77-784

Documentation for Subroutine REDUC3, an Algorithm for the Linear Filtering of Gridded Magnetic Data

Richard J. Blakely
U.S. Geological Survey
345 Middlefield Road
Menlo Park, California 94025

1. Abstract

Subroutine REDUC3 transforms a total field anomaly $h_1(x,y)$, measured on a horizontal and rectangular grid, into a new anomaly $h_2(x,y)$. This new anomaly is produced by the same source as $h_1(x,y)$, but (1) is observed at a different elevation, (2) has a source with a different direction of magnetization, and/or (3) has a different direction of residual field. Case 1 is tantamount to upward or downward continuation. Cases 2 and 3 are "reduction to the pole", if the new inclinations of both the magnetization and regional field are 90 degrees. REDUC3 is a filtering operation applied in the wave-number domain. It first Fourier transforms $h_1(x,y)$, multiplies by the appropriate filter, and inverse Fourier transforms the result to obtain $h_2(x,y)$. No assumptions are required about the shape of the source or how the intensity of magnetization varies within it.

2. Mathematical background

Consider a horizontal lamina (Fig. 1) located at depth w and spread with a distribution of magnetization

$$\vec{m}(x,y) = m(x,y)\ \hat{m}$$

The unit vector $\hat{m}$ has a constant direction over the plane, but the intensity of magnetization $m(x,y)$ varies arbitrarily. The direction of the residual field $\hat{h}$ is also assumed constant throughout the survey area. The total field anomaly observed on the w=0 plane is $h_1(x,y)$ and its Fourier transform is $H_1(k_x,k_y)$ where k_x and k_y are wavenumbers. There exists a response function $G_1(k_x,k_y)$ such that

$$H_1(k_x,k_y) = G_1(k_x,k_y)\ M(k_x,k_y) \qquad (1)$$

where $M(k_x,k_y)$ is the Fourier transform of $m(x,y)$.

Now consider an anomaly $h_2(x,y)$ produced by precisely the same distribution of magnetic material $m(x,y)$, but located either at a different depth w, magnetized in a different direction $\hat{m}$, and/or measured in a regional field with a different direction $\hat{h}$. For this new anomaly

$$H_2(k_x,k_y) = G_2(k_x,k_y)\ M(k_x,k_y) \qquad (2)$$

and combining equations (1) and (2) to eliminate the common term $M(k_x,k_y)$ yields

$$H_2(k_x,k_y) = H_1(k_x,k_y)\ G_2(k_x,k_y)/G_1(k_x,k_y) \qquad (3)$$

The ratio G_2/G_1 is a filter which transforms a measured anomaly $h_1(x,y)$ into the desired anomaly $h_2(x,y)$. The new anomaly is produced by the same planar source but differs either in w, $\hat{m}$, $\hat{h}$, or any combination of these.

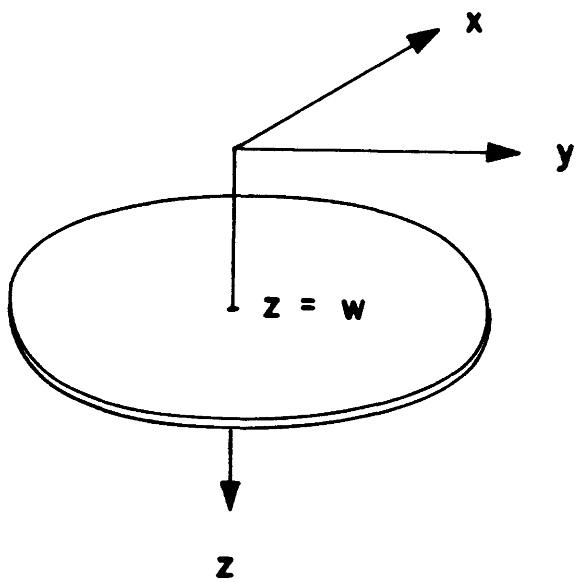

Figure 1. Source geometry to derive linear filters for magnetic anomalies.

Invoking the superposition principle, the anomaly produced by a volume of magnetic material is the integral of the anomalies produced by all of its laminae. Moreover, the Fourier transform of a sum of functions is equal to the sum of the individual Fourier transforms. Consequently, equation (3) exists not only for each of the infinite set of planes that compose a volume of material, but also for that volume. No knowledge is required about the shape of the volume or about the manner in which the magnetization is distributed inside the volume (Fig. 2). The required assumption is that $\hat{m}$ and $\hat{h}$ remain constant throughout the survey region.

The problem is thus reduced to the determination of $G(k_x,k_y)$, the response function of a planar source. A response function is the Fourier transform of the impulse response of the source. Because the source is an infinitely thin layer of dipoles, the impulsive source is a single dipole at depth w and the impulse response is the anomaly that it produces. Hence, $G(k_x,k_y)$ is the Fourier transform of the anomaly produced by a single dipole. The potential of a dipole observed at P(x,y,0) is

$$V = -\hat{m} \cdot \nabla_p (1/r)$$

where

$$r^2 = x^2 + y^2 + w^2$$

The total field anomaly is

$$\begin{aligned} g(x,y) &= -\hat{h} \cdot \nabla_p V \\ &= \hat{h} \cdot \nabla_p (\hat{m} \cdot \nabla_p (1/r)) \\ &= \frac{\partial^2}{\partial h\, \partial m} (1/r) \qquad (4) \end{aligned}$$

where $\frac{\partial}{\partial h}$ and $\frac{\partial}{\partial m}$ represent partial derivatives in the $\hat{h}$ and $\hat{m}$ directions respectively and are given by

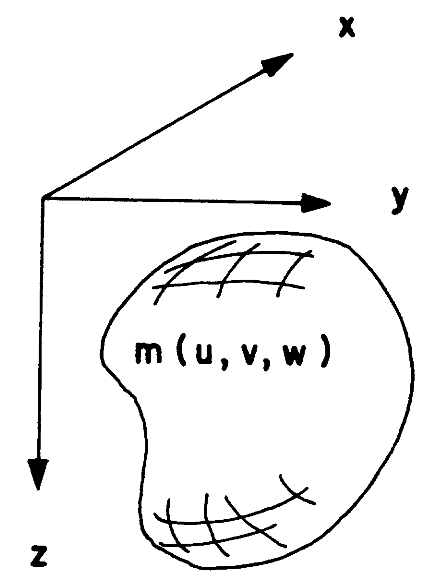

Figure 2. General magnetic source

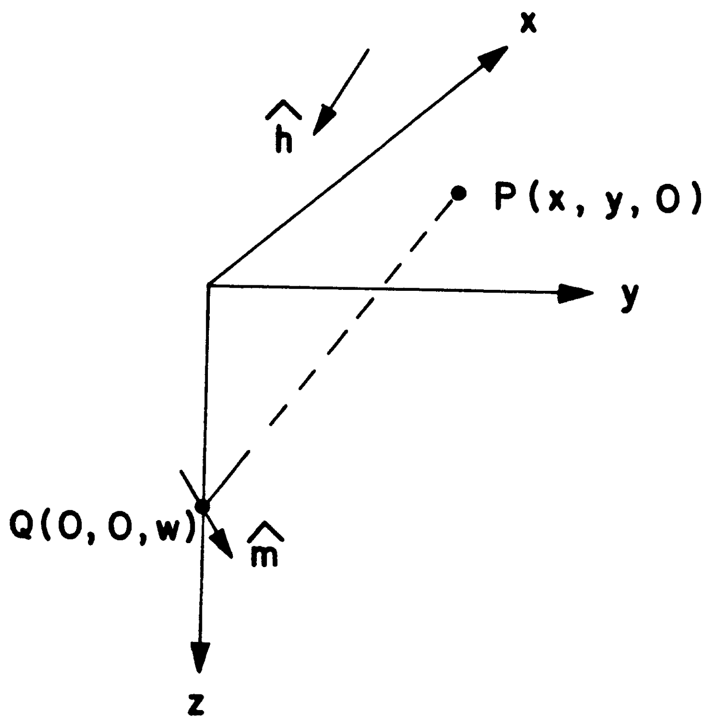

Figure 3. Geometry for dipolar source

$$\frac{\partial}{\partial h} = \hat{h}_x \frac{\partial}{\partial x} + \hat{h}_y \frac{\partial}{\partial y} + \hat{h}_z \frac{\partial}{\partial_z} \tag{5}$$

$$\frac{\partial}{\partial m} = \hat{m}_x \frac{\partial}{\partial x} + \hat{m}_y \frac{\partial}{\partial y} + \hat{m}_z \frac{\partial}{\partial z} \tag{6}$$

The traditional assumption, that the anomalous field is much smaller than the regional field, has been invoked in equation (4). We require the Fourier transform of g(x,y), which can be found from the transform of (1/r) using relations like

$$\begin{aligned}
\mathcal{F}\left[\frac{\partial^n}{\partial x^n}\frac{1}{r}\right] &= \left(ik_x\right)^n \mathcal{F}\left[\frac{1}{r}\right] \\
\mathcal{F}\left[\frac{\partial^n}{\partial y^n}\frac{1}{r}\right] &= \left(ik_y\right)^n \mathcal{F}\left[\frac{1}{r}\right] \\
\mathcal{F}\left[\frac{\partial^2}{\partial x \partial y}\frac{1}{r}\right] &= -k_x k_y \mathcal{F}\left[\frac{1}{r}\right] \\
\mathcal{F}\left[\frac{\partial}{\partial z}\frac{1}{r}\right] &= \frac{\partial}{\partial z} \mathcal{F}\left[\frac{1}{r}\right] \\
&= -\frac{\partial}{\partial w} \mathcal{F}\left[\frac{1}{r}\right]
\end{aligned} \tag{7}$$

(Bracewell, 1965) where $\mathcal{F}\left[f\right]$ denotes the two-dimensional Fourier transform of f. Now

$$\mathcal{F}\left[\frac{1}{r}\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{(x^2+y^2+w^2)^{\frac{1}{2}}} e^{-i(k_x x+k_y y)} dx\, dy \tag{8}$$

and converting to polar coordinates

$$\mathcal{F}\left[\frac{1}{r}\right] = \int_0^{2\pi}\int_0^{\infty} \frac{1}{(a^2+w^2)^{\frac{1}{2}}}\, e^{-i\,a\,k\cos\theta}\, a\, da\, d\theta \tag{9}$$

where

$$\vec{a} = (x,y)$$

$$\vec{k} = (k_x,k_y)$$

$$\vec{a} \cdot \vec{k} = a\,k\cos\theta = k_x\,x + k_y\,y$$

$$a = (x^2 + y^2)^{\frac{1}{2}}$$

$$k = (k_x{}^2 + k_y{}^2)^{\frac{1}{2}}$$

Substituting the zero-order Bessel function

$$J_o(z) = \frac{1}{2\pi}\int_0^{2\pi} e^{-i\,z\cos\theta}\, d\theta$$

equation (9) becomes

$$\mathcal{F}\left[\frac{1}{r}\right] = 2\pi\int_0^{\infty} \frac{1}{(a^2+w^2)^{\frac{1}{2}}}\, J_o(ak)\; a\; da \tag{10}$$

and the complicated two-dimensional Fourier transform of equation (8) has been reduced to a Hankel transform of zero order. Equation (10) is given by (see Bracewell (1965, p. 249) for example)

$$\mathcal{F}\left[\frac{1}{r}\right] = \frac{2\pi e^{-wk}}{k} \tag{11}$$

Combining equations (4), (5), (6), (7), and (11), and rearranging terms yields the Fourier transform of g(x,y):

$$G(k_x,k_y) = -\ 2\pi k \exp(-w\ k)\ \ \emptyset\ (k_x,k_y) \tag{12}$$

where

$$\emptyset\ (k_x,k_y) = \frac{a_1 k_x^2 + a_2 k_y^2 + a_3 k_x k_y}{k^2} + \frac{i\ (b_1 k_x + b_2 k_y)}{k}$$

$$a_1 = \hat{m}_x\ \hat{h}_x - \hat{m}_z\ \hat{h}_z$$

$$a_2 = \hat{m}_y\ \hat{h}_y - \hat{m}_z\ \hat{h}_z$$

$$a_3 = \hat{m}_x\ \hat{h}_y - \hat{m}_y\ \hat{h}_x$$

$$b_1 = -\ \hat{m}_x\ \hat{h}_z - \hat{m}_z\ \hat{h}_x$$

$$b_2 = -\ \hat{m}_y\ \hat{h}_z - \hat{m}_z\ \hat{h}_y$$

This is precisely the same result as equation (12) of Gunn (1975) and equation (43) of Spector and Bhattacharyya (1966) (except for an incorrect sign of their imaginary part).

Equation (12) describes the Fourier transform of the total field anomaly produced by a single dipole. This can be used with equation (3) to solve the initial problems. Consider upward continuation for example. If the new level of observation is higher than the initial level by an amount Δz, then

$$G_1(k_x,k_y) = -2\ \pi\ k \exp(-w\ k)\ \ \emptyset(k_x,k_y)$$

$$G_2(k_x,k_y) = -2\ \pi\ k \exp\ (-(w + \Delta z)k)\ \ \emptyset(k_x,k_y)$$

$$G_2/G_1 = \exp(-\ \Delta z\ k) \tag{13}$$

and equation (13) describes a filter for upward continuation. As stated earlier, this expression is valid for any three-dimensional source.

As a second example, consider reduction to the pole:

$$G_1(k_x,k_y) = -2\pi k \exp(-w k)\ \emptyset\ (k_x,k_y)$$

$$G_2(k_x,k_y) = -2\pi k \exp(-w k)\ (-1)$$

$$G_2/G_1 = -1/\emptyset(k_x,k_y) \tag{14}$$

and equation (14) describes a filter for reduction to the pole.

3. Program considerations

To perform the operation described by equation (3), REDUC3 must Fourier transform $h_1(x,y)$, calculate and apply G_2/G_1, and inverse Fourier transform to obtain $h_2(x,y)$. Fourier transforms are made with an external subroutine called FFT, developed by R. C. Singleton of Stanford Research Institute (Singleton, 1969). The two-dimensional anomaly $h_1(x,y)$ is input as a one-dimensional array H(I), I=1,2,...(NX) (NY), where the subscript increases first in the y direction and secondly in the x direction (fig. 4). Other input parameters are as follows:

NX - number of points in the x direction,

NY - number of points in the y direction,

DELTX - sample interval in the x direction,

DELTY - sample interval in the y direction,

AZIM - angle between the x axis and geodetic north,

MI1 - original inclination of magnetization,

MD1 - original declination of magnetization,

FI1 - original inclination of the regional field,

FD1 - original declination of the regional field,

MI2 - new inclination of magnetization,

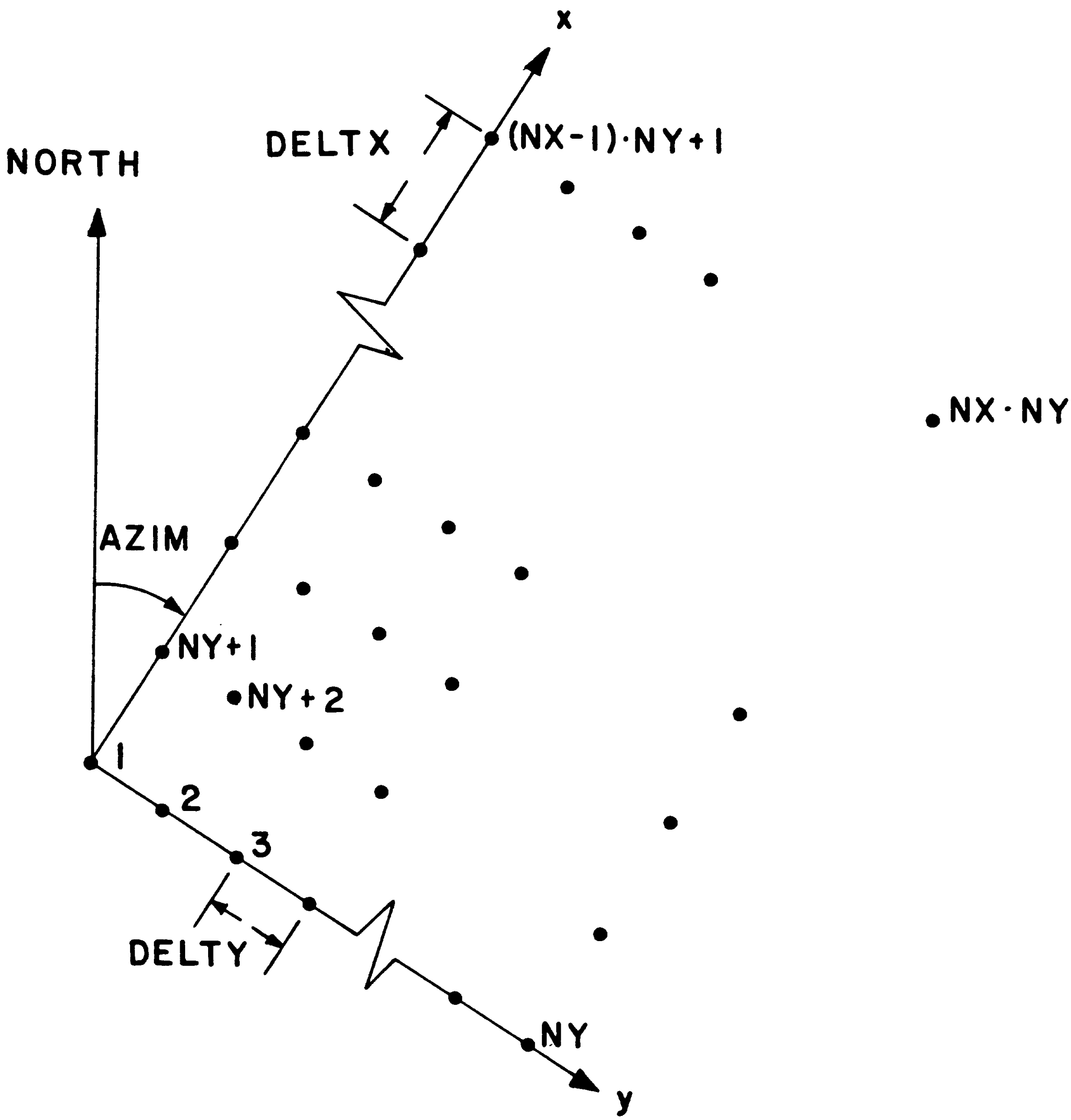

Figure 4. Definition of input array.

MD2 - new declination of magnetization,

FI2 - new inclination of the regional field,

FD2 - new declination of the regional field, and

DELTZ - new depth of source relative to original source.

Units of length (for parameters DELTX, DELTY, and DELTZ) are unimportant but must be consistent. Units of angle (for parameters MI1, MD1, FI1, FD1, MI2, MD2, FI2, FD2, and AZIM) are degrees. DELTZ should be positive for upward continuation.

Before Fourier transforming, H(I), I=1,2,...(NX)(NY) is copied to an internal array HREAL(I,J), I=1,2,...(NX). J=1,2,...(NY). It is this array along with a zero array HIMAG(I,J), I=1,2,...(NX), J=1,2,...(NY), that enters the Fourier transform algorithm. After inverse Fourier transforming, H(I) is replaced with the contents of HREAL(I,J) which corresponds to the new data $h_2(x,y)$. HREAL and HIMAG are presently dimensioned 64 x 64. This means that, in general, HREAL and HIMAG may be largely zeroes (if NX<64 and/or NY<64). This should not be a problem computationally, and because FFT is faster when the arrays are of length equal to a power of two, should not be a time consideration.

4. Testing of the subroutine

A driver program was written to test subroutine REDUC3. It computes a 19 x 19 "true" array of the vertical field due to a single vertical dipole. The dipole is at a depth of 2 km. Then it computes a number of "test" arrays with various dipole-depths w, magnetization directions $\hat{m}$ and regional field directions $\hat{h}$, and REDUC3 is called to try and convert the test data into the true data.

To compare the relative success of REDUC3, an error R is computed where R is given by

$$R^2 = \frac{\sum_{I=1}^{(NX)(NY)} [HTRUE(I)-H(I)]^2}{\sum_{I=1}^{(NX)(NY)} [HTRUE(I)]^2} \qquad (15)$$

The output from this series of tests is reproduced in section 8 following.

It can be seen from the following graphs (Fig. 5) that upward continuation is virtually error-free. The test included input values of DELTZ = .2, .4, .6, .8, and 1.0 km. The error is roughly linear with respect to DELTZ reaching a maximum of 2.4% at DELTZ = 1.

Reduction to the pole, on the other hand, suffers from substantial error when starting with fields with low inclinations (Fig. 5). Presumably this is due to the very singular nature of the filter near the origin. This can best be illustrated by rewriting equation (14) in polar coordinates

$$G_2/G_1 = -1/\emptyset\ (k_x,k_y)$$

$$= -(a_1 \cos^2 \lambda + a_2 \sin^2 \lambda + a_3 \cos \lambda \sin \lambda$$

$$+ i\ (b_1 \cos \lambda + b_2 \sin \lambda\ )$$

where

$$k_x = \rho \cos \lambda$$

$$k_y = \rho \sin \lambda$$

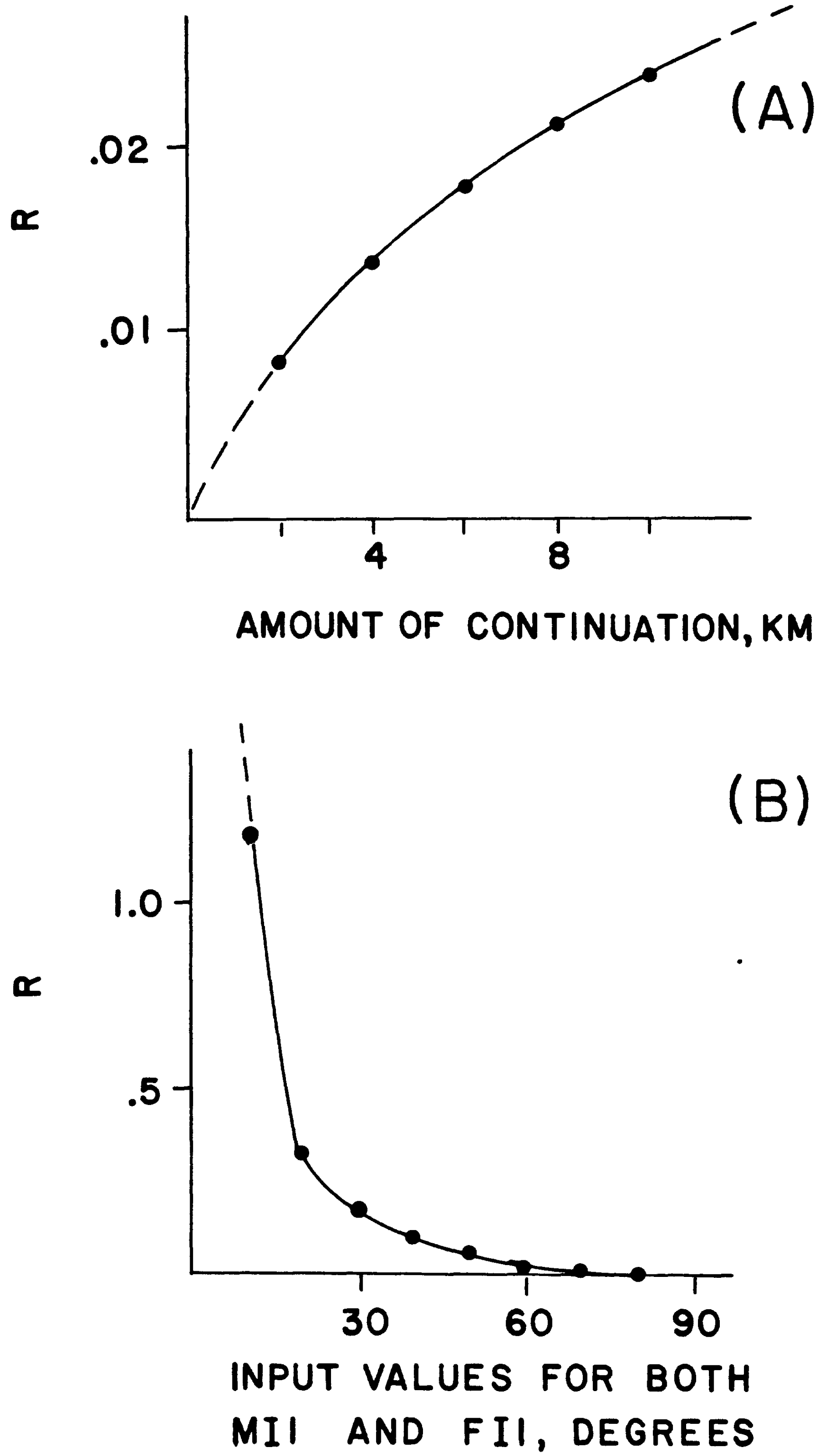

Figure 5. Error as a function of input values. a: upward continuation. b: reduction to the pole.

From equation (15), the filter G_2/G_1 has an infinite number of values at the origin depending on the direction from which the origin is approached. REDUC3 approximates G_2/G_1 at the origin by using the average of the values obtained when approaching the origin along the k_x and k_y axes. Nevertheless, G_2/G_1 is varying rapidly as a function of k_x and k_y in the vicinity of the origin. It is the inaccuracy of the digital approximation to G_2/G_1 in this region that produces the errors when the input anomaly has low inclinations.

5. Cost

In the test sample shown in section 8, a total of 13 different 19 x 19 arrays were operated on by REDUC3. This test took .02 minutes to compile and .06 minutes to execute on Stanford University's IBM 370/168. The total charge was $3.83. However, a major fraction of this cost is attributed to the driver program, which calculates the initial 19 x 19 arrays and determines relative errors.

6. References

Bracewell, Ron, The Fourier Transform and Its Applications, McGraw-Hill, New York, 381 p., 1965.

Gunn, P. J., Linear transformation of gravity and magnetic fields, Geophys. Prosp., 23, 300-312, 1975.

Singleton, R. C., An algorithm for computing the mixed radix fast Fourier transform, IEEE Trans. Audio and Electroacoustics, AU-17, 93-103, 1969.

Spector, A. and B. K. Bhattacharyya, Energy density spectrum and autocorrelation function of anomalies due to simple magnetic models, Geophys. Prosp., 14, 242-272, 1966.

7. Listing of REDUC3 (plus driver program and test data)

```
// JOB B44$XA,BLAKELY
/*JOBPARM LINECT=0
// EXEC FORTCG
//FORT.SYSIN DD *


      REAL MI,MD
      DIMENSION HTRUE(400),H(400),ERROR(400)
C
C     THIS PROGRAM IS A DRIVER TO TEST SUBROUTINE REDUC3.  IT FIRST
C   COMPUTES A TWO DIMENSIONAL ANOMALY DUE TO A SINGLE DIPOLE AT A
C   DEPTH OF 2 KM WITH VERTICAL INCLINATIONS OF REGIONAL FIELD AND
C   MAGNETIZATION.  IT THEN INPUTS A SET OF DIPOLE PARAMETERS (DEPTH,
C   INCLINATIONS, DECLINATIONS) WHICH ARE COMPUTED INTO A TEST ANOMALY.
C   REDUC3 IS THEN CALLED TO TRANSFORM THE TEST ANOMALY INTO THE
C   TRUE ANOMALY AND THE RMS ERROR IS COMPUTED.  IT LOOPS BACK FOR
C   A NEW SET OF PARAMETERS, AND CONTINUES IN THIS FASHION UNTIL AN
C   END-OF-FILE IS ENCOUNTERED.  DATA ARRAYS ARE 19X19 IN SIZE,
C   BUT ONLY THE CENTRAL 13X13 BLOCK IS PRINTED.
C
      CALL DIPOLE(90.,0.,90.,0.,2.,-4.5,.5,19,-4.5,.5,19,HTRUE)
      SQET=0.
      DO 10 K=1,361
   10 SQET=SQET+HTRUE(K)**2
      WRITE(6,100)
  100 FORMAT(1H1,' TEST OF REDUC3...',//,'  TRUE ANOMALY...'/)
      DO 50 I=4,16
      K1=(I-1)*19+4
      K2=K1+12
   50 WRITE(6,101)(HTRUE(K),K=K1,K2)
  101 FORMAT(1X,13F7.0)
 1001 READ(5,102,END=1000)MI,MD,FI,FD,DEPTH
  102 FORMAT(5F5.0)
      WRITE(6,103)MI,MD,FI,FD,DEPTH
  103 FORMAT(//,' INPUT PARAMETERS TO CREATE TEST ARRAY...',//,
     1  5X,'MI = ',F5.0,'  MD = ',F5.0,'  FI = ',F5.0,'  FD = ',
     2  F5.0,'  DEPTH = ',F5.1/)
      CALL DIPOLE(MI,MD,FI,FD,DEPTH,-4.5,.5,19,-4.5,.5,19,H)
      DZ=2.-DEPTH
      CALL REDUC3(19,19,.5,.5,MI,MD,FI,FD,90.,0.,90.,0.,0.,DZ,H,RATIO)
      SQEE=0.
      DO 20 K=1,361
      ERROR(K)=HTRUE(K)-H(K)
   20 SQEE=SQEE+ERROR(K)**2
      RMS1=SQRT(SQEE/SQET)
      WRITE(6,105)RMS1
  105 FORMAT(/,' NORMALIZED RMS=',G10.3,//,'  ERROR ARRAY...'/)
      DO 60 I=4,16
      K1=(I-1)*19+4
      K2=K1+12
   60 WRITE(6,101)(ERROR(K),K=K1,K2)
      GO TO 1001
 1000 STOP
      END


      SUBROUTINE DIPOLE(MI,MD,FI,FD,DEPTH,XSTART,XDELT,NX,YSTART,YDELT,
     1  NY,H)
      DIMENSION H(400)
      REAL MI,MD,MX,MY,MZ,MIC,MDC
      DATA CONV/.0174533/
C
C     SUBROUTINE DIPOLE COMPUTES THE TOTAL FIELD ANOMALY IN GAMMAS ON A
C   RECTANGULAR GRID.  MI AND MD ARE THE INCLINATION AND DECLINATION OF
C   THE DIPOLE AND FI AND FD ARE THE INCLINATION AND DECLINATION OF THE
C   REGIONAL FIELD.  DEPTH IS THE DEPTH TO THE DIPOLE.  XSTART, XDELT,
C   AND NX ARE THE STARTING VALUE, THE SAMPLE INTERVAL AND THE NUMBER OF
C   X COORDINATES.  YSTART, YDELT AND NY ARE THE STARTING VALUE, THE
C   SAMPLE INTERVAL AND THE NUMBER OF Y COORDINATES.  H IS THE OUTPUT
```

```
C  ARRAY WHICH INCREASES FIRST IN THE Y DIRECTION AND SECONDLY IN THE
C  X DIRECTION.
C
      INDEX(I,J,N)=(I-1)*N+J
      MIC=MI*CONV
      MDC=MD*CONV
      FIC=FI*CONV
      FDC=FD*CONV
      MX=COS(MIC)*COS(MDC)
      MY=COS(MIC)*SIN(MDC)
      MZ=SIN(MIC)
      HX=COS(FIC)*COS(FDC)
      HY=COS(FIC)*SIN(FDC)
      HZ=SIN(FIC)
      DO 10 I=1,NX
      X=XSTART+(I-1)*XDELT
      DO 10 J=1,NY
      Y=YSTART+(J-1)*YDELT
      R=SQRT(X*X+Y*Y+DEPTH*DEPTH)
      R3=R*R*R
      RX=X/R
      RY=Y/R
      RZ=-DEPTH/R
      DOT=RX*MX+RY*MY+RZ*MZ
      BX=(3.*DOT*RX-MX)/R3
      BY=(3.*DOT*RY-MY)/R3
      BZ=(3.*DOT*RZ-MZ)/R3
      K=INDEX(I,J,NY)
      H(K)=BX*HX+BY*HY+BZ*HZ
   10 H(K)=H(K)*(10.**5)
      RETURN
      END



      SUBROUTINE REDUC3(NX,NY,DX,DY,IM1,DM1,IF1,DF1,IM2,DM2,IF2,DF2,
     1  AZIM,DELTZ,H,RATIO)
C
C     SUBROUTINE REDUC3 PERFORMS LINEAR TRANSFORMATION OF A TWO-
C  DIMENSIONAL MAGNETIC ANOMALY.  SUCH TRANSFORMATIONS INCLUDE UPWARD
C  CONTINUATION, REDUCTION TO THE POLE, AND CALCULATION OF COMPONENTS.
C  H(I), I=1,2,...NX*NY, IS BOTH THE INPUT AND OUTPUT ARRAY.  ARRAY
C  H SHOULD BE FILLED AS FOLLOWS:  H(I)=H(M,N) WHERE M=1,2,...NX,
C  N=1,2,...NY, AND I=(M-1)*NY+N.  THE Z AXIS IS POSITIVE DOWN.  OTHER
C  INPUTS ARE...
C
C     DX     - SAMPLE INTERVAL IN THE X DIRECTION
C     DY     - SAMPLE INTERVAL IN THE Y DIRECTION
C     IM1    - INITIAL INCLINATION OF THE MAGNETIZATION
C     DM1    - INITIAL DECLINATION OF THE MAGNETIZATION
C     IF1    - INITIAL INCLINATION OF THE REGIONAL FIELD
C     DF1    - INITIAL DECLINATION OF THE REGIONAL FIELD
C     IM2    - NEW INCLINATION OF THE MAGNETIZATION
C     DM2    - NEW DECLINATION OF THE MAGNETIZATION
C     IF2    - NEW INCLINATION OF THE REGIONAL FIELD
C     DF2    - NEW DECLINATION OF THE REGIONAL FIELD
C     DELTZ  - THE AMOUNT OF UPWARD (OR DOWNWARD) CONTINUATION
C                REQUIRED - POSITIVE FOR UPWARD CONTINUATION
C     AZIM   - AZIMUTH OF THE X AXIS (POSITIVE TO THE EAST)
C
C  UNITS OF DX,DY, AND DELTZ ARE UNIMPORTANT SO LONG AS THEY ARE
C  CONSISTENT.  UNITS OF ALL ANGLES ARE IN DEGREES.  RATIO IS THE
C  RATIO OF THE SQUARE OF THE IMAGINARY PART TO THE SQUARE OF THE
C  REAL PART OF THE TRANSFORMED ARRAY.  IF ONLY UPWARD (OR DOWNWARD)
C  CONTINUATION IS DESIRED, SET MI1, MI2, FI1, AND FI2 TO 90.  IF
C  NO UPWARD (OR DOWNWARD) CONTINUATION IS DESIRED, SET DELTZ TO 0.
C
      DIMENSION H(400),HREAL(64,64),HIMAG(64,64)
      REAL IM1,IF1,IM2,IF2,MX,MY,MZ
      COMPLEX HC,THETA1,THETA2,CMPLX
      DATA PI/3.14159265/,CONV/.01745329/,LX/64/,LY/64/
      INDEX(I,J,N)=(I-1)*N+J
      DO 95 I=1,LX
      DO 95 J=1,LY
      HREAL(I,J)=0.
   95 HIMAG(I,J)=0.
C
C  --  REPLACE PART OF THE ZEROED ARRAYS WITH DATA H
C
      DO 10 I=1,NX
      DO 10 J=1,NY
      IJ=INDEX(I,J,NY)
   10 HREAL(I,J)=H(IJ)
      NTOT=LX*LY
```

```
C
C  --  FOURIER TRANSFORM
C
      CALL FFT(HREAL,HIMAG,NTOT,LX,LX,-1)
      CALL FFT(HREAL,HIMAG,NTOT,LY,NTOT,-1)
C
C  --  COMPUTE THE FIVE FILTER COEFFICIENTS FOR INITIAL PARAMETERS
C
      IPOLE=1
      IF(IM1.NE.90..OR.IF1.NE.90.)GO TO 60
      A11=-1.
      A21=-1.
      A31=0.
      B11=0.
      B21=0.
      GO TO 61
   60 CONTINUE
      IPOLE=0
      IM1=IM1*CONV
      DM1=(DM1-AZIM)*CONV
      IF1=IF1*CONV
      DF1=(DF1-AZIM)*CONV
      MX=COS(IM1)*COS(DM1)
      MY=COS(IM1)*SIN(DM1)
      MZ=SIN(IM1)
      HX=COS(IF1)*COS(DF1)
      HY=COS(IF1)*SIN(DF1)
      HZ=SIN(IF1)
      A11=MX*HX-MZ*HZ
      A21=MY*HY-MZ*HZ
      A31=MX*HY+MY*HX
      B11=-MX*HZ-MZ*HX
      B21=-MY*HZ-MZ*HY
   61 CONTINUE
C
C  --  COMPUTE THE FIVE FILTER COEFFICIENTS FOR NEW PARAMETERS
C
      IF(IM2.NE.90..OR.IF2.NE.90.)GO TO 70
      A12=-1.
      A22=-1.
      A32=0.
      B12=0.
      B22=0.
      GO TO 71
   70 CONTINUE
      IPOLE=0
      IM2=IM2*CONV
      DM2=(DM2-AZIM)*CONV
      IF2=IF2*CONV
      DF2=(DF2-AZIM)*CONV
      MX=COS(IM2)*COS(DM2)
      MY=COS(IM2)*SIN(DM2)
      MZ=SIN(IM2)
      HX=COS(IF2)*COS(DF2)
      HY=COS(IF2)*SIN(DF2)
      HZ=SIN(IF2)
      A12=MX*HX-MZ*HZ
      A22=MY*HY-MZ*HZ
      A32=MX*HY+MY*HX
      B12=-MX*HZ-MZ*HX
      B22=-MY*HZ-MZ*HY
   71 CONTINUE
      DELTKX=2.*PI/(LX*DX)
      DELTKY=2.*PI/(LY*DY)
      NNX=LX/2+1
      NNY=LY/2+1
C
C  --  START LOOPS TO APPLY FILTER INDEXING FOUR QUADRANTS DIFFERENTLY
C
      DO 20 I=1,LX
      IF(I-NNX)40,40,41
   40 FKX=(I-1)*DELTKX
      GO TO 42
   41 FKX=(I-LX-1)*DELTKX
   42 CONTINUE
      DO 20 J=1,LY
      IF(I.EQ.1.AND.J.EQ.1)GO TO 20
      IF(J-NNY)50,50,51
   50 FKY=(J-1)*DELTKY
      GO TO 52
   51 FKY=(J-LY-1)*DELTKY
   52 CONTINUE
C
C  --  GIVE SPECIAL ATTENTION TO NYQUIST ROW AND COLUMN
C
      IF(J.EQ.NNY.AND.I.GT.NNX)FKY=(J-LY-1)*DELTKY
      IF(I.EQ.NNX.AND.J.GT.NNY)FKX=(I-LX-1)*DELTKX
```

```
C
C --  COMPUTE COMPLEX FILTER FOR REDUCTION TO POLE
C
      FKXFKX=FKX*FKX
      FKYFKY=FKY*FKY
      FKXFKY=FKX*FKY
      FKFK=FKXFKX+FKYFKY
      FK=SQRT(FKFK)
      IF(IPOLE.EQ.1)GO TO 90
      THR1=FKXFKX*A11+FKYFKY*A21+FKXFKY*A31
      THR2=FKXFKX*A12+FKYFKY*A22+FKXFKY*A32
      THI1=(FKX*B11+FKY*B21)*FK
      THI2=(FKX*B12+FKY*B22)*FK
      THETA1=CMPLX(THR1,THI1)
      THETA2=CMPLX(THR2,THI2)
C
C --  APPLY REDUCTION-TO-POLE FILTER TO COMPLEX DATA
C
      HC=CMPLX(HREAL(I,J),HIMAG(I,J))
      HC=HC*THETA2/THETA1
      HREAL(I,J)=REAL(HC)
      HIMAG(I,J)=AIMAG(HC)
   90 CONTINUE
C
C --  COMPUTE AND APPLY UPWARD-CONTINUATION FILTER
C
      IF(DELTZ.EQ.0.)GO TO 80
      UP=EXP(-DELTZ*FK)
      HREAL(I,J)=HREAL(I,J)*UP
      HIMAG(I,J)=HIMAG(I,J)*UP
   80 CONTINUE
   20 CONTINUE
C
C --  TREAT ZERO AND NYQUIST WAVENUMBERS SPECIALLY
C
      IF(IPOLE.EQ.1)GO TO 81
      R1=REAL(CMPLX(A12,B12)/CMPLX(A11,B11))
      R2=REAL(CMPLX(A22,B22)/CMPLX(A21,B21))
      FZERO=.5*(R1+R2)
      HREAL(1,1)=HREAL(1,1)*FZERO
   81 CONTINUE
      HIMAG(1,1)=0.
      HIMAG(NNX,NNY)=0.
      HIMAG(NNX,1)=0.
      HIMAG(1,NNY)=0.
C
C --  INVERSE TRANSFORM

C
      CALL FFT(HREAL,HIMAG,NTOT,LX,LX,1)
      CALL FFT(HREAL,HIMAG,NTOT,LY,NTOT,1)
C
C --  COMPARE ENERGIES OF REAL AND IMAGINARY ARRAYS AND REPLACE
C      H WITH REDUCED DATA
C
      SUMR=0.
      SUMI=0.
      DO 30 I=1,NX
      DO 30 J=1,NY
      IJ=INDEX(I,J,NY)
      SUMR=SUMR+HREAL(I,J)**2
      SUMI=SUMI+HIMAG(I,J)**2
   30 H(IJ)=HREAL(I,J)/NTOT
      RATIO=SQRT(SUMI/SUMR)
      RETURN
      END
```

```
      SUBROUTINE FFT(A,B,NTOT,N,NSPAN,ISN)
C ********************** START OF FFT **************************
C                                        UP TO 2/4/73
C
C ** FOR REAL DATA, USE TOGETHER WITH REALTR, A SEPARATE ROUTINE
C      -SEE REALTR FOR DETAILS
C
C ** DOES NOT RUN UNDER WATFIV
C
C ** AS AN EXAMPLE OF HOW TO CALL THIS PROGRAM,  A TEST DRIVER
C         IS AVAILABLE IN A SEPARATE LIBRARY
C
C MULTIVARIATE COMPLEX FOURIER TRANSFORM, COMPUTED IN PLACE
C   USING MIXED-RADIX FAST FOURIER TRANSFORM ALGORITHM.
C BY R. C. SINGLETON, STANFORD RESEARCH INSTITUTE, SEPT. 1968
```

```
C  ARRAYS A AND B ORIGINALLY HOLD THE REAL AND IMAGINARY
C    COMPONENTS OF THE DATA, AND RETURN THE REAL AND
C    IMAGINARY COMPONENTS OF THE RESULTING FOURIER COEFFICIENTS.
C  MULTIVARIATE DATA IS INDEXED ACCORDING TO THE FORTRAN
C    ARRAY ELEMENT SUCCESSOR FUNCTION, WITHOUT LIMIT
C    ON THE NUMBER OF IMPLIED MULTIPLE SUBSCRIPTS.
C    THE SUBROUTINE IS CALLED ONCE FOR EACH VARIATE.
C    THE CALLS FOR A MULTIVARIATE TRANSFORM MAY BE IN ANY ORDER.
C  NTOT IS THE TOTAL NUMBER OF COMPLEX DATA VALUES.
C  N IS THE DIMENSION OF THE CURRENT VARIABLE.
C  NSPAN/N IS THE SPACING OF CONSECUTIVE DATA VALUES
C    WHILE INDEXING THE CURRENT VARIABLE.
C  THE SIGN OF ISN DETERMINES THE SIGN OF THE COMPLEX
C    EXPONENTIAL, AND THE MAGNITUDE OF ISN IS NORMALLY ONE.
C  A TRI-VARIATE TRANSFORM WITH A(N1,N2,N3), B(N1,N2,N3)
C    IS COMPUTED BY
C      CALL FFT(A,B,N1*N2*N3,N1,N1,1)
C      CALL FFT(A,B,N1*N2*N3,N2,N1*N2,1)
C      CALL FFT(A,B,N1*N2*N3,N3,N1*N2*N3,1)
C  FOR A SINGLE-VARIATE TRANSFORM,
C    NTOT = N = NSPAN = (NUMBER OF COMPLEX DATA VALUES), E.G.
C      CALL FFT(A,B,N,N,N,1)
C  THE DATA CAN ALTERNATIVELY BE STORED IN A SINGLE COMPLEX ARRAY C
C   IN STANDARD FORTRAN FASHION, I.E., ALTERNATING REAL AND
C   IMAGINARY PARTS.  THEN WITH MOST FORTRAN COMPILERS, THE
C   COMPLEX ARRAY C CAN BE EQUIVALENCED TO A REAL ARRAY A, THE
C   MAGNITUDE OF ISN CHANGED TO TWO TO GIVE THE CORRECT INDEXING
C   INCREMENT, AND A AND A(2) USED TO PASS THE INITIAL ADDRESSES
C   FOR THE SEQUENCES OF REAL AND IMAGINARY VALUES, E.G.,
C      COMPLEX C(NTOT)
C      REAL A(2*NTOT)
C      EQUIVALENCE (C(1),A(1))
C      CALL FFT(A,A(2),NTOT,N,NSPAN,2)
C  ARRAYS AT(MAXF), CK(MAXF), BT(MAXF), SK(MAXF), AND NP(MAXP)
C    ARE USED FOR TEMPORARY STORAGE.  IF THE AVAILABLE STORAGE
C    IS INSUFFICIENT, THE PROGRAM IS TERMINATED BY A STOP.
C    MAXF MUST BE .GE. THE MAXIMUM PRIME FACTOR OF N.
C    MAXP MUST BE .GT. THE NUMBER OF PRIME FACTORS OF N.
C    IN ADDITION, IF THE SQUARE-FREE PORTION K OF N HAS TWO OR
C    MORE PRIME FACTORS, THEN MAXP MUST BE .GE. K-1.
      DIMENSION A(1),B(1)
C  ARRAY STORAGE IN NFAC FOR A MAXIMUM OF 15 PRIME FACTORS OF N.
C  IF N HAS MORE THAN ONE SQUARE-FREE FACTOR, THE PRODUCT OF THE
C    SQUARE-FREE FACTORS MUST BE .LE. 210
      DIMENSION NFAC(11),NP(209)
C  ARRAY STORAGE FOR MAXIMUM PRIME FACTOR OF 23
      DIMENSION AT(23),CK(23),BT(23),SK(23)
      EQUIVALENCE (I,II)
C  THE FOLLOWING TWO CONSTANTS SHOULD AGREE WITH THE ARRAY DIMENSIONS.
      MAXF=23
      MAXP=209
      IF(N .LT. 2) RETURN
      INC=ISN
      C72=0.30901699437494742D0
      S72=0.95105651629515357D0
      S120=0.86602540378443865D0
      RAD=6.2831853071796D0
      IF(ISN .GE. 0) GO TO 10
      S72=-S72
      S120=-S120
      RAD=-RAD
      INC=-INC
   10 NT=INC*NTOT
      KS=INC*NSPAN
      KSPAN=KS
      NN=NT-INC
      JC=KS/N
      RADF=RAD*FLOAT(JC)*0.5
      I=0
      JF=0
C  DETERMINE THE FACTORS OF N
      M=0
      K=N
      GO TO 20
   15 M=M+1
      NFAC(M)=4
      K=K/16
   20 IF(K-(K/16)*16 .EQ. 0) GO TO 15
      J=3
      JJ=9
      GO TO 30
   25 M=M+1
      NFAC(M)=J
      K=K/JJ
   30 IF(MOD(K,JJ) .EQ. 0) GO TO 25
      J=J+2
```

```
      JJ=J**2
      IF(JJ .LE. K) GO TO 30
      IF(K .GT. 4) GO TO 40
      KT=M
      NFAC(M+1)=K
      IF(K .NE. 1) M=M+1
      GO TO 80
   40 IF(K-(K/4)*4 .NE. 0) GO TO 50
      M=M+1
      NFAC(M)=2
      K=K/4
   50 KT=M
      J=2
   60 IF(MOD(K,J) .NE. 0) GO TO 70
      M=M+1
      NFAC(M)=J
      K=K/J
   70 J=((J+1)/2)*2+1
      IF(J .LE. K) GO TO 60
   80 IF(KT .EQ. 0) GO TO 100
      J=KT
   90 M=M+1
      NFAC(M)=NFAC(J)
      J=J-1
      IF(J .NE. 0) GO TO 90
C  COMPUTE FOURIER TRANSFORM
  100 SD=RADF/FLOAT(KSPAN)
      CD=2.0*SIN(SD)**2
      SD=SIN(SD+SD)
      KK=1
      I=I+1
      IF(NFAC(I) .NE. 2) GO TO 400
C  TRANSFORM FOR FACTOR OF 2 (INCLUDING ROTATION FACTOR)
      KSPAN=KSPAN/2
      K1=KSPAN+2
  210 K2=KK+KSPAN
      AK=A(K2)
      BK=B(K2)
      A(K2)=A(KK)-AK
      B(K2)=B(KK)-BK
      A(KK)=A(KK)+AK
      B(KK)=B(KK)+BK
      KK=K2+KSPAN
      IF(KK .LE. NN) GO TO 210
      KK=KK-NN
      IF(KK .LE. JC) GO TO 210
      IF(KK .GT. KSPAN) GO TO 800
  220 C1=1.0-CD
      S1=SD
  230 K2=KK+KSPAN
      AK=A(KK)-A(K2)
      BK=B(KK)-B(K2)
      A(KK)=A(KK)+A(K2)
      B(KK)=B(KK)+B(K2)
      A(K2)=C1*AK-S1*BK
      B(K2)=S1*AK+C1*BK
      KK=K2+KSPAN
      IF(KK .LT. NT) GO TO 230
      K2=KK-NT
      C1=-C1
      KK=K1-K2
      IF(KK .GT. K2) GO TO 230
      AK=C1-(CD*C1+SD*S1)
      S1=(SD*C1-CD*S1)+S1
      C1=2.0-(AK**2+S1**2)
      S1=C1*S1
      C1=C1*AK
      KK=KK+JC
      IF(KK .LT. K2) GO TO 230
      K1=K1+INC+INC
      KK=(K1-KSPAN)/2+JC
      IF(KK .LE. JC+JC) GO TO 220
      GO TO 100
C  TRANSFORM FOR FACTOR OF 3 (OPTIONAL CODE)
  320 K1=KK+KSPAN
      K2=K1+KSPAN
      AK=A(KK)
      BK=B(KK)
      AJ=A(K1)+A(K2)
      BJ=B(K1)+B(K2)
      A(KK)=AK+AJ
      B(KK)=BK+BJ
      AK=-0.5*AJ+AK
      BK=-0.5*BJ+BK
      AJ=(A(K1)-A(K2))*S120
      BJ=(B(K1)-B(K2))*S120
      A(K1)=AK-BJ
      B(K1)=BK+AJ
      A(K2)=AK+BJ
      B(K2)=BK-AJ
      KK=K2+KSPAN
      IF(KK .LT. NN) GO TO 320
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 320
      GO TO 700
C  TRANSFORM FOR FACTOR OF 4
  400 IF(NFAC(I) .NE. 4) GO TO 600
      KSPNN=KSPAN
      KSPAN=KSPAN/4
  410 C1=1.0
      S1=0
  420 K1=KK+KSPAN
      K2=K1+KSPAN
      K3=K2+KSPAN
      AKP=A(KK)+A(K2)
      AKM=A(KK)-A(K2)
      AJP=A(K1)+A(K3)
      AJM=A(K1)-A(K3)
      A(KK)=AKP+AJP
      AJP=AKP-AJP
      BKP=B(KK)+B(K2)
      BKM=B(KK)-B(K2)
      BJP=B(K1)+B(K3)
      BJM=B(K1)-B(K3)
      B(KK)=BKP+BJP
      BJP=BKP-BJP
      IF(ISN .LT. 0) GO TO 450
      AKP=AKM-BJM
      AKM=AKM+BJM
      BKP=BKM+AJM
      BKM=BKM-AJM
      IF(S1 .EQ. 0) GO TO 460
  430 A(K1)=AKP*C1-BKP*S1
      B(K1)=AKP*S1+BKP*C1
      A(K2)=AJP*C2-BJP*S2
      B(K2)=AJP*S2+BJP*C2
      A(K3)=AKM*C3-BKM*S3
      B(K3)=AKM*S3+BKM*C3
      KK=K3+KSPAN
      IF(KK .LE. NT) GO TO 420
  440 C2=C1-(CD*C1+SD*S1)
      S1=(SD*C1-CD*S1)+S1
      C1=2.0-(C2**2+S1**2)
      S1=C1*S1
      C1=C1*C2
      C2=C1**2-S1**2
      S2=2.0*C1*S1
      C3=C2*C1-S2*S1
      S3=C2*S1+S2*C1
      KK=KK-NT+JC
      IF(KK .LE. KSPAN) GO TO 420
      KK=KK-KSPAN+INC
      IF(KK .LE. JC) GO TO 410
      IF(KSPAN .EQ. JC) GO TO 800
      GO TO 100
  450 AKP=AKM+BJM
      AKM=AKM-BJM
      BKP=BKM-AJM
      BKM=BKM+AJM
      IF(S1 .NE. 0) GO TO 430
  460 A(K1)=AKP
      B(K1)=BKP
      A(K2)=AJP
      B(K2)=BJP
      A(K3)=AKM
      B(K3)=BKM
      KK=K3+KSPAN
      IF(KK .LE. NT) GO TO 420
      GO TO 440
C  TRANSFORM FOR FACTOR OF 5 (OPTIONAL CODE)
  510 C2=C72**2-S72**2
      S2=2.0*C72*S72
  520 K1=KK+KSPAN
      K2=K1+KSPAN
      K3=K2+KSPAN
      K4=K3+KSPAN
      AKP=A(K1)+A(K4)
      AKM=A(K1)-A(K4)
      BKP=B(K1)+B(K4)
      BKM=B(K1)-B(K4)
      AJP=A(K2)+A(K3)
      AJM=A(K2)-A(K3)
      BJP=B(K2)+B(K3)
      BJM=B(K2)-B(K3)
```

```
      AA=A(KK)
      BB=B(KK)
      A(KK)=AA+AKP+AJP
      B(KK)=BB+BKP+BJP
      AK=AKP*C72+AJP*C2+AA
      BK=BKP*C72+BJP*C2+BB
      AJ=AKM*S72+AJM*S2
      BJ=BKM*S72+BJM*S2
      A(K1)=AK-BJ
      A(K4)=AK+BJ
      B(K1)=BK+AJ
      B(K4)=BK-AJ
      AK=AKP*C2+AJP*C72+AA
      BK=BKP*C2+BJP*C72+BB
      AJ=AKM*S2-AJM*S72
      BJ=BKM*S2-BJM*S72
      A(K2)=AK-BJ
      A(K3)=AK+BJ
      B(K2)=BK+AJ
      B(K3)=BK-AJ
      KK=K4+KSPAN
      IF(KK .LT. NN) GO TO 520
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 520
      GO TO 700
C  TRANSFORM FOR ODD FACTORS
  600 K=NFAC(I)
      KSPNN=KSPAN
      KSPAN=KSPAN/K
      IF(K .EQ. 3) GO TO 320
      IF(K .EQ. 5) GO TO 510
      IF(K .EQ. JF) GO TO 640
      JF=K
      S1=RAD/FLOAT(K)
      C1=COS(S1)
      S1=SIN(S1)
      IF(JF .GT. MAXF) GO TO 998
      CK(JF)=1.0
      SK(JF)=0.0
      J=1
  630 CK(J)=CK(K)*C1+SK(K)*S1
      SK(J)=CK(K)*S1-SK(K)*C1
      K=K-1
      CK(K)=CK(J)
      SK(K)=-SK(J)
      J=J+1
      IF(J .LT. K) GO TO 630
  640 K1=KK
      K2=KK+KSPNN
      AA=A(KK)
      BB=B(KK)
      AK=AA
      BK=BB
      J=1
      K1=K1+KSPAN
  650 K2=K2-KSPAN
      J=J+1
      AT(J)=A(K1)+A(K2)
      AK=AT(J)+AK
      BT(J)=B(K1)+B(K2)
      BK=BT(J)+BK
      J=J+1
      AT(J)=A(K1)-A(K2)
      BT(J)=B(K1)-B(K2)
      K1=K1+KSPAN
      IF(K1 .LT. K2) GO TO 650
      A(KK)=AK
      B(KK)=BK
      K1=KK
      K2=KK+KSPNN
      J=1
  660 K1=K1+KSPAN
      K2=K2-KSPAN
      JJ=J
      AK=AA
      BK=BB
      AJ=0.0
      BJ=0.0
      K=1
  670 K=K+1
      AK=AT(K)*CK(JJ)+AK
      BK=BT(K)*CK(JJ)+BK
      K=K+1
      AJ=AT(K)*SK(JJ)+AJ
      BJ=BT(K)*SK(JJ)+BJ
      JJ=JJ+J
      IF(JJ .GT. JF) JJ=JJ-JF
      IF(K .LT. JF) GO TO 670
      K=JF-J
      A(K1)=AK-BJ
      B(K1)=BK+AJ
      A(K2)=AK+BJ
      B(K2)=BK-AJ
      J=J+1
      IF(J .LT. K) GO TO 660
      KK=KK+KSPNN
      IF(KK .LE. NN) GO TO 640
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 640
C  MULTIPLY BY ROTATION FACTOR
  700 IF(I .EQ. M) GO TO 800
      KK=JC+1
  710 C2=1.0-CD
      S1=SD
  720 C1=C2
      S2=S1
      KK=KK+KSPAN
  730 AK=A(KK)
      A(KK)=C2*AK-S2*B(KK)
      B(KK)=S2*AK+C2*B(KK)
      KK=KK+KSPNN
      IF(KK .LE. NT) GO TO 730
      AK=S1*S2
      S2=S1*C2+C1*S2
      C2=C1*C2-AK
      KK=KK-NT+KSPAN
      IF(KK .LE. KSPNN) GO TO 730
      C2=C1-(CD*C1+SD*S1)
      S1=S1+(SD*C1-CD*S1)
      C1=2.0-(C2**2+S1**2)
      S1=C1*S1
      C2=C1*C2
      KK=KK-KSPNN+JC
      IF(KK .LE. KSPAN) GO TO 720
      KK=KK-KSPAN+JC+INC
      IF(KK .LE. JC+JC) GO TO 710
      GO TO 100
C  PERMUTE THE RESULTS TO NORMAL ORDER
C  PERMUTATION FOR SQUARE FACTORS OF N
  800 NP(1)=KS
      IF(KT .EQ. 0) GO TO 890
      K=KT+KT+1
      IF(M .LT. K) K=K-1
      J=1
      NP(K+1)=JC
  810 NP(J+1)=NP(J)/NFAC(J)
      NP(K)=NP(K+1)*NFAC(J)
      J=J+1
      K=K-1
      IF(J .LT. K) GO TO 810
      K3=NP(K+1)
      KSPAN=NP(2)
      KK=JC+1
      K2=KSPAN+1
      J=1
      IF(N .NE. NTOT) GO TO 850
C  PERMUTATION FOR SINGLE-VARIATE TRANSFORM
  820 AK=A(KK)
      A(KK)=A(K2)
      A(K2)=AK
      BK=B(KK)
      B(KK)=B(K2)
      B(K2)=BK
      KK=KK+INC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 820
  830 K2=K2-NP(J)
      J=J+1
      K2=NP(J+1)+K2
      IF(K2 .GT. NP(J)) GO TO 830
      J=1
  840 IF(KK .LT. K2) GO TO 820
      KK=KK+INC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 840
      IF(KK .LT. KS) GO TO 830
      JC=K3
      GO TO 890
C  PERMUTATION FOR MULTIVARIATE TRANSFORM
  850 K=KK+JC
  860 AK=A(KK)
      A(KK)=A(K2)
      A(K2)=AK
      BK=B(KK)
```

```
      B(KK)=B(K2)
      B(K2)=BK
      KK=KK+INC
      K2=K2+INC
      IF(KK .LT. K) GO TO 860
      KK=KK+KS-JC
      K2=K2+KS-JC
      IF(KK .LT. NT) GO TO 850
      K2=K2-NT+KSPAN
      KK=KK-NT+JC
      IF(K2 .LT. KS) GO TO 850
  870 K2=K2-NP(J)
      J=J+1
      K2=NP(J+1)+K2
      IF(K2 .GT. NP(J)) GO TO 870
      J=1
  880 IF(KK .LT. K2) GO TO 850
      KK=KK+JC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 880
      IF(KK .LT. KS) GO TO 870
      JC=K3
  890 IF(2*KT+1 .GE. M) RETURN
      KSPNN=NP(KT+1)
C  PERMUTATION FOR SQUARE-FREE FACTORS OF N
      J=M-KT
      NFAC(J+1)=1
  900 NFAC(J)=NFAC(J)*NFAC(J+1)
      J=J-1
      IF(J .NE. KT) GO TO 900
      KT=KT+1
      NN=NFAC(KT)-1
      IF(NN .GT. MAXP) GO TO 998
      JJ=0
      J=0
      GO TO 906
  902 JJ=JJ-K2
      K2=KK
      K=K+1
      KK=NFAC(K)
  904 JJ=KK+JJ
      IF(JJ .GE. K2) GO TO 902
      NP(J)=JJ
  906 K2=NFAC(KT)
      K=KT+1
      KK=NFAC(K)
      J=J+1
      IF(J .LE. NN) GO TO 904
C  DETERMINE THE PERMUTATION CYCLES OF LENGTH GREATER THAN 1
      J=0
      GO TO 914
  910 K=KK
      KK=NP(K)
      NP(K)=-KK
      IF(KK .NE. J) GO TO 910
      K3=KK
  914 J=J+1
      KK=NP(J)
      IF(KK .LT. 0) GO TO 914
      IF(KK .NE. J) GO TO 910
      NP(J)=-J
      IF(J .NE. NN) GO TO 914
      MAXF=INC*MAXF
C  REORDER A AND B, FOLLOWING THE PERMUTATION CYCLES
      GO TO 950
  924 J=J-1
      IF(NP(J) .LT. 0) GO TO 924
      JJ=JC
  926 KSPAN=JJ
      IF(JJ .GT. MAXF) KSPAN=MAXF
      JJ=JJ-KSPAN
      K=NP(J)
      KK=JC*K+II+JJ
      K1=KK+KSPAN
      K2=0
  928 K2=K2+1
      AT(K2)=A(K1)
      BT(K2)=B(K1)
      K1=K1-INC
      IF(K1 .NE. KK) GO TO 928
  932 K1=KK+KSPAN
      K2=K1-JC*(K+NP(K))
      K=-NP(K)
  936 A(K1)=A(K2)
      B(K1)=B(K2)
      K1=K1-INC
      K2=K2-INC
      IF(K1 .NE. KK) GO TO 936
      KK=K2
      IF(K .NE. J) GO TO 932
      K1=KK+KSPAN
      K2=0
  940 K2=K2+1
      A(K1)=AT(K2)
      B(K1)=BT(K2)
      K1=K1-INC
      IF(K1 .NE. KK) GO TO 940
      IF(JJ .NE. 0) GO TO 926
      IF(J .NE. 1) GO TO 924
  950 J=K3+1
      NT=NT-KSPNN
      II=NT-INC+1
      IF(NT .GE. 0) GO TO 924
      RETURN
C  ERROR FINISH, INSUFFICIENT ARRAY STORAGE
  998 ISN=0
      PRINT 999
      STOP
  999 FORMAT(44H0ARRAY BOUNDS EXCEEDED WITHIN SUBROUTINE FI
      END

//GO.SYSIN DD *

 80.   0.  80.   0.   2.
 70.   0.  70.   0.   2.
 60.   0.  60.   0.   2.
 50.   0.  50.   0.   2.
 40.   0.  40.   0.   2.
 30.   0.  30.   0.   2.
 20.   0.  20.   0.   2.
 10.   0.  10.   0.   2.
 90.   0.  90.   0.   1.8
 90.   0.  90.   0.   1.6
 90.   0.  90.   0.   1.4
 90.   0.  90.   0.   1.2
 90.   0.  90.   0.   1.
/*
```

8. Output from test of REDUC3

\014 TEST OF REDUC3...

TRUE ANOMALY...

-440.	-446.	-420.	-358.	-273.	-196.	-164.	-196.	-273.	-358.	-420.	-446.	-440.
-446.	-407.	-294.	-91.	177.	420.	520.	420.	177.	-91.	-294.	-407.	-446.
-420.	-294.	-0.	520.	1235.	1918.	2210.	1918.	1235.	520.	-0.	-294.	-420.
-358.	-91.	520.	1662.	3356.	5106.	5888.	5106.	3356.	1662.	520.	-91.	-358.
-273.	177.	1235.	3356.	6804.	10688.	12522.	10688.	6804.	3356.	1235.	177.	-273.
-196.	420.	1918.	5106.	10688.	17459.	20813.	17459.	10688.	5106.	1918.	420.	-196.
-164.	520.	2210.	5888.	12522.	20813.	25000.	20813.	12522.	5888.	2210.	520.	-164.
-196.	420.	1918.	5106.	10688.	17459.	20813.	17459.	10688.	5106.	1918.	420.	-196.
-273.	177.	1235.	3356.	6804.	10688.	12522.	10688.	6804.	3356.	1235.	177.	-273.
-358.	-91.	520.	1662.	3356.	5106.	5888.	5106.	3356.	1662.	520.	-91.	-358.
-420.	-294.	0.	520.	1235.	1918.	2210.	1918.	1235.	520.	0.	-294.	-420.
-446.	-407.	-294.	-91.	177.	420.	520.	420.	177.	-91.	-294.	-407.	-446.
-440.	-446.	-420.	-358.	-273.	-196.	-164.	-196.	-273.	-358.	-420.	-446.	-440.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 80. MD = 0. FD = 80. FD = 0. DEPTH = 2.0

NORMALIZED RMS= 0.829e-02

ERROR ARRAY...

4.	1.	-0.	-2.	-3.	-3.	-4.	-4.	-4.	-3.	-2.	-0.	2.
3.	2.	1.	0.	0.	1.	1.	1.	1.	2.	2.	4.	5.
8.	6.	5.	3.	2.	2.	1.	1.	2.	2.	3.	4.	6.
8.	7.	7.	6.	6.	6.	7.	7.	7.	7.	8.	9.	10.
11.	9.	8.	7.	6.	5.	5.	5.	5.	6.	6.	7.	9.
12.	11.	11.	11.	11.	11.	11.	11.	12.	12.	12.	13.	14.
12.	11.	10.	9.	9.	9.	8.	8.	8.	8.	9.	9.	10.
14.	14.	15.	15.	15.	15.	15.	16.	16.	16.	16.	16.	16.
13.	13.	13.	12.	12.	12.	12.	11.	11.	11.	11.	11.	11.
18.	19.	20.	20.	21.	21.	21.	22.	22.	22.	21.	20.	20.
15.	16.	16.	16.	16.	16.	16.	16.	15.	15.	14.	14.	13.
23.	26.	28.	30.	31.	32.	32.	32.	32.	31.	30.	28.	26.
19.	21.	22.	23.	24.	24.	24.	24.	23.	22.	21.	19.	17.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 70. MD = 0. FD = 70. FD = 0. DEPTH = 2.0

NORMALIZED RMS= 0.201e-01

ERROR ARRAY...

34.	32.	32.	32.	32.	32.	32.	31.	31.	31.	30.	30.	31.
41.	42.	44.	46.	49.	50.	51.	51.	49.	47.	45.	44.	43.
40.	37.	35.	33.	32.	32.	31.	31.	31.	32.	33.	35.	38.
47.	46.	47.	48.	49.	49.	50.	50.	49.	49.	48.	48.	50.
45.	41.	38.	36.	34.	34.	33.	33.	33.	35.	36.	39.	42.
52.	51.	51.	51.	51.	51.	52.	52.	52.	52.	52.	53.	55.
47.	44.	40.	38.	37.	37.	37.	36.	36.	37.	39.	42.	45.
56.	56.	56.	56.	56.	56.	56.	57.	57.	57.	57.	58.	59.
48.	46.	43.	42.	40.	40.	41.	40.	40.	40.	42.	44.	46.
61.	61.	63.	64.	65.	65.	65.	65.	66.	65.	64.	63.	63.
49.	49.	47.	47.	46.	46.	46.	46.	45.	45.	46.	47.	47.
69.	73.	77.	79.	82.	82.	83.	83.	82.	80.	78.	75.	72.
54.	56.	57.	57.	58.	58.	58.	57.	57.	56.	55.	54.	52.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 60. MD = 0. FD = 60. FD = 0. DEPTH = 2.0

NORMALIZED RMS= 0.394e-01

ERROR ARRAY...

92.	100.	108.	117.	124.	129.	131.	129.	125.	118.	109.	100.	92.
113.	123.	135.	147.	157.	163.	166.	163.	156.	146.	134.	123.	113.
97.	98.	100.	103.	105.	108.	108.	108.	106.	104.	101.	98.	97.
116.	120.	125.	131.	136.	139.	140.	139.	135.	130.	124.	119.	116.
102.	99.	97.	97.	97.	98.	98.	98.	98.	98.	98.	99.	102.
122.	121.	122.	125.	127.	128.	128.	128.	126.	124.	121.	121.	122.
107.	101.	97.	94.	93.	94.	95.	94.	94.	95.	98.	101.	107.
126.	124.	124.	125.	125.	125.	124.	125.	125.	124.	123.	123.	126.
108.	102.	97.	94.	93.	93.	94.	93.	93.	95.	98.	102.	108.
129.	128.	128.	129.	130.	130.	129.	130.	130.	129.	128.	128.	129.
106.	102.	99.	96.	95.	95.	96.	95.	96.	97.	100.	102.	106.
135.	138.	141.	143.	145.	146.	146.	146.	145.	142.	140.	137.	135.
106.	106.	105.	104.	104.	104.	105.	104.	105.	105.	106.	106.	106.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 50. MD = 0. FD = 50. FD = 0. DEPTH = 2.0

NORMALIZED RMS= 0.683e-01

ERROR ARRAY...

176.	205.	236.	266.	290.	307.	314.	309.	294.	270.	241.	210.	180.
210.	242.	275.	307.	333.	350.	355.	348.	330.	303.	270.	237.	206.
177.	193.	210.	228.	242.	253.	257.	255.	245.	232.	215.	198.	181.
210.	227.	246.	265.	280.	289.	292.	288.	277.	260.	241.	222.	205.
185.	190.	197.	206.	214.	221.	224.	222.	217.	210.	202.	195.	190.
217.	223.	232.	242.	250.	255.	255.	253.	247.	238.	227.	218.	213.
194.	190.	190.	193.	197.	201.	204.	203.	200.	198.	195.	195.	199.
224.	223.	225.	230.	234.	235.	234.	234.	231.	226.	220.	218.	220.
199.	190.	186.	185.	186.	189.	191.	190.	189.	190.	191.	195.	203.
228.	224.	224.	226.	227.	228.	226.	226.	224.	221.	219.	219.	223.
196.	188.	183.	181.	180.	181.	182.	183.	183.	185.	188.	193.	200.
228.	227.	228.	229.	231.	231.	230.	229.	228.	225.	223.	222.	223.
187.	184.	182.	180.	179.	180.	181.	182.	183.	184.	187.	189.	192.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 40. MD = 0. FD = 40. FD = 0. DEPTH = 2.0

NORMALIZED RMS= .109

ERROR ARRAY...

278.	347.	418.	487.	544.	584.	599.	587.	550.	495.	427.	357.	287.
323.	393.	464.	532.	588.	624.	635.	620.	581.	524.	455.	383.	314.
279.	328.	379.	428.	469.	497.	509.	501.	475.	436.	388.	338.	289.
323.	369.	418.	466.	503.	527.	534.	524.	497.	458.	409.	360.	314.
295.	324.	356.	390.	418.	439.	447.	442.	424.	398.	365.	333.	304.
338.	363.	393.	424.	449.	464.	467.	461.	443.	416.	384.	354.	329.
316.	327.	343.	364.	382.	397.	404.	400.	388.	372.	353.	336.	325.
357.	364.	380.	398.	413.	422.	422.	419.	407.	390.	370.	355.	348.
334.	330.	336.	346.	357.	366.	371.	370.	363.	355.	345.	340.	343.
370.	366.	372.	381.	389.	394.	394.	390.	383.	373.	362.	357.	361.
339.	330.	329.	333.	338.	344.	347.	347.	345.	341.	339.	339.	349.
371.	365.	366.	370.	374.	376.	376.	373.	368.	362.	357.	355.	361.
329.	322.	320.	321.	324.	327.	330.	330.	330.	329.	330.	331.	339.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 30. MD = 0. FD = 30. FD = 0. DEPTH = 2.0

NORMALIZED RMS= .173

ERROR ARRAY...

425.	552.	685.	814.	923.	999.	1027.	1003.	931.	824.	696.	564.	437.
471.	599.	730.	858.	963.	1033.	1055.	1029.	955.	848.	719.	587.	459.
429.	538.	648.	756.	846.	908.	931.	912.	853.	767.	660.	550.	441.
476.	580.	685.	788.	870.	925.	941.	921.	862.	778.	673.	568.	464.

ERROR ARRAY...

-26.	-24.	-22.	-22.	-21.	-21.	-21.	-21.	-21.	-22.	-22.	-24.	-26.
-24.	-21.	-19.	-18.	-18.	-18.	-18.	-18.	-18.	-18.	-19.	-21.	-24.
-22.	-19.	-17.	-16.	-15.	-15.	-14.	-15.	-15.	-16.	-17.	-19.	-22.
-22.	-18.	-16.	-15.	-14.	-14.	-13.	-14.	-14.	-15.	-16.	-18.	-22.
-21.	-18.	-15.	-14.	-13.	-13.	-13.	-13.	-13.	-14.	-15.	-18.	-21.
-21.	-18.	-15.	-14.	-13.	-11.	-12.	-11.	-13.	-14.	-15.	-18.	-21.
-21.	-18.	-14.	-13.	-13.	-12.	-14.	-12.	-13.	-13.	-14.	-18.	-21.
-21.	-18.	-15.	-14.	-13.	-11.	-12.	-11.	-13.	-14.	-15.	-18.	-21.
-21.	-18.	-15.	-14.	-13.	-13.	-13.	-13.	-13.	-14.	-15.	-18.	-21.
-22.	-18.	-16.	-15.	-14.	-14.	-13.	-14.	-14.	-15.	-16.	-18.	-22.
-22.	-19.	-17.	-16.	-15.	-15.	-14.	-15.	-15.	-16.	-17.	-19.	-22.
-24.	-21.	-19.	-18.	-18.	-18.	-18.	-18.	-18.	-18.	-19.	-21.	-24.
-26.	-24.	-22.	-22.	-21.	-21.	-21.	-21.	-21.	-22.	-22.	-24.	-26.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 90. MD = 0. FD = 90. FD = 0. DEPTH = 1.4

NORMALIZED RMS= 0.179e-01

ERROR ARRAY...

-42.	-38.	-36.	-35.	-34.	-34.	-34.	-34.	-34.	-35.	-36.	-38.	-42.
-38.	-33.	-30.	-29.	-28.	-28.	-28.	-28.	-28.	-29.	-30.	-33.	-38.
-36.	-30.	-27.	-25.	-24.	-23.	-23.	-23.	-24.	-25.	-27.	-30.	-36.
-35.	-29.	-25.	-23.	-22.	-21.	-21.	-21.	-22.	-23.	-25.	-29.	-35.
-34.	-28.	-24.	-22.	-20.	-20.	-20.	-20.	-20.	-22.	-24.	-28.	-34.
-34.	-28.	-23.	-21.	-20.	-17.	-19.	-17.	-20.	-21.	-23.	-28.	-34.
-34.	-28.	-23.	-21.	-20.	-19.	-25.	-19.	-20.	-21.	-23.	-28.	-34.
-34.	-28.	-23.	-21.	-20.	-17.	-19.	-17.	-20.	-21.	-23.	-28.	-34.
-34.	-28.	-24.	-22.	-20.	-20.	-20.	-20.	-20.	-22.	-24.	-28.	-34.
-35.	-29.	-25.	-23.	-22.	-21.	-21.	-21.	-22.	-23.	-25.	-29.	-35.
-36.	-30.	-27.	-25.	-24.	-23.	-23.	-23.	-24.	-25.	-27.	-30.	-36.
-38.	-33.	-30.	-29.	-28.	-28.	-28.	-28.	-28.	-29.	-30.	-33.	-38.
-42.	-38.	-36.	-35.	-34.	-34.	-34.	-34.	-34.	-35.	-36.	-38.	-42.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 90. MD = 0. FD = 90. FD = 0. DEPTH = 1.2

NORMALIZED RMS= 0.211e-01

ERROR ARRAY...

-57.	-52.	-49.	-48.	-47.	-47.	-46.	-47.	-47.	-48.	-49.	-52.	-57.
-52.	-45.	-41.	-40.	-39.	-38.	-38.	-38.	-39.	-40.	-41.	-45.	-52.
-49.	-41.	-37.	-35.	-33.	-32.	-32.	-32.	-33.	-35.	-37.	-41.	-49.
-48.	-40.	-35.	-32.	-31.	-30.	-30.	-30.	-31.	-32.	-35.	-40.	-48.
-47.	-39.	-33.	-31.	-28.	-27.	-28.	-27.	-28.	-31.	-33.	-39.	-47.
-47.	-38.	-32.	-30.	-27.	-25.	-30.	-25.	-27.	-30.	-32.	-38.	-47.
-46.	-38.	-32.	-30.	-28.	-30.	-41.	-30.	-28.	-30.	-32.	-38.	-46.
-47.	-38.	-32.	-30.	-27.	-25.	-30.	-25.	-27.	-30.	-32.	-38.	-47.
-47.	-39.	-33.	-31.	-28.	-27.	-28.	-27.	-28.	-31.	-33.	-39.	-47.
-48.	-40.	-35.	-32.	-31.	-30.	-30.	-30.	-31.	-32.	-35.	-40.	-48.
-49.	-41.	-37.	-35.	-33.	-32.	-32.	-32.	-33.	-35.	-37.	-41.	-49.
-52.	-45.	-41.	-40.	-39.	-38.	-38.	-38.	-39.	-40.	-41.	-45.	-52.
-57.	-52.	-49.	-48.	-47.	-47.	-46.	-47.	-47.	-48.	-49.	-52.	-57.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 90. MD = 0. FD = 90. FD = 0. DEPTH = 1.0

NORMALIZED RMS= 0.238e-01

ERROR ARRAY...

-71.	-65.	-62.	-61.	-60.	-59.	-59.	-59.	-60.	-61.	-62.	-65.	-71.
-65.	-57.	-53.	-51.	-50.	-50.	-50.	-50.	-50.	-51.	-53.	-57.	-65.
-62.	-53.	-48.	-45.	-44.	-43.	-42.	-43.	-44.	-45.	-48.	-53.	-62.
-61.	-51.	-45.	-42.	-41.	-41.	-42.	-41.	-41.	-42.	-45.	-51.	-61.
-60.	-50.	-44.	-41.	-38.	-38.	-41.	-38.	-38.	-41.	-44.	-50.	-60.
-59.	-50.	-43.	-41.	-38.	-41.	-55.	-41.	-38.	-41.	-43.	-50.	-59.
-59.	-50.	-42.	-42.	-41.	-55.	-84.	-55.	-41.	-42.	-42.	-50.	-59.
-59.	-50.	-43.	-41.	-38.	-41.	-55.	-41.	-38.	-41.	-43.	-50.	-59.
-60.	-50.	-44.	-41.	-38.	-38.	-41.	-38.	-38.	-41.	-44.	-50.	-60.
-61.	-51.	-45.	-42.	-41.	-41.	-42.	-41.	-41.	-42.	-45.	-51.	-61.
-62.	-53.	-48.	-45.	-44.	-43.	-42.	-43.	-44.	-45.	-48.	-53.	-62.
-65.	-57.	-53.	-51.	-50.	-50.	-50.	-50.	-50.	-51.	-53.	-57.	-65.
-71.	-65.	-62.	-61.	-60.	-59.	-59.	-59.	-60.	-61.	-62.	-65.	-71.

455.	539.	625.	713.	784.	835.	853.	839.	791.	723.	637.	551.	467.
505.	579.	658.	738.	801.	843.	854.	839.	794.	728.	646.	567.	493.
497.	550.	613.	680.	734.	774.	789.	778.	742.	690.	624.	563.	509.
548.	590.	643.	702.	749.	780.	787.	776.	741.	692.	632.	577.	536.
543.	567.	607.	654.	694.	724.	736.	728.	702.	664.	619.	579.	555.
590.	603.	635.	674.	707.	729.	735.	725.	699.	663.	624.	591.	578.
580.	580.	603.	633.	662.	683.	692.	687.	670.	643.	615.	592.	591.
614.	610.	628.	650.	672.	686.	690.	681.	664.	639.	616.	598.	602.
591.	583.	596.	613.	633.	647.	655.	652.	641.	623.	607.	595.	603.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 20. MD = 0. FD = 20. FD = 0. DEPTH = 2.0

NORMALIZED RMS= .331

ERROR ARRAY...

828.	1054.	1291.	1521.	1718.	1853.	1902.	1857.	1725.	1531.	1302.	1065.	840.
859.	1092.	1329.	1563.	1757.	1889.	1932.	1885.	1750.	1553.	1319.	1081.	848.
826.	1050.	1272.	1495.	1677.	1806.	1850.	1810.	1684.	1504.	1283.	1061.	837.
863.	1085.	1302.	1520.	1695.	1817.	1853.	1813.	1688.	1511.	1291.	1074.	852.
852.	1060.	1262.	1469.	1633.	1751.	1790.	1755.	1640.	1478.	1273.	1072.	863.
899.	1096.	1287.	1485.	1639.	1748.	1779.	1744.	1632.	1475.	1276.	1084.	888.
908.	1083.	1258.	1443.	1587.	1693.	1726.	1697.	1595.	1452.	1269.	1094.	919.
965.	1119.	1281.	1452.	1586.	1681.	1708.	1677.	1579.	1443.	1270.	1107.	954.
989.	1114.	1262.	1418.	1544.	1634.	1664.	1638.	1551.	1427.	1273.	1126.	1000.
1051.	1150.	1283.	1422.	1537.	1616.	1641.	1612.	1530.	1412.	1272.	1138.	1040.
1080.	1151.	1270.	1394.	1504.	1577.	1605.	1581.	1511.	1404.	1281.	1163.	1091.
1134.	1182.	1285.	1392.	1489.	1551.	1574.	1547.	1482.	1382.	1274.	1171.	1123.
1155.	1185.	1276.	1371.	1464.	1522.	1548.	1526.	1471.	1381.	1287.	1196.	1167.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 10. MD = 0. FD = 10. FD = 0. DEPTH = 2.0

NORMALIZED RMS= 1.18

ERROR ARRAY...

3458.	4172.	4870.	5493.	6003.	6335.	6456.	6338.	6008.	5498.	4877.	4179.	3465.
3448.	4180.	4882.	5522.	6040.	6381.	6499.	6379.	6035.	5517.	4876.	4173.	3441.
3407.	4152.	4847.	5498.	6015.	6367.	6483.	6369.	6019.	5504.	4853.	4159.	3414.
3402.	4164.	4852.	5517.	6030.	6389.	6498.	6386.	6025.	5511.	4846.	4157.	3395.
3378.	4150.	4825.	5498.	6005.	6372.	6478.	6374.	6009.	5503.	4831.	4157.	3384.
3385.	4165.	4827.	5506.	6005.	6375.	6473.	6372.	6001.	5500.	4821.	4158.	3379.
3384.	4158.	4808.	5487.	5980.	6352.	6449.	6354.	5984.	5493.	4814.	4165.	3390.
3414.	4173.	4812.	5485.	5971.	6339.	6431.	6336.	5966.	5479.	4806.	4166.	3407.
3441.	4171.	4803.	5464.	5947.	6310.	6404.	6312.	5951.	5470.	4809.	4178.	3448.
3497.	4184.	4810.	5452.	5931.	6281.	6377.	6278.	5927.	5446.	4804.	4177.	3490.
3552.	4187.	4811.	5430.	5910.	6248.	6351.	6250.	5915.	5436.	4817.	4194.	3559.
3625.	4201.	4818.	5408.	5885.	6204.	6311.	6202.	5881.	5402.	4811.	4194.	3618.
3696.	4213.	4825.	5389.	5867.	6173.	6288.	6175.	5872.	5395.	4831.	4219.	3702.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 90. MD = 0. FD = 90. FD = 0. DEPTH = 1.8

NORMALIZED RMS= 0.847e-02

ERROR ARRAY...

-12.	-11.	-10.	-10.	-9.	-10.	-9.	-10.	-9.	-10.	-10.	-11.	-12.
-11.	-10.	-9.	-9.	-8.	-9.	-8.	-9.	-8.	-9.	-9.	-10.	-11.
-10.	-9.	-7.	-7.	-7.	-7.	-6.	-7.	-7.	-7.	-7.	-9.	-10.
-10.	-9.	-7.	-7.	-6.	-7.	-6.	-7.	-6.	-7.	-7.	-9.	-10.
-9.	-8.	-7.	-6.	-6.	-6.	-6.	-6.	-6.	-6.	-7.	-8.	-9.
-10.	-9.	-7.	-7.	-6.	-6.	-6.	-6.	-6.	-7.	-7.	-9.	-10.
-9.	-8.	-6.	-6.	-6.	-6.	-6.	-6.	-6.	-6.	-6.	-8.	-9.
-10.	-9.	-7.	-7.	-6.	-6.	-6.	-6.	-6.	-7.	-7.	-9.	-10.
-9.	-8.	-7.	-6.	-6.	-6.	-6.	-6.	-6.	-6.	-7.	-8.	-9.
-10.	-9.	-7.	-7.	-6.	-7.	-6.	-7.	-6.	-7.	-7.	-9.	-10.
-10.	-9.	-7.	-7.	-7.	-7.	-6.	-7.	-7.	-7.	-7.	-9.	-10.
-11.	-10.	-9.	-9.	-8.	-9.	-8.	-9.	-8.	-9.	-9.	-10.	-11.
-12.	-11.	-10.	-10.	-9.	-10.	-9.	-10.	-9.	-10.	-10.	-11.	-12.

INPUT PARAMETERS TO CREATE TEST ARRAY...

MI = 90. MD = 0. FD = 90. FD = 0. DEPTH = 1.6

NORMALIZED RMS= 0.139e-01