```
c************************************************************
c     program maginv  for intensity
c                                           coded by J. Hara
c                                           arranged by RIE
c************************************************************
c
      implicit real*8 (a-h,o-y)
c
      common /comar1/f(3,1000)
      common /comin1/fmax,fmin,nd
      common /comche/h(200),d(200),offset
      common /comin3/a(200),b(200),Y0(200),c,e,X0
      common /comin5/taiji(200)
      common /comin6/nmb,iend
      common /compai/pai
      common /comar3/g(1000)
      common /comar5/gp(401,401)
      common /comar6/dp(401)
c
      common /comin0/fymx,fymn
cc
      pai=3.14159265358979323846
c
      ym=0
c
      fmax=-99999.
      fmin=99999.
      nn=0
c
      open(10,file='I:¥airmag-db¥chkget.txt')
   13 continue
        read(10,*,end=95) no,dis,utmn,utmet,amag
         nn=nn+1
         f(1,nn)=dis-ym
         f(2,nn)=0.0
         f(3,nn)=amag
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 13
   95 continue
      close(10)
c
      nd=nn
      write(6,*) 'in data: nd ',nd
c       read(5,*) ians
c       ians=ians
c
      write(6,*) 'fmax: x,  y,  f ',fxmx,fymx,fmax
      write(6,*) 'fmin: x,  y,  f ',fxmn,fymn,fmin
       read(5,*) ians
       ians=ians
cc
c
c
    1 continue
c
c     write(6,*) '   input inclination in degree   '
      cc=56
c     write(6,*) '   input declination in degree   '
      ee=-8.5
      c=cc*pai/180
c     write(6,*) 'angl: azimth of X-axis positive east of true North'
      angl=-28
      e=(ee-angl)*pai/180
c
c     height:survey altitude in km
c     tjj:magnetization A/m
      x0=0.0
      tjj=1.5
      height=0.5
c
      open(20,file='I:¥airmag-db¥modelt0.txt',status='old')
      read(20,*) nmb
      do 100 i=1,nmb
      read(20,*) y0(i),a(i),b(i),hh,d(i)
      taiji(i)=tjj*100
      h(i)=hh/1000+height
  100 continue
c
      close(20)
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
      iend=10
      do 10 icount=1, iend
c
      call riron
      call inv(icount)
c
   10 continue
c
      stop
      end
c
c
c     ****************
      subroutine riron
c     ****************
c
      implicit real * 8 (a-h,o-y)
c
      common /comin1/fmax,fmin,nd
      common /comin0/fymx,fymn
      common /comin6/nmb,iend
      common /comar1/f(3,1000)
      common /comar3/g(1000)
      common /comar5/gp(401,401)
      common /comar6/dp(401)
c
      dimension hen(401)
c
      do 10 i=1,nmb+1
         do 20 j=1,nmb+1
            gp(i,j)=0
   20    continue
         dp(i)=0
   10 continue
c
      gmax=-99999.
      gmin=99999.
```

```
c
      do 100 nn=1,nd
c
         call keisan(nn)
            if(f(1,nn).eq.fymx) then
               write(6,*) ' fmax ',f(1,nn), f(3,nn), g(nn)
            end if
            if(f(1,nn).eq.fymn) then
               write(6,*) ' fmin ',f(1,nn), f(3,nn), g(nn)
            end if
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             write(6,*) ' riron check '
c
         do 210 i=1,nmb+1
            do 220 j=i,nmb+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+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(200),d(200),offset
      common /comin3/a(200),b(200),Y0(200),c,e,X0
      common /comin5/taiji(200)
      common /comin6/nmb,iend
      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
         if(r1.le.p1 .or. r1.le.q1) then
            write(6,*) 'k ',x,y,p1,q1
            read(5,*) ians
            ians=ians
         end if
         if(r3.le.p1 .or. r3.le.q2) then
            write(6,*) 'k ',x,y,p1,q2
            read(5,*) ians
            ians=ians
         end if
         if(r2.le.p2 .or. r2.le.q2) then
            write(6,*) 'k ',x,y,p2,q2
            read(5,*) ians
            ians=ians
         end if
         if(r4.le.p2 .or. r4.le.q1) then
            write(6,*) 'k ',x,y,p2,q1
            read(5,*) ians
            ians=ians
         end if
         if(hh.eq.0) then
           write(6,*) 'check hh ',i
            read(5,*) ians
            ians=ians
         end if
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
c
         g0=(g1+g2+g3+g4+g5+g6)
         g00=g0*taiji(i)
c
         if(iii.EQ.1) then
            g(nn)=g(nn)+g00
            iii=0
            hh=h(i)+d(i)
            go to 10
         end if
         g(nn)=g(nn)-g00
c
  100 continue
c
c          write(6,*) 'check 101 keisan  ',nn,g(nn),iii
c
      g(nn)=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(200),d(200),offset
      common /comin3/a(200),b(200),Y0(200),c,e,X0
      common /comin5/taiji(200)
      common /comin6/nmb,iend
c     common /comar1/f(3,1000)
c
      dimension hen(401)
c
      x=XX-X0
      cc=cos(c)
      sc=sin(c)
      ce=cos(e)
      se=sin(e)
c
      gg=0.0
      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
        iii=1
        r1=r11
        r2=r21
        r3=r31
        r4=r41
        hh=h1
   10 continue
c
         if(r1.le.p1 .or. r1.le.q1) then
            write(6,*) xx,yy,p1,q1
            read(5,*) ians
            ians=ians
         end if
         if(r3.le.p1 .or. r3.le.q2) then
            write(6,*) xx,yy,p1,q2
            read(5,*) ians
            ians=ians
         end if
         if(r2.le.p2 .or. r2.le.q2) then
            write(6,*) xx,yy,p2,q2
            read(5,*) ians
            ians=ians
         end if
         if(r4.le.p2 .or. r4.le.q1) then
            write(6,*) xx,yy,p2,q1
            read(5,*) ians
            ians=ians
         end if
         if(hh.eq.0) then
           write(6,*) 'check hh ',i
            read(5,*) ians
            ians=ians
         end if
c
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
           gg=gg+g0
           iii=0
           r1=r12
           r2=r22
           r3=r32
           r4=r42
           hh=h2
           go to 10
        end if
        gg=gg-g0
c
      hen(i)=gg
c
  100 continue
c
      hen(nmb+1)=1.
c
      return
      end
c
c     ********************
      subroutine inv(icount)
c     ********************
c
      implicit real * 8 (a-h,o-y)
c
```

```
      common /comar1/f(3,1000)
      common /comar3/g(1000)
      common /comar5/gp(401,401)
      common /comar6/dp(401)
      common /comin1/fmax,fmin,nd
      common /comin3/a(200),b(200),Y0(200),c,e,X0
      common /comin6/nmb,iend
      common /comche/h(200),d(200),offset
      common /comin5/taiji(200)
c
      dimension gpp(401,401),qq(5),pmp(401),pmpp(5,401),
     -          pm(5,402),wk(401),t0(200)
c
      pp=0.0
      do 10 i=1,nmb+1
         pp=pp+gp(i,i)
   10 continue
      pp=pp/float(nmb+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
         t0(i)=taiji(i)
   20 continue
c
      offs=offset
c
      do 100 i=1,5
c
         ramuda=pp*qq(i)
         do 150 ii=1,nmb+1
            do 160 jj=1,nmb+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+1
         call dlf2m(gpp,nmb2,nmb2,pmp,0.0,1,wk,ier)
         write(6,*) ' ier = ', ier
c
         do 180 jj=1,nmb+1
            pmpp(i,jj)=pmp(jj)
  180    continue
c
         do 190 j=1,nmb
            pm(i,j)=pmp(j)+t0(j)
            go to 188
c---------------------------------------
            if((pm(i,j)).le.0) then
               pm(i,j)=0.01
               d(j)=d(j)-0.2
            end if
            if(pm(i,j).le.0.1 .and. d(j).le.0) then
               d(j)=0.1
               h(j)=h(j)+0.2
            end if
c---------------------------------------
  188       continue
c
```

```
            taiji(j)=pm(i,j)
  190    continue
c
      pm(i,nmb+1)=pmp(nmb+1)+offs
      offset=pm(i,nmb+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)=0
c
         do 300 nn=1,nd
            pq=g(nn)-f(3,nn)
            pm(i,nmb+2)=pm(i,nmb+2)+pq*pq
  300    continue
c
         pm(i,nmb+2)=sqrt(pm(i,nmb+2)/nd)
c
  100 continue
c
      write(6,*)
c
      imin=1
      do 400 i=2,5
         if(pm(i,nmb+2).LT.pm(imin,nmb+2)) then
            imin=i
         end if
  400 continue
c
      write(6,*)
      write(6,*) 'saitekichi!!    ',icount
      write(6,*)
      write(6,*) (pmpp(imin,j),j=1,nmb)
      write(6,*)
      write(6,*)  pmpp(imin,nmb+1)
      write(6,*)
      write(6,*) '----------'
      write(6,*) (pm(imin,j),j=1,nmb)
      write(6,*)
      write(6,*) pm(imin,nmb+1)
      write(6,*) pm(imin,nmb+2),qq(imin)
      write(6,*)
c     write(6,*) ' short stop at line 544 '
c
      if(icount.eq.iend) then
         open(3,file='I:¥result-inv.dat')
         write(3,*) icount,nmb
         do 990 k1=1,nmb
          write(3,*) y0(k1),a(k1),b(k1),pm(imin,k1),h(k1),d(k1)
  990    continue
          write(3,*) pm(imin,nmb+1)
         do 991 k2=1,nd
          write(3,*) f(1,k2),f(3,k2),g(k2)
  991    continue
         close(3)
      end if
  881   format(f12.2,2f8.1,3f12.2)
  882   format(3f12.2)
c
  888 format (5(3x,f12.2))
  999 format (3x,f12.2)
  777 format (2(3x,f12.2))
c
      do 500 j=1,nmb
         taiji(j)=pm(imin,j)
  500 continue
c
      offset=pm(imin,nmb+1)
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
c                           assumed.
c              iopt     - input.
c                           iopt=1,modified cholesky decomposition and
c                                  solution of the equation.
c                           iopt=2,modified cholesky decomposition
c                                  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(401,401), b(2500), wk(2500)
      data one,sixtn /1.0d0, 16.0d0 /
c     data ueps /z3410000000000000/
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 continue
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
c     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
```