
Y=Y1 00022080
DO 110 I=129,146 00022090
Y=Y*E 00022100
C=FUN(SNGL(Y))*WT(I) 00022110
ZHANK0=ZHANK0+C 00022120

```

	TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))	00022130
	TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))	00022140
110	CONTINUE	00022150
	IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150	00022160
	CMAX=TOL*CMAX	00022170
	DO 120 I=147,193	00022180
	Y=Y*E	00022190
	C=FUN(SNGL(Y))*WT(I)	00022200
	ZHANKO=ZHANKO+C	00022210
	L=L+1	00022220
	IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130	00022230
120	CONTINUE	00022240
130	Y=Y1*E	00022250
	DO 140 I=1,128	00022260
	Y=Y*ER	00022270
	C=FUN(SNGL(Y))*WT(129-I)	00022280
	ZHANKO=ZHANKO+C	00022290
	L=L+1	00022300
	IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190	00022310
140	CONTINUE	00022320
	GO TO 190	00022330
150	Y=A	00022340
	DO 160 I=1,128	00022350
	Y=Y*E	00022360
	C=FUN(SNGL(Y))*WT(I)	00022370
	ZHANKO=ZHANKO+C	00022380
	L=L+1	00022390
	IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170	00022400
160	CONTINUE	00022410
170	Y=A*0.7089667994071963201 D+17	00022420
	DO 180 I=1,47	00022430
	Y=Y*ER	00022440
	C=FUN(SNGL(Y))*WT(194-I)	00022450
	ZHANKO=ZHANKO+C	00022460
	L=L+1	00022470
	IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190	00022480
180	CONTINUE	00022490
190	RETURN	00022500
	END	00022510

COMPLEX FUNCTION ZLAGHO(X,FUN,TOL,L,NEW) 00022520

C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE	00022530
C INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*JO(G*B)*DG' DEFINED AS THE	00022540
C COMPLEX HANKEL TRANSFORM OF ORDER 0 AND ARGUMENT X(=ALOG(B))	00022550
C BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND	00022560
C USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....	00022570
C	00022580
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.	00022590
C	00022600
C--PARAMETERS:	00022610
C	00022620
C * X = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM	00022630

```

C      'ZLAGHO' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E., 00022640
C      SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT, 00022650
C      THEN SUBPROGRAM 'ZHANKO' IS ADVISED FOR GENERAL USE. 00022660
C      (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW). 00022670
C      FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED) 00022680
C      OF A REAL ARGUMENT G. 00022690
C      NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN 00022700
C      CALLING PROGRAM AND IN SUBPROGRAM FUN. 00022710
C      THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE 00022720
C      DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE... 00022730
C      FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RLAGHO' IS ADVISED; 00022740
C      HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE 00022750
C      INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)) 00022760
C      TOL= REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E., 00022770
C      IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED. 00022780
C      THIS IS DONE AT BOTH ENDS OF FILTER. TYPICALLY, 00022790
C      TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON 00022800
C      THE FUNCTION FUN AND PARAMETER X...IN GENERAL, 00022810
C      A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY' 00022820
C      BUT WITH 'MORE WEIGHTS' BEING USED. TOL IS NOT DIRECTLY 00022830
C      RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00022840
C      APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B, 00022850
C      ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE... 00022860
C      L= RESULTING NO. FILTER WTS. USED IN THE VARIABLE 00022870
C      CONVOLUTION (L DEPENDS ON TOL AND FUN). 00022880
C      MIN.L=20 AND MAX.L=193--WHICH COULD 00022890
C      OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING 00022900
C      VERY FAST... 00022910
C      * NEW= 1 IS NECESSARY 1ST TIME OR BRAND NEW X. 00022920
C      0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20 00022930
C      IS ASSUMED INTERNALLY BY THIS ROUTINE. 00022940
C      NOTE: IF THIS IS NOT TRUE, ROUTINE WILL 00022950
C      STILL ASSUME X=(LAST X)-0.20 ANYWAY... 00022960
C      IT IS THE USERS RESPONSIBILITY TO NORMALIZE 00022970
C      BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW). 00022980
C      THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT 00022990
C      TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A 00023000
C      SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00023010
C      ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1... 00023020
C      THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED 00023030
C      KERNELS WILL BE USED IN THE LAGGED CONVOLUTION 00023040
C      WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS 00023050
C      WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN) 00023060
C      00023070
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZLAGHO; THE HANKEL 00023080
C TRANSFORM IS THEN ZLAGHO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM 00023090
C THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL... 00023100
C 00023110
C--USAGE-- 'ZLAGHO' IS CALLED AS FOLLOWS: 00023120
C ... 00023130
C COMPLEX Z,ZLAGHO,ZF 00023140
C EXTERNAL ZF 00023150

```

```

C      ...                                00023160
C      Z=ZLAGH0(ALOG(B),ZF,TOL,L,NEW)/B    00023170
C      ...                                00023180
C      END                                00023190
C      COMPLEX FUNCTION ZF(G)              00023200
C      ...USER SUPPLIED CODE...           00023210
C      END                                00023220
C                                          00023230
C--NOTES:                                00023240
C      (1). EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM 00023250
C      BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS 00023260
C      ANY & ALL EXP-UNDERFLOW'S TO 0.0.... 00023270
C      (2). AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION 00023280
C      METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN      00023290
C      THAT THE ACTUAL NUMBER OF B'S IS NB=AIN(5.*ALOG(BMAX/BMIN))+1, 00023300
C      PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED' 00023310
C      BMINA=BMAX*EXP(-.2*(NB-1)). THE METHOD GENERATES THE DECREASING 00023320
C      ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA). 00023330
C      FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:             00023340
C      ...                                00023350
C      NB=AIN(5.*ALOG(BMAX/BMIN))+1      00023360
C      NB1=NB+1                          00023370
C      X0=ALOG(BMAX)+.2                  00023380
C      NEW=1                             00023390
C      DO 1 J=1,NB                      00023400
C      I=NB1-J                          00023410
C      X=X0-.2*J                        00023420
C      ARG(I)=EXP(X)                    00023430
C      Z(I)=ZLAGH0(X,ZF,TOL,L,NEW)/ARG(I) 00023440
C      1 NEW=0                          00023450
C      ...                                00023460
C      (3). IF RESULTS ARE STORED IN ARRAYS ARG(I),Z(I),I=1,NB FOR 00023470
C      ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE, 00023480
C      TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)      00023490
C      SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.           00023500
C      (4). IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY      00023510
C      ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW          00023520
C      BMAX,BMIN AND BY SETTING NEW=1.... 00023530
C      (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE 00023540
C                                          00023550
C                                          00023560
C      COMPLEX FUN,C,CMAX,SAVE           00023570
C      DIMENSION KEY(193),SAVE(193),T(2),TMAX(2) 00023580
C      DIMENSION YT(193),Y1(76),Y2(76),Y3(41)    00023590
C      EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))        00023600
C      EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1)) 00023610
C--JO-EXTENDED FILTER WEIGHT ARRAYS:          00023620
C      DATA Y1/                             00023630
C      1 5.8565723E-08, 7.1143477E-11,-7.8395565E-11, 8.7489547E-11, 00023640
C      2-8.9007811E-11, 9.8790055E-11,-9.8675347E-11, 1.1118797E-10, 00023650
C      3-1.0893474E-10, 1.2543400E-10,-1.1979399E-10, 1.4200767E-10, 00023660
C      4-1.3106341E-10, 1.6153229E-10,-1.4238602E-10, 1.8486236E-10, 00023670

```

```

5-1.5315381E-10, 2.1319755E-10, -1.6238115E-10, 2.4824144E-10, 00023680
6-1.6850378E-10, 2.9243813E-10, -1.6909302E-10, 3.4934366E-10, 00023690
7-1.6043759E-10, 4.2417082E-10, -1.3690001E-10, 5.2458440E-10, 00023700
8-8.9946096E-11, 6.6188220E-10, -6.6964033E-12, 8.5276151E-10, 00023710
9 1.3222770E-10, 1.1219600E-09, 3.5591442E-10, 1.5061956E-09, 00023720
1 7.0795382E-10, 2.0600379E-09, 1.2535947E-09, 2.8646623E-09, 00023730
2 2.0904225E-09, 4.0409101E-09, 3.3642886E-09, 5.7687700E-09, 00023740
3 5.2930786E-09, 8.3164338E-09, 8.2021809E-09, 1.2083635E-08, 00023750
4 1.2577400E-08, 1.7666303E-08, 1.9143895E-08, 2.5953011E-08, 00023760
5 2.8983953E-08, 3.8268851E-08, 4.3712685E-08, 5.6590075E-08, 00023770
6 6.5740136E-08, 8.3864288E-08, 9.8662323E-08, 1.2448811E-07, 00023780
7 1.4784461E-07, 1.8501974E-07, 2.2129198E-07, 2.7524203E-07, 00023790
8 3.3094739E-07, 4.0974828E-07, 4.9462868E-07, 6.1030809E-07, 00023800
9 7.3891802E-07, 9.0939667E-07, 1.1034727E-06, 1.3554600E-06, 00023810
1 1.6474556E-06, 2.0207696E-06, 2.4591294E-06, 3.0131400E-06/ 00023820
DATA Y2/ 00023830
1 3.6701680E-06, 4.4934101E-06, 5.4770076E-06, 6.7015208E-06, 00023840
2 8.1726989E-06, 9.9954201E-06, 1.2194425E-05, 1.4909101E-05, 00023850
3 1.8194388E-05, 2.2239184E-05, 2.7145562E-05, 3.3174088E-05, 00023860
4 4.0499452E-05, 4.9486730E-05, 6.0421440E-05, 7.3822001E-05, 00023870
5 9.0141902E-05, 1.1012552E-04, 1.3448017E-04, 1.6428337E-04, 00023880
6 2.0062570E-04, 2.4507680E-04, 2.9930366E-04, 3.6560582E-04, 00023890
7 4.4651421E-04, 5.4541300E-04, 6.6612648E-04, 8.1365181E-04, 00023900
8 9.9374786E-04, 1.2138120E-03, 1.4824945E-03, 1.8107657E-03, 00023910
9 2.2115938E-03, 2.7012675E-03, 3.2991969E-03, 4.0295817E-03, 00023920
1 4.9214244E-03, 6.0106700E-03, 7.3405529E-03, 8.9643708E-03, 00023930
2 1.0946310E-02, 1.3365017E-02, 1.6314985E-02, 1.9910907E-02, 00023940
3 2.4289325E-02, 2.9612896E-02, 3.6070402E-02, 4.3876936E-02, 00023950
4 5.3264829E-02, 6.4465091E-02, 7.7664144E-02, 9.2918324E-02, 00023960
5 1.1000121E-01, 1.2811102E-01, 1.4543025E-01, 1.5832248E-01, 00023970
6 1.6049224E-01, 1.4170064E-01, 8.8788108E-02, -1.1330934E-02, 00023980
7-1.5331864E-01, -2.9094670E-01, -2.9084655E-01, -2.9708834E-02, 00023990
8 3.9009601E-01, 1.7999785E-01, -4.1858139E-01, 1.5317216E-01, 00024000
9 6.5184953E-02, -1.0751806E-01, 7.8429567E-02, -4.6019124E-02, 00024010
1 2.5309571E-02, -1.3904823E-02, 7.8187120E-03, -4.5190369E-03/ 00024020
DATA Y3/ 00024030
1 2.6729062E-03, -1.6073718E-03, 9.7715622E-04, -5.9804407E-04, 00024040
2 3.6749320E-04, -2.2635296E-04, 1.3960805E-04, -8.6172618E-05, 00024050
3 5.3212947E-05, -3.2867888E-05, 2.0304203E-05, -1.2543926E-05, 00024060
4 7.7499633E-06, -4.7882430E-06, 2.9584108E-06, -1.8278645E-06, 00024070
5 1.1293571E-06, -6.9778174E-07, 4.3113019E-07, -2.6637753E-07, 00024080
6 1.6458373E-07, -1.0168954E-07, 6.2829807E-08, -3.8819969E-08, 00024090
7 2.3985272E-08, -1.4819520E-08, 9.1563774E-09, -5.6573541E-09, 00024100
8 3.4954514E-09, -2.1597005E-09, 1.3343946E-09, -8.2447148E-10, 00024110
9 5.0941033E-10, -3.1474631E-10, 1.9447072E-10, -1.2015685E-10, 00024120
1 7.4241055E-11, -4.5871468E-11, 2.8343095E-11, -1.7513137E-11, 00024130
2 6.9049613E-12/ 00024140
C--$ENDATA 00024150
C 00024160
IF(NEW) 10,30,10 00024170
LAG=-1 00024180
X0=-X-26.30455704 00024190

```


	DO 20 IR=1,193	00024200
20	KEY(IR)=0	00024210
30	LAG=LAG+1	00024220
	ZLAGH0=(0.0,0.0)	00024230
	CMAX=(0.0,0.0)	00024240
	L=0	00024250
	ASSIGN 110 TO M	00024260
	I=129	00024270
	GO TO 200	00024280
110	TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))	00024290
	TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))	00024300
	I=I+1	00024310
	IF(I.LE.146) GO TO 200	00024320
	IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150	00024330
	CMAX=TOL*CMAX	00024340
	ASSIGN 120 TO M	00024350
	I=128	00024360
	GO TO 200	00024370
120	IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130	00024380
	I=I-1	00024390
	IF(I.GT.0) GO TO 200	00024400
130	ASSIGN 140 TO M	00024410
	I=147	00024420
	GO TO 200	00024430
140	IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190	00024440
	I=I+1	00024450
	IF(I.LE.193) GO TO 200	00024460
	GO TO 190	00024470
150	ASSIGN 160 TO M	00024480
	I=1	00024490
	GO TO 200	00024500
160	IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170	00024510
	I=I+1	00024520
	IF(I.LE.128) GO TO 200	00024530
170	ASSIGN 180 TO M	00024540
	I=193	00024550
	GO TO 200	00024560
180	IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190	00024570
	I=I-1	00024580
	IF(I.GE.147) GO TO 200	00024590
190	RETURN	00024600
	C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)	00024610
200	LOOK=I+LAG	00024620
	IQ=LOOK/194	00024630
	IR=MOD(LOOK,194)	00024640
	IF(IR.EQ.0) IR=1	00024650
	IROLL=IQ*193	00024660
	IF(KEY(IR).LE.IROLL) GO TO 220	00024670
210	C=SAVE(IR)*YT(I)	00024680
	ZLAGH0=ZLAGH0+C	00024690
	L=L+1	00024700
	GO TO M,(110,120,140,160,180)	00024710

220	KEY(IR)=IROLL+IR	00024720
	SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))	00024730
	GO TO 210	00024740
	END	00024750
	 SUBROUTINE ZQUAD1(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)	00024760
	COMPLEX F,RESULT,FUNCT,FZERO,ACUM	00024770
	DIMENSION FUNCT(127),P(381),RESULT(8)	00024780
	COMMON/ZQUADP/P	00024790
C--	FOLLOWING CALL ONLY FOR MULTICS SYSTEM:	00024800
	CALL ZBLOCK	00024810
	ICHECK = 0	00024820
C	CHECK FOR TRIVIAL CASE.	00024830
	IF (A.EQ.B) GO TO 70	00024840
C	SCALE FACTORS.	00024850
	SUM = (B+A)/2.0	00024860
	DIFF = (B-A)/2.0	00024870
C	1-POINT GAUSS	00024880
	FZERO = F(SUM)	00024890
	RESULT(1) = 2.0*FZERO*DIFF	00024900
	I = 0	00024910
	IOLD = 0	00024920
	INEW = 1	00024930
	K = 2	00024940
	ACUM = (0.0,0.0)	00024950
	GO TO 30	00024960
10	IF (K.EQ.8) GO TO 50	00024970
	IF(INEW+IOLD.GE.MEV) GO TO 60	00024980
	K = K + 1	00024990
	ACUM = (0.0,0.0)	00025000
C	CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.	00025010
	DO 20 J=1,IOLD	00025020
	I = I + 1	00025030
	ACUM = ACUM + P(I)*FUNCT(J)	00025040
20	CONTINUE	00025050
C	CONTRIBUTION FROM NEW FUNCTION VALUES.	00025060
30	IOLD = IOLD + INEW	00025070
	DO 40 J=INEW,IOLD	00025080
	I = I + 1	00025090
	X = P(I)*DIFF	00025100
	FUNCT(J) = F(SUM+X) + F(SUM-X)	00025110
	I = I + 1	00025120
	ACUM = ACUM + P(I)*FUNCT(J)	00025130
40	CONTINUE	00025140
	INEW = IOLD + 1	00025150
	I = I + 1	00025160
	RESULT(K) = (ACUM+P(I)*FZERO)*DIFF	00025170
C	CHECK FOR CONVERGENCE.	00025180
	IF(ABS(REAL(RESULT(K))-REAL(RESULT(K-1)))) .LE. EPSIL*	00025190
	\$ABS(REAL(RESULT(K))).AND.	00025200
	\$ABS(AIMAG(RESULT(K))-AIMAG(RESULT(K-1)))) .LE. EPSIL*	00025210
	\$ABS(AIMAG(RESULT(K)))) GO TO 60	00025220

GO TO 10	00025230
C CONVERGENCE NOT ACHIEVED.	00025240
50 ICHECK = 1	00025250
C NORMAL TERMINATION.	00025260
60 NPTS = INEW + IOLD	00025270
RETURN	00025280
C TRIVIAL CASE	00025290
70 K = 2	00025300
RESULT(1) = (0.0,0.0)	00025310
RESULT(2) = (0.0,0.0)	00025320
NPTS = 0	00025330
RETURN	00025340
END	00025350
SUBROUTINE ZQUAD2(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)	00025360
COMPLEX F,RESULT,FUNCT,FZERO,ACUM	00025370
DIMENSION FUNCT(127), P(381), RESULT(8)	00025380
COMMON/ZQUADP/P	00025390
C--FOLLOWING CALL ONLY FOR MULTICS SYSTEM:	00025400
CALL ZBLOCK	00025410
ICHECK = 0	00025420
C CHECK FOR TRIVIAL CASE.	00025430
IF (A.EQ.B) GO TO 70	00025440
C SCALE FACTORS.	00025450
SUM = (B+A)/2.0	00025460
DIFF = (B-A)/2.0	00025470
C 1-POINT GAUSS	00025480
FZERO = F(SUM)	00025490
RESULT(1) = 2.0*FZERO*DIFF	00025500
I = 0	00025510
IOLD = 0	00025520
INEW = 1	00025530
K = 2	00025540
ACUM = (0.0,0.0)	00025550
GO TO 30	00025560
10 IF (K.EQ.8) GO TO 50	00025570
IF(INEW+IOLD.GE.MEV) GO TO 60	00025580
K = K + 1	00025590
ACUM = (0.0,0.0)	00025600
C CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.	00025610
DO 20 J=1,IOLD	00025620
I = I + 1	00025630
ACUM = ACUM + P(I)*FUNCT(J)	00025640
20 CONTINUE	00025650
C CONTRIBUTION FROM NEW FUNCTION VALUES.	00025660
30 IOLD = IOLD + INEW	00025670
DO 40 J=INEW,IOLD	00025680
I = I + 1	00025690
X = P(I)*DIFF	00025700
FUNCT(J) = F(SUM+X) + F(SUM-X)	00025710
I = I + 1	00025720
ACUM = ACUM + P(I)*FUNCT(J)	00025730

40 CONTINUE	00025740
INEW = IOLD + 1	00025750
I = I + 1	00025760
RESULT(K) = (ACUM+P(I)*FZERO)*DIFF	00025770
C CHECK FOR CONVERGENCE.	00025780
IF(ABS(REAL(RESULT(K))-REAL(RESULT(K-1)))) .LE. EPSIL*	00025790
\$ABS(REAL(RESULT(K))).AND.	00025800
\$ ABS(AIMAG(RESULT(K))-AIMAG(RESULT(K-1))) .LE. EPSIL*	00025810
\$ABS(AIMAG(RESULT(K)))) GO TO 60	00025820
GO TO 10	00025830
C CONVERGENCE NOT ACHIEVED.	00025840
50 ICHECK = 1	00025850
C NORMAL TERMINATION.	00025860
60 NPTS = INEW + IOLD	00025870
RETURN	00025880
C TRIVIAL CASE	00025890
70 K = 2	00025900
RESULT(1) = (0.0,0.0)	00025910
RESULT(2) = (0.0,0.0)	00025920
NPTS = 0	00025930
RETURN	00025940
END	00025950
 COMPLEX FUNCTION ZSUB1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)	00025960
COMPLEX RELERR,F,RESULT,ESTIM,COMP	00025970
C THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION	00025980
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION	00025990
C ALGORITHM ZQUAD1, TOGETHER WITH, IF NECESSARY, A NON-	00026000
C ADAPTIVE SUBDIVISION PROCESS.	00026010
DIMENSION RESULT(8)	00026020
INTEGER BAD, OUT	00026030
LOGICAL RHS	00026040
EXTERNAL F	00026050
DATA NMAX/4096/	00026060
CALL ZQUAD1(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)	00026070
ZSUB1 = RESULT(K)	00026080
RELERR = (0.0,0.0)	00026090
IF(REAL(ZSUB1).NE.0.0.AND.AIMAG(ZSUB1).NE.0.0) RELERR=	00026100
\$ CMLPX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUB1),	00026110
\$ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUB1))	00026120
C CHECK IF SUBDIVISION IS NEEDED.	00026130
IF (ICHECK.EQ.0) RETURN	00026140
C SUBDIVIDE	00026150
ESTIM=ZSUB1*EPSIL	00026160
ESTIM=CMLPX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))	00026170
IC = 1	00026180
RHS = .FALSE.	00026190
N = 1	00026200
H = B - A	00026210
BAD = 1	00026220
10 ZSUB1 = (0.0,0.0)	00026230
RELERR = (0.0,0.0)	00026240

H = H*0.5	00026250
N = N + N	00026260
C INTERVAL (A,B) DIVIDED INTO N EQUAL SUBINTERVALS.	00026270
C INTEGRATE OVER SUBINTERVALS BAD TO (BAD+1) WHERE TROUBLE	00026280
C HAS OCCURRED.	00026290
M1 = BAD	00026300
M2 = BAD + 1	00026310
OUT = 1	00026320
GO TO 50	00026330
C INTEGRATE OVER SUBINTERVALS 1 TO (BAD-1)	00026340
20 M1 = 1	00026350
M2 = BAD - 1	00026360
RHS = .FALSE.	00026370
OUT = 2	00026380
GO TO 50	00026390
C INTEGRATE OVER SUBINTERVALS (BAD+2) TO N.	00026400
30 M1 = BAD + 2	00026410
M2 = N	00026420
OUT = 3	00026430
GO TO 50	00026440
C SUBDIVISION RESULT	00026450
40 ICHECK = IC	00026460
IF (REAL(ZSUB1).EQ.0.0) GO TO 42	00026470
IF (AIMAG(ZSUB1).EQ.0.0) GO TO 44	00026480
RELERR=CMPLX (REAL (RELERR)/ABS (REAL (ZSUB1))),	00026490
\$ AIMAG (RELERR)/ABS (AIMAG (ZSUB1)))	00026500
RETURN	00026510
42 IF (AIMAG (ZSUB1).EQ.0.0) GO TO 46	00026520
RELERR=CMPLX (0.0, AIMAG (RELERR)/ABS (AIMAG (ZSUB1)))	00026530
RETURN	00026540
44 RELERR=CMPLX (REAL (RELERR)/ABS (REAL (ZSUB1))), 0.0)	00026550
RETURN	00026560
46 RELERR=(0.0,0.0)	00026570
RETURN	00026580
C INTEGRATE OVER SUBINTERVALS M1 TO M2.	00026590
50 IF (M1.GT.M2) GO TO 90	00026600
DO 80 JJ=M1,M2	00026610
J = JJ	00026620
C EXAMINE FIRST THE LEFT OR RIGHT HALF OF THE SUBDIVIDED	00026630
C TROUBLESOME INTERVAL DEPENDING ON THE OBSERVED TREND.	00026640
IF (RHS) J = M2 + M1 - JJ	00026650
ALPHA = A + H*(J-1)	00026660
BETA = ALPHA + H	00026670
CALL ZQUAD1 (ALPHA, BETA, RESULT, M, EPSIL, NF, ICHECK, F, MEV)	00026680
COMP = (RESULT(M)-RESULT(M-1))	00026690
COMP=CMPLX (ABS (REAL (COMP)), ABS (AIMAG (COMP)))	00026700
NPTS = NPTS + NF	00026710
IF (NPTS.GE.MEV) GO TO 70	00026720
IF (ICHECK.NE.1) GO TO 70	00026730
IF (REAL (COMP).LE.REAL (ESTIM).AND.	00026740
\$ AIMAG (COMP).LE.AIMAG (ESTIM)) GO TO 100	00026750
C SUBINTERVAL J HAS CAUSED TROUBLE.	00026760

C CHECK IF FURTHER SUBDIVISION SHOULD BE CARRIED OUT.	00026770
IF (N.EQ.NMAX) GO TO 60	00026780
BAD = 2*J - 1	00026790
RHS = .FALSE.	00026800
IF ((J-2*(J/2)).EQ.0) RHS = .TRUE.	00026810
GO TO 10	00026820
60 IC = -IABS(IC)	00026830
70 ZSUB1 = ZSUB1 + RESULT(M)	00026840.
80 CONTINUE	00026850
RELERR = RELERR + COMP	00026860
90 IF(OUT-2) 20,30,40	00026870
C RELAXED CONVERGENCE	00026880
100 IC = ISIGN(2,IC)	00026890
GO TO 70	00026900
END	00026910
COMPLEX FUNCTION ZSUB2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)	
COMPLEX RELERR,F,RESULT,ESTIM,COMP	00026930
C THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION	00026940
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION	00026950
C ALGORITHM ZQUAD2, TOGETHER WITH, IF NECESSARY, A NON-	00026960
C ADAPTIVE SUBDIVISION PROCESS.	00026970
DIMENSION RESULT(8)	00026980
INTEGER BAD, OUT	00026990
LOGICAL RHS	00027000
EXTERNAL F	00027010
DATA NMAX/4096/	00027020
CALL ZQUAD2(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)	00027030
ZSUB2 = RESULT(K)	00027040
RELERR = (0.0,0.0)	00027050
IF(REAL(ZSUB2).NE.0.0.AND.AIMAG(ZSUB2).NE.0.0) RELERR=	00027060
\$ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUB2),	00027070
\$ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUB2))	00027080
C CHECK IF SUBDIVISION IS NEEDED.	00027090
IF (ICHECK.EQ.0) RETURN	00027100
C SUBDIVIDE	00027110
ESTIM=ZSUB2*EPSIL	00027120
ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))	00027130
IC = 1	00027140
RHS = .FALSE.	00027150
N = 1	00027160
H = B - A	00027170
BAD = 1	00027180
10 ZSUB2 = (0.0,0.0)	00027190
RELERR = (0.0,0.0)	00027200
H = H*0.5	00027210
N = N + N	00027220
C INTERVAL (A,B) DIVIDED INTO N EQUAL SUBINTERVALS.	00027230
C INTEGRATE OVER SUBINTERVALS BAD TO (BAD+1) WHERE TROUBLE	00027240
C HAS OCCURRED.	00027250
M1 = BAD	00027260
M2 = BAD + 1	00027270

OUT = 1	00027280
GO TO 50	00027290
C INTEGRATE OVER SUBINTERVALS 1 TO (BAD-1)	00027300
20 M1 = 1	00027310
M2 = BAD - 1	00027320
RHS = .FALSE.	00027330
OUT = 2	00027340
GO TO 50	00027350
C INTEGRATE OVER SUBINTERVALS (BAD+2) TO N.	00027360
30 M1 = BAD + 2	00027370
M2 = N	00027380
OUT = 3	00027390
GO TO 50	00027400
C SUBDIVISION RESULT	00027410
40 ICHECK = IC	00027420
IF (REAL(ZSUB2).EQ.0.0) GO TO 42	00027430
IF (AIMAG(ZSUB2).EQ.0.0) GO TO 44	00027440
RELERR=CMPLX (REAL (RELERR)/ABS (REAL (ZSUB2))),	00027450
\$ AIMAG (RELERR)/ABS (AIMAG (ZSUB2)))	00027460
RETURN	00027470
42 IF (AIMAG(ZSUB2).EQ.0.0) GO TO 46	00027480
RELERR=CMPLX (0.0, AIMAG (RELERR)/ABS (AIMAG (ZSUB2)))	00027490
RETURN	00027500
44 RELERR=CMPLX (REAL (RELERR)/ABS (REAL (ZSUB2))), 0.0)	00027510
RETURN	00027520
46 RELERR=(0.0,0.0)	00027530
RETURN	00027540
C INTEGRATE OVER SUBINTERVALS M1 TO M2.	00027550
50. IF (M1.GT.M2) GO TO 90	00027560
DO 80 JJ=M1,M2	00027570
J = JJ	00027580
C EXAMINE FIRST THE LEFT OR RIGHT HALF OF THE SUBDIVIDED	00027590
C TROUBLESOME INTERVAL DEPENDING ON THE OBSERVED TREND.	00027600
IF (RHS) J = M2 + M1 - JJ	00027610
ALPHA = A + H*(J-1)	00027620
BETA = ALPHA + H	00027630
CALL ZQUAD2 (ALPHA, BETA, RESULT, M, EPSIL, NF, ICHECK, F, MEV)	00027640
COMP = (RESULT(M)-RESULT(M-1))	00027650
COMP=CMPLX (ABS (REAL (COMP)), ABS (AIMAG (COMP)))	00027660
NPTS = NPTS + NF	00027670
IF (NPTS.GE.MEV) GO TO 70	00027680
IF (ICHECK.NE.1) GO TO 70	00027690
IF (REAL (COMP).LE.REAL (ESTIM).AND.	00027700
\$ AIMAG (COMP).LE.AIMAG (ESTIM)) GO TO 100	00027710
C SUBINTERVAL J HAS CAUSED TROUBLE.	00027720
C CHECK IF FURTHER SUBDIVISION SHOULD BE CARRIED OUT.	00027730
IF (N.EQ.NMAX) GO TO 60	00027740
BAD = 2*J - 1	00027750
RHS = .FALSE.	00027760
IF ((J-2*(J/2)).EQ.0) RHS = .TRUE.	00027770
GO TO 10	00027780
60 IC = -IABS (IC)	00027790

70	ZSUB2 = ZSUB2 + RESULT(M)	00027800
80	CONTINUE	00027810
	RELERR = RELERR + COMP	00027820
90	IF(OUT-2) 20,30,40	00027830
C	RELAXED CONVERGENCE	00027840
100	IC = ISIGN(2,IC)	00027850
	GO TO 70	00027860
	END	00027870
	COMPLEX FUNCTION ZSUBA1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)	00027880
	COMPLEX RELERR,F,RESULT,ESTIM,COMP	00027890
C	THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION	00027900
C	OVER A FINITE INTERVAL USING THE BASIC INTEGRATION	00027910
C	ALGORITHM ZQUAD1 TOGETHER WITH, IF NECESSARY AN ADAPTIVE	00027920
C	SUBDIVISION PROCESS. IT IS GENERALLY MORE EFFICIENT THAN	00027930
C	THE NON-ADAPTIVE ALGORITHM ZSUB1 BUT IS LIKELY TO BE LESS	00027940
C	RELIABLE(SEE COMP.J.,14,189,1971).	00027950
	DIMENSION RESULT(8), STACK(100)	00027960
	EXTERNAL F	00027970
	DATA ISMAX/100/	00027980
	CALL ZQUAD1(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)	00027990
	ZSUBA1 = RESULT(K)	00028000
	RELERR = (0.0,0.0)	00028010
	IF(REAL(ZSUBA1).NE.0.0.AND.AIMAG(ZSUBA1).NE.0.0) RELERR=	00028020
	\$ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUBA1),	00028030
	\$ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUBA1))	00028040
C	CHECK IF SUBDIVISION IS NEEDED	00028050
	IF (ICHECK.EQ.0) RETURN	00028060
C	SUBDIVIDE	00028070
	ESTIM=ZSUBA1*EPSIL	00028080
	ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))	00028090
	RELERR = (0.0,0.0)	00028100
	ZSUBA1 = (0.0,0.0)	00028110
	IS = 1	00028120
	IC = 1	00028130
	SUB1 = A	00028140
	SUB3 = B	00028150
10	SUB2 = (SUB1+SUB3)*0.5	00028160
	CALL ZQUAD1(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)	00028170
	NPTS = NPTS + NF	00028180
	IF(NPTS.GE.MEV) GO TO 50	00028190
	COMP = (RESULT(K)-RESULT(K-1))	00028200
	COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))	00028210
	IF (ICHECK.EQ.0) GO TO 30	00028220
	IF(REAL(COMP).LE.REAL(ESTIM).AND.	00028230
	\$ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 70	00028240
	IF (IS.GE.ISMAX) GO TO 20	00028250
C	STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION	00028260
	STACK(IS) = SUB1	00028270
	IS = IS + 1	00028280
	STACK(IS) = SUB2	00028290
	IS = IS + 1	00028300

<p>GO TO 40</p> <p>20 IC = -IABS(IC)</p> <p>30 ZSUBA1 = ZSUBA1 + RESULT(K)</p> <p>RELERR = RELERR + COMP</p> <p>40 CALL ZQUAD1(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)</p> <p>NPTS = NPTS + NF</p> <p>IF(NPTS.GE.MEV) GO TO 50</p> <p>COMP = (RESULT(K)-RESULT(K-1))</p> <p>COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))</p> <p>IF (ICHECK.EQ.0) GO TO 50</p> <p>IF(REAL(COMP).LE.REAL(ESTIM).AND.</p> <p>\$ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 80</p> <p>C SUBDIVIDE INTERVAL (SUB2,SUB3)</p> <p>SUB1 = SUB2</p> <p>GO TO 10</p> <p>50 ZSUBA1 = ZSUBA1 + RESULT(K)</p> <p>RELERR = RELERR + COMP</p> <p>IF(NPTS.GE.MEV) RETURN</p> <p>IF (IS.EQ.1) GO TO 60</p> <p>C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED</p> <p>IS = IS - 1</p> <p>SUB3 = STACK(IS)</p> <p>IS = IS - 1</p> <p>SUB1 = STACK(IS)</p> <p>GO TO 10</p> <p>C SUBDIVISION RESULT</p> <p>60 ICHECK = IC</p> <p>IF(REAL(ZSUBA1).EQ.0.0) GO TO 62</p> <p>IF(AIMAG(ZSUBA1).EQ.0.0) GO TO 64</p> <p>RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA1)),</p> <p>\$ AIMAG(RELERR)/ABS(AIMAG(ZSUBA1)))</p> <p>RETURN</p> <p>62 IF(AIMAG(ZSUBA1).EQ.0.0) GO TO 66</p> <p>RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUBA1)))</p> <p>RETURN</p> <p>64 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA1)),0.0)</p> <p>RETURN</p> <p>66 RELERR=(0.0,0.0)</p> <p>RETURN</p> <p>C RELAXED CONVERGENCE</p> <p>70 IC = ISIGN(2,IC)</p> <p>GO TO 30</p> <p>80 IC = ISIGN(2,IC)</p> <p>GO TO 50</p> <p>END</p>	<p>00028310</p> <p>00028320</p> <p>00028330</p> <p>00028340</p> <p>00028350</p> <p>00028360</p> <p>00028370</p> <p>00028380</p> <p>00028390</p> <p>00028400</p> <p>00028410</p> <p>00028420</p> <p>00028430</p> <p>00028440</p> <p>00028450</p> <p>00028460</p> <p>00028470</p> <p>00028480</p> <p>00028490</p> <p>00028500</p> <p>00028510</p> <p>00028520</p> <p>00028530</p> <p>00028540</p> <p>00028550</p> <p>00028560</p> <p>00028570</p> <p>00028580</p> <p>00028590</p> <p>00028600</p> <p>00028610</p> <p>00028620</p> <p>00028630</p> <p>00028640</p> <p>00028650</p> <p>00028660</p> <p>00028670</p> <p>00028680</p> <p>00028690</p> <p>00028700</p> <p>00028710</p> <p>00028720</p> <p>00028730</p> <p>00028740</p> <p>00028750</p>
<p>COMPLEX FUNCTION ZSUBA2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)</p> <p>COMPLEX RELERR,F,RESULT,ESTIM,COMP</p> <p>C THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION</p> <p>C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION</p> <p>C ALGORITHM ZQUAD2 TOGETHER WITH, IF NECESSARY AN ADAPTIVE</p> <p>C SUBDIVISION PROCESS. IT IS GENERALLY MORE EFFICIENT THAN</p>	<p>00028760</p> <p>00028770</p> <p>00028780</p> <p>00028790</p> <p>00028800</p> <p>00028810</p>

C THE NON-ADAPTIVE ALGORITHM ZSUB2 BUT IS LIKELY TO BE LESS	00028820
C RELIABLE(SEE COMP.J.,14,189,1971).	00028830
DIMENSION RESULT(8), STACK(100)	00028840
EXTERNAL F	00028850
DATA ISMAX/100/	00028860
CALL ZQUAD2(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)	00028870
ZSUBA2 = RESULT(K)	00028880
RELERR = (0.0,0.0)	00028890
IF(REAL(ZSUBA2).NE.0.0.AND.AIMAG(ZSUBA2).NE.0.0) RELERR=	00028900
\$ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUBA2),	00028910
\$ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUBA2))	00028920
C CHECK IF SUBDIVISION IS NEEDED	00028930
IF (ICHECK.EQ.0) RETURN	00028940
C SUBDIVIDE	00028950
ESTIM=ZSUBA2*EPSIL	00028960
ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))	00028970
RELERR = (0.0,0.0)	00028980
ZSUBA2 = (0.0,0.0)	00028990
IS = 1	00029000
IC = 1	00029010
SUB1 = A	00029020
SUB3 = B	00029030
10 SUB2 = (SUB1+SUB3)*0.5	00029040
CALL ZQUAD2(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)	00029050
NPTS = NPTS + NF	00029060
IF(NPTS.GE.MEV) GO TO 50	00029070
COMP = (RESULT(K)-RESULT(K-1))	00029080
COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))	00029090
IF (ICHECK.EQ.0) GO TO 30	00029100
IF(REAL(COMP).LE.REAL(ESTIM).AND.	00029110
\$ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 70	00029120
IF (IS.GE.ISMAX) GO TO 20	00029130
C STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION	00029140
STACK(IS) = SUB1	00029150
IS = IS + 1	00029160
STACK(IS) = SUB2	00029170
IS = IS + 1	00029180
GO TO 40	00029190
20 IC = -IABS(IC)	00029200
30 ZSUBA2 = ZSUBA2 + RESULT(K)	00029210
RELERR = RELERR + COMP	00029220
40 CALL ZQUAD2(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)	00029230
NPTS = NPTS + NF	00029240
IF(NPTS.GE.MEV) GO TO 50	00029250
COMP = (RESULT(K)-RESULT(K-1))	00029260
COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))	00029270
IF (ICHECK.EQ.0) GO TO 50	00029280
IF(REAL(COMP).LE.REAL(ESTIM).AND.	00029290
\$ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 80	00029300
C SUBDIVIDE INTERVAL (SUB2,SUB3)	00029310
SUB1 = SUB2	00029320
GO TO 10	00029330

50 ZSUBA2 = ZSUBA2 + RESULT(K)	00029340
RELERR = RELERR + COMP	00029350
IF(NPTS.GE.MEV) RETURN	00029360
IF (IS.EQ.1) GO TO 60	00029370
C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED	00029380
IS = IS - 1	00029390
SUB3 = STACK(IS)	00029400
IS = IS - 1	00029410
SUB1 = STACK(IS)	00029420
GO TO 10	00029430
C SUBDIVISION RESULT	00029440
60 ICHECK = IC	00029450
IF-REAL(ZSUBA2).EQ.0.0) GO TO 62	00029460
IF-AIMAG(ZSUBA2).EQ.0.0) GO TO 64	00029470
RELERR=CMPLX-REAL(RELERR)/ABS-REAL(ZSUBA2)),	00029480
\$ AIMAG(RELERR)/ABS-AIMAG(ZSUBA2)))	00029490
RETURN	00029500
62 IF-AIMAG(ZSUBA2).EQ.0.0) GO TO 66	00029510
RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS-AIMAG(ZSUBA2)))	00029520
RETURN	00029530
64 RELERR=CMPLX-REAL(RELERR)/ABS-REAL(ZSUBA2)),0.0)	00029540
RETURN	00029550
66 RELERR=(0.0,0.0)	00029560
RETURN	00029570
C RELAXED CONVERGENCE	00029580
70 IC = ISIGN(2,IC)	00029590
GO TO 30	00029600
80 IC = ISIGN(2,IC)	00029610
. GO TO 50	00029620
END	00029630