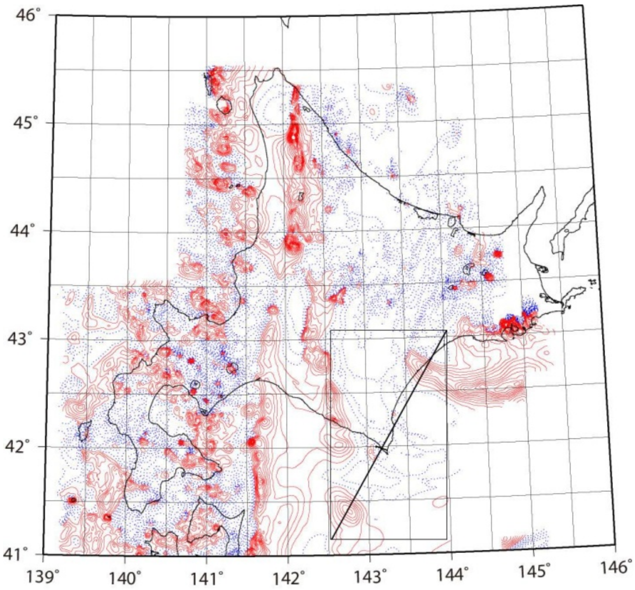
モデル計算の例―つづき

１．データを抽出する

→http://staff.aist.go.jp/r-morijiri/memomemo/inversion.html を参照

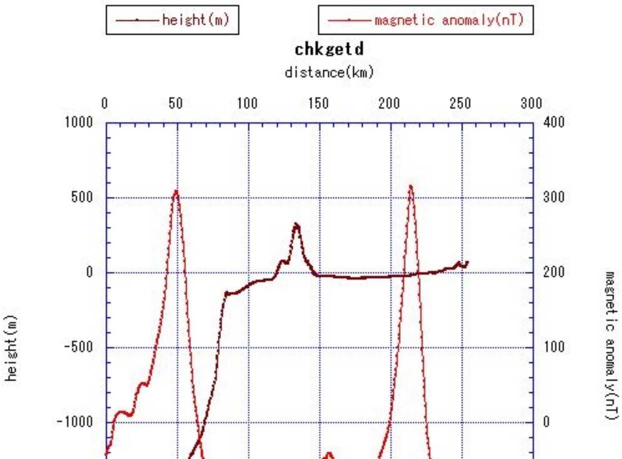


xxmn=626.1543

xxmx=744.533

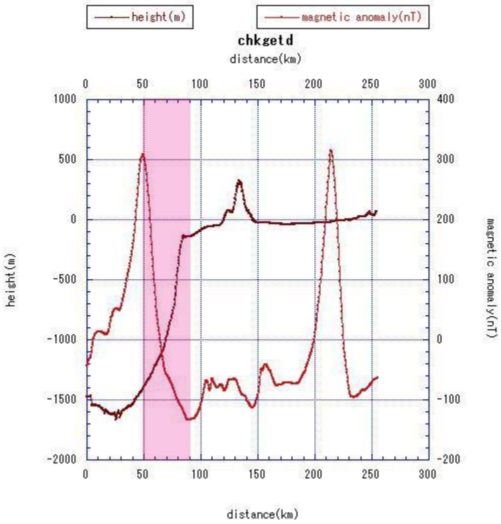
yymn=4539.8406

yymx=4765.1829

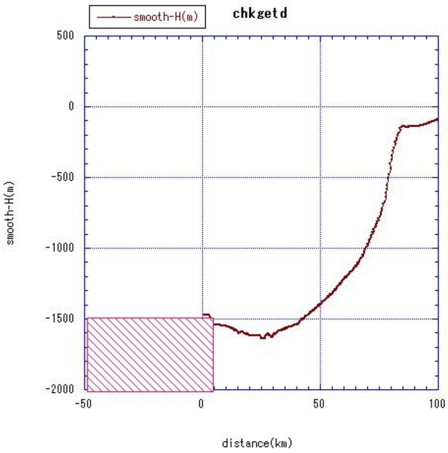


２．初期モデルを作る

目立つ磁気異常には、正負のピークをプリズムの角に合わせる。



打切り誤差の影響を避けるために、両端には広めのプリズムを置く。



初期モデル

上面深度はとりあえず地形の最深部1500mより少し深い2km＋測定高度500mとした。

与える磁化はすべてのプリズムで一様。外部磁場に平行で強度を適当に与える。

Y軸の中心にあたる点のIGRFの値：サンプルデータの場合は143度15分、42度0分

計算結果の例（とりあえず2000年で計算した。1点であれば京都大学のサイトが便利）

北緯 42.000 度、東経 143.250 度、標高: 0.0m での 2000 年 の地磁気要素（IGRF-11）

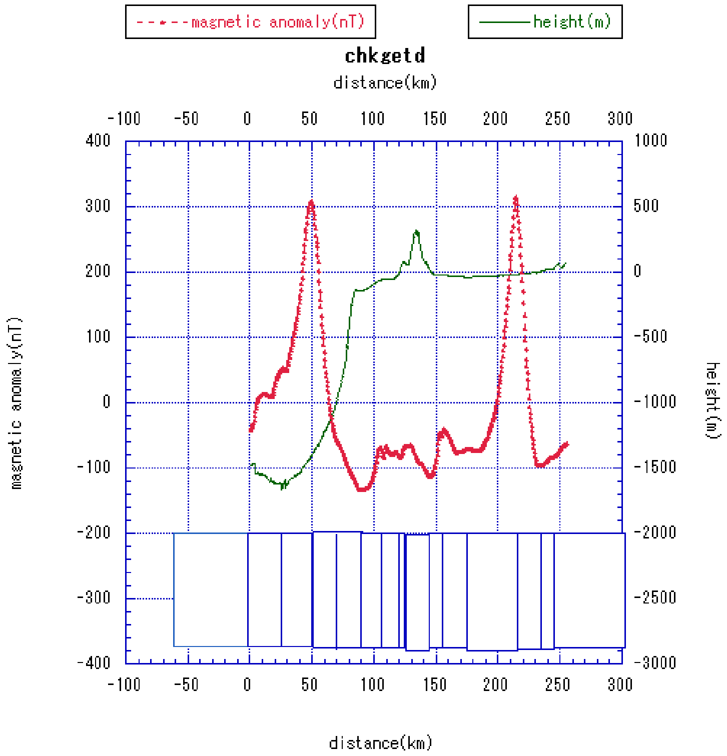
全磁力（F）48821.3 nT、偏角（D）-8.599度、伏角（I)55.904度

ここから偏角-8度、伏角56度を与える。

サンプルデータに見られる2つの大きな磁気異常は、正負のピークの間が20kmで振幅はざ

っと 400nT になる。ここで、20km×20km のプリズムが磁気異常を作るとして振幅を 400nT にするにはフォワードモデルで、磁化強度を 1.5A/m、上面深度を 2km（データ取得高度は 500m）とすればよい。この値を参考にして磁化強度は 1.5A/m とした。

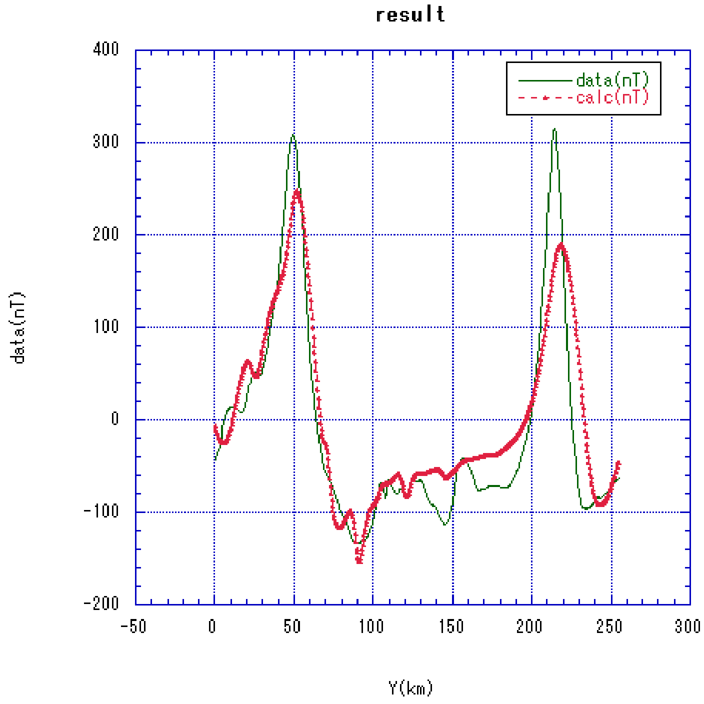
プリズムの幅はある程度、磁気異常プロファイルに従う。

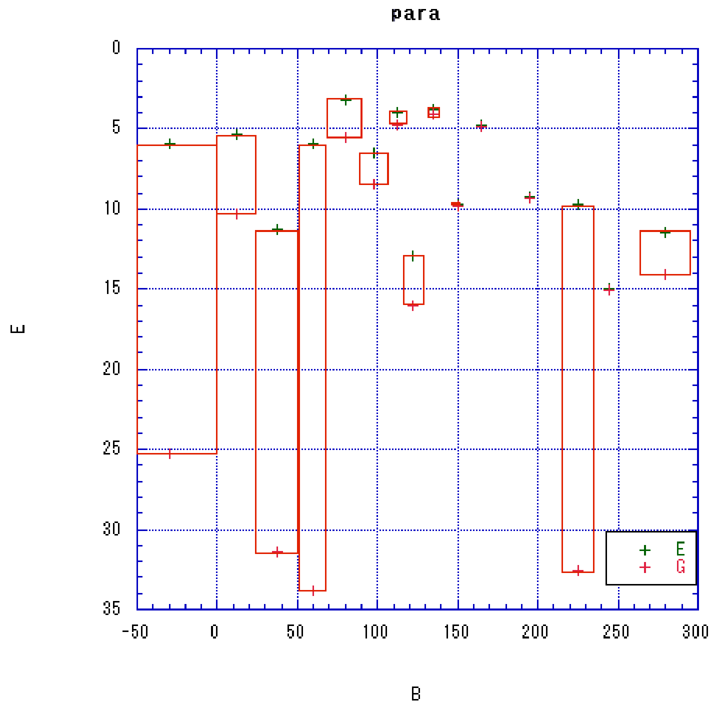


1 回目の結果

上面深度とプリズムの厚さを計算する。

結果は図のようになるが、これを見て、不自然なところは磁化強度を見直すなどフォワードモデルで修正を加える。





プログラム

```
c*******************************************************
c        program maginv for line-depth of surface and thickness
c                                                coded by J. Hara
c                                                arranged by RIE
```

```
c***********************************************************
c
c     input data:
c        magnetic anomaly data, profile : { No, distance(km), utm-easting(km), utm-northing(km), magnetic anomaly(nT)}
c     initial model : { no, x0, y0, a, b, h, d }
c
      implicit real*8 (a-h,o-y)
c
      common /comar1/f(3,1000)
      common /comin1/fmax,fmin,nd
      common /comche/h(20),d(20),offset
      common /comin3/a(20),b(20),Y0(20),c,e,X0
      common /comin4/hh(20),dd(20)
      common /comin5/tij
      common /comin6/nmb
      common /compai/pai
      common /comar3/g(1000)
      common /comar5/gp(41,41)
      common /comar6/dp(41)
c
      common /comin2/cxy
c
c
      pai=3.14159265358979323846
c
c     ***************
c     data read
c     ***************
c
      write(6,*) '         data input         '
c
      fmax=-99999.
      fmin=99999.
      nn=0
c
```

```
      open(10,file='I:¥airmag-db¥chkget.txt')
  100   continue
        read(10,*,end=99) nn,dis,ax,ay,dat
          f(1,nn)=dis
          f(2,nn)=0
          f(3,nn)=dat
c
        if(f(3,nn).gt.fmax) then
               fmax=f(3,nn)
               fxmx=f(2,nn)
               fymx=f(1,nn)
          else if(f(3,nn).lt.fmin) then
               fmin=f(3,nn)
               fxmn=f(2,nn)
               fymn=f(1,nn)
        end if
      go to 100
   99 continue
      close(10)
c
      nd=nn
      write(6,*) 'in data: nd ',nd
c
      write(6,*) 'fmax: x, y, f ',fxmx,fymx,fmax
      write(6,*) 'fmin: x, y, f ',fxmn,fymn,fmin
c
c     # angle in degree bewteen i-axis and Y-axis:cc
c     #データを抽出したときの utm(km)で表された x と y の最大値と最小値
       xxmn=626.1543
       xxmx=744.533
       yymn=4539.8406
       yymx=4765.1829
      dx=xxmx-xxmn
      dy=yymx-yymn
      cxy=atan2(dx,dy)
      cc=cxy*180/pai
   10 continue
c     # input inclination in degree
      cc=56
      c=cc*pai/1.8d2
c     write(6,*) '   input declination in degree   '
      dec=-8
c     angle between positive x-axis and mag. north
      ee=dec*pai/1.8d2
      e=cxy-ee
c
c    # blocks set on Y-axis:nmb
      nmb=15
c
      x0=0.0
      open(2,file='I:¥airmag-db¥para.txt')
c
      do 150 i=1,nmb
      read(2,*) no, y0(i), a(i), b(i),hh(i),dd(i)
      h(i)=hh(i)
      d(i)=dd(i)
  150 continue
c
      close(2)
c
c----------------------------------
      ff=0.
      do 880 in=1, nd
        ff=ff+f(3,in)
  880 continue
      ff=ff/float(nd)
      offset=ff
      write(6,*) ' init2 : offset   ', offset
c----------------------------------
c
c     #   m-intensity in A/m
      ti0=1.5
      ti1=ti0*0.001
      tij=ti1*100000
c
      icount=0
c    # iend: iteration
      iend=20
      do 15 i=1, iend
c--------------------------------------------------
      icount=icount+1
c
      call riron
c
      call inv(icount,iend)
c
c--------------------------------------------------
   15 continue
c
      stop
      end
c
c     ****************
      subroutine riron
c     ****************
c
      implicit real * 8 (a-h,o-y)
c
      common /comin1/fmax,fmin,nd
      common /comin6/nmb
      common /comar1/f(3,1000)
      common /comar3/g(1000)
      common /comar5/gp(41,41)
      common /comar6/dp(41)
c
      dimension hen(41)
c
      do 10 i=1,nmb*2+1
```

```
            do 20 j=1,nmb*2+1
               gp(i,j)=0
   20       continue
            dp(i)=0
   10 continue
         gmax=-99999.
         gmin=99999.
c
         do 100 nn=1,nd
c
            call keisan(nn)
c
            if(g(nn).GT.gmax) then
                  gmax=g(nn)
                  Xmax=f(2,nn)
                  Ymax=f(1,nn)
               else if(g(nn).LT.gmin) then
                  gmin=g(nn)
                  Xmin=f(2,nn)
                  Ymin=f(1,nn)
            end if
c
            call henbun(hen,f(1,nn),f(2,nn))
c
            do 210 i=1,nmb*2+1
               do 220 j=i,nmb*2+1
                  gp(i,j)=gp(i,j)+hen(i)*hen(j)
  220          continue
               dp(i)=dp(i)+(f(3,nn)-g(nn))*hen(i)
  210       continue
c
  100 continue
c
         write(6,*) 'gmax: x, y, g ',xmax, ymax, gmax
         write(6,*) 'gmin: x, y, g ',xmin, ymin, gmin
c
```

```
      do 30 i=2,nmb*2+1
        do 40 j=1,i-1
          gp(i,j)=gp(j,i)
 40     continue
 30 continue
c
      return
      end
c
c     ********************
      subroutine keisan(nn)
c     ********************
c
      implicit real * 8 (a-h,o-y)
c
      common /comche/h(20),d(20),offset
      common /comin3/a(20),b(20),Y0(20),c,e,X0
      common /comin5/tij
      common /comin6/nmb
      common /comar1/f(3,1000)
      common /comar3/g(1000)
c
      x=f(2,nn)-X0
c
      cc=cos(c)
      sc=sin(c)
      ce=cos(e)
      se=sin(e)
c
      g(nn)=0.0
      do 100 i=1,nmb
c
        y=f(1,nn)-Y0(i)
        p1=-x+a(i)
        p2=-x-a(i)
        q1=-y+b(i)
      q2=-y-b(i)
c
      hh=h(i)
      iii=1
  10  continue
c
      r1=sqrt(p1*p1+q1*q1+hh*hh)
      r2=sqrt(p2*p2+q2*q2+hh*hh)
      r3=sqrt(p1*p1+q2*q2+hh*hh)
      r4=sqrt(p2*p2+q1*q1+hh*hh)
c
      g1=cc*sc*se*(log((r1-p1)/(r1+p1))+log((r2-p2)/(r2+p2))
     -              -log((r3-p1)/(r3+p1))-log((r4-p2)/(r4+p2)))
      g2=cc*sc*ce*(log((r1-q1)/(r1+q1))+log((r2-q2)/(r2+q2))
     -              -log((r3-q2)/(r3+q2))-log((r4-q1)/(r4+q1)))
      g3=-2*cc*cc*se*ce*(log(r1+hh)+log(r2+hh)
     -                    -log(r3+hh)-log(r4+hh))
      g4=-cc*cc*ce*ce*
     -   (atan(p1*q1/(r1*hh+p1*p1+hh*hh))
     -   +atan(p2*q2/(r2*hh+p2*p2+hh*hh))
     -   -atan(p1*q2/(r3*hh+p1*p1+hh*hh))
     -   -atan(p2*q1/(r4*hh+p2*p2+hh*hh)))
      g5=-cc*cc*se*se*
     -   (atan(p1*q1/(r1*hh+q1*q1+hh*hh))
     -   +atan(p2*q2/(r2*hh+q2*q2+hh*hh))
     -   -atan(p1*q2/(r3*hh+q2*q2+hh*hh))
     -   -atan(p2*q1/(r4*hh+q1*q1+hh*hh)))
      g6=sc*sc*(atan(p1*q1/hh/r1)+atan(p2*q2/hh/r2)
     -          -atan(p1*q2/hh/r3)-atan(p2*q1/hh/r4))
c
      g0=(g1+g2+g3+g4+g5+g6)
c
      if(iii.EQ.1) then
        g(nn)=g(nn)+g0
        iii=0
        hh=h(i)+d(i)
```

```
            goto 10
         end if
         g(nn)=g(nn)-g0
c
  100 continue
c
      g(nn)=tij*g(nn)+offset
c
      return
      end
c
c     ************************************************
      subroutine henbun(hen,YY,XX)
c     ************************************************
c
      implicit real * 8 (a-h,o-y)
c
      common /comche/h(20),d(20),offset
      common /comin3/a(20),b(20),Y0(20),c,e,X0
      common /comin5/tij
      common /comin6/nmb
c
      dimension hen(41)
c
      gh1(p,q,r,hh)=2*hh*p/(q*q+hh*hh)/r
      gh2(p,q,r,hh)=2*hh*q/(p*p+hh*hh)/r
      gh4(p,q,r,hh)=-p*q/(p*p+hh*hh)/r
      gh5(p,q,r,hh)=-p*q/(q*q+hh*hh)/r
      gh6(p,q,r,hh)=-p*q*(p*p+q*q+2*hh*hh)
     -              /r/(p*p+hh*hh)/(q*q+hh*hh)
c
      x=XX-X0
      cc=cos(c)
      sc=sin(c)
      ce=cos(e)
      se=sin(e)
```

```
c
      do 100 i=1,nmb
c
        y=YY-Y0(i)
c
        p1=-x+a(i)
        p2=-x-a(i)
        q1=-y+b(i)
        q2=-y-b(i)
        h1=h(i)
        h2=h(i)+d(i)
        r11=sqrt(p1*p1+q1*q1+h1*h1)
        r12=sqrt(p1*p1+q1*q1+h2*h2)
        r21=sqrt(p2*p2+q2*q2+h1*h1)
        r22=sqrt(p2*p2+q2*q2+h2*h2)
        r31=sqrt(p1*p1+q2*q2+h1*h1)
        r32=sqrt(p1*p1+q2*q2+h2*h2)
        r41=sqrt(p2*p2+q1*q1+h1*h1)
        r42=sqrt(p2*p2+q1*q1+h2*h2)
c
        gh01=cc*sc*se*(gh1(p1,q1,r11,h1)-gh1(p1,q1,r12,h2)
     -                 +gh1(p2,q2,r21,h1)-gh1(p2,q2,r22,h2)
     -                 -gh1(p1,q2,r31,h1)+gh1(p1,q2,r32,h2)
     -                 -gh1(p2,q1,r41,h1)+gh1(p2,q1,r42,h2))
        gh02=cc*sc*ce*(gh2(p1,q1,r11,h1)-gh2(p1,q1,r12,h2)
     -                 +gh2(p2,q2,r21,h1)-gh2(p2,q2,r22,h2)
     -                 -gh2(p1,q2,r31,h1)+gh2(p1,q2,r32,h2)
     -                 -gh2(p2,q1,r41,h1)+gh2(p2,q1,r42,h2))
        gh03=-2*cc*cc*se*ce*(1/r11-1/r12+1/r21-1/r22
     -                       -1/r31+1/r32-1/r41+1/r42)
        gh04=-cc*cc*ce*ce*(gh4(p1,q1,r11,h1)-gh4(p1,q1,r12,h2)
     -                     +gh4(p2,q2,r21,h1)-gh4(p2,q2,r22,h2)
     -                     -gh4(p1,q2,r31,h1)+gh4(p1,q2,r32,h2)
     -                     -gh4(p2,q1,r41,h1)+gh4(p2,q1,r42,h2))
        gh05=-cc*cc*se*se*(gh5(p1,q1,r11,h1)-gh5(p1,q1,r12,h2)
     -                     +gh5(p2,q2,r21,h1)-gh5(p2,q2,r22,h2)
     -                              -gh5(p1,q2,r31,h1)+gh5(p1,q2,r32,h2)
     -                              -gh5(p2,q1,r41,h1)+gh5(p2,q1,r42,h2))
         gh06=sc*sc*(gh6(p1,q1,r11,h1)-gh6(p1,q1,r12,h2)
     -                     +gh6(p2,q2,r21,h1)-gh6(p2,q2,r22,h2)
     -                     -gh6(p1,q2,r31,h1)+gh6(p1,q2,r32,h2)
     -                     -gh6(p2,q1,r41,h1)+gh6(p2,q1,r42,h2))
         gd01=cc*sc*se*(-gh1(p1,q1,r12,h2)-gh1(p2,q2,r22,h2)
     -                         +gh1(p1,q2,r32,h2)+gh1(p2,q1,r42,h2))
         gd02=cc*sc*ce*(-gh2(p1,q1,r12,h2)-gh2(p2,q2,r22,h2)
     -                         +gh2(p1,q2,r32,h2)+gh2(p2,q1,r42,h2))
         gd03=-2*cc*cc*se*ce*(-1/r12-1/r22+1/r32+1/r42)
         gd04=-cc*cc*ce*ce*(-gh4(p1,q1,r12,h2)-gh4(p2,q2,r22,h2)
     -                              +gh4(p1,q2,r32,h2)+gh4(p2,q1,r42,h2))
         gd05=-cc*cc*se*se*(-gh5(p1,q1,r12,h2)-gh5(p2,q2,r22,h2)
     -                              +gh5(p1,q2,r32,h2)+gh5(p2,q1,r42,h2))
         gd06=sc*sc*(-gh6(p1,q1,r12,h2)-gh6(p2,q2,r22,h2)
     -                      +gh6(p1,q2,r32,h2)+gh6(p2,q1,r42,h2))
c
      hen(i+nmb)=tij*(gd01+gd02+gd03+gd04+gd05+gd06)
      hen(i)=tij*(gh01+gh02+gh03+gh04+gh05+gh06)
c
  100 continue
c
      hen(nmb*2+1)=1.
c
      return
      end
c
c     *********************
      subroutine inv(icount,iend)
c     *********************
c
      implicit real * 8 (a-h,o-y)
c
      common /comar1/f(3,1000)
      common /comar3/g(1000)
      common /comar5/gp(41,41)
      common /comar6/dp(41)
      common /comin1/fmax,fmin,nd
      common /comin3/a(20),b(20),Y0(20),c,e,X0
      common /comin6/nmb
      common /comche/h(20),d(20),offset
c
      dimension gpp(41,41),qq(5),pmp(41),pmpp(5,41),
     -          pm(5,42),wk(41),h0(20),d0(20)
c
      pp=0.0
      do 10 i=1,nmb*2+1
         pp=pp+gp(i,i)
   10 continue
      pp=pp/float(nmb*2+1)
c
      qq(1)=1.0d-2
      qq(2)=1.0d-1
      qq(3)=1.0d0
      qq(4)=1.0d1
      qq(5)=1.0d2
c
      do 20 i=1,nmb
         h0(i)=h(i)
         d0(i)=d(i)
   20 continue
c
      offs=offset
c
      poff=1
c        write(6,*) (dp(ii), ii=1, nmb*2+1)
c
      do 100 i=1,5
c
c
         ramuda=pp*qq(i)
      do 150 ii=1,nmb*2+1
        do 160 jj=1,nmb*2+1
          if(ii.EQ.jj) then
              gpp(ii,jj)=gp(ii,jj)+ramuda
            else
              gpp(ii,jj)=gp(ii,jj)
          end if
 160    continue
        pmp(ii)=dp(ii)
 150  continue
c
      nmb2=nmb*2+1
      call dlf2m(gpp,nmb2,nmb2,pmp,0.0,1,wk,ier)
c
      do 180 jj=1,nmb*2+1
        pmpp(i,jj)=pmp(jj)
 180  continue
c
      do 190 j=1,nmb
        pm(i,j)=pmp(j)+h0(j)
        pm(i,j+nmb)=pmp(j+nmb)+d0(j)
c
        if(pm(i,j+nmb).LE.0) then
          pm(i,j+nmb)=0.1
        end if
c
        h(j)=pm(i,j)
        d(j)=pm(i,j+nmb)
 190  continue
c
    pmp(nmb*2+1)=pmp(nmb*2+1)*poff
    pm(i,nmb*2+1)=pmp(nmb*2+1)+offs
    offset=pm(i,nmb*2+1)
c
      gmax=-99999.
      gmin=99999.
```

```
c
         do 200 nn=1,nd
c
            call keisan(nn)
c
            if(g(nn).GT.gmax) then
                  gmax=g(nn)
              else if(g(nn).LT.gmin) then
                  gmin=g(nn)
            end if
  200    continue
c
         pm(i,nmb*2+2)=0
c
         do 300 nn=1,nd
            pq=g(nn)-f(3,nn)
            pm(i,nmb*2+2)=pm(i,nmb*2+2)+pq*pq
  300    continue
c
         pm(i,nmb*2+2)=sqrt(pm(i,nmb*2+2)/nd)
c
  100 continue
c
      write(6,*)
c
      imin=1
      do 400 i=2,5
         if(pm(i,nmb*2+2).LT.pm(imin,nmb*2+2)) then
            imin=i
         end if
  400 continue
c
      write(6,*)
      write(6,*) 'saitekichi!!        ',icount
      write(6,*)
      write(6,888) (pmpp(imin,j),j=1,nmb)
      write(6,*)
      write(6,888) (pmpp(imin,j),j=nmb+1,nmb*2)
      write(6,*)
      write(6,999)   pmpp(imin,nmb*2+1)
      write(6,*)
      write(6,*) '----------'
      write(6,888) (pm(imin,j),j=1,nmb)
      write(6,*)
      write(6,888) (pm(imin,j),j=nmb+1,nmb*2)
      write(6,*)
      write(6,999) pm(imin,nmb*2+1)
      write(6,777) pm(imin,nmb*2+2),qq(imin)
      write(6,*)
c
  888 format (5(3x,f7.2))
  999 format (3x,f7.2)
  777 format (2(3x,f7.2))
c
      do 500 j=1,nmb
         h(j)=pm(imin,j)
         d(j)=pm(imin,j+nmb)
  500 continue
c
      offset=pm(imin,nmb*2+1)
c
      if(icount.eq.iend) then
         open(3,file='I:¥airmag-db¥result-inv.dat')
          write(3,*) icount,nmb
          write(3,*) (pm(imin,j),j=1,nmb)
          write(3,*) (pm(imin,j),j=nmb+1,nmb*2)
          write(3,*) pm(imin,nmb*2+1)
          write(3,*) pm(imin,nmb*2+2),qq(imin)
         do 991 k2=1,nd
          write(3,*) f(1,k2),f(3,k2),g(k2)
  991    continue
         close(3)
      end if
c
      return
      end
c
      subroutine dlf2m(a,n,na,b,eps,iopt,wk,ier)
c
c---------------------------------------------------------------------
c
c    all rights reserved,copyright(c)1980,hitachi,ltd.s-1511-1
c    licensed material of hitachi,ltd.
c
c    name                  - dlf2m : double precision
c
c    usage                 - call dlf2m(a,n,na,b,eps,iopt,wk,ier)
c
c    function              - by the modified cholesky method, we solve the
c                               system of linear equations in n unknowns wi-
c                               th real symmetric coefficient matrix.
c
c    arguments   a(na,n)   - input. matrix formed by left-side coeffi-
c                               cients of the equation. only elements of the
c                               upper triangular matrix may be given.
c                             output. cholesky-factorized results are given.
c                n         - input. number of unknowns (0<n<na).
c                na        - input. number of rows of the matrix a in the
c                               dimension statement of a main program.
c                b(n)      - input. right-side vector.
c                             output. solutions of the equation.
c                eps       - input. criterion for the singularity
c                               (eps>=0.0).
c                               when eps<0.0 is given, standard value is assumed.
c                iopt      - input.
c                               iopt=1,modified cholesky decomposition and
c                                      solution of the equation.
c                               iopt=2,modified cholesky decomposition only.
```

```
c                                 iopt=3,solution of the equation only.
c                  wk(n)      - work area.
c                  ier        - error indicator.
c                                 ier=     0,no error was detected.
c                                 ier=1000,the coefficient matrix is not
c                                             positive definite.
c                                 ier=2000,n<1,n>na,iopt<1 or iopt>3.
c                                 ier=3000,the matrix is nearly singular.
c
c
c    status                   - s-1511-1 05-02
c
c    history                  - date.      1979.12
c                                            1980.11
c                                            1982. 4
c                                            1986. 6
c
c-----------------------------------------------------------------------
c
      implicit real*8(a-h,o-z)
c     generic
      dimension   a(41,41), b(100), wk(100)
      data one/1.0d0/
      data sixtn/16.0d0/
c
c          check the input data.
c
      if(iopt.lt.1 .or. iopt.gt.3) go to 10
      if(n.ge.1 .and. n.le.na) go to 20
   10 continue
      go to 9999
   20 if(iopt.eq.3) go to 550
      nm1 = n-1
      ier = 0
      seps=eps
      if( eps ) 40, 50, 50
   40 seps = n*sixtn
c  40 seps = n*sixtn*ueps
   50 if(a(1,1)) 60, 60, 70
   60 j=1
      continue
c           initialization.
   70 continue
      wk(1) = one/a(1,1)
      if( nm1 ) 170, 170, 90
c
c           modified cholesky decomposition of the real symmetric matrix
c           (a) into a product of a lower trianguar matrix (l) that has
c           1 as the diagonal elements, a diagonal matrix (d) and (l)'s
c           transposed matrix (l(t)).
c
   90 do 500 j=2, n
      jsub1=j-1
      if( j-2 ) 130, 130, 100
  100 do 120 i=2, jsub1
      s=0.0
      isub1=i-1
      do 110 k=1, isub1
      s=s+a(k,i)*a(k,j)
c                       sum of l(i,k)*( (k,j) element of d*l(t) ).
  110 continue
      a(i,j)=a(i,j)-s
c                       = (i,j) element of d*l(t).
  120 continue
  130 s=0.0
      do 140 i=1, jsub1
      t=a(i,j)
      a(i,j)=wk(i)*t
c                       = (i,j) element of l(t).
      s=s+a(i,j)*t
  140 continue
      t=a(j,j)-s
      if( abs(t)-abs(a(j,j))*seps ) 150, 150, 160
  150 continue
      go to 9999
  160 wk(j)= one/t
      if(t) 165,165,500
  165 continue
c                   = 1/d(j,j)
  500 a(j,j)=t
c                   = d(j,j)
c
c         modified cholesky decomposition is completed.
c           now ,   d(i,i), 1/d(i,i), (i,j) of l(t) are stored
c                   in a(i,i), wk(i), a(i,j) over ( i=1,2,...,n ;
c                   j=i,i+1,...,n ) , respectively.
c
      if(iopt.eq.2) go to 9999
  550 continue
      if(ier.ge.3000) go to 9999
c
c         forward substitution.
c
      if( nm1 ) 170, 170, 180
          170 if(iopt.eq.2) go to 9999
      b(1) = b(1)*wk(1)
      go to 9999
  180 do 200 i=2, n
      s=0.0
      isub1=i-1
      do 190 k=1, isub1
      s=s+a(k,i)*b(k)
c                   sum of l(k,i)*b(k).
  190 continue
      b(i)=b(i)-s
  200 continue
      do 210 i=1,n
      b(i)=b(i)*wk(i)
  210 continue
c
c           backward substitution.
c
      np2=n+2
      do 240 k=2, n
      j=np2-k
      t=b(j)
      if( t ) 220, 240, 220
  220 jsub1=j-1
      do 230 i=1, jsub1
      b(i)=b(i)-a(i,j)*t
  230 continue
  240 continue
c
 9999 continue
      continue
      return
      end
```