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)

$$Z = Q(r_{\alpha a}) - Q(r_{\alpha b}) + Q(r_{\beta b}) - Q(r_{\beta a})$$

$$+ \cos(\theta) \int_a^b \int_\alpha^\beta P(r)\,dS\,ds, \qquad (1)$$

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

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

- Z — EM coupling or mutual impedance,
- i — $=\sqrt{(-1)}$ ,
- g — Hankel transform variable,
- $\alpha, \beta$ — 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,
- $\theta$ — 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 = $r/\delta_1$ ,

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

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

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

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

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

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

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

$f_3(g) = gV_1(g)(1-F_1(g))/[(g+V_1(g)F_1(g))(g+V_1(g))]$ , and

$f_7(g) = iV_1(g)(L_1(g)-1)/g +$
$2V_1(g)(1-F_1(g))/[(g+V_1(g)F_1(g))(g+V_1(g))]$ .

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

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

where $Z_S$ is the mutual impedance for this configuration,

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

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

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 $J_o$ 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

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

The relation between the transform integral and the convolution integral becomes

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

where f(g) is the kernel function, and

h(gB) is the transform function (either cosine or $J_o$ Bessel function).

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

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

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*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,

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

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.

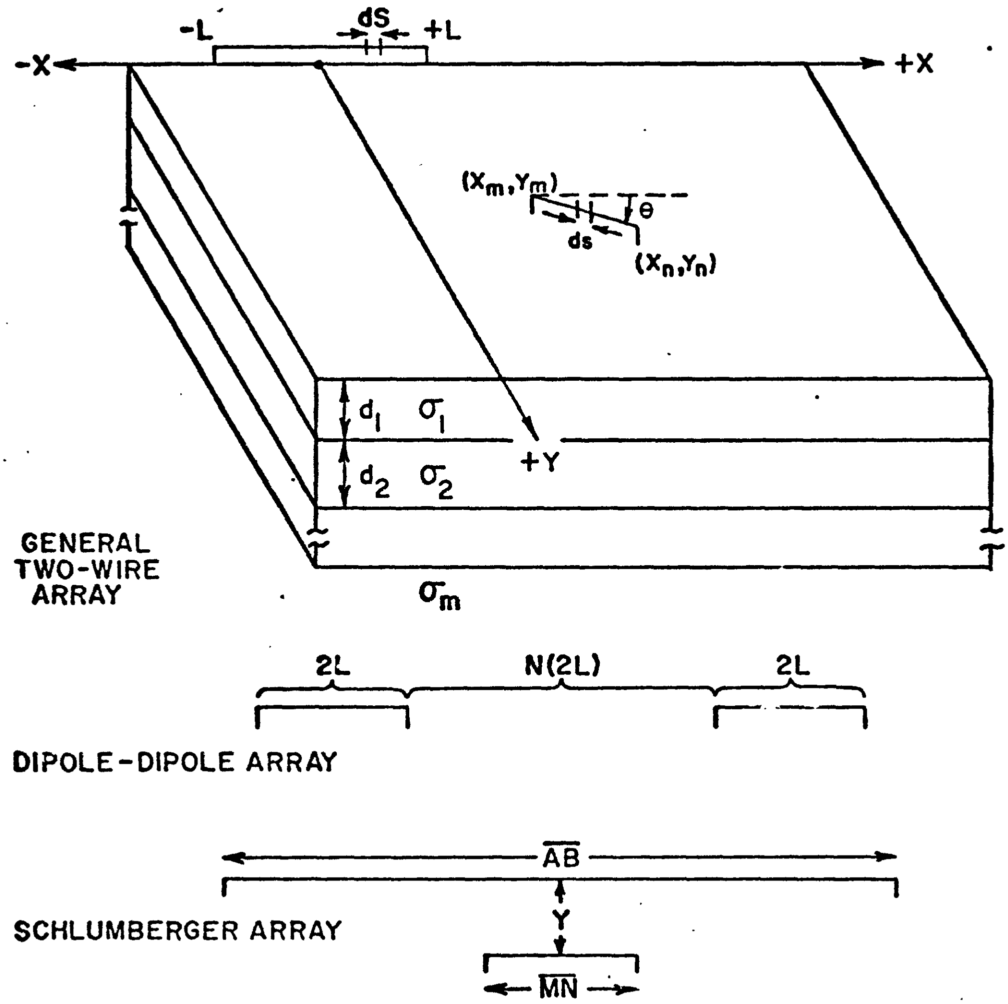


Fig. 1. Diagram defining general, dipole-dipole, and Schlumberger wire geometries and earth model parameters.

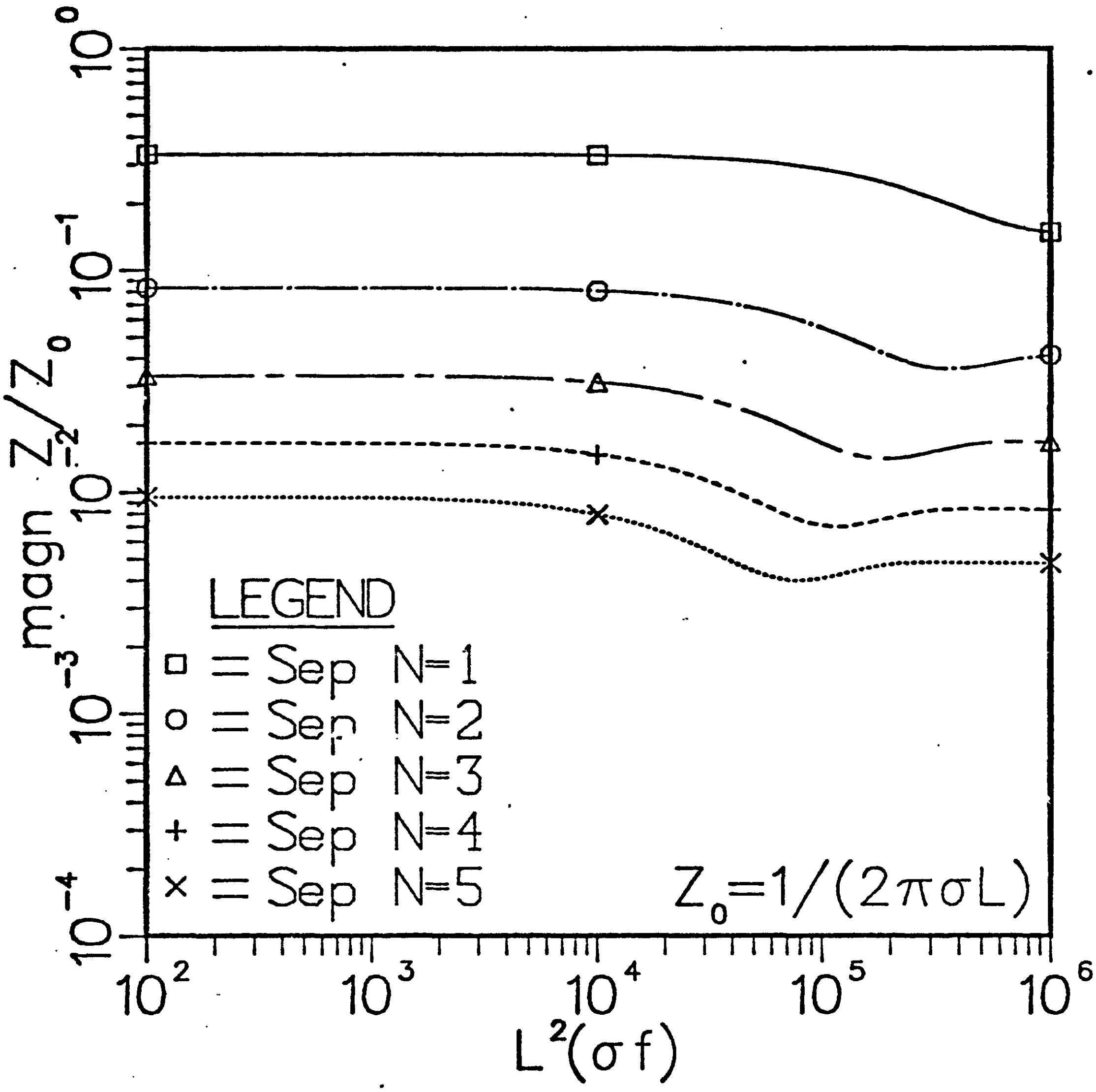


Fig. 2. Amplitude of normalized mutual impedance for a dipole-dipole configuration over a homogeneous halfspace (compare with Dey and Morrison, 1973, Figure 4).

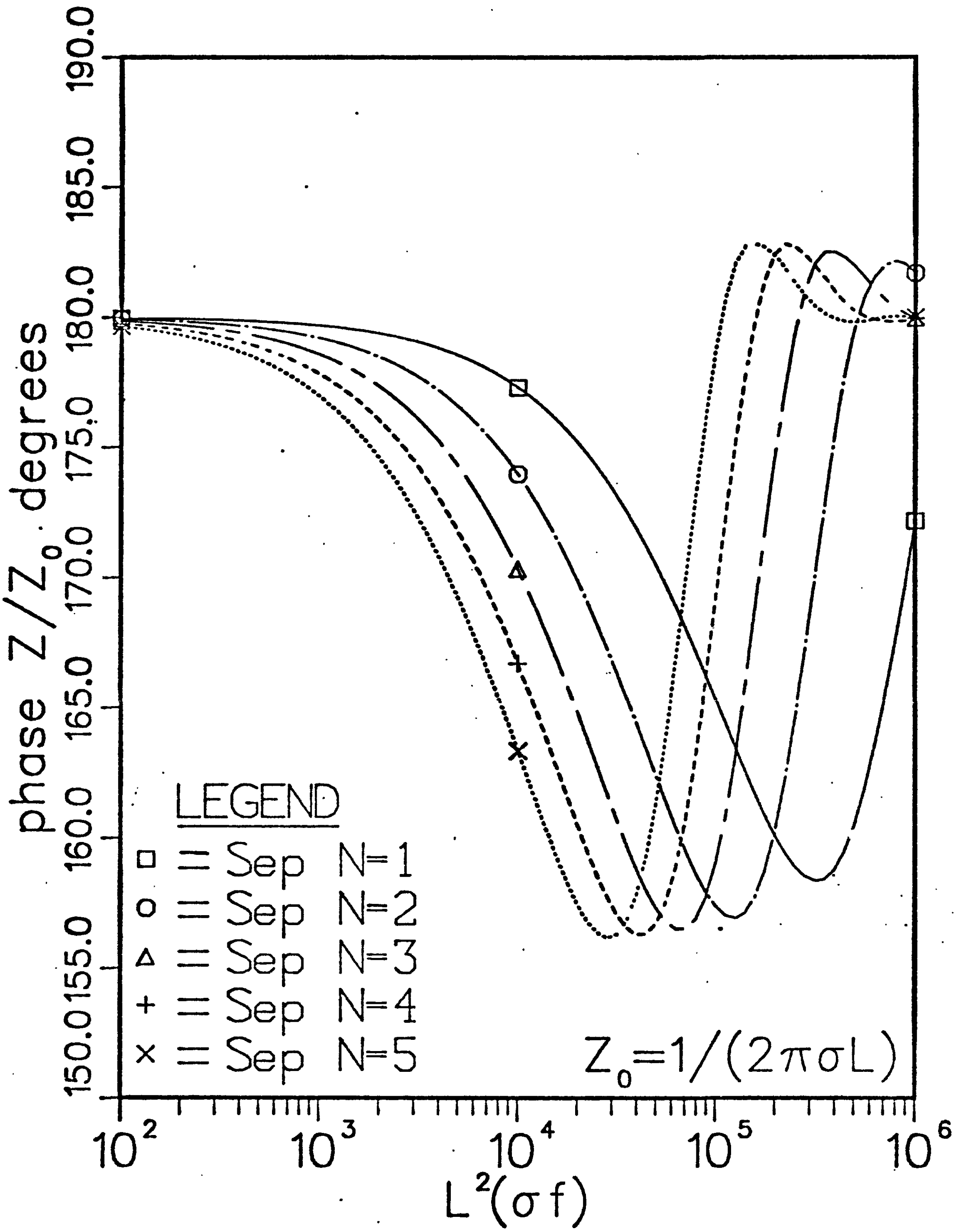


Fig. 3. Phase of normalized mutual impedance for a dipole-dipole configuration over a homogeneous halfspace (compare with Dey and Morrison, 1973, Figure 5).

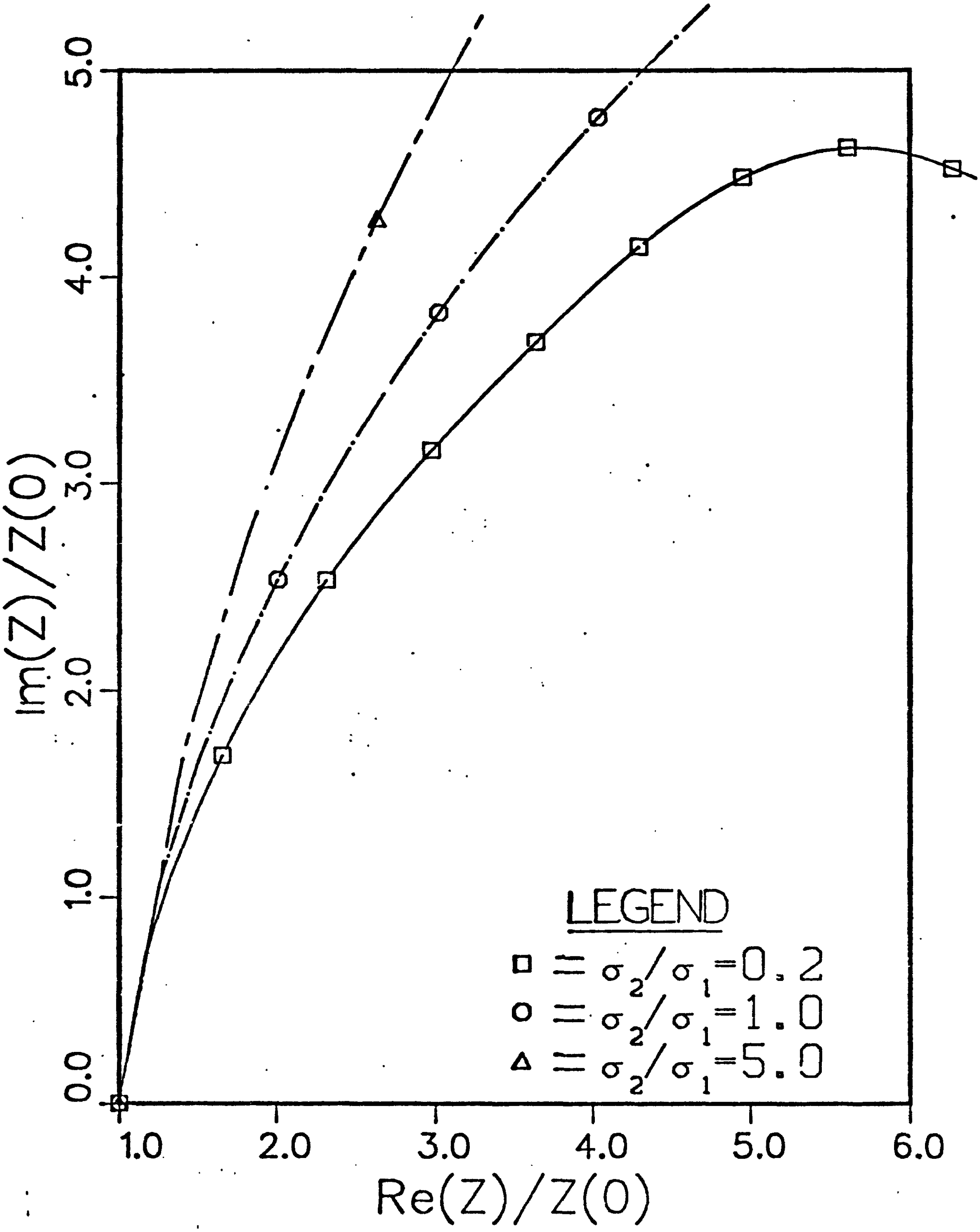


Fig. 4. EM coupling for the Wenner array as a function of resistivity of a two-layer earth with all other parameters held constant (compare with Wynn and Zonge, 1977, Figure 2). Source wire is 450 m long and receiver wire is 150 m long and 75 m away from the source wire. Conductivities for the first 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, |
| * | FO,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:

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

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

sig1(omega) = sig1(0)/[1 - m*( 1 - 1/(1 + (i*omega*tau)**c))],

where sig1(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 $PARMS PARAMETERS$
$PARMS M=1,ICMPLX=1,L=50,YM=0,YN=0,XM=150,XN=250,FM=100,FO=.01,NF=4$

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 |
| ZFOURO | | 00018780 |
| ZHANKO | | 00020770 |
| ZLAGHO | | 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)
C--COMPUTES FINITE INTEGRAL OVER (RMIN,RMAX) OF
C  COMPLEX FUNCTION FUNINT (SPLINE INTERPOLATOR)
C  ASSUMES PRIOR CALL TO SETSPL ...........
      REAL L
      EXTERNAL FUNINT
      COMPLEX ESUM,ZSUBA2,ZSUB2
      COMMON/FIN/R1,R2,R,L,SIG1,XX,YY
      COMMON/CORRF/Y,RMIN,RMAX
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW
      XX=X
      YY=Y
C  ADAPTIVE GAUSSIAN QUADRATURE INTEGRATION
      IF(INTYPE.EQ.1) CORFUN=ZSUBA2(RMIN-X,RMAX-X,FINTOL,NEV,ICK,ESUM,
     1 FUNINT,MEV)
C  NON-ADAPTIVE GAUSSIAN QUADRATURE INTEGRATION
      IF(INTYPE.EQ.2) CORFUN=ZSUB2(RMIN-X,RMAX-X,FINTOL,NEV,ICK,ESUM,
     1 FUNINT,MEV)
      RETURN
      END

      REAL FUNCTION CSPLNT(T)
C--CUBIC SPLINE INTERPOLATOR OF INDEPENDENT VARIABLE Y
C  FOR DEPENDENT VARIABLE X.  IF
C         T >=X(N) CSPLNT=Y(N)
C         T <=X(1) CSPLNT=Y(1)
C--ASSUMES PRIOR CALL TO SPLIN1
      COMMON/CSPLINE/A(50),B(50),C(50),N,X(50),Y(50)
      IF(T.LT.X(1).OR.T.GT.X(N)) GO TO 2
      N1=N-1
      DO 1 I=1,N1
           J=I
           IF(T.LT.X(I+1)) GO TO 3
    1      CONTINUE
    2 IF(T.GT.X(N)) CS=Y(N)
      IF(T.LT.X(1)) CS=Y(1)
      CSPLNT=CS
      RETURN
    3 Z=T-X(J)
      CS=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z
      CSPLNT=CS
      RETURN
      END

      COMPLEX FUNCTION CZEX(B,NEW,R)
C--CZEX COMPUTES THE P(R) TERM WHICH IS
C  DOUBLE INTEGRATED OVER FINITE LIMITS.
C  IT IS PART OF THE EQUATION FOR THE
C  ELECTRIC FIELD OF AN ELECTRIC DIPOLE.
C
C      B   INDUCTION NUMBER
C      R   DISTANCE
```

```
C      NEW CONTROLS ZLAGHO INTEGRATION
C
C--CZEX IS IDENTICAL TO ZEX EXCEPT THAT
C  IT ALLOWS FOR COMPLEX CONDUCTIVITIES
C  WHICH ARE COMPUTED BY USER-DEFINED COMPLEX ROUTINE
C  SIGMA
      COMPLEX ZLAGHO,TWODEL3,ONE,SIGMA,SIGMA1,CB,CK
      EXTERNAL F3
      COMMON/CFLAG/CK(10),ICMPLX
      COMMON/PARM/ISTEP,A1,A2,A3,SIG1,A5,M,TOL
      COMMON/CONST/DEL,DEL2,TWODEL3
      DATA ONE/(1.0,0.0)/
      CZEX=CMPLX(0.0,0.0)
      CB=CMPLX(B,B)*CSQRT(CK(1))
      IF(M.EQ.1) GO TO 2
      CZEX=ZLAGHO(ALOG(B),F3,TOL,LW,NEW)/B
    2 CZEX=TWODEL3*CZEX+(ONE-(ONE+CB)*CEXP(-CB))/R**3
      RETURN
      END

C  *************** PROGRAM EMCUPL ****************
C--PROGRAM EMCUPL CALCULATES THE ELECTROMAGNETIC COUP-
C  LING BETWEEN TWO STRAIGHT GROUNDED WIRES OF
C  ARBITRARY LENGTH AND ORIENTATION ON THE SURFACE OF A LAYERED
C  EARTH.  ONE WIRE (REFERRED TO AS THE SOURCE WIRE) IS
C  CONSTRAINED TO LIE ALONG THE X-AXIS BETWEEN -L AND L.
C  THE OTHER WIRE (RECEIVER WIRE)
C  MUST BE SPECIFIED AS TWO PAIRS OF X,Y COORDINATES- ONE
C  PAIR FOR EACH WIRE END.  THE INPUT PARAMETERS ARE:
C
C  SIG(I)     CONDUCTIVITY OF THE ITH LAYER
C  D(I)       THICKNESS OF THE ITH LAYER
C  M          NUMBER OF LAYERS
C  XM,YM      COORDINATES OF ONE RECEIVER WIRE END
C  XN,YN      COORDINATES OF THE OTHER END
C  L          SOURCE WIRE HALFLENGTH
C  DX         INCREMENT FOR STEPPING RECEIVER POSITION
C  XMAX       MAXIMUM X-VALUE CONSIDERED
C  DY         INCREMENT FOR STEPPING RECEIVER POSITION
C  YMAX       MAXIMUM Y-VALUE CONSIDERED
C  NF         >0, NUMBER OF FREQUENCIES DESIRED PER DECADE
C             BETWEEN F0 AND FM
C             <0, NUMBER OF SPECIFIED FREQUENCIES IN FNF
C  F0,FM      MINIMUM AND MAXIMUM FREQUENCIES DESIRED
C  FNF        SPECIFIED FREQUENCIES
C  TOL        TOLERANCE FOR HANKEL TRANSFORM CALCULATIONS
C  FINTL1     TOLERANCE FOR INTEGRATION ALONG SOURCE WIRE
C  FINTL2     TOLERANCE FOR INTEGRATION ALONG RECEIVER WIRE
C  IN1,IN2    =1 FOR ADAPTIVE QUADRATURE INTEGRATION
C             =2 FOR NON-ADAPTIVE QUADRATURE INTEGRATION
C             IN1 IS FOR THE OUTER INTEGRAL ACROSS THE RECEIVER
C             IN2 IS FOR THE INNER INTEGRAL ACROSS THE SOURCE
```

```
C  NFIN      INTERVAL IN LOG-SPACE WITH WHICH THE SPLINE
C            NODES FOR FINITE WIRE INTEGRATION ARE
C            CALCULATED, E. G. INTERVAL=0.2/NFIN
C            (DEFAULT=1)
C  MEV1,MEV2 MAXIMUM NUMBER OF FUNCTION EVALUATIONS FOR
C            RESPECTIVE INTEGRATION ROUTINES
C  TMAX,TMIN MAXIMUM AND MINIMUM TIME VALUES DESIRED
C  TFLAG     =0 COMPUTES FREQUENCY RESPONSE ALONE
C            =1 COMPUTES FREQUENCY AND TRANSIENT RESPONSE
C            =2 COMPUTES TRANSIENT RESPONSE ALONE
C            =3 COMPUTES TRANSIENT RESPONSE FOR A
C            FREQUENCY RESPONSE PREVIOUSLY COMPUTED
C  RC        TIME CONSTANT OF SINGLE POLE LOW-PASS
C            FILTER TO BE CONVOLVED WITH FREQUENCY
C            RESPONSE FOR TRANSIENT RESPONSE
C            CALCULATIONS
C            NOTE:  TRANSIENT RESPONSE IS THE STEP
C            RESPONSE
C  ICMPLX    =0 COMPUTES THE COUPLING USING THE REAL
C               CONDUCTIVITIES IN SIG ARRAY,
C            =1 COMPUTES THE COUPLING USING THE COMPLEX
C               CONDUCTIVITIES COMPUTED BY THE USER-
C               DEFINED FUNCTION SIGMA(J,1./(DEL*DEL))
C  INFILE    (DEFAULT=5) INPUT FILE NUMBER
C  OUFILE    (DEFAULT=6) OUTPUT FILE NUMBER
C
C  PROGRAM ORGANIZATION IS AS FOLLOWS:
C
C          EMCUPL
C            *
C            *             DOUBLE       INTEGRATION      TRANSIENT CALC
C      *************************************************************
C      *        *            *        *      *           *        *
C    SIGMA    FINQDF        SETSPL  ZSUB1  ZSUBA1      SPLIN1   RLAGF1
C               *            *        *       *                   *
C               *         *******     *********                   *
C               *        *   *   *        *                       *
C             FINQ     ZEX  CZEX QUINT  FINFUN                  CSPLNT
C               *       *   * *           *
C               *       ***** *        *********
C               *         *   *        *       *
C            ZHANKO    ZLAGHO SIGMA  ZSUB2  ZSUBA2
C               *         *            *       *
C               *         *            *********
C               *         *               *
C              F7G        F3           FUNINT
C               *         *              *
C            RECUR2     RECURS         QPOINT
C
C
      COMMON/CFLAG/CK(10),ICMPLX
      COMMON/FINERR/HAKTOL,FINTL1,IN2,NFIN,NEV2,MEV2,ZERR2,LW
```

```
      COMMON/FIN/RMAX,RMIN,R0,L,SIG1,XX,YY
      COMMON/ENDS/XXM,YYM,XXN,YYN,DS
      COMMON/THICK/D(9)
      COMMON/MODEL/RK(10),DD(9),M
      COMMON/PARM/IS,A1,A2,A3,SSIG1,A5,NLYR,TOL
      COMMON/CSPLINE/D1(50),D2(50),D3(50),NN,BB(50),SPEC(50)
      INTEGER OUFILE
      REAL L,SIG(10),FNF(50),P(50),S(50),PS(2),T(50),V(50)
      COMPLEX ZSUB1,ZSUBA1,ZERR1,ZERR2,ECOPL,FINQDF,EC(30,30),
     1 CSPEC,SIGMA,SIGMA1,CK
      EXTERNAL ZEX,FINFUN,CSPLNT,CZEX
      NAMELIST/PARMS/SIG,M,D,XM,YM,XN,YN,L,TOL,FINTL1,FINTL2
     1,IN1,IN2,NFIN,MEV1,MEV2,FM,F0,NF,FNF,DX,DY,XMAX,YMAX,TMAX,TMIN,
     2 TFLAG,RC,ICMPLX,INFILE,OUFILE
      DATA DEG/57.29577951/
      DATA PS,TWOPI/0.0,0.0,6.283185308/
C
C--ASSIGN NAMELIST PARAMETER DEFAULTS
C
      INFILE=5
      OUFILE=6
      MEV1=300
      MEV2=300
      NFIN=1
      IN1=1
      IN2=1
      FINTL1=1.E-6
      FINTL2=1.E-4
      TOL=1.E-8
      M=1
      DX=0.0
      DY=0.0
      XMAX=0.0
      YMAX=0.0
      TFLAG=0
      ICMPLX=0
      RC=0.0
C
      WRITE(OUFILE,501)
  501 FORMAT(25H ENTER $PARMS PARAMETERS$)
  100 READ(INFILE,PARMS)
      IF(M.EQ.0) STOP
C--TFLAG = 3 ASSUMES THAT FREQUENCY VALUES HAVE ALREADY
C  BEEN COMPUTED IN A PREVIOUS RUN
      IF(TFLAG.EQ.3) GO TO 21
      IF(DX.EQ.0.0) XMAX=0.0
      IF(DY.EQ.0.0) YMAX=0.0
C
C--DEFINE EQUIVALENT COMMON PARAMETERS
C
      NLYR=M
      HAKTOL=TOL
```

```
C
      IF(ICMPLX.EQ.1) GO TO 300
      SIG1=SIG(1)
      DO 10 I=1,M
   10     RK(I)=SIG(I)/SIG1
      GO TO 301
  300 SIG1=REAL(SIGMA(1,0.0))
      RK(1)=SIG1
  301 M1=M-1
      SSIG1=SIG1
C
C--CHECK THAT ICMPLX = 0 OR 1 ONLY
C
      IF(ICMPLX.EQ.0.OR.ICMPLX.EQ.1) GO TO 302
      WRITE(OUFILE,502)
  502 FORMAT(34H ICMPLX MUST BE SET TO 0 OR 1 ONLY)
      GO TO 100
  302 TWL=2.*L
C
C--PRINT MODEL PARAMETERS
C
      WRITE(OUFILE,503) TWL,M
  503 FORMAT(/16H SOURCE LENGTH =,E15.3//5X,I3,12H LAYER MODEL)
      IF(ICMPLX.EQ.0) WRITE(OUFILE,504) (SIG(I),I=1,M)
  504 FORMAT(5H SIG=,12E12.4)
  505 FORMAT(5H D  =,12E12.4)
      IF(ICMPLX.EQ.1) WRITE(OUFILE,506)
  506 FORMAT(28H COMPLEX CONDUCTIVITIES USED)
      IF(M.GT.1) WRITE(OUFILE,505) (D(I),I=1,M1)
C--MAKE SURE XM IS LESS THAN XN
      IF(XM.LT.XN) GO TO 1
      X=XM
      XM=XN
      XN=X
      Y=YM
      YM=YN
      YN=Y
C
C--CALCULATE MAXIMUM AND MINIMUM DISTANCE BETWEEN WIRES
C  IN ORDER TO DEFINE ARGUMENT RANGE FOR ZEX OR CZEX
C  EVALUATIONS
C
    1 Y2=YM*YM
      R1=SQRT((XM+L)**2+Y2)
      R2=SQRT((XM-L)**2+Y2)
      Y2=YN*YN
      R3=SQRT((XN+L)**2+Y2)
      R4=SQRT((XN-L)**2+Y2)
      RMAX=(L+AMAX1(ABS(XN),ABS(XM),ABS(XMAX)))**2
      RMAX=SQRT(RMAX+AMAX1(ABS(YN),ABS(YM),ABS(YMAX))**2)
      RMIN=AMIN1(R1,R2,R3,R4)
      XX=XM+0.5*(XN-XM)
```

```
      CXX=XX
      YY=YM+0.5*(YN-YM)
      RO=SQRT(XX*XX+YY*YY)
      IF(XN.NE.XM) DS=(YN-YM)/(XN-XM)
      CDS=COS(ATAN(DS))
    3 ITRUE=0
      IF(ABS(XM).LE.L) ITRUE=1
      IF(ABS(XN).LE.L) ITRUE=ITRUE+2
      IF(ITRUE.EQ.0) GO TO 6
      IF(ITRUE.LT.3) GO TO 5
    4 RMIN=AMIN1(YM,YN)
      GO TO 6
    5 IF((ITRUE.EQ.1.AND.YN.LT.YM).OR.(ITRUE.EQ.2.AND.YM.LT.YN)) GO TO 4
      RL=L*SIGN(1.,ITRUE-1.5)
      XP=(-YM/DS)+XM-RL
      YP=-(XP+RL)/DS
      RMIN=SQRT(XP*XP+YP*YP)
    6 CONTINUE
      CON=-1./(SIG1*6.283185308)
      XB=SQRT(1./(SIG1*3.9478417E-6))
      NX=0
      NY=0
      IF(DX.NE.0.0) NX=1+(XMAX-AMAX1(XM,XN))/DX
      IF(DY.NE.0.0) NY=1+(YMAX-AMAX1(YM,YN))/DY
      IF(NF.LT.0) GO TO 13
      NN=NF*ALOG10(FM/F0)+1
      F=F0
      DELX=EXP(2.30258509/FLOAT(NF))
      GO TO 14
   13 NN=-NF
C
C--FIRST LOOP OVER FREQUENCIES
C
   14 DO 20 JJ=1,NN
          IF(NF.GT.0) GO TO 11
          F=FNF(JJ)
          GO TO 12
   11     IF(JJ.GT.1) F=F*DELX
   12     BB(JJ)=F
          DEL=XB/SQRT(F)
C--COMPUTE COMPLEX SIGMA1
          CF=1./(DEL*DEL)
          IF(ICMPLX.EQ.1) SIGMA1=SIGMA(1,CF)/SIG1
          DO 15 I=1,M
              IF(ICMPLX.EQ.1) CK(I)=SIGMA(I,CF)/SIG1
   15         DD(I)=2.*D(I)/DEL
          IPOS=0
C--SET SPLINE COEFFICIENTS UNLESS WIRES ARE
C  PERPENDICULAR
          IF(XN.EQ.XM) GO TO 16
          IF(ICMPLX.EQ.0) CALL SETSPL(ZEX,DEL,RMAX,RMIN)
          IF(ICMPLX.EQ.1) CALL SETSPL(CZEX,DEL,RMAX,RMIN)
```

```
  16     NEW=1
C
C--LOOP OVER X COORDINATE FIRST, THEN Y COORDINATE
C
         DO 20 IX=1,NX
             XXM=XM+DX*(IX-1)
             XXN=XN+DX*(IX-1)
             DO 20 IY=1,NY
                 YYM=YM+DY*(IY-1)
                 YYN=YN+DY*(IY-1)
                 Y2=YYM*YYM
                 R1=SQRT((XXM+L)**2+Y2)
                 R2=SQRT((XXM-L)**2+Y2)
                 Y2=YYN*YYN
                 R3=SQRT((XXN+L)**2+Y2)
                 R4=SQRT((XXN-L)**2+Y2)
  18             IPOS=IPOS+1
                 ECOPL=CMPLX(0.0,0.0)
                 IF(XXN.EQ.XXM) GO TO 19
C--CALCULATE WIRE COUPLING (DOUBLE INTEGRAL) UNLESS WIRES ARE
C  PERPENDICULAR
                 IF(DS.EQ.0.0.AND.CXX.EQ.0.0) GO TO 24
                 IF(IN1.EQ.1) ECOPL=CDS*ZSUBA1(XXM,XXN,FINTL1,NEV1,ICK,
     1           ZERR1,FINFUN,MEV1)
                 IF(IN1.EQ.2) ECOPL=CDS*ZSUB1(XXM,XXN,FINTL1,NEV1,ICK,
     1           ZERR1,FINFUN,MEV1)
                 GO TO 25
  24             IF(IN1.EQ.1) ECOPL=2.*CDS*ZSUBA1(0.0,XXN,FINTL1,NEV1,
     1           ICK,ZERR1,FINFUN,MEV1)
                 IF(IN1.EQ.2) ECOPL=2.*CDS*ZSUB1(0.0,XXN,FINTL1,NEV1,
     1           ICK,ZERR1,FINFUN,MEV1)
  25             IF(MEV1.GE.NEV1-1.AND.ICK.GE.0) GO TO 19
                 WRITE(OUFILE,520) NEV1,MEV1,ICK,BB(JJ),IX,IY
 520             FORMAT(40H GAUSS QUADRATURE: COMPUTED INTEGRAL MAY,
     1           13H BE ERRONEOUS/6H NEV1=,I4,6H MEV1=,I4,5H ICK=,I2,
     2           6H FREQ=,E15.4,9H POS: IX=,I3,4H IY=,I3/)
  19             EC(JJ,IPOS)=CON*(ECOPL-FINQDF(DEL,R4,R3,R2,R1,
     1           FINTL1,NEW))
                 IF(ICMPLX.EQ.0) GO TO 20
                 EC(JJ,IPOS)=EC(JJ,IPOS)/SIGMA1
  20             NEW=0
      IF(TFLAG.EQ.0) GO TO 22
  21  NT=AINT(5.*ALOG(TMAX/TMIN))+1
      NT1=NT+1
C--CALCULATE IN NORMALIZED TIME
C  NORM TIME =TWOPI * REAL TIME
      X0=ALOG(TMAX*TWOPI)+0.2
  22  IPOS=0
C
C--PRINT OUTPUT
C
      DO 41 IX=1,NX
```

```
         XA=XM+DX*(IX-1)
         XB=XN+DX*(IX-1)
         DO 41 IY=1,NY
             YA=YM+DY*(IY-1)
             YB=YN+DY*(IY-1)
             WRITE(OUFILE,507) XA,YA,XB,YB
  507        FORMAT(/25H RECEIVER ELECTRODES AT (,E10.4,1H,,E10.4,1H)/
     1        24X,1H(,E10.4,1H,,E10.4,1H)/)
             IPOS=IPOS+1
             IF(TFLAG.GT.1) GO TO 49
C
C--PRINT FREQUENCY RESPONSE COMPUTATION
C
             WRITE(OUFILE,200)
  200        FORMAT(4X,9HFREQUENCY,5X,4HREAL,8X,4HIMAG,8X,4HMAGN,8X,
     1       4HPFE ,8X,5HPHASE)
             PFE0=CABS(EC(1,IPOS))
             DO 40 IF=1,NN
                 AMP=CABS(EC(IF,IPOS))
                 PFE=100.*(1.-AMP/PFE0)
                 PHZ=DEG*ATAN2(AIMAG(EC(IF,IPOS)),REAL(EC(IF,IPOS)))
   40            WRITE(OUFILE,509) BB(IF),EC(IF,IPOS),AMP,PFE,PHZ
  509        FORMAT(1H ,6E12.4)
             IF(TFLAG.EQ.0) GO TO 41
C
C--COMPUTE THE STEP TRANSIENT RESPONSE USING A CUBICALLY
C  SPLINED FREQUENCY RESPONSE FUNCTION
C
   49        DC=REAL(EC(1,IPOS))
             DO 50 II=1,NN
C
C--MULTIPLY THE FREQUENCY RESPONSE (DC VALUE SUBTRACTED)
C  BY THE SOURCE FUNCTION (1/BB FOR STEP RESPONSE)
C
                 SPEC(II)=(REAL(EC(II,IPOS))-DC)/BB(II)
                 IF(RC.EQ.0.0) GO TO 50
C
C--MULTIPLY FREQUENCY RESPONSE BY LOW-PASS
C  FILTER TRANSFER FUNCTION IF RC GT 0
C
                 TCON=BB(II)*RC
                 CSPEC=EC(II,IPOS)*CMPLX(1.,-TCON)
                 SPEC(II)=(REAL(CSPEC)-DC)/BB(II)/(1.+TCON*TCON)
   50            CONTINUE
             CALL SPLIN1(NN,0,BB,SPEC,D1,D2,D3,0,PS,P,S)
             NEW=1
             WRITE(OUFILE,510)
  510        FORMAT(/6X,9HTIME(SEC),4X,15HOBS VOLTAGE/AMP)
C
C--COMPUTE SINE TRANSFORM (RLAGF1) OF CUBICALLY SPLINED
C  FREQUENCY FUNCTION (SPLINE INTERPOLATOR-CSPLNT)
C
```

```
            DO 60 J=1,NT
                I=NT1-J
                X=X0-0.2*J
                T(I)=EXP(X)
                FDC=1.0
                IF(RC.GT.0.0) FDC=1.-EXP(-T(I)/(TWOPI*RC))
                V(I)=0.636619772*RLAGF1(X,CSPLNT,TOL,LW,NEW)/T(I)+
     1          DC*FDC
                T(I)=T(I)/TWOPI
                WRITE(OUFILE,509) T(I),V(I)
   60           NEW=0
   41       WRITE(OUFILE,511)
  511 FORMAT(1H )
      WRITE(OUFILE,512)
  512 FORMAT(27H ENTER $PARMS CHANGES ONLY$)
      GO TO 100
      END

      COMPLEX FUNCTION F2MOD(G)
C--KERNEL FUNCTION FOR SINE INTEGRAL IN ROUTINE INFNEX
C--CALLS RECURS
      COMPLEX V1,F1,C
      CALL RECURS(G,V1,F1)
      C=G
      F2MOD=CMPLX(1.0,0.0)/(C+V1*F1)
      RETURN
      END

      COMPLEX FUNCTION F3(G)
      COMPLEX V1,F1,C,ONE
      DATA ONE/(1.0,0.0)/
      CALL RECURS(G,V1,F1)
      C=G
      F3=(V1*C*(ONE-F1))/((C+V1*F1)*(C+V1))
      RETURN
      END

      COMPLEX FUNCTION F7G(G)
C--KERNEL OF HANKEL TRANSFORM USED BY
C  ROUTINES FINQ AND SCHCOPL
C--CALLS RECUR2
      COMPLEX V1,F1,L1,I1,ONE,TWO,C
      DATA I1,ONE,TWO/(0.0,1.0),(1.0,0.0),(2.0,0.0)/
      CALL RECUR2(G,V1,F1,L1)
      C=G
      F7G=I1*V1*(L1-ONE)+(TWO*V1*(ONE-F1))/((C+V1*F1)*(C+V1))
      F7G=F7G/G
      RETURN
      END

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

```
C  (L PASSED IN THROUGH COMMON AREA FIN) OF
C  COMPLEX FUNCTION FUNINT (SPLINE INTERPOLATOR)
C  AT FIELD POINT (XX,YY).
C  ASSUMES PRIOR CALL TO SETSPL ...........
C--CALLS FUNINT
C        ZSUBA2 - ADAPTIVE GAUSSIAN INTEGRATION
C        ZSUB2  - NON-ADAPTIVE GAUSSIAN INTEGRATION
      REAL L
      EXTERNAL FUNINT
      COMPLEX ESUM,ZSUBA2,ZSUB2
      COMMON/FIN/R1,R2,R,L,SIG1,XX,YY
      COMMON/ENDS/XM,YM,XN,YN,DS
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW
C  CHECK TO SEE THAT X IS IN THE RANGE XM -> XN
      IF(X.LE.XN.AND.X.GE.XM) GO TO 1
      WRITE(6,100) X
  100 FORMAT(34H FINFUN: X NOT IN PROPER RANGE, X=,E12.4)
      STOP
    1 XX=X
      YY=YM+DS*(X-XM)
      IF(ABS(X).LT.L) GO TO 8
      IF(INTYPE.EQ.1) FINFUN=ZSUBA2(X-L,X+L,FINTOL,NEV,ICK,ESUM,FUNINT,
     1 MEV)
      IF(INTYPE.EQ.2) FINFUN=ZSUB2(X-L,X+L,FINTOL,NEV,ICK,ESUM,FUNINT,
     1 MEV)
      GO TO 10
    8 XMIN=AMIN1(ABS(X-L),ABS(X+L))
      XMAX=AMAX1(ABS(X-L),ABS(X+L))
      IF(INTYPE.EQ.1) FINFUN=2.*ZSUBA2(0.,XMIN,FINTOL,NEV,ICK,ESUM,
     1 FUNINT,MEV)
      IF(INTYPE.EQ.2) FINFUN=2.*ZSUB2(0.,XMIN,FINTOL,NEV,ICK,ESUM,
     1 FUNINT,MEV)
      IF(X.EQ.0.0) GO TO 10
      IF(INTYPE.EQ.1) FINFUN=FINFUN+ZSUBA2(XMIN,XMAX,FINTOL,NEV1,ICK,
     1 ESUM,FUNINT,MEV)
      IF(INTYPE.EQ.2) FINFUN=FINFUN+ZSUB2(XMIN,XMAX,FINTOL,NEV1,ICK,
     1 ESUM,FUNINT,MEV)
      NEV=NEV+NEV1
   10 RETURN
      END

      COMPLEX FUNCTION FINQ(DEL,R,TOL)
C--FINQ CALCULATES THE J0 HANKEL TRANSFORM
C  (USING DIGITAL FILTER ROUTINE ZHANKO)
C  REQUIRED BY PROGRAM EMCUPL
C--CALLS ZHANKO, F7G
      COMMON/MODEL/RK(10),D(9),M
      COMPLEX ZHANKO,ES
      EXTERNAL F7G
      B=R/DEL
      FINQ=CMPLX(0.0,0.0)
      IF(M.EQ.1) GO TO 1
```

```
      FINQ=ZHANKO(ALOG(B),F7G,TOL,LW)/B
      FINQ=FINQ*CMPLX(0.0,1./DEL)
    1 FINQ=FINQ-1./R
      RETURN
      END

      COMPLEX FUNCTION FINQDF(DEL,R1,R2,R3,R4,TOL,NEW)
C--COMPUTES FINQ AT FOUR SEPARATIONS USING
C  PREVIOUS CALCULATED RESULTS IF NEW=0
C    DEL   SKIN DEPTH IN FIRST LAYER
C    R1,R2,R3,R4  THE FOUR SEPARATIONS
C    NEW   = 1    FIRST CALL FOR A MODEL
C          = 0    SUBSEQUENT CALL WHERE PREVIOUS
C                 RESULTS MAY BE USED
C
C    FINQDF = FINQ(R1)-FIN1(R2) - (FINQ(R3)-FINQ(R4))
C
C--CALLS FINQ
      COMPLEX FINQ,Q1,Q2,Q3,Q4,ZERO,QOLD(4)
      REAL ROLD(4)
      INTEGER ISET(4)
      COMMON/SAVEQ/ROLD,QOLD
      DATA ZERO/(0.0,0.0)/
      FINQDF=ZERO
      Q1=ZERO
      Q2=ZERO
      Q3=ZERO
      Q4=ZERO
      DO 1 I=1,4
    1      ISET(I)=0
      IF(NEW.EQ.0) GO TO 10
      DO 2 I=1,4
           ROLD(I)=0E0
    2      QOLD(I)=ZERO
           GO TO 100
C  CHECK TO SEE IF Q VALUE WAS CALCULATED
C  IN PREVIOUS CALL TO FINQDF
   10 DO 20 I=1,4
           IF(ROLD(I).NE.R1) GO TO 11
           Q1=QOLD(I)
           ISET(1)=I+1
           GO TO 20
   11      IF(ROLD(I).NE.R2) GO TO 12
           Q2=QOLD(I)
           ISET(2)=I+1
           GO TO 20
   12      IF(ROLD(I).NE.R3) GO TO 13
           Q3=QOLD(I)
           ISET(3)=I+1
           GO TO 20
   13      IF(ROLD(I).NE.R4) GO TO 20
           Q4=QOLD(I)
```

```
          ISET(4)=I+1
 20       CONTINUE
 100 IF(R1.EQ.R2) GO TO 150
     IF(R1.EQ.R3) GO TO 400
 101 IF(ISET(1).GT.0) GO TO 110
     ISET(1)=1
     Q1=FINQ(DEL,R1,TOL)
 110 IF(ISET(2).GT.0) GO TO 120
     ISET(2)=1
     Q2=FINQ(DEL,R2,TOL)
 120 FINQDF=Q1-Q2
 150 IF(R3.EQ.R4) GO TO 300
     IF(ISET(3).GT.0) GO TO 250
     IF(R3.NE.R1.OR.ISET(1).EQ.0) GO TO 155
     Q3=Q1
     GO TO 250
 155 IF(R3.NE.R2.OR.ISET(2).EQ.0) GO TO 160
     Q3=Q2
     GO TO 250
 160 Q3=FINQ(DEL,R3,TOL)
     ISET(3)=1
 250 IF(ISET(4).GT.0) GO TO 290
     IF(R4.NE.R1.OR.ISET(1).EQ.0) GO TO 255
     Q4=Q1
     GO TO 290
 255 IF(R4.NE.R2.OR.ISET(2).EQ.0) GO TO 260
     Q4=Q2
     GO TO 290
 260 Q4=FINQ(DEL,R3,TOL)
     ISET(4)=1
 290 FINQDF=FINQDF-(Q3-Q4)
 300 GO TO 500
 400 IF(R2.EQ.R4) GO TO 500
     IF(ISET(2).GT.0) GO TO 410
     ISET(2)=1
     Q2=FINQ(DEL,R2,TOL)
 410 IF(ISET(4).GT.0) GO TO 420
     ISET(4)=1
     Q4=FINQ(DEL,R4,TOL)
 420 FINQDF=Q4-Q2
C
C  SAVE CALCULATED (ISET(I)=1) VALUES
C
 500 IF(ISET(1).NE.1) GO TO 510
     ROLD(1)=R1
     QOLD(1)=Q1
 510 IF(ISET(2).NE.1) GO TO 520
     ROLD(2)=R2
     QOLD(2)=Q2
 520 IF(ISET(3).NE.1) GO TO 530
     ROLD(3)=R3
     QOLD(3)=Q3
```

```
  530 IF(ISET(4).NE.1) GO TO 540
      ROLD(4)=R4
      QOLD(4)=Q4
  540 RETURN
      END

      COMPLEX FUNCTION FUNINT(X)
C--COMPLEX FUNCTION INTERPOLATION BY QUINTIC SPLINE VIA
C  CALL TO 'QPOINT', WHERE THE QUINTIC SPLINE
C  COEFFICIENTS AR,BR,CR,DR,ER, AI,BI,CI,DI,EI WERE
C  PREVIOUSLY OBTAINED BY SUBR 'QUINT'.
C
      DIMENSION SR(80),AR(80),BR(80),CR(80),DR(80),ER(80),
     & SI(80),AI(80),BI(80),CI(80),DI(80),EI(80)
      COMMON/SPLN80/SR,AR,BR,CR,DR,ER,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,NL
      COMMON/FIN/R1,R2,R0,XL,SIG1,XX,Y
      R=ALOG(SQRT(X*X+Y*Y))
      CALL QPOINT(NL,SR,AR,BR,CR,DR,ER,RLM1,DELRLM,R,YR)
      CALL QPOINT(NL,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,R,YI)
      FUNINT=CMPLX(YR,YI)
      RETURN
      END

      COMPLEX FUNCTION INFNEX(B)
C--INFNEX COMPUTES THE ELECTRIC FIELD PARALLEL TO AN
C  INFINITELY LONG WIRE SOURCE AT THE EARTH"S SURFACE
C--CALLS ZFOURO(SINE TRANSFORM) AND F2MOD
      EXTERNAL F2MOD
      COMPLEX ZFOURO
      COMMON/PARM/IS,X,Y,R,SIG1,BNYQ,M,TOL
      SB=SQRT(B)
      DEL=R/SB
      DEL2=DEL*DEL
      INFNEX=CMPLX(0.0,0.0)
      INFNEX=ZFOURO(ALOG(SB),F2MOD,TOL,LW)*CMPLX(0.0,-2./(SB*DEL2))
      INFNEX=INFNEX/(3.1415927*SIG1)
      RETURN
      END

      SUBROUTINE QPOINT(NY,Y,B,C,D,E,F,X1,DELX,XX,YY)
C  GIVEN THE QUINTIC SPLINE COEFF"S B(*),C(*),D(*),E(*),F(*) AS
C  OBTAINED FROM SUBR 'QUINT', AND GIVEN NY OBS. DATA Y(NY) EQUALLY
C  SPACED BY DELX STARTING AT X1, THEN 'QPOINT' INTERPOLATES
C  YY AT ANY XX IN (X1,X1+(NY-1)*DELX).
C
      DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)
      XMAX=X1+(NY-1)*DELX
      IF(XX.LT.X1.OR.XX.GT.XMAX) GO TO 2
      I=(XX-X1)/DELX+1
      XI=X1+(I-1)*DELX
      T=(XX-XI)/DELX
      YY=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I)
```

```
1     RETURN
2     WRITE(6,3) XX,X1,XMAX
3     FORMAT('0QPOINT ERROR-- XX=',E16.8,' NOT IN CLOSED INTERVAL (',
     & E16.8,',',E16.8,')')
      GO TO 1
      END

      SUBROUTINE QUINT(NY,Y,B,C,D,E,F)
C--COMPUTES COEFFICIENTS OF A QUINTIC NATURAL SPLINE S(X) GIVEN
C  THE ORDINATES Y(I) AT ASSUMED EQUIDISTANT POINTS X(I),I=1 TO NY.
C
C  TRANSLATED FROM ALGOL TO FORTRAN BY
C   W.L. ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.
C  REF: ACM TRANSACTIONS ON MATH. SOFTWARE, SEPT 1976, V.2, N. 3,
C       PP.281-289.
C
C  PARAMETERS:
C
C  NY =  NUMBER OF DATA POINTS GIVEN IN Y(NY), NY.GT.2.
C  Y()=  ARRAY OF NY GIVEN ORDINATES (DIM.GE.NY).
C        Y() POINTS ASSUMED EQUALLY SPACED IN X-DIRECTION.
C  B,C,D,E,F() = RESULTING ARRAYS (EACH DIM.GE.NY) OF
C        QUINTIC SPLINE COEFFICIENTS, WHERE
C        FOR ANY XX IN [X(I),X(I+1)):
C        S(XX)=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I) WITH
C        T=(XX-X(I))/DELX,  DELX=(X(I+1)-X(I)) FOR ANY I.
C NOTE:  SEE PROC 'QPOINT' TO EVAL THE QUINTIC SPLINE AFTER
C           'QUINT' IS CALLED.
C
      DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)
      IF(NY.LE.2) GO TO 4
      N=NY-3
      P=0.0
      Q=0.0
      R=0.0
      S=0.0
      T=0.0
      DO 1 I=1,N
      U=P*R
      B(I)=1.0/(66.0-U*R-Q)
      R=26.0-U
      C(I)=R
      D(I)=Y(I+3)-3.0*(Y(I+2)-Y(I+1))-Y(I)-U*S-Q*T
      Q=P
      P=B(I)
      T=S
      S=D(I)
1     CONTINUE
      D(N+2)=0.0
      N1=N+1
      D(N1)=0.0
      DO 2 J=1,N
```

```
      I=N1-J
      D(I)=(D(I)-C(I)*D(I+1)-D(I+2))*B(I)
2     CONTINUE
      N=NY-1
      Q=0.0
      V=D(1)
      T=V
      R=V
      DO 3 I=2,N
      P=Q
      Q=R
      R=D(I)
      S=T
      T=P-Q-Q+R
      F(I)=T
      U=5.0*(-P+Q)
      E(I)=U
      D(I)=10.0*(P+Q)
      C(I)=0.5*(Y(I+1)+Y(I-1)+S-T)-Y(I)-U
      B(I)=0.5*(Y(I+1)-Y(I-1)-S-T)-D(I)
3     CONTINUE
      F(1)=V
      E(1)=0.0
      E(NY)=0.0
      D(1)=0.0
      D(NY)=0.0
      C(1)=C(2)-10.0*V
      C(NY)=C(NY-1)+10.0*T
      B(1)=Y(2)-Y(1)-C(1)-V
      B(NY)=Y(NY)-Y(NY-1)+C(NY)-T
4     RETURN
      END

      SUBROUTINE RECUR2(G,V1,F1,L1)
C  RECUR2 RECURSIVELY COMPUTES THE ELEMENTS WHICH ARE USED
C  IN THE LAYERED-EARTH HANKEL TRANSFORM KERNELS
C--ORIGINAL ALGORITHM FROM ANDERSON(1974) HAS BEEN
C  MODIFIED TO USE COMPLEX CONDUCTIVITIES IF ICMPLX=1
C
      COMMON/MODEL/K,D,M
      COMMON/CFLAG/CK,ICMPLX
      REAL K(10),D(9)
      COMPLEX C,VM,V1,F1,L1,E,ONE,CK(10)
      DATA ONE/(1.0,0.0)/
      F1=ONE
      L1=ONE
      G2=G*G
      IF(ICMPLX.EQ.0) VM=CSQRT(CMPLX(G2,2.*K(M)))
      IF(ICMPLX.EQ.1) VM=CSQRT(CMPLX(G2-2.*AIMAG(CK(M)),2.*REAL(CK(M))))
      IF(M.EQ.1) GO TO 2
      J=M-1
    1 IF(ICMPLX.EQ.0) V1=CSQRT(CMPLX(G2,2.*K(J)))
```

```
      IF(ICMPLX.EQ.1) V1=CSQRT(CMPLX(G2-2.*AIMAG(CK(J)),2.*REAL(CK(J))))
      E=CEXP(-V1*D(J))
      C=(ONE-E)/(ONE+E)
      F1=(VM*F1+V1*C)/(V1+VM*F1*C)
      IF(ICMPLX.EQ.0) E=K(J+1)*V1+K(J)*VM*L1*C
      IF(ICMPLX.EQ.1) E=CK(J+1)*V1+CK(J)*VM*L1*C
      IF(REAL(E).EQ.0.0.AND.AIMAG(E).EQ.0.0) E=CMPLX(1.E-30,1.E-30)
      IF(ICMPLX.EQ.0) L1=(K(J)*VM*L1+K(J+1)*V1*C)/E
      IF(ICMPLX.EQ.1) L1=(CK(J)*VM*L1+CK(J+1)*V1*C)/E
      IF(J.EQ.1) GO TO 3
      J=J-1
      VM=V1
      GO TO 1
    2 V1=VM
    3 RETURN
      END

      SUBROUTINE RECURS(G,V1,F1)
C  RECURS RECURSIVELY COMPUTES THE ELEMENTS WHICH ARE USED
C  IN THE LAYERED-EARTH HANKEL TRANSFORM KERNELS
C--ORIGINAL ALGORITHM FROM ANDERSON(1974) HAS BEEN
C--MODIFIED TO USE COMPLEX CONDUCTIVITIES IF ICMPLX=1
C
      COMPLEX C,VM,V1,F1,EVD,ONE,T,CK(10)
      REAL K(10),D(9)
      COMMON/MODEL/K,D,M
      COMMON/CFLAG/CK,ICMPLX
      DATA ONE/(1.0,0.0)/
      F1=ONE
      G2=G*G
      IF(ICMPLX.EQ.0) VM=CSQRT(CMPLX(G2,2.0*K(M)))
      IF(ICMPLX.EQ.1) VM=CSQRT(CMPLX(G2-2.*AIMAG(CK(M)),2.*REAL(CK(M))))
      IF(M.EQ.1) GO TO 30
      J=M-1
   10 IF(ICMPLX.EQ.0) V1=CSQRT(CMPLX(G2,2.0*K(J)))
      IF(ICMPLX.EQ.1) V1=CSQRT(CMPLX(G2-2.*AIMAG(CK(J)),2.*REAL(CK(J))))
      EVD=-V1*D(J)
      EVD=CEXP(EVD)
   20 C=(ONE-EVD)/(ONE+EVD)
      T=VM*F1
      F1=(T+V1*C)/(V1+T*C)
      IF(J.EQ.1) GO TO 40
      J=J-1
      VM=V1
      GO TO 10
   30 V1=VM
   40 RETURN
      END

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

```
C  REAL FOURIER SINE TRANSFORM WITH ARGUMENT X(=ALOG(B))
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....
C
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.
C
C--PARAMETERS:
C
C     * X     = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM
C                'RLAGF1' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,
C                THEN SUBPROGRAM 'RFOUR1' IS ADVISED FOR GENERAL USE.
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).
C               MIN.L=20 AND MAX.L=266--WHICH COULD
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING
C                 VERY FAST...
C     * NEW=    1 IS NECESSARY 1ST TIME OR BRAND NEW X.
C               0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20
C                 IS ASSUMED INTERNALLY BY THIS ROUTINE.
C                 NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL
C                 STILL ASSUME X=(LAST X)-0.20 ANYWAY...
C                 IT IS THE USERS RESPONSIBILITY TO NORMALIZE
C                 BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).
C               THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT
C                 TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A
C                 SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING
C                 ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...
C                 THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED
C                 KERNELS WILL BE USED IN THE LAGGED CONVOLUTION
C                 WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS
C                 WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)
C
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF1; THE FOURIER
C  TRANSFORM IS THEN RLAGF1/B WHICH IS TO BE COMPUTED AFTER EXIT FROM
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...
```

```
C
C--USAGE-- 'RLAGF1' IS CALLED AS FOLLOWS:
C     ...
C     EXTERNAL RF
C     ...
C     R=RLAGF1(ALOG(B),RF,TOL,L,NEW)/B
C     ...
C     END
C     REAL FUNCTION RF(G)
C     ...USER SUPPLIED CODE...
C     END
C
C--NOTES:
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:
C               ...
C               NB=AINT(5.*ALOG(BMAX/BMIN))+1
C               NB1=NB+1
C               X0=ALOG(BMAX)+.2
C               NEW=1
C               DO 1 J=1,NB
C               I=NB1-J
C               X=X0-.2*J
C               ARG(I)=EXP(X)
C               ANS(I)=RLAGF1(X,RF,TOL,L,NEW)/ARG(I)
C     1         NEW=0
C               ...
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW
C     BMAX,BMIN AND BY SETTING NEW=1....
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE
C
C
      DIMENSION KEY(266),SAVE(266)
      DIMENSION WT(266),W1(76),W2(76),W3(76),W4(38)
      EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),
     1 (WT(229),W4(1))
C--SIN-EXTENDED FILTER WEIGHT ARRAYS:
      DATA W1/
     1-1.1113940E-09,-1.3237246E-12, 1.5091739E-12,-1.6240954E-12,
```

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

```
    5 8.0869203E-01,-6.2757149E-01, 3.4062630E-01,-1.5885304E-01,
    6 7.0472984E-02,-3.1624462E-02, 1.4894068E-02,-7.4821176E-03,
    7 4.0035936E-03,-2.2543784E-03, 1.3160358E-03,-7.8636604E-04,
    8 4.7658745E-04,-2.9125817E-04, 1.7885105E-04,-1.1012416E-04,
    9 6.7910334E-05,-4.1914054E-05, 2.5881544E-05,-1.5985851E-05,
    1 9.8751880E-06,-6.1008526E-06, 3.7692543E-06,-2.3287953E-06/
     DATA W4/
    1 1.4388425E-06,-8.8899353E-07, 5.4926991E-07,-3.3937048E-07,
    2 2.0968284E-07,-1.2955437E-07, 8.0046336E-08,-4.9457371E-08,
    3 3.0557711E-08,-1.8880390E-08, 1.1665454E-08,-7.2076428E-09,
    4 4.4533423E-09,-2.7515696E-09, 1.7001092E-09,-1.0504494E-09,
    5 6.4904567E-10,-4.0102999E-10, 2.4778763E-10,-1.5310321E-10,
    6 9.4600354E-11,-5.8453314E-11, 3.6119400E-11,-2.2320056E-11,
    7 1.3793460E-11,-8.5242656E-12, 5.2675102E-12,-3.2543076E-12,
    8 2.0097689E-12,-1.2405412E-12, 7.6530538E-13,-4.7191929E-13,
    9 2.9084993E-13,-1.7923661E-13, 1.1018948E-13,-6.7885902E-14,
    1 4.2025050E-14,-2.1314731E-14/
C--$$ENDATA
C
      IF(NEW) 10,30,10
10    LAG=-1
      X0=-X-38.30455704
      DO 20 IR=1,266
20    KEY(IR)=0
30    LAG=LAG+1
      RLAGF1=0.0
      CMAX=0.0
      L=0
      ASSIGN 110 TO M
      I=191
      GO TO 200
110   CMAX=AMAX1(ABS(C),CMAX)
      I=I+1
      IF(I.LE.208) GO TO 200
      IF(CMAX.EQ.0.0) GO TO 150
      CMAX=TOL*CMAX
      ASSIGN 120 TO M
      I=190
      GO TO 200
120   IF(ABS(C).LE.CMAX) GO TO 130
      I=I-1
      IF(I.GT.0) GO TO 200
130   ASSIGN 140 TO M
      I=209
      GO TO 200
140   IF(ABS(C).LE.CMAX) GO TO 190
      I=I+1
      IF(I.LE.266) GO TO 200
      GO TO 190
150   ASSIGN 160 TO M
      I=1
      GO TO 200
```

```
160   IF(C.EQ.0.0) GO TO 170
      I=I+1
      IF(I.LE.190) GO TO 200
170   ASSIGN 180 TO M
      I=266
      GO TO 200
180   IF(C.EQ.0.0) GO TO 190
      I=I-1
      IF(I.GE.209) GO TO 200
190   RETURN
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)
200   LOOK=I+LAG
      IQ=LOOK/267
      IR=MOD(LOOK,267)
      IF(IR.EQ.0) IR=1
      IROLL=IQ*266
      IF(KEY(IR).LE.IROLL) GO TO 220
210   C=SAVE(IR)*WT(I)
      RLAGF1=RLAGF1+C
      L=L+1
      GO TO M,(110,120,140,160,180)
220   KEY(IR)=IROLL+IR
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))
      GO TO 210
      END

C ****************** PROGRAM SCHCOPL ******************
C--PROGRAM SCHCOPL CALCULATES THE ELECTROMAGNETIC COUPLING
C  BETWEEN TWO STRAIGHT PARALLEL WIRES WHICH ARE VERY
C  CLOSE TOGETHER IN A SCHLUMBERGER OR WENNER TYPE ARRAY.
C  THE SOURCE WIRE IS ASSUMED TO LIE ALONG THE X-AXIS
C  CENTERED AT THE ORIGIN AND EXTENDED BETWEEN +-AB WHILE
C  THE RECEIVER WIRE IS EXTENDED BETWEEN +-MN AND SEPARATED
C  FROM THE SOURCE WIRE BY DISTANCE Y .  THE INPUT PARAMETERS ARE:
C
C  SIG(I)     CONDUCTIVITY OF THE ITH LAYER
C  D(I)       THICKNESS OF THE ITH LAYER
C  M          NUMBER OF LAYERS
C  AB         ARRAY OF SOURCE HALFLENGTHS
C  MN         ARRAY OF RECEIVER HALFLENGTHS
C  NSP        NUMBER OF DIFFERENT VALUES IN
C             MN AND AB ARRAYS
C  Y          SEPARATION BETWEEN WIRES
C  RMAX       (DEFAULT=1000*(LARGEST AB USED)) IS UPPER LIMIT OF
C             DOUBLE INTEGRAL WHICH IS USED TO CORRECT INFINITE
C             LINE INTEGRAL
C  NF         >0, NUMBER OF FREQUENCIES DESIRED PER DECADE
C             BETWEEN F0 AND FM
C             <0, NUMBER OF SPECIFIED FREQUENCIES IN FNF
C  F0,FM      MINIMUM AND MAXIMUM FREQUENCIES DESIRED
C  FNF        SPECIFIED FREQUENCIES
C  TOL        TOLERANCE FOR HANKEL TRANSFORM CALCULATIONS
```

```
C  FINTL1     TOLERANCE FOR INTEGRATION ALONG SOURCE WIRE
C  FINTL2     TOLERANCE FOR INTEGRATION ALONG RECEIVER WIRE
C  IN1,IN2    =1 FOR ADAPTIVE QUADRATURE INTEGRATION
C             =2 FOR NON-ADAPTIVE QUADRATURE INTEGRATION
C             IN1 IS FOR THE OUTER INTEGRAL ACROSS THE RECEIVER
C             IN2 IS FOR THE INNER INTEGRAL ACROSS THE SOURCE
C  NFIN       INTERVAL IN LOG-SPACE WITH WHICH THE SPLINE
C             NODES FOR FINITE WIRE INTEGRATION ARE
C             CALCULATED, E. G. INTERVAL=0.2/NFIN
C             (DEFAULT=1)
C  MEV1,MEV2 MAXIMUM NUMBER OF FUNCTION EVALUATIONS FOR
C             RESPECTIVE INTEGRATION ROUTINES
C  TMAX,TMIN MAXIMUM AND MINIMUM TIME VALUES DESIRED
C  TFLAG      =0 COMPUTES FREQUENCY RESPONSE ALONE
C             =1 COMPUTES FREQUENCY AND TRANSIENT RESPONSE
C             =2 COMPUTES TRANSIENT RESPONSE ALONE
C             =3 COMPUTES TRANSIENT RESPONSE FOR A
C             FREQUENCY RESPONSE PREVIOUSLY COMPUTED
C  RC         TIME CONSTANT OF SINGLE POLE LOW-PASS
C             FILTER TO BE CONVOLVED WITH FREQUENCY
C             RESPONSE FOR TRANSIENT RESPONSE
C             CALCULATIONS
C             NOTE:  TRANSIENT RESPONSE IS THE STEP
C             RESPONSE
C  ICMPLX     =0 COMPUTES THE COUPLING USING THE REAL
C                CONDUCTIVITIES IN SIG ARRAY,
C             =1 COMPUTES THE COUPLING USING THE COMPLEX
C                CONDUCTIVITIES COMPUTED BY THE USER-
C                DEFINED FUNCTION SIGMA(J,1./(DEL*DEL))
C  INFILE     (DEFAULT=5) INPUT FILE NUMBER
C  OUFILE     (DEFAULT=6) OUTPUT FILE NUMBER
C
C  THE SUBPROGRAMS ARE ORGANIZED AS FOLLOWS:
C
C                   SCHCOPL
C                      *
C                      *   INFINITE   DOUBLE                 TRANSIENT
C                      *     WIRE    INTEGRAL                 RESPONSE
C    *************************************************************
C    *        *              *       *     *      *      *      *
C  SIGMA    SETSPL         INFNEX  ZSUB1 ZSUBA1  ZHANKO SPLIN1 RLAGF1
C             *              *       *     *      *             *
C      **************        *       *******      *             *
C      *      *      *       *          *         *             *
C     ZEX    CZEX   QUINT  ZFOURO     CORFUN     F7G          CSPLNT
C      *      *  *           *          *         *
C      ******** *            *       *******      *
C         *     *            *       *     *      *
C       ZLAGHO  SIGMA      F2MOD   ZSUB2 ZSUBA2  RECUR2
C         *                  *       *     *
C        F3                  *       *******
C         *                  *          *
```

```
C        RECURS                  RECURS     FUNINT***QPOINT
C
      COMPLEX INFNEX,FINQ,ZHANKO,SCH(30,30),ECOPL,CSPEC,
     1 ZERR1,ZERR2,CORECT,ZSUB1,ZSUBA1,SIGMA,SIGMA1,CK(10)
      REAL MN(30),AB(30),K(10),D(9),DD(9),SIG(10)
      REAL P(50),S(50),PS(2),FNF(50)
      INTEGER OUFILE
      EXTERNAL F7G,CSPLNT,CORFUN,ZEX,CZEX
      COMMON/CFLAG/CK,ICMPLX
      COMMON/FINERR/TL,FINTL1,IN1,NFIN,NEV1,MEV1,ZERR1,LW
      COMMON/CORRF/YY,XMIN,RMAX
      COMMON/CSPLINE/D1(50),D2(50),D3(50),NN,FREQ(50),SPEC(50)
      COMMON/MODEL/K,DD,M
      COMMON/THICK/D
      COMMON/PARM/IS,X,Y,R,SIG1,BNYQ,NLYR,TOL
      NAMELIST/PARMS/SIG,M,D,MN,AB,FM,NF,FO,NSP,Y,TMAX,TMIN,TFLAG,RC
     1,TOL,FINTL1,FINTL2,IN1,IN2,NFIN,MEV1,MEV2,RMAX,ICMPLX,FNF,
     2 INFILE,OUFILE
      DATA DEG/57.29577951/
      DATA PS,TWOPI/0.0,0.0,6.283185308/
C
C--DEFAULT AB/2 AND MN/2 VALUES
C
      DATA AB/10,13,16,20,20,25,30,40,50,65,80,100,100,130,160,200,200,
     1 250,300,400,500,650,800,1000,1000,1300,1600,2000,2000,2500/
      DATA MN/2,2,2,2,4,4,4,4,4,4,4,4,20,20,20,20,40,40,40,40,40,40,
     1 40,40,200,200,200,200,400,400/
C
C--ASSIGN NAMELIST DEFAULT PARAMETERS
C
      INFILE=5
      OUFILE=6
      X=0
      IS=0
      Y=.01
      BNYQ=1.E30
      TOL=1.E-6
      M=1
      TFLAG=0
      FINTL1=1.E-3
      FINTL2=1.E-5
      MEV1=300
      MEV2=300
      TOL=1.E-6
      IN1=1
      IN2=1
      NFIN=1
      RC=0.0
      ICMPLX=0
      RMAX=0.0
C
      WRITE(OUFILE,501)
```

```
  501 FORMAT(25H ENTER $PARMS PARAMETERS$)
  100 READ(INFILE,PARMS)
      IF(M.EQ.0) STOP
      IF(TFLAG.EQ.3) GO TO 20
C
C--TFLAG=3 ASSUMES THAT THE FREQUENCY FUNCTION TO BE TRANSFORMED
C  HAS ALREADY BEEN COMPUTED AND RESIDES IN THE SCH ARRAY
C
C--DEFINE EQUIVALENT COMMON PARAMETERS
      NLYR=M
      R=Y
      YY=Y
C
      SIG1=SIG(1)
      IF(ICMPLX.EQ.1) GO TO 2
      DO 10 I=1,M
   10     K(I)=SIG(I)/SIG1
      GO TO 3
    2 SIG1=REAL(SIGMA(1,0.0))
      K(1)=SIG1
    3 Y2=Y*Y
      M1=M-1
C
C--PRINT MODEL PARAMETERS
C
      WRITE(OUFILE,503) M
  503 FORMAT(/5X,I3,12H LAYER MODEL)
      IF(ICMPLX.EQ.0) WRITE(OUFILE,504) (SIG(I),I=1,M)
  504 FORMAT(5H SIG=,12E12.4)
  505 FORMAT(5H D  =,12E12.4)
      IF(ICMPLX.EQ.1) WRITE(OUFILE,506)
  506 FORMAT(28H COMPLEX CONDUCTIVITIES USED)
      IF(M.GT.1) WRITE(OUFILE,505) (D(I),I=1,M1)
C
C--COMPUTE MAXIMUM AND MINIMUM DISTANCE BETWEEN WIRES
C  IN ORDER TO DEFINE ARGUMENT RANGE FOR ZEX OR CZEX
C  EVALUATIONS
C
      RMIN=AB(1)-MN(1)
      IF(RMAX.EQ.0.0) RMAX=100.*AB(NSP)
      CON=1./(SIG1*6.283185308)
      XB=R*R*SIG1*39.47841763E-7
      IF(NF.LT.0) GO TO 41
      NN=NF*ALOG10(FM/F0)+1
      DX=EXP(2.30258509/FLOAT(NF))
      B=XB*F0
      FREQ(1)=F0
      GO TO 42
   41 NN=-NF
C
C--FIRST LOOP OVER NN FREQUENCIES
C
```

```
   42 DO 7 JJ=1,NN
         IF(NF.LT.0) GO TO 43
         IF(JJ.GT.1) B=B*DX
         IF(JJ.GT.1) FREQ(JJ)=FREQ(JJ-1)*DX
         GO TO 44
   43    B=XB*FNF(JJ)
         FREQ(JJ)=FNF(JJ)
   44    SB=SQRT(B)
         DEL=R/SB
         CF=1./(DEL*DEL)
         IF(ICMPLX.EQ.1) SIGMA1=SIGMA(1,CF)/SIG1
         DO 8 I=1,M
             IF(ICMPLX.EQ.1) CK(I)=SIGMA(I,CF)/SIG1
    8        DD(I)=2.*D(I)/DEL
C--COMPUTE THE WIRE COUPLING
C  ASSUMING INFINITELY-LONG SOURCE WIRE
         ECOPL=INFNEX(B)
C
C--COMPUTE SPLINE COEFFICIENTS
C
         IF(ICMPLX.EQ.0) CALL SETSPL(ZEX,DEL,RMAX,RMIN)
         IF(ICMPLX.EQ.1) CALL SETSPL(CZEX,DEL,RMAX,RMIN)
C
C--NEXT LOOP OVER NSP SPACINGS
C
         DO 7 ISP=1,NSP
             R1=SQRT((AB(ISP)-MN(ISP))**2+Y2)
             R2=SQRT((AB(ISP)+MN(ISP))**2+Y2)
             FINQ=CMPLX(0.0,0.0)
             IF(M.EQ.1) GO TO 5
             B1=R1/DEL
             B2=R2/DEL
             FINQ=ZHANKO(ALOG(B1),F7G,TOL,LW)/B1
             FINQ=FINQ-ZHANKO(ALOG(B2),F7G,TOL,LW)/B2
             FINQ=FINQ*CMPLX(0.0,1./DEL)
    5        FINQ=FINQ-CMPLX((1./R1-1./R2),0.0)
             XMIN=AB(ISP)
C
C--COMPUTE DOUBLE INTEGRAL CORRECTION TERM
C
             IF(IN1.EQ.1) CORECT=2.*ZSUBA1(-MN(ISP),MN(ISP),FINTL1,
     1       NEV1,ICK1,ZERR1,CORFUN,MEV1)
             IF(IN1.EQ.2) CORECT=2.*ZSUB1(-MN(ISP),MN(ISP),FINTL1,
     1       NEV1,ICK1,ZERR1,CORFUN,MEV1)
C
             SCH(JJ,ISP)=2.*MN(ISP)*ECOPL+CON*CORECT+2.*CON*FINQ
    7        IF(ICMPLX.EQ.1) SCH(JJ,ISP)=SCH(JJ,ISP)/SIGMA1
      IF(TFLAG.EQ.0) GO TO 21
   20 NT=AINT(5.*ALOG(TMAX/TMIN))+1
      NT1=NT+1
C--CALCULATED IN NORMALIZED TIME
C  NORM TIME = TWOPI * REAL TIME
```

```
      X0=ALOG(TMAX*TWOPI)+0.2
C
   21 DO 40 IPOS=1,NSP
          WRITE(OUFILE,507) AB(IPOS),MN(IPOS)
  507     FORMAT(/3X,6HAB/2 =,E12.4/3X,6HMN/2 =,E12.4/)
          IF(TFLAG.GT.1) GO TO 35
          WRITE(OUFILE,200)
  200     FORMAT(4X,9HFREQUENCY,5X,4HREAL,8X,4HIMAG,8X,4HMAGN,8X,3HPFE,
     1    8X,5HPHASE)
          PFE0=CABS(SCH(1,IPOS))
          DO 30 J=1,NN
              AMP=CABS(SCH(J,IPOS))
              PFE=100.*(1.-AMP/PFE0)
              PHZ=DEG*ATAN2(AIMAG(SCH(J,IPOS)),REAL(SCH(J,IPOS)))
   30         WRITE(OUFILE,509) FREQ(J),SCH(J,IPOS),AMP,PFE,PHZ
  509     FORMAT(1H ,6E12.4)
          IF(TFLAG.EQ.0) GO TO 40
C
C--COMPUTE THE STEP TRANSIENT RESPONSE USING A CUBICALLY
C  SPLINED FREQUENCY RESPONSE FUNCTION
C
   35     DC=REAL(SCH(1,IPOS))
          DO 50 II=1,NN
C
C--MULTIPLY THE FREQUENCY RESPONSE (DC VALUE SUBTRACTED)
C  BY THE SOURCE FUNCTION (1/FREQ FOR STEP FUNCTION)
C
              SPEC(II)=(REAL(SCH(II,IPOS))-DC)/FREQ(II)
              IF(RC.EQ.0.0) GO TO 50
C
C--MULTIPLY THE FREQUENCY RESPONSE BY LOW-PASS
C  FILTER TRANSFER FUNCTION IF RC GT 0
C
              TCON=FREQ(II)*RC
              CSPEC=SCH(II,IPOS)*CMPLX(1.,-TCON)
              SPEC(II)=(REAL(CSPEC)-DC)/FREQ(II)/(1.+TCON*TCON)
   50         CONTINUE
          CALL SPLIN1(NN,0,FREQ,SPEC,D1,D2,D3,0,PS,P,S)
          NEW=1
          WRITE(OUFILE,510)
  510     FORMAT(/6X,9HTIME(SEC),4X,15HOBS VOLTAGE/AMP)
          DO 60 J=1,NT
              I=NT1-J
              X=X0-0.2*J
              T=EXP(X)
              FDC=1.
              IF(RC.GT.0.0) FDC=1.-EXP(-T/(RC*TWOPI))
C
C--COMPUTE SINE TRANSFORM (RLAGF1) OF CUBICALLY SPLINED
C  FREQUENCY FUNCTION (SPLINE INTERPOLATOR CSPLNT)
C
              V=0.636619772*RLAGF1(X,CSPLNT,TOL,LW,NEW)/T+DC*FDC
```

```
             T=T/TWOPI
             WRITE(OUFILE,509) T,V
   60        NEW=0
   40    WRITE(OUFILE,512)
  512 FORMAT(1H )
      WRITE(OUFILE,513)
  513 FORMAT(27H ENTER $PARMS CHANGES ONLY$)
      GO TO 100
      END

      SUBROUTINE SETSPL(FUNC,DEL,RMAX,RMIN)
C--COMPUTE THE QUINTIC SPLINE COEFFICIENTS TO
C  REPRESENT FUNC(R) FOR THE RANGE RMAX TO RMIN.
C  'SETSPL' CALLS 'FUNC' (WHICH CALLS 'ZLAGH1 OR ZLAGH0') AND 'QUINT'.
C
C  PARAMETERS:
C
C  FUNC =  EXTERNAL DECLARED COMPLEX FUNCTION DEFINING THE DIPOLE FIELD
C          FUNCTION WITH CALLING SEQ: FUNC(B,NEW,R), WHERE
C           B = ANY IND. NO.
C           NEW = 1 FIRST TIME, 0 OTHERWISE (REF: ZLAGH1 OR ZLAGH0)
C           R = B*DEL FOR ANY B OR DEL (SKIN DEPTH).
C  DEL  =  SKIN DEPTH.
C  RMAX = MAXIMUM R AT WHICH FUNC WILL NEED TO BE EVALUATED
C  RMIN = MINIMUM R AT WHICH FUNC WILL NEED TO BE EVALUATED
C
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW
      COMMON/SPLN80/FDR(80),AR(80),BR(80),CR(80),DR(80),ER(80),
     & FDI(80),AI(80),BI(80),CI(80),DI(80),EI(80),RLM1,DELRLM,NB
      COMMON/CONST/DELL,DEL2,Z2DEL3
      COMPLEX FUNC,ESUM,FD,Z2DEL3
C--ISIZE IS THE MAXIMUM POSSIBLE NUMBER OF NODES IN QUINTIC SPLINE
C  AND ALSO IS THE DIMENSION OF ALL ARRAYS IN COMMON AREA SPLN80.
      DATA ISIZE/80/
      DELL=DEL
      DEL2=DEL*DEL
      Z2DEL3=CMPLX(0.0,2./(DEL2*DEL))
      BMAX=RMAX/DEL
      BMIN=RMIN/DEL
      NB=AINT(5.*ALOG(BMAX/BMIN))+2
      NB=MAX(NB,3)
      X0=ALOG(BMIN)+NB*0.2
      NB=NB+3
C--RANGE OF RMIN,RMAX EXTENDED BY AT LEAST 2 ON EACH
C  END IN ORDER TO MAKE THE END CONDITIONS CHOSEN FOR
C  THE SPLINE IRRELEVANT TO THE REAL RANGE OF INTEREST.
      NRMAX=ISIZE/NB
      IF(NFIN.LE.NRMAX) GO TO 3
      IF(NRMAX.GT.0.0) GO TO 2
      WRITE(6,100)
  100 FORMAT(43H ERROR IN SETSPL: INSUFFICIENT SPLINE NODES)
      STOP
```

```
    2 NFIN=NRMAX
      WRITE(6,110) NFIN
  110 FORMAT(43H ERROR IN SETSPL: NFIN TOO LARGE, RESET TO ,I2)
3     DELRLM=.2/FLOAT(NFIN)
      X0=X0-DELRLM
      DO 5 ITIME=1,NFIN
          NEW=1
          X0=X0+DELRLM
          DO 5 J=1,NB
              I=(NB+1)-J
              I=NFIN*(I-1)+ITIME
              XX=X0-0.2*J
              BM=EXP(XX)
              RM=BM*DEL
              IF(I.EQ.1) RLM1=ALOG(RM)
              FD=FUNC(BM,NEW,RM)
              FDR(I)=REAL(FD)
              FDI(I)=AIMAG(FD)
    5         NEW=0
      NB=NFIN*NB
      CALL QUINT(NB,FDR,AR,BR,CR,DR,ER)
      CALL QUINT(NB,FDI,AI,BI,CI,DI,EI)
   10 RETURN
      END

      COMPLEX FUNCTION SIGMA(J,CF)
C--THIS IS AN EXAMPLE FUNCTION WHICH
C  COMPUTES A COMPLEX CONDUCTIVITY.
C  IT WAS USED FOR A HALFSPACE MODEL AND IGNORES THE
C  J ARGUMENT.
      COMMON/PARM/IS,A1,A2,A3,SIG1,A5,M,A7
      COMPLEX ONE
      DATA ONE/(1.0,0.0)/
      RM=0.3
      TAU=0.4
      IF(SIG1.EQ.0.0.AND.CF.EQ.0.0) SIG1=1.0
      OMEGA=CF/(SIG1*6.28318531E-7)
      SIGMA=CSQRT(CSQRT(CMPLX(0.0,OMEGA*TAU)))
      SIGMA=ONE/(ONE+SIGMA)
      SIGMA=ONE-RM*(ONE-SIGMA)
      SIGMA=0.1/SIGMA
      RETURN
      END

      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION.
C
C         BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO
C
C  PARMS--- M= NUMBER OF DATA POINTS .GT. 2
C           H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE),
C              UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE)
```

```
C           X= INDEP.VAR WHEN H=0. (DIM .GE. M).
C           Y= DEPENDENT VARIABLE  (DIM .GE. M).
C           A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)
C                 RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C.
C                 ALSO USED AS WORK ARRAYS DURING EXECUTION.
C           IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE
C              IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR
C              IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS.
C           D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY.
C           P,S= WORK ARRAYS (EACH DIM=M).
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.
C  THE RESULTING CUBIC SPLINE IS OF THE FORM:
C     Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3
C       FOR I=1,2,...,M-1
C
C
      REAL*4  X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL
      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999
      N=M-1
      IF(IT.EQ.0) GO TO 20
C--1ST DERIVATIVE BOUNDARIES GIVEN
      NE=N-1
      IF(H) 999,11,1
C--EQUAL SPACING H .GT. 0. AND IT=1
    1 HH=3.0/H
      DO 2 I=1,NE
      B(I)=4.0
      C(I)=1.0
      A(I)=1.0
    2 P(I)=HH*(Y(I+2)-Y(I))
      P(1)=P(1)-D(1)
      P(NE)=P(NE)-D(2)
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE
    3 C(1)=C(1)/B(1)
      P(1)=P(1)/B(1)
      DO 4 I=2,NE
      MUL=1.0/(B(I)-A(I)*C(I-1))
      C(I)=MUL*C(I)
    4 P(I)=MUL*(P(I)-A(I)*P(I-1))
C--OBTAIN SPLINE COEFFICIENTS
      A(NE+IT)=P(NE)
      I=NE-1
    5 A(I+IT)=P(I)-C(I)*A(I+IT+1)
      I=I-1
       IF(I.GE.1) GO TO 5
      IF(IT.EQ.0) GO TO 6
      A(1)=D(1)
      A(M)=D(2)
    6 IF(H.EQ.0.) GO TO 14
      HH=1.0/H
      DO 7 I=1,N
      MUL=HH*(Y(I+1)-Y(I))
```

```
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))
    7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))
      RETURN
C--UNEQUAL SPACING H=0.. AND IT=1
   11 DO 12 I=1,N
   12 S(I+1)=X(I+1)-X(I)
      DO 13 I=1,NE
      B(I)=2.0*(S(I+1)+S(I+2))
      C(I)=S(I+1)
      A(I)=S(I+2)
   13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/
     $ (S(I+1)*S(I+2))
      P(1)=P(1)-S(3)*D(1)
      P(NE)=P(NE)-S(N)*D(2)
      GO TO 3
   14 DO 15 I=1,N
      HH=1.0/S(I+1)
      MUL=(Y(I+1)-Y(I))*HH**2
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH
   15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2
      RETURN
C--2ND DERIVATIVE BOUNDARIES GIVEN
   20 NE=N+1
      IF(H) 999,31,21
C--EQUAL SPACING H .GT. 0 AND IT=0
   21 HH=3.0/H
      DO 22 I=2,N
      B(I)=4.0
      C(I)=1.0
      A(I)=1.0
   22 P(I)=HH*(Y(I+1)-Y(I-1))
      B(1)=2.0
      B(NE)=2.0
      C(1)=1.0
      C(NE)=1.0
      A(NE)=1.0
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)
      GO TO 3
C--UNEQUAL SPACING H=0 AND IT=0
   31 DO 32 I=1,N
   32 S(I+1)=X(I+1)-X(I)
      N1=N-1
      DO 33 I=1,N1
      B(I+1)=2.0*(S(I+1)+S(I+2))
      C(I+1)=S(I+1)
      A(I+1)=S(I+2)
   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)))/
     *      (S(I+1)*S(I+2))
      B(1)=2.0
      B(NE)=2.0
      C(1)=1.0
```

```
      C(NE)=1.0
      A(NE)=1.0
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)
      GO TO 3
  999 M=-IABS(M)
      RETURN
      END

C--ZQUAD PACKAGE (ZBLOCK,ZQUAD1,ZSUB1,ZSUBA1,ZQUAD2,ZSUB2,ZSUBA2)
C  FOR AUTOMATIC COMPLEX GAUSSIAN DOUBLE INTEGRATION OVER A
C  FINITE INTERVAL.
C
C--MODIFIED BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO
C  12/30/75.
C
C--USAGE:
C
C  USE 'ZSUB1' OR 'ZSUBA1' FOR 1ST COMPLEX INTEGRATION (CALLS ZQUAD1)
C  AND 'ZSUB2' OR 'ZSUBA2' FOR 2ND COMPLEX INTEGRATION (CALLS ZQUAD2)
C
C--REFERENCES:
C
C     (1)       PATTERSON,T.N.L, 1973, ALGORITHM FOR AUTOMATIC
C               NUMERICAL INTEGRATION OVER A FINITE INTERVAL [D1]
C               ACM COMM. V.16, NO.11, P.694-699.
C     (2)       ANDERSON,W.L., 1974, ELECTROMAGNETIC FIELDS ABOUT A
C               FINITE ELECTRIC WIRE SOURCE:
C               N.T.I.S REPORT PB-238199, 209P.
C
C--NOTES:
C
C     (A).  SEE REF(1) FOR A COMPLETE DISCUSSION OF THE BASIC
C               ALGORITHM(S) AS ORIGINALLY DEVELOPED FOR
C               SINGLE REAL FUNCTION AUTOMATIC GAUSSIAN INTEGRATION.
C     (B).  SEE REF(2) FOR A MODIFIED VERSION FOR SINGLE COMPLEX
C               FUNCTION AUTOMATIC GAUSSIAN INTEGRATION.
C     (C).  ALL CALLING PARMS USED BELOW IN THE ZQUAD PACKAGE ARE
C               IDENTICAL TO THOSE USED IN REF(2).  THEREFORE, SEE
C               REF(2) FOR COMMENTS ON THESE ANALOGOUS ROUTINES.
C               REF(1) MAY ALSO BE USED FOR DEFINITIONS OF MOST OF
C               THE PARMS...
C
C-- MULTICS VERSION USES CALL ZBLOCK TO INITILIZE COMMON/ZQUADP
C   FOR OTHER SYSTEMS, CHANGE SUBROUINTE ZBLOCK TO A
C   BLOCK DATA SUBPROGRAM -- AND REMOVE THE ASSIGNMENTS STATEMENTS.
C
      SUBROUTINE ZBLOCK
      DIMENSION P(381)
      COMMON/ZQUADP/Q(381)
      DATA MULTICS/0/
      DATA
```

```
* P( 1),P( 2),P( 3),P( 4),P( 5),P( 6),P( 7),
* P( 8),P( 9),P(10),P(11),P(12),P(13),P(14),
* P(15),P(16),P(17),P(18),P(19),P(20),P(21),
* P(22),P(23),P(24),P(25),P(26),P(27),P(28)/
*  0.77459666924148337704E 00,0.55555555555555555556E 00,
*  0.88888888888888888889E 00,0.26848808986833344073E 00,
*  0.96049126870802028342E 00,0.10465622602646726519E 00,
*  0.43424374934680255800E 00,0.40139741477596222291E 00,
*  0.45091653865847414235E 00,0.13441525524378422036E 00,
*  0.51603282997079739697E-01,0.20062852937698902103E 00,
*  0.99383196321275502221E 00,0.17001719629940260339E-01,
*  0.88845923287225699889E 00,0.92927195315124537686E-01,
*  0.62110294673722640294E 00,0.17151190913639138079E 00,
*  0.22338668642896688163E 00,0.21915685840158749640E 00,
*  0.22551049979820668739E 00,0.67207754295990703540E-01,
*  0.25807598096176653565E-01,0.10031427861179557877E 00,
*  0.84345657393211062463E-02,0.46462893261757986541E-01,
*  0.85755920049990351154E-01,0.10957842105592463824E 00/
 DATA
* P(29),P(30),P(31),P(32),P(33),P(34),P(35),
* P(36),P(37),P(38),P(39),P(40),P(41),P(42),
* P(43),P(44),P(45),P(46),P(47),P(48),P(49),
* P(50),P(51),P(52),P(53),P(54),P(55),P(56)/
*  0.99909812496766759766E 00,0.25447807915618744154E-02,
*  0.98153114955374010687E 00,0.16446049854387810934E-01,
*  0.92965485742974005667E 00,0.35957103307129322097E-01,
*  0.83672593816886873550E 00,0.56979509494123357412E-01,
*  0.70249620649152707861E 00,0.76879620499003531043E-01,
*  0.53131974364437562397E 00,0.93627109981264473617E-01,
*  0.33113539325797683309E 00,0.10566989358023480974E 00,
*  0.11248894313318662575E 00,0.11195687302095345688E 00,
*  0.11275525672076869161E 00,0.33603877148207730542E-01,
*  0.12903800100351265626E-01,0.50157139305899537414E-01,
*  0.42176304415588548391E-02,0.23231446639910269443E-01,
*  0.42877960025007734493E-01,0.54789210527962865032E-01,
*  0.12651565562300680114E-02,0.82230079572359296693E-02,
*  0.17978551568128270333E-01,0.28489754745833548613E-01/
 DATA
* P(57),P(58),P(59),P(60),P(61),P(62),P(63),
* P(64),P(65),P(66),P(67),P(68),P(69),P(70),
* P(71),P(72),P(73),P(74),P(75),P(76),P(77),
* P(78),P(79),P(80),P(81),P(82),P(83),P(84)/
*  0.38439810249455532039E-01,0.46813554990628012403E-01,
*  0.52834946790116519862E-01,0.55978436510476319408E-01,
*  0.99987288812035761194E 00,0.36322148184553065969E-03,
*  0.99720625937222195908E 00,0.25790497946856882724E-02,
*  0.98868475754742947994E 00,0.61155068221172463397E-02,
*  0.97218287474858179658E 00,0.10498246909621321898E-01,
*  0.94634285837340290515E 00,0.15406750466559497802E-01,
*  0.91037115695700429250E 00,0.20594233915912711149E-01,
*  0.86390793819369047715E 00,0.25869679327214746911E-01,
*  0.80694053195021761186E 00,0.31073551111687964880E-01,
```

```
*  0.73975604435269475868E 00,0.36064432780782572640E-01,
*  0.66290966002478059546E 00,0.40715510116944318934E-01,
*  0.57719571005204581484E 00,0.44914531653632197414E-01,
*  0.48361802694584102756E 00,0.48564330406673198716E-01/
 DATA
* P( 85),P( 86),P( 87),P( 88),P( 89),P( 90),P( 91),
* P( 92),P( 93),P( 94),P( 95),P( 96),P( 97),P( 98),
* P( 99),P(100),P(101),P(102),P(103),P(104),P(105),
* P(106),P(107),P(108),P(109),P(110),P(111),P(112)/
*  0.38335932419873034692E 00,0.51583253952048458777E-01,
*  0.27774982202182431507E 00,0.53905499335266063927E-01,
*  0.16823525155220746498E 00,0.55481404356559363988E-01,
*  0.56344313046592789972E-01,0.56277699831254301273E-01,
*  0.56377628360384717388E-01,0.16801938574103865271E-01,
*  0.64519000501757369228E-02,0.25078569652949768707E-01,
*  0.21088152457266328793E-02,0.11615723319955134727E-01,
*  0.21438980012503867246E-01,0.27394605263981432516E-01,
*  0.63260731936263354422E-03,0.41115039786546930472E-02,
*  0.89892757840641357233E-02,0.14244877372916774306E-01,
*  0.19219905124727766019E-01,0.23406777495314006201E-01,
*  0.26417473395058259931E-01,0.27989218255238159704E-01,
*  0.18073956444538835782E-03,0.12895240826104173921E-02,
*  0.30577534101755311361E-02,0.52491234548088591251E-02/
 DATA
* P(113),P(114),P(115),P(116),P(117),P(118),P(119),
* P(120),P(121),P(122),P(123),P(124),P(125),P(126),
* P(127),P(128),P(129),P(130),P(131),P(132),P(133),
* P(134),P(135),P(136),P(137),P(138),P(139),P(140)/
*  0.77033752332797418482E-02,0.10297116957956355524E-01,
*  0.12934839663607373455E-01,0.15536775555843982440E-01,
*  0.18032216390391286320E-01,0.20357755058472159467E-01,
*  0.22457265826816098707E-01,0.24282165203336599358E-01,
*  0.25791626976024229388E-01,0.26952749667633031963E-01,
*  0.27740702178279681994E-01,0.28138849915627150636E-01,
*  0.99998243035489159858E 00,0.50536095207862517625E-04,
*  0.99959879967191068325E 00,0.37774664632698466027E-03,
*  0.99831663531840739253E 00,0.93836984854238150079E-03,
*  0.99572410469840718851E 00,0.16811428654214699063E-02,
*  0.99149572117810613240E 00,0.25687649437940203731E-02,
*  0.98537149959852037111E 00,0.35728927835172996494E-02,
*  0.97714151463970571416E 00,0.46710503721143217474E-02,
*  0.96663785155841656709E 00,0.58434498758356395076E-02/
 DATA
* P(141),P(142),P(143),P(144),P(145),P(146),P(147),
* P(148),P(149),P(150),P(151),P(152),P(153),P(154),
* P(155),P(156),P(157),P(158),P(159),P(160),P(161),
* P(162),P(163),P(164),P(165),P(166),P(167),P(168)/
*  0.95373000642576113641E 00,0.70724899954335554680E-02,
*  0.93832039777959288365E 00,0.83428387539681577056E-02,
*  0.92034002547001242073E 00,0.96411777297025366953E-02,
*  0.89974489977694003664E 00,0.10955733387837901648E-01,
*  0.87651341448470526974E 00,0.12275830560082770087E-01,
```

```
*  0.8506444947683502797 6E 00,0.13591571009765546790E-01,
*  0.82215625436498040737E 00,0.14893641664815182035E-01,
*  0.79108493379984836143E 00,0.16173218729577719942E-01,
*  0.75748396638051363793E 00,0.17421930159464173747E-01,
*  0.72142308537009891548E 00,0.18631848256138790186E-01,
*  0.68298743109107922809E 00,0.19795495048097499488E-01,
*  0.64227664250975951377E 00,0.20905851445812023852E-01,
*  0.59940393024224289297E 00,0.21956366305317824939E-01,
*  0.55449513263193254887E 00,0.22940964229387748761E-01/
 DATA
* P(169),P(170),P(171),P(172),P(173),P(174),P(175),
* P(176),P(177),P(178),P(179),P(180),P(181),P(182),
* P(183),P(184),P(185),P(186),P(187),P(188),P(189),
* P(190),P(191),P(192),P(193),P(194),P(195),P(196)/
*  0.50768775753371660215E 00,0.23854052106038540080E-01,
*  0.45913001198983233287E 00,0.24690524744487676909E-01,
*  0.40897982122988867241E 00,0.25445769965464765813E-01,
*  0.35740383783153215238E 00,0.26115673376706097680E-01,
*  0.30457644155671404334E 00,0.26696622927450359906E-01,
*  0.25067873030348317661E 00,0.27185513229624791819E-01,
*  0.19589750271110015392E 00,0.27579749566481873035E-01,
*  0.14042423315256017459E 00,0.27877251476613701609E-01,
*  0.84454040083710883710E-01,0.28076455793817246607E-01,
*  0.28184648949745694339E-01,0.28176319033016602131E-01,
*  0.28188814180192358694E-01,0.84009692870519326354E-02,
*  0.32259500250878684614E-02,0.12539284826474884353E-01,
*  0.10544076228633167722E-02,0.58078616599775673635E-02,
*  0.10719490006251933623E-01,0.13697302631990716258E-01/
 DATA
* P(197),P(198),P(199),P(200),P(201),P(202),P(203),
* P(204),P(205),P(206),P(207),P(208),P(209),P(210),
* P(211),P(212),P(213),P(214),P(215),P(216),P(217),
* P(218),P(219),P(220),P(221),P(222),P(223),P(224)/
*  0.31630366082226447689E-03,0.20557519893273465236E-02,
*  0.44946378920320678616E-02,0.71224386864583871532E-02,
*  0.96099525623638830097E-02,0.11703388747657003101E-01,
*  0.13208736697529129966E-01,0.13994609127619079852E-01,
*  0.90372734658751149261E-04,0.64476204130572477933E-03,
*  0.15288767050877655684E-02,0.26245617274044295626E-02,
*  0.38516876166398709241E-02,0.51485584789781777618E-02,
*  0.64674198318036867274E-02,0.77683877779219912200E-02,
*  0.90161081951956431600E-02,0.10178877529236079733E-01,
*  0.11228632913408049354E-01,0.12141082601668299679E-01,
*  0.12895813488012114694E-01,0.13476374833816515982E-01,
*  0.13870351089139840997E-01,0.14069424957813575318E-01,
*  0.25157870384280661489E-04,0.18887326450650491366E-03,
*  0.46918492424785040975E-03,0.84057143271072246365E-03/
 DATA
* P(225),P(226),P(227),P(228),P(229),P(230),P(231),
* P(232),P(233),P(234),P(235),P(236),P(237),P(238),
* P(239),P(240),P(241),P(242),P(243),P(244),P(245),
* P(246),P(247),P(248),P(249),P(250),P(251),P(252)/
```

```
*  0.12843824718970101768E-02,0.17864463917586498247E-02,
*  0.23355251860571608737E-02,0.29217249379178197538E-02,
*  0.35362449977167777340E-02,0.41714193769840788528E-02,
*  0.48205888648512683476E-02,0.54778666939189508240E-02,
*  0.61379152800413850435E-02,0.67957855048827733948E-02,
*  0.74468208324075910174E-02,0.80866093647888599710E-02,
*  0.87109650797320868736E-02,0.93159241280693950932E-02,
*  0.98977475240487497440E-02,0.10452925722906011926E-01,
*  0.10978183152658912470E-01,0.11470482114693874380E-01,
*  0.11927026053019270040E-01,0.12345262372243838455E-01,
*  0.12722884982732382906E-01,0.13057836688353048840E-01,
*  0.13348311463725179953E-01,0.13592756614812395910E-01,
*  0.13789874783240936517E-01,0.13938625738306850804E-01,
*  0.14038227896908623303E-01,0.14088159516508301065E-01/
 DATA
* P(253),P(254),P(255),P(256),P(257),P(258),P(259),
* P(260),P(261),P(262),P(263),P(264),P(265),P(266),
* P(267),P(268),P(269),P(270),P(271),P(272),P(273),
* P(274),P(275),P(276),P(277),P(278),P(279),P(280)/
*  0.99999759637974846462E 00,0.69379364324108267170E-05,
*  0.99994399620705437576E 00,0.53275293669780613125E-04,
*  0.99976049092443204733E 00,0.13575491094922871973E-03,
*  0.99938033802502358193E 00,0.24921240048299729402E-03,
*  0.99874561446809511470E 00,0.38974528447328229322E-03,
*  0.99780535449595727456E 00,0.55429531493037471492E-03,
*  0.99651414591489027385E 00,0.74028280424450333046E-03,
*  0.99483150280062100052E 00,0.94536151685852538246E-03,
*  0.99272134428278861533E 00,0.11674841174299594077E-02,
*  0.99015137040077015918E 00,0.14049079956551446427E-02,
*  0.98709252795403406719E 00,0.16561127281544526052E-02,
*  0.98351865757863272876E 00,0.19197129710138724125E-02,
*  0.97940628167086268381E 00,0.21944069253638388388E-02,
*  0.97473445975240266776E 00,0.24789582266575679307E-02/
 DATA
* P(281),P(282),P(283),P(284),P(285),P(286),P(287),
* P(288),P(289),P(290),P(291),P(292),P(293),P(294),
* P(295),P(296),P(297),P(298),P(299),P(300),P(301),
* P(302),P(303),P(304),P(305),P(306),P(307),P(308)/
*  0.96948465950245923177E 00,0.27721957645934509940E-02,
*  0.96364062156981213252E 00,0.30730184347025783234E-02,
*  0.95718821610986096274E 00,0.33803979910869203823E-02,
*  0.95011529752129487656E 00,0.36933779170256508183E-02,
*  0.94241156519108305981E 00,0.40110687240750233989E-02,
*  0.93406843615772578800E 00,0.43326409680929828545E-02,
*  0.92507893290707565236E 00,0.46573172997568547773E-02,
*  0.91543758715576504064E 00,0.49843645647655386012E-02,
*  0.90514035881326159519E 00,0.53130866051870565663E-02,
*  0.89418456833555902286E 00,0.56428181013844441585E-02,
*  0.88256884024734190684E 00,0.59729195655081658049E-02,
*  0.87029305554811390585E 00,0.63027734490857587172E-02,
*  0.85735831088623215653E 00,0.66317812429018878941E-02,
*  0.84376688267270860104E 00,0.69593614093904229394E-02/
```

```
      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
1     Q(I)=P(I)
      MULTICS=1
      RETURN
      END

      COMPLEX FUNCTION ZEX(B,NEW,R)
C--ZEX COMPUTES THE P(R) TERM WHICH IS
C  DOUBLE INTEGRATED OVER FINITE LIMITS.
C  IT IS PART OF THE EQUATION FOR THE
C  ELECTRIC FIELD OF AN ELECTRIC DIPOLE.
C
C       B   INDUCTION NUMBER
C       R   DISTANCE
C       NEW CONTROLS ZLAGHO INTEGRATION
C
      COMPLEX ZLAGHO,TWODEL3,ONE
      EXTERNAL F3
      COMMON/PARM/ISTEP,A1,A2,A3,A4,A5,M,TOL
      COMMON/CONST/DEL,DEL2,TWODEL3
      DATA ONE/(1.0,0.0)/
      ZEX=CMPLX(0.0,0.0)
      IF(M.EQ.1) GO TO 2
      ZEX=ZLAGHO(ALOG(B),F3,TOL,LW,NEW)/B
    2 ZEX=TWODEL3*ZEX+(ONE-(ONE+CMPLX(B,B))*CEXP(-CMPLX(B,B)))/R**3
      RETURN
      END

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

```
C             TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON
C             THE FUNCTION FUN AND PARAMETER X...IN GENERAL,
C           A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'
C           BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY
C           RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN
C           APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,
C           ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...
C       L=    RESULTING NO. FILTER WTS. USED IN THE VARIABLE
C             CONVOLUTION (L DEPENDS ON TOL AND FUN).
C           MIN.L=24 AND MAX.L=281--WHICH COULD
C             OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING
C             VERY FAST...
C
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZFOURO; THE FOURIER
C  TRANSFORM IS THEN ZFOURO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM
C  THIS ROUTINE....
C
C--USAGE-- 'ZFOURO' IS CALLED AS FOLLOWS:
C     ...
C     COMPLEX Z,ZFOURO,ZF
C     EXTERNAL ZF
C     ...
C     Z=ZFOURO(ALOG(B),ZF,TOL,L)/B
C     ...
C     END
C     COMPLEX FUNCTION ZF(G)
C     ...USER SUPPLIED CODE...
C     END
C
C--NOTES:
C       (1). EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THIS SUBPROGRAM;
C            THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS
C            EXP-UNDERFLOW'S TO 0.0......
C       (2).   THIS SUBPROGRAM IS AN ANSI REVISION OF THE ORIGINAL
C              PUBLISHED VERSION IN NTIS REPT. PB-242-800, P.54-58;
C              IMPROVEMENTS HAVE BEEN MADE IN OVERALL EXECUTION TIME,
C              HOWEVER, THE CALLING SEQUENCE AND FILTER WEIGHTS
C              WERE NOT CHANGED.
C       (3).  ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED
C             TO SAVE STORAGE.
        COMPLEX FUN,C,CMAX
        DOUBLE PRECISION A,E,ER,Y1,Y
        DIMENSION T(2),TMAX(2)
        DIMENSION WT(281),W1(76),W2(76),W3(76),W4(53)
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),
     1 (WT(229),W4(1))
        EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))
        DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/
C--COS-EXTENDED FILTER WEIGHT ARRAYS:
      DATA W1/
     1 5.1178101E-14, 2.9433849E-14, 2.5492522E-14, 1.9034819E-14,
     2 6.4179780E-14, 1.3085746E-15, 1.1989957E-13,-1.2216234E-14,
```

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

```
    6 1.5875917E-06,-1.1980090E-06, 9.0398030E-07,-6.8208296E-07,
    7 5.1458650E-07,-3.8817581E-07, 2.9272267E-07,-2.2067921E-07,
    8 1.6623514E-07,-1.2514102E-07, 9.4034535E-08,-7.0556837E-08,
    9 5.2741581E-08,-3.9298610E-08, 2.9107255E-08,-2.1413893E-08,
    1 1.5742032E-08,-1.1498608E-08, 8.7561571E-09,-7.2959446E-09/
     DATA W4/
    1 6.8816619E-09,-8.9679825E-09, 1.4258275E-08,-1.9564299E-08,
    2 2.0235313E-08,-1.4725545E-08, 5.4632820E-09, 3.5995580E-09,
    3-9.5287133E-09, 1.1460041E-08,-1.0250532E-08, 7.4641748E-09,
    4-4.4703465E-09, 2.0499053E-09,-4.4806353E-10,-4.0374336E-10,
    5 7.0321001E-10,-6.7067960E-10, 4.9130404E-10,-2.8840747E-10,
    6 1.2373144E-10,-1.5260443E-11,-4.2027559E-11, 6.1885474E-11,
    7-5.9273937E-11, 4.6588766E-11,-3.2054182E-11, 1.9831637E-11,
    8-1.1210098E-11, 5.9567021E-12,-3.2427812E-12, 2.1353868E-12,
    9-1.8476851E-12, 1.8438474E-12,-1.8362842E-12, 1.7241847E-12,
    1-1.5161479E-12, 1.2627657E-12,-1.0129176E-12, 7.9578625E-13,
    2-6.2131435E-13, 4.8745900E-13,-3.8703630E-13, 3.1172547E-13,
    3-2.5397802E-13, 2.0824130E-13,-1.7123163E-13, 1.4113344E-13,
    4-1.1687986E-13, 9.7664016E-14,-8.2977176E-14, 7.2515267E-14,
    5-5.6047478E-14/
C--$$ENDATA
C
      A=DBLE(EXP(-X-30.3025124))
      ZFOURO=(0.0,0.0)
      CMAX=(0.0,0.0)
      L=22
      Y1=A*0.71633581334461667811 D+13
      Y=Y1
      DO 110 I=149,170
      Y=Y*E
      C=FUN(SNGL(Y))*WT(I)
      ZFOURO=ZFOURO+C
      TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))
  110 CONTINUE
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150
      CMAX=TOL*CMAX
      DO 120 I=171,281
      Y=Y*E
      C=FUN(SNGL(Y))*WT(I)
      ZFOURO=ZFOURO+C
      L=L+1
      IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130
  120 CONTINUE
  130 Y=Y1*E
      DO 140 I=1,148
      Y=Y*ER
      C=FUN(SNGL(Y))*WT(149-I)
      ZFOURO=ZFOURO+C
      L=L+1
      IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190
  140 CONTINUE
```

```
      GO TO 190
  150 Y=A
      DO 160 I=1,148
      Y=Y*E
      C=FUN(SNGL(Y))*WT(I)
      ZFOUR0=ZFOUR0+C
      L=L+1
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170
  160 CONTINUE
  170 Y=A*0.3120389295208079937 D+25
      DO 180 I=1,111
      Y=Y*ER
      C=FUN(SNGL(Y))*WT(282-I)
      ZFOUR0=ZFOUR0+C
      L=L+1
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190
  180 CONTINUE
  190 RETURN
      END

      COMPLEX FUNCTION ZHANK0(X,FUN,TOL,L)
C--REVISED VERSION: 12-13-76 -- SEE NOTE(2) BELOW.
C--INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*J0(G*B)*DG' DEFINED AS THE
C  COMPLEX HANKEL TRANSFORM OF ORDER 0 AND ARGUMENT X(=ALOG(B))
C  BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....
C
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.
C
C--PARAMETERS:
C
C     X     = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)
C                OF A REAL ARGUMENT G.
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN
C                  CALLING PROGRAM AND IN SUBPROGRAM FUN.
C                  THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE
C                  DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...
C                FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RHANK0' IS ADVISED;
C                HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE
C                INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G))
C       TOL=       REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,
C                  IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.
C                  THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,
C                  TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON
C                  THE FUNCTION FUN AND PARAMETER X...IN GENERAL,
C                A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'
C                BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY
C                RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN
C                APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,
C                ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...
C       L=         RESULTING NO. FILTER WTS. USED IN THE VARIABLE
```

```
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).
C               MIN.L=20 AND MAX.L=193--WHICH COULD
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING
C                 VERY FAST...
C
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZHANKO; THE HANKEL
C  TRANSFORM IS THEN ZHANKO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM
C  THIS ROUTINE....
C
C--USAGE-- 'ZHANKO' IS CALLED AS FOLLOWS:
C     ...
C     COMPLEX Z,ZHANKO,ZF
C     EXTERNAL ZF
C     ...
C     Z=ZHANKO(ALOG(B),ZF,TOL,L)/B
C     ...
C     END
C     COMPLEX FUNCTION ZF(G)
C     ...USER SUPPLIED CODE...
C     END
C
C--NOTES:
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THIS SUBPROGRAM;
C             THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS
C             EXP-UNDERFLOW'S TO 0.0......
C       (2).  THIS SUBPROGRAM IS AN ANSI REVISION OF THE ORIGINAL
C             PUBLISHED VERSION IN NTIS REPT. PB-242-800, P.45-48;
C             IMPROVEMENTS HAVE BEEN MADE IN OVERALL EXECUTION TIME,
C             HOWEVER, THE CALLING SEQUENCE AND FILTER WEIGHTS
C             WERE NOT CHANGED.
C       (3).  ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED
C             TO SAVE STORAGE.
        COMPLEX FUN,C,CMAX
        DOUBLE PRECISION A,E,ER,Y1,Y
        DIMENSION T(2),TMAX(2)
        DIMENSION WT(193),W1(76),W2(76),W3(41)
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1))
        EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))
        DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/
C--J0-EXTENDED FILTER WEIGHT ARRAYS:
      DATA W1/
     1 5.8565723E-08, 7.1143477E-11,-7.8395565E-11, 8.7489547E-11,
     2-8.9007811E-11, 9.8790055E-11,-9.8675347E-11, 1.1118797E-10,
     3-1.0893474E-10, 1.2543400E-10,-1.1979399E-10, 1.4200767E-10,
     4-1.3106341E-10, 1.6153229E-10,-1.4238602E-10, 1.8486236E-10,
     5-1.5315381E-10, 2.1319755E-10,-1.6238115E-10, 2.4824144E-10,
     6-1.6850378E-10, 2.9243813E-10,-1.6909302E-10, 3.4934366E-10,
     7-1.6043759E-10, 4.2417082E-10,-1.3690001E-10, 5.2458440E-10,
     8-8.9946096E-11, 6.6188220E-10,-6.6964033E-12, 8.5276151E-10,
     9 1.3222770E-10, 1.1219600E-09, 3.5591442E-10, 1.5061956E-09,
     1 7.0795382E-10, 2.0600379E-09, 1.2535947E-09, 2.8646623E-09,
     2 2.0904225E-09, 4.0409101E-09, 3.3642886E-09, 5.7687700E-09,
```

```
     3 5.2930786E-09, 8.3164338E-09, 8.2021809E-09, 1.2083635E-08,
     4 1.2577400E-08, 1.7666303E-08, 1.9143895E-08, 2.5953011E-08,
     5 2.8983953E-08, 3.8268851E-08, 4.3712685E-08, 5.6590075E-08,
     6 6.5740136E-08, 8.3864288E-08, 9.8662323E-08, 1.2448811E-07,
     7 1.4784461E-07, 1.8501974E-07, 2.2129198E-07, 2.7524203E-07,
     8 3.3094739E-07, 4.0974828E-07, 4.9462868E-07, 6.1030809E-07,
     9 7.3891802E-07, 9.0939667E-07, 1.1034727E-06, 1.3554600E-06,
     1 1.6474556E-06, 2.0207696E-06, 2.4591294E-06, 3.0131400E-06/
      DATA W2/
     1 3.6701680E-06, 4.4934101E-06, 5.4770076E-06, 6.7015208E-06,
     2 8.1726989E-06, 9.9954201E-06, 1.2194425E-05, 1.4909101E-05,
     3 1.8194388E-05, 2.2239184E-05, 2.7145562E-05, 3.3174088E-05,
     4 4.0499452E-05, 4.9486730E-05, 6.0421440E-05, 7.3822001E-05,
     5 9.0141902E-05, 1.1012552E-04, 1.3448017E-04, 1.6428337E-04,
     6 2.0062570E-04, 2.4507680E-04, 2.9930366E-04, 3.6560582E-04,
     7 4.4651421E-04, 5.4541300E-04, 6.6612648E-04, 8.1365181E-04,
     8 9.9374786E-04, 1.2138120E-03, 1.4824945E-03, 1.8107657E-03,
     9 2.2115938E-03, 2.7012675E-03, 3.2991969E-03, 4.0295817E-03,
     1 4.9214244E-03, 6.0106700E-03, 7.3405529E-03, 8.9643708E-03,
     2 1.0946310E-02, 1.3365017E-02, 1.6314985E-02, 1.9910907E-02,
     3 2.4289325E-02, 2.9612896E-02, 3.6070402E-02, 4.3876936E-02,
     4 5.3264829E-02, 6.4465091E-02, 7.7664144E-02, 9.2918324E-02,
     5 1.1000121E-01, 1.2811102E-01, 1.4543025E-01, 1.5832248E-01,
     6 1.6049224E-01, 1.4170064E-01, 8.8788108E-02,-1.1330934E-02,
     7-1.5331864E-01,-2.9094670E-01,-2.9084655E-01,-2.9708834E-02,
     8 3.9009601E-01, 1.7999785E-01,-4.1858139E-01, 1.5317216E-01,
     9 6.5184953E-02,-1.0751806E-01, 7.8429567E-02,-4.6019124E-02,
     1 2.5309571E-02,-1.3904823E-02, 7.8187120E-03,-4.5190369E-03/
      DATA W3/
     1 2.6729062E-03,-1.6073718E-03, 9.7715622E-04,-5.9804407E-04,
     2 3.6749320E-04,-2.2635296E-04, 1.3960805E-04,-8.6172618E-05,
     3 5.3212947E-05,-3.2867888E-05, 2.0304203E-05,-1.2543926E-05,
     4 7.7499633E-06,-4.7882430E-06, 2.9584108E-06,-1.8278645E-06,
     5 1.1293571E-06,-6.9778174E-07, 4.3113019E-07,-2.6637753E-07,
     6 1.6458373E-07,-1.0168954E-07, 6.2829807E-08,-3.8819969E-08,
     7 2.3985272E-08,-1.4819520E-08, 9.1563774E-09,-5.6573541E-09,
     8 3.4954514E-09,-2.1597005E-09, 1.3343946E-09,-8.2447148E-10,
     9 5.0941033E-10,-3.1474631E-10, 1.9447072E-10,-1.2015685E-10,
     1 7.4241055E-11,-4.5871468E-11, 2.8343095E-11,-1.7513137E-11,
     2 6.9049613E-12/
C--$$ENDATA
C
      A=DBLE(EXP(-X-26.3045570))
      ZHANK0=(0.0,0.0)
      CMAX=(0.0,0.0)
      L=18
      Y1=A*0.131201480802876898 8 D+12
      Y=Y1
      DO 110 I=129,146
      Y=Y*E
      C=FUN(SNGL(Y))*WT(I)
      ZHANK0=ZHANK0+C
```

```
      TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))
  110 CONTINUE
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150
      CMAX=TOL*CMAX
      DO 120 I=147,193
      Y=Y*E
      C=FUN(SNGL(Y))*WT(I)
      ZHANK0=ZHANK0+C
      L=L+1
      IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130
  120 CONTINUE
  130 Y=Y1*E
      DO 140 I=1,128
      Y=Y*ER
      C=FUN(SNGL(Y))*WT(129-I)
      ZHANK0=ZHANK0+C
      L=L+1
      IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190
  140 CONTINUE
      GO TO 190
  150 Y=A
      DO 160 I=1,128
      Y=Y*E
      C=FUN(SNGL(Y))*WT(I)
      ZHANK0=ZHANK0+C
      L=L+1
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170
  160 CONTINUE
  170 Y=A*0.70896679940719632O1 D+17
      DO 180 I=1,47
      Y=Y*ER
      C=FUN(SNGL(Y))*WT(194-I)
      ZHANK0=ZHANK0+C
      L=L+1
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190
  180 CONTINUE
  190 RETURN
      END

      COMPLEX FUNCTION ZLAGH0(X,FUN,TOL,L,NEW)
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*J0(G*B)*DG' DEFINED AS THE
C  COMPLEX HANKEL TRANSFORM OF ORDER 0 AND ARGUMENT X(=ALOG(B))
C  BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....
C
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.
C
C--PARAMETERS:
C
C     * X     = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM
```

```
C              'ZLAGHO' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,
C              SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,
C              THEN SUBPROGRAM 'ZHANKO' IS ADVISED FOR GENERAL USE.
C              (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)
C              OF A REAL ARGUMENT G.
C            NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN
C                CALLING PROGRAM AND IN SUBPROGRAM FUN.
C                THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE
C                DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...
C              FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RLAGHO' IS ADVISED;
C              HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G))
C       TOL=     REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,
C                IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.
C                THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,
C                TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON
C                THE FUNCTION FUN AND PARAMETER X...IN GENERAL,
C              A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'
C              BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY
C              RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN
C              APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,
C              ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...
C       L=       RESULTING NO. FILTER WTS. USED IN THE VARIABLE
C                CONVOLUTION (L DEPENDS ON TOL AND FUN).
C              MIN.L=20 AND MAX.L=193--WHICH COULD
C                OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING
C                VERY FAST...
C     * NEW=     1 IS NECESSARY 1ST TIME OR BRAND NEW X.
C              0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20
C                IS ASSUMED INTERNALLY BY THIS ROUTINE.
C                NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL
C                STILL ASSUME X=(LAST X)-0.20 ANYWAY...
C                IT IS THE USERS RESPONSIBILITY TO NORMALIZE
C                BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).
C              THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT
C                TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A
C                SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING
C                ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...
C                THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED
C                KERNELS WILL BE USED IN THE LAGGED CONVOLUTION
C                WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS
C                WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)
C
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZLAGHO; THE HANKEL
C  TRANSFORM IS THEN ZLAGHO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...
C
C--USAGE-- 'ZLAGHO' IS CALLED AS FOLLOWS:
C     ...
C     COMPLEX Z,ZLAGHO,ZF
C     EXTERNAL ZF
```

```
C     ...
C     Z=ZLAGHO(ALOG(B),ZF,TOL,L,NEW)/B
C     ...
C     END
C     COMPLEX FUNCTION ZF(G)
C     ...USER SUPPLIED CODE...
C     END
C
C--NOTES:
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:
C               ...
C               NB=AINT(5.*ALOG(BMAX/BMIN))+1
C               NB1=NB+1
C               X0=ALOG(BMAX)+.2
C               NEW=1
C               DO 1 J=1,NB
C               I=NB1-J
C               X=X0-.2*J
C               ARG(I)=EXP(X)
C               Z(I)=ZLAGHO(X,ZF,TOL,L,NEW)/ARG(I)
C     1         NEW=0
C               ...
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),Z(I),I=1,NB FOR
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW
C     BMAX,BMIN AND BY SETTING NEW=1....
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE
C
C
        COMPLEX FUN,C,CMAX,SAVE
        DIMENSION KEY(193),SAVE(193),T(2),TMAX(2)
        DIMENSION YT(193),Y1(76),Y2(76),Y3(41)
        EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))
        EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1))
C--J0-EXTENDED FILTER WEIGHT ARRAYS:
      DATA Y1/
     1 5.8565723E-08, 7.1143477E-11,-7.8395565E-11, 8.7489547E-11,
     2-8.9007811E-11, 9.8790055E-11,-9.8675347E-11, 1.1118797E-10,
     3-1.0893474E-10, 1.2543400E-10,-1.1979399E-10, 1.4200767E-10,
     4-1.3106341E-10, 1.6153229E-10,-1.4238602E-10, 1.8486236E-10,
```

```
     5-1.5315381E-10, 2.1319755E-10,-1.6238115E-10, 2.4824144E-10,
     6-1.6850378E-10, 2.9243813E-10,-1.6909302E-10, 3.4934366E-10,
     7-1.6043759E-10, 4.2417082E-10,-1.3690001E-10, 5.2458440E-10,
     8-8.9946096E-11, 6.6188220E-10,-6.6964033E-12, 8.5276151E-10,
     9 1.3222770E-10, 1.1219600E-09, 3.5591442E-10, 1.5061956E-09,
     1 7.0795382E-10, 2.0600379E-09, 1.2535947E-09, 2.8646623E-09,
     2 2.0904225E-09, 4.0409101E-09, 3.3642886E-09, 5.7687700E-09,
     3 5.2930786E-09, 8.3164338E-09, 8.2021809E-09, 1.2083635E-08,
     4 1.2577400E-08, 1.7666303E-08, 1.9143895E-08, 2.5953011E-08,
     5 2.8983953E-08, 3.8268851E-08, 4.3712685E-08, 5.6590075E-08,
     6 6.5740136E-08, 8.3864288E-08, 9.8662323E-08, 1.2448811E-07,
     7 1.4784461E-07, 1.8501974E-07, 2.2129198E-07, 2.7524203E-07,
     8 3.3094739E-07, 4.0974828E-07, 4.9462868E-07, 6.1030809E-07,
     9 7.3891802E-07, 9.0939667E-07, 1.1034727E-06, 1.3554600E-06,
     1 1.6474556E-06, 2.0207696E-06, 2.4591294E-06, 3.0131400E-06/
      DATA Y2/
     1 3.6701680E-06, 4.4934101E-06, 5.4770076E-06, 6.7015208E-06,
     2 8.1726989E-06, 9.9954201E-06, 1.2194425E-05, 1.4909101E-05,
     3 1.8194388E-05, 2.2239184E-05, 2.7145562E-05, 3.3174088E-05,
     4 4.0499452E-05, 4.9486730E-05, 6.0421440E-05, 7.3822001E-05,
     5 9.0141902E-05, 1.1012552E-04, 1.3448017E-04, 1.6428337E-04,
     6 2.0062570E-04, 2.4507680E-04, 2.9930366E-04, 3.6560582E-04,
     7 4.4651421E-04, 5.4541300E-04, 6.6612648E-04, 8.1365181E-04,
     8 9.9374786E-04, 1.2138120E-03, 1.4824945E-03, 1.8107657E-03,
     9 2.2115938E-03, 2.7012675E-03, 3.2991969E-03, 4.0295817E-03,
     1 4.9214244E-03, 6.0106700E-03, 7.3405529E-03, 8.9643708E-03,
     2 1.0946310E-02, 1.3365017E-02, 1.6314985E-02, 1.9910907E-02,
     3 2.4289325E-02, 2.9612896E-02, 3.6070402E-02, 4.3876936E-02,
     4 5.3264829E-02, 6.4465091E-02, 7.7654144E-02, 9.2918324E-02,
     5 1.1000121E-01, 1.2811102E-01, 1.4543025E-01, 1.5832248E-01,
     6 1.6049224E-01, 1.4170064E-01, 8.8788108E-02,-1.1330934E-02,
     7-1.5331864E-01,-2.9094670E-01,-2.9084655E-01,-2.9708834E-02,
     8 3.9009601E-01, 1.7999785E-01,-4.1858139E-01, 1.5317216E-01,
     9 6.5184953E-02,-1.0751806E-01, 7.8429567E-02,-4.6019124E-02,
     1 2.5309571E-02,-1.3904823E-02, 7.8187120E-03,-4.5190369E-03/
      DATA Y3/
     1 2.6729062E-03,-1.6073718E-03, 9.7715622E-04,-5.9804407E-04,
     2 3.6749320E-04,-2.2635296E-04, 1.3960805E-04,-8.6172618E-05,
     3 5.3212947E-05,-3.2867888E-05, 2.0304203E-05,-1.2543926E-05,
     4 7.7499633E-06,-4.7882430E-06, 2.9584108E-06,-1.8278645E-06,
     5 1.1293571E-06,-6.9778174E-07, 4.3113019E-07,-2.6637753E-07,
     6 1.6458373E-07,-1.0168954E-07, 6.2829807E-08,-3.8819969E-08,
     7 2.3985272E-08,-1.4819520E-08, 9.1563774E-09,-5.6573541E-09,
     8 3.4954514E-09,-2.1597005E-09, 1.3343946E-09,-8.2447148E-10,
     9 5.0941033E-10,-3.1474631E-10, 1.9447072E-10,-1.2015685E-10,
     1 7.4241055E-11,-4.5871468E-11, 2.8343095E-11,-1.7513137E-11,
     2 6.9049613E-12/
C--$$ENDATA
C
      IF(NEW) 10,30,10
10    LAG=-1
      X0=-X-26.30455704
```

```
      DO 20 IR=1,193
20    KEY(IR)=0
30    LAG=LAG+1
      ZLAGH0=(0.0,0.0)
      CMAX=(0.0,0.0)
      L=0
      ASSIGN 110 TO M
      I=129
      GO TO 200
110   TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))
      I=I+1
      IF(I.LE.146) GO TO 200
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150
      CMAX=TOL*CMAX
      ASSIGN 120 TO M
      I=128
      GO TO 200
120   IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130
      I=I-1
      IF(I.GT.0) GO TO 200
130   ASSIGN 140 TO M
      I=147
      GO TO 200
140   IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190
      I=I+1
      IF(I.LE.193) GO TO 200
      GO TO 190
150   ASSIGN 160 TO M
      I=1
      GO TO 200
160   IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170
      I=I+1
      IF(I.LE.128) GO TO 200
170   ASSIGN 180 TO M
      I=193
      GO TO 200
180   IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190
      I=I-1
      IF(I.GE.147) GO TO 200
190   RETURN
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)
200   LOOK=I+LAG
      IQ=LOOK/194
      IR=MOD(LOOK,194)
      IF(IR.EQ.0) IR=1
      IROLL=IQ*193
      IF(KEY(IR).LE.IROLL) GO TO 220
210   C=SAVE(IR)*YT(I)
      ZLAGH0=ZLAGH0+C
      L=L+1
      GO TO M,(110,120,140,160,180)
```

```
220   KEY(IR)=IROLL+IR
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))
      GO TO 210
      END

      SUBROUTINE ZQUAD1(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)
      COMPLEX F,RESULT,FUNCT,FZERO,ACUM
      DIMENSION FUNCT(127),P(381),RESULT(8)
      COMMON/ZQUADP/P
C--FOLLOWING CALL ONLY FOR MULTICS SYSTEM:
      CALL ZBLOCK
      ICHECK = 0
C CHECK FOR TRIVIAL CASE.
      IF (A.EQ.B) GO TO 70
C SCALE FACTORS.
      SUM = (B+A)/2.0
      DIFF = (B-A)/2.0
C 1-POINT GAUSS
      FZERO = F(SUM)
      RESULT(1) = 2.0*FZERO*DIFF
      I = 0
      IOLD = 0
      INEW = 1
      K = 2
      ACUM = (0.0,0.0)
      GO TO 30
   10 IF (K.EQ.8) GO TO 50
      IF(INEW+IOLD.GE.MEV) GO TO 60
      K = K + 1
      ACUM = (0.0,0.0)
C CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.
      DO 20 J=1,IOLD
        I = I + 1
        ACUM = ACUM + P(I)*FUNCT(J)
   20 CONTINUE
C CONTRIBUTION FROM NEW FUNCTION VALUES.
   30 IOLD = IOLD + INEW
      DO 40 J=INEW,IOLD
        I = I + 1
        X = P(I)*DIFF
        FUNCT(J) = F(SUM+X) + F(SUM-X)
        I = I + 1
        ACUM = ACUM + P(I)*FUNCT(J)
   40 CONTINUE
      INEW = IOLD + 1
      I = I + 1
      RESULT(K) = (ACUM+P(I)*FZERO)*DIFF
C CHECK FOR CONVERGENCE.
      IF(ABS(REAL(RESULT(K))-REAL(RESULT(K-1))).LE.EPSIL*
     $ABS(REAL(RESULT(K))).AND.
     $ ABS(AIMAG(RESULT(K))-AIMAG(RESULT(K-1))).LE.EPSIL*
     $ABS(AIMAG(RESULT(K)))) GO TO 60
```

```
      GO TO 10
C CONVERGENCE NOT ACHIEVED.
   50 ICHECK = 1
C NORMAL TERMINATION.
   60 NPTS = INEW + IOLD
      RETURN
C TRIVIAL CASE
   70 K = 2
      RESULT(1) = (0.0,0.0)
      RESULT(2) = (0.0,0.0)
      NPTS = 0
      RETURN
      END

      SUBROUTINE ZQUAD2(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)
      COMPLEX F,RESULT,FUNCT,FZERO,ACUM
      DIMENSION FUNCT(127), P(381), RESULT(8)
      COMMON/ZQUADP/P
C--FOLLOWING CALL ONLY FOR MULTICS SYSTEM:
      CALL ZBLOCK
      ICHECK = 0
C CHECK FOR TRIVIAL CASE.
      IF (A.EQ.B) GO TO 70
C SCALE FACTORS.
      SUM = (B+A)/2.0
      DIFF = (B-A)/2.0
C 1-POINT GAUSS
      FZERO = F(SUM)
      RESULT(1) = 2.0*FZERO*DIFF
      I = 0
      IOLD = 0
      INEW = 1
      K = 2
      ACUM = (0.0,0.0)
      GO TO 30
   10 IF (K.EQ.8) GO TO 50
      IF(INEW+IOLD.GE.MEV) GO TO 60
      K = K + 1
      ACUM = (0.0,0.0)
C CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.
      DO 20 J=1,IOLD
        I = I + 1
        ACUM = ACUM + P(I)*FUNCT(J)
   20 CONTINUE
C CONTRIBUTION FROM NEW FUNCTION VALUES.
   30 IOLD = IOLD + INEW
      DO 40 J=INEW,IOLD
        I = I + 1
        X = P(I)*DIFF
        FUNCT(J) = F(SUM+X) + F(SUM-X)
        I = I + 1
        ACUM = ACUM + P(I)*FUNCT(J)
```

```
   40 CONTINUE
      INEW = IOLD + 1
      I = I + 1
      RESULT(K) = (ACUM+P(I)*FZERO)*DIFF
C CHECK FOR CONVERGENCE.
      IF(ABS(REAL(RESULT(K))-REAL(RESULT(K-1))).LE.EPSIL*
     $ABS(REAL(RESULT(K))).AND.
     $ ABS(AIMAG(RESULT(K))-AIMAG(RESULT(K-1))).LE.EPSIL*
     $ABS(AIMAG(RESULT(K)))) GO TO 60
      GO TO 10
C CONVERGENCE NOT ACHIEVED.
   50 ICHECK = 1
C NORMAL TERMINATION.
   60 NPTS = INEW + IOLD
      RETURN
C TRIVIAL CASE
   70 K = 2
      RESULT(1) = (0.0,0.0)
      RESULT(2) = (0.0,0.0)
      NPTS = 0
      RETURN
      END

      COMPLEX FUNCTION ZSUB1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)
      COMPLEX RELERR,F,RESULT,ESTIM,COMP
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION
C ALGORITHM ZQUAD1, TOGETHER WITH, IF NECESSARY, A NON-
C ADAPTIVE SUBDIVISION PROCESS.
      DIMENSION RESULT(8)
      INTEGER BAD, OUT
      LOGICAL RHS
      EXTERNAL F
      DATA NMAX/4096/
      CALL ZQUAD1(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)
      ZSUB1 = RESULT(K)
      RELERR = (0.0,0.0)
      IF(REAL(ZSUB1).NE.0.0.AND.AIMAG(ZSUB1).NE.0.0) RELERR=
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUB1),
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUB1))
C CHECK IF SUBDIVISION IS NEEDED.
      IF (ICHECK.EQ.0) RETURN
C SUBDIVIDE
      ESTIM=ZSUB1*EPSIL
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))
      IC = 1
      RHS = .FALSE.
      N = 1
      H = B - A
      BAD = 1
   10 ZSUB1 = (0.0,0.0)
      RELERR = (0.0,0.0)
```

```
      H = H*0.5
      N = N + N
C INTERVAL (A,B) DIVIDED INTO N EQUAL SUBINTERVALS.
C INTEGRATE OVER SUBINTERVALS BAD TO (BAD+1) WHERE TROUBLE
C HAS OCCURRED.
      M1 = BAD
      M2 = BAD + 1
      OUT = 1
      GO TO 50
C INTEGRATE OVER SUBINTERVALS 1 TO (BAD-1)
   20 M1 = 1
      M2 = BAD - 1
      RHS = .FALSE.
      OUT = 2
      GO TO 50
C INTEGRATE OVER SUBINTERVALS (BAD+2) TO N.
   30 M1 = BAD + 2
      M2 = N
      OUT = 3
      GO TO 50
C SUBDIVISION RESULT
   40 ICHECK = IC
      IF(REAL(ZSUB1).EQ.0.0) GO TO 42
      IF(AIMAG(ZSUB1).EQ.0.0) GO TO 44
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB1)),
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUB1)))
      RETURN
   42 IF(AIMAG(ZSUB1).EQ.0.0) GO TO 46
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUB1)))
      RETURN
   44 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB1)),0.0)
      RETURN
   46 RELERR=(0.0,0.0)
      RETURN
C INTEGRATE OVER SUBINTERVALS M1 TO M2.
   50 IF (M1.GT.M2) GO TO 90
      DO 80 JJ=M1,M2
        J = JJ
C EXAMINE FIRST THE LEFT OR RIGHT HALF OF THE SUBDIVIDED
C TROUBLESOME INTERVAL DEPENDING ON THE OBSERVED TREND.
        IF (RHS) J = M2 + M1 - JJ
        ALPHA = A + H*(J-1)
        BETA = ALPHA + H
        CALL ZQUAD1(ALPHA, BETA, RESULT, M, EPSIL, NF, ICHECK, F,MEV)
        COMP = (RESULT(M)-RESULT(M-1))
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))
        NPTS = NPTS + NF
      IF(NPTS.GE.MEV) GO TO 70
        IF (ICHECK.NE.1) GO TO 70
      IF(REAL(COMP).LE.REAL(ESTIM).AND.
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 100
C SUBINTERVAL J HAS CAUSED TROUBLE.
```

```
C CHECK IF FURTHER SUBDIVISION SHOULD BE CARRIED OUT.
         IF (N.EQ.NMAX) GO TO 60
         BAD = 2*J - 1
         RHS = .FALSE.
         IF ((J-2*(J/2)).EQ.0) RHS = .TRUE.
         GO TO 10
   60    IC = -IABS(IC)
   70    ZSUB1 = ZSUB1 + RESULT(M)
   80 CONTINUE
      RELERR = RELERR + COMP
90    IF(OUT-2) 20,30,40
C RELAXED CONVERGENCE
  100 IC = ISIGN(2,IC)
      GO TO 70
      END

      COMPLEX FUNCTION ZSUB2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)
      COMPLEX RELERR,F,RESULT,ESTIM,COMP
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION
C ALGORITHM ZQUAD2, TOGETHER WITH, IF NECESSARY, A NON-
C ADAPTIVE SUBDIVISION PROCESS.
      DIMENSION RESULT(8)
      INTEGER BAD, OUT
      LOGICAL RHS
      EXTERNAL F
      DATA NMAX/4096/
      CALL ZQUAD2(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)
      ZSUB2 = RESULT(K)
      RELERR = (0.0,0.0)
      IF(REAL(ZSUB2).NE.0.0.AND.AIMAG(ZSUB2).NE.0.0) RELERR=
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUB2),
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUB2))
C CHECK IF SUBDIVISION IS NEEDED.
      IF (ICHECK.EQ.0) RETURN
C SUBDIVIDE
      ESTIM=ZSUB2*EPSIL
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))
      IC = 1
      RHS = .FALSE.
      N = 1
      H = B - A
      BAD = 1
   10 ZSUB2 = (0.0,0.0)
      RELERR = (0.0,0.0)
      H = H*0.5
      N = N + N
C INTERVAL (A,B) DIVIDED INTO N EQUAL SUBINTERVALS.
C INTEGRATE OVER SUBINTERVALS BAD TO (BAD+1) WHERE TROUBLE
C HAS OCCURRED.
      M1 = BAD
      M2 = BAD + 1
```

```
      OUT = 1
      GO TO 50
C INTEGRATE OVER SUBINTERVALS 1 TO (BAD-1)
   20 M1 = 1
      M2 = BAD - 1
      RHS = .FALSE.
      OUT = 2
      GO TO 50
C INTEGRATE OVER SUBINTERVALS (BAD+2) TO N.
   30 M1 = BAD + 2
      M2 = N
      OUT = 3
      GO TO 50
C SUBDIVISION RESULT
   40 ICHECK = IC
      IF(REAL(ZSUB2).EQ.0.0) GO TO 42
      IF(AIMAG(ZSUB2).EQ.0.0) GO TO 44
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB2)),
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUB2)))
      RETURN
   42 IF(AIMAG(ZSUB2).EQ.0.0) GO TO 46
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUB2)))
      RETURN
   44 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB2)),0.0)
      RETURN
   46 RELERR=(0.0,0.0)
      RETURN
C INTEGRATE OVER SUBINTERVALS M1 TO M2.
   50 IF (M1.GT.M2) GO TO 90
      DO 80 JJ=M1,M2
        J = JJ
C EXAMINE FIRST THE LEFT OR RIGHT HALF OF THE SUBDIVIDED
C TROUBLESOME INTERVAL DEPENDING ON THE OBSERVED TREND.
        IF (RHS) J = M2 + M1 - JJ
        ALPHA = A + H*(J-1)
        BETA = ALPHA + H
        CALL ZQUAD2(ALPHA, BETA, RESULT, M, EPSIL, NF, ICHECK, F,MEV)
        COMP = (RESULT(M)-RESULT(M-1))
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))
        NPTS = NPTS + NF
      IF(NPTS.GE.MEV) GO TO 70
        IF (ICHECK.NE.1) GO TO 70
      IF(REAL(COMP).LE.REAL(ESTIM).AND.
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 100
C SUBINTERVAL J HAS CAUSED TROUBLE.
C CHECK IF FURTHER SUBDIVISION SHOULD BE CARRIED OUT.
        IF (N.EQ.NMAX) GO TO 60
        BAD = 2*J - 1
        RHS = .FALSE.
        IF ((J-2*(J/2)).EQ.0) RHS = .TRUE.
        GO TO 10
   60   IC = -IABS(IC)
```

```
   70   ZSUB2 = ZSUB2 + RESULT(M)
   80 CONTINUE
      RELERR = RELERR + COMP
90    IF(OUT-2) 20,30,40
C RELAXED CONVERGENCE
  100 IC = ISIGN(2,IC)
      GO TO 70
      END

      COMPLEX FUNCTION ZSUBA1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)
      COMPLEX RELERR,F,RESULT,ESTIM,COMP
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION
C ALGORITHM ZQUAD1 TOGETHER WITH,  IF NECESSARY AN ADAPTIVE
C SUBDIVISION PROCESS.  IT IS GENERALLY MORE EFFICIENT THAN
C THE NON-ADAPTIVE ALGORITHM ZSUB1 BUT IS LIKELY TO BE LESS
C RELIABLE(SEE COMP.J.,14,189,1971).
      DIMENSION RESULT(8), STACK(100)
      EXTERNAL F
      DATA ISMAX/100/
      CALL ZQUAD1(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)
      ZSUBA1 = RESULT(K)
      RELERR = (0.0,0.0)
      IF(REAL(ZSUBA1).NE.0.0.AND.AIMAG(ZSUBA1).NE.0.0) RELERR=
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUBA1),
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUBA1))
C CHECK IF SUBDIVISION IS NEEDED
      IF (ICHECK.EQ.0) RETURN
C SUBDIVIDE
      ESTIM=ZSUBA1*EPSIL
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))
      RELERR = (0.0,0.0)
      ZSUBA1 = (0.0,0.0)
      IS = 1
      IC = 1
      SUB1 = A
      SUB3 = B
   10 SUB2 = (SUB1+SUB3)*0.5
      CALL  ZQUAD1(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)
      NPTS = NPTS + NF
      IF(NPTS.GE.MEV) GO TO 50
      COMP = (RESULT(K)-RESULT(K-1))
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))
      IF (ICHECK.EQ.0) GO TO 30
      IF(REAL(COMP).LE.REAL(ESTIM).AND.
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 70
      IF (IS.GE.ISMAX) GO TO 20
C STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION
      STACK(IS) = SUB1
      IS = IS + 1
      STACK(IS) = SUB2
      IS = IS + 1
```

```
      GO TO 40
   20 IC = -IABS(IC)
   30 ZSUBA1 = ZSUBA1 + RESULT(K)
      RELERR = RELERR + COMP
   40 CALL ZQUAD1(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)
      NPTS = NPTS + NF
      IF(NPTS.GE.MEV) GO TO 50
      COMP = (RESULT(K)-RESULT(K-1))
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))
      IF (ICHECK.EQ.0) GO TO 50
      IF(REAL(COMP).LE.REAL(ESTIM).AND.
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 80
C SUBDIVIDE INTERVAL (SUB2,SUB3)
      SUB1 = SUB2
      GO TO 10
   50 ZSUBA1 = ZSUBA1 + RESULT(K)
      RELERR = RELERR + COMP
      IF(NPTS.GE.MEV) RETURN
      IF (IS.EQ.1) GO TO 60
C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED
      IS = IS - 1
      SUB3 = STACK(IS)
      IS = IS - 1
      SUB1 = STACK(IS)
      GO TO 10
C SUBDIVISION RESULT
   60 ICHECK = IC
      IF(REAL(ZSUBA1).EQ.0.0) GO TO 62
      IF(AIMAG(ZSUBA1).EQ.0.0) GO TO 64
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA1)),
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUBA1)))
      RETURN
   62 IF(AIMAG(ZSUBA1).EQ.0.0) GO TO 66
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUBA1)))
      RETURN
   64 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA1)),0.0)
      RETURN
   66 RELERR=(0.0,0.0)
      RETURN
C RELAXED CONVERGENCE
   70 IC = ISIGN(2,IC)
      GO TO 30
   80 IC = ISIGN(2,IC)
      GO TO 50
      END

      COMPLEX FUNCTION ZSUBA2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)
      COMPLEX RELERR,F,RESULT,ESTIM,COMP
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION
C ALGORITHM ZQUAD2 TOGETHER WITH, IF NECESSARY AN ADAPTIVE
C SUBDIVISION PROCESS.  IT IS GENERALLY MORE EFFICIENT THAN
```

```
C THE NON-ADAPTIVE ALGORITHM ZSUB2 BUT IS LIKELY TO BE LESS
C RELIABLE(SEE COMP.J.,14,189,1971).
      DIMENSION RESULT(8), STACK(100)
      EXTERNAL F
      DATA ISMAX/100/
      CALL ZQUAD2(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)
      ZSUBA2 = RESULT(K)
      RELERR = (0.0,0.0)
      IF(REAL(ZSUBA2).NE.0.0.AND.AIMAG(ZSUBA2).NE.0.0) RELERR=
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUBA2),
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUBA2))
C CHECK IF SUBDIVISION IS NEEDED
      IF (ICHECK.EQ.0) RETURN
C SUBDIVIDE
      ESTIM=ZSUBA2*EPSIL
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))
      RELERR = (0.0,0.0)
      ZSUBA2 = (0.0,0.0)
      IS = 1
      IC = 1
      SUB1 = A
      SUB3 = B
   10 SUB2 = (SUB1+SUB3)*0.5
      CALL  ZQUAD2(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)
      NPTS = NPTS + NF
      IF(NPTS.GE.MEV) GO TO 50
      COMP = (RESULT(K)-RESULT(K-1))
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))
      IF (ICHECK.EQ.0) GO TO 30
      IF(REAL(COMP).LE.REAL(ESTIM).AND.
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 70
      IF (IS.GE.ISMAX) GO TO 20
C STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION
      STACK(IS) = SUB1
      IS = IS + 1
      STACK(IS) = SUB2
      IS = IS + 1
      GO TO 40
   20 IC = -IABS(IC)
   30 ZSUBA2 = ZSUBA2 + RESULT(K)
      RELERR = RELERR + COMP
   40 CALL ZQUAD2(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)
      NPTS = NPTS + NF
      IF(NPTS.GE.MEV) GO TO 50
      COMP = (RESULT(K)-RESULT(K-1))
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))
      IF (ICHECK.EQ.0) GO TO 50
      IF(REAL(COMP).LE.REAL(ESTIM).AND.
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 80
C SUBDIVIDE INTERVAL (SUB2,SUB3)
      SUB1 = SUB2
      GO TO 10
```

```
   50 ZSUBA2 = ZSUBA2 + RESULT(K)
      RELERR = RELERR + COMP
      IF(NPTS.GE.MEV) RETURN
      IF (IS.EQ.1) GO TO 60
C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED
      IS = IS - 1
      SUB3 = STACK(IS)
      IS = IS - 1
      SUB1 = STACK(IS)
      GO TO 10
C SUBDIVISION RESULT
   60 ICHECK = IC
      IF(REAL(ZSUBA2).EQ.0.0) GO TO 62
      IF(AIMAG(ZSUBA2).EQ.0.0) GO TO 64
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA2)),
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUBA2)))
      RETURN
   62 IF(AIMAG(ZSUBA2).EQ.0.0) GO TO 66
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUBA2)))
      RETURN
   64 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA2)),0.0)
      RETURN
   66 RELERR=(0.0,0.0)
      RETURN
C RELAXED CONVERGENCE
   70 IC = ISIGN(2,IC)
      GO TO 30
   80 IC = ISIGN(2,IC)
      GO TO 50
      END
```