

```

IMPLICIT REAL*8(A-H,O-Z)
C -----
  open(20,file='I:\grav-DB\xyll_conic_230.txt')
C -----
  open(10,file='I:\grav-DB\original\grid_land-sea_230.dat')
C -----
  olat=3600.00
  olon=13600.00
  QLAT = DEG(OLAT)
  QLON = DEG(OLON)
  1 continue
  read(10,100,end=2) no,xcor,ycor,bg
100 format(17,2F10.1,F10.3)
C -----
  CALL IKDO(YCOR,XCOR,ALAT,ALON,QLAT,QLON)
  plat=alat
  plon=alon
C -----
  WRITE(20,120) PLon,PLat,bg
120 format(f12.6,f11.6,f9.3)
  GO TO 1
  2 CONTINUE
  close(10)
  close(20)
  STOP
  END
C *****
  DOUBLE PRECISION FUNCTION DEG(Pdm)
  DOUBLE PRECISION Pdm,p,pl
  pl = 1.0d0
  p = pdm
  if( pdm .lt. 0.d0 ) pl = - pl
  if( pdm .lt. 0.d0 ) p = - p
  I = P/100.d0 + 0.0001d0
  DEG = DFLOAT(I) + ( P - 100.d0*DFLOAT(I) )/60.d0
  deg = pl * deg
  RETURN
  END
C *****
  DOUBLE PRECISION FUNCTION DEMI(Pdeg)
  DOUBLE PRECISION Pdeg,p,Q,pl
  pl = 1.0d0
  p = pdeg
  if( pdeg .lt. 0.d0 ) pl = - pl
  if( pdeg .lt. 0.d0 ) p = - p
  I = IFIX(SNGL( P * 1.00001d0 ))
  Q = ( P - DFLOAT(I) )*60.d0
  DEMI = 100.d0*DFLOAT(I) + Q
  demi = pl * demi
  RETURN
  END
C *****
  SUBROUTINE IKDO(XCOR,YCOR,PLAT,PLON,OLAT,OLON)

```

```

DOUBLE PRECISION SE,PLAT,PLON,XCOR,YCOR,OLAT,OLON,
& XPI,YPK,X,Y,DX,DY,AX,AY
C SE = 0.000000001
SE = 0.1
PLAT = OLAT + 1.d0
PLON = OLON + 1.d0
CALL XYCORD(OLAT,OLON,X,Y,OLAT,OLON)
CALL XYCORD(PLAT,PLON,XPI,YPK,OLAT,OLON)
XPI = XPI - X
YPK = YPK - Y
C -----
PLAT = OLAT + (XCOR-X)/XPI
PLON = OLON + (YCOR-Y)/YPK
1 CALL XYCORD(PLAT,PLON,X,Y,OLAT,OLON)
C -----
DX = XCOR - X
DY = YCOR - Y
AX = DABS(DX)
AY = DABS(DY)
IF( AX .LT. SE .AND. AY .LT. SE ) GO TO 2
PLAT = PLAT + DX/XPI
PLON = PLON + DY/YPK
GO TO 1
2 CONTINUE
RETURN
END
C *****
subroutine XYCORD(dlat,dlon,y,x,olat,olon)
implicit real*8(a-h,o-z)
iscale = 1000
c -----
c ar:Semimajor Axis
c fl:Flattening
c ep:Squre root of Eccentricity, ep**2 = 2*fl - fl**2
c -----
c ----- Bessel(Tokyo datum) -----
cc ar = 6377397.15500D0
cc fl = 1.0d0/299.1528128d0
c -----
c ----- GRS80(WGS84) -----
ar = 6378137.000D0
fl = 1.0d0/298.257223563d0
c -----
ramd0=olon
phai0=olat
e2 = 2*fl - fl**2
e4=e2*e2
e6=e4*e2
e8=e6*e2
e10=e8*e2
pai=3.141592653589d0
radian=pai/180.
phai0=phai0*radian

```

```

a = 1.+0.75*e2+45./64.*e4+175./256.*e6+11025./16384.*e8
&   +43659./65536.*e10
b = 0.75*e2+15./16.*e4+525./512.*e6+2205./2048.*e8
&   +72765./65536.*e10
c = 15./64.*e4+105./256.*e6+2205./4096.*e8+10395./16384.*e10
d = 35./512.*e6+315./2048.*e8+31185./131072.*e10
e = 315./16384.*e8+3465./65536.*e10
f = 693./131072.*e10
c...dsx = dsin( x*phai0 )
      ds2 = dsin( 2.0*phai0 )
      ds4 = dsin( 4.0*phai0 )
      ds6 = dsin( 6.0*phai0 )
      ds8 = dsin( 8.0*phai0 )
      ds10= dsin( 10.0*phai0 )
c
c...input parameters handling
c
      dlr =dlat*radian
      dlrc = dcos( dlr )
      dlrs = dsin( dlr )
      rrate=1000.d0 / iscale
      s2=dsin(2.0*dlr )-ds2
      s4=dsin(4.0*dlr )-ds4
      s6=dsin(6.0*dlr )-ds6
      s8=dsin(8.0*dlr )-ds8
      s10=dsin(10.*dlr )-ds10
      sss=a*(dlr -phai0)-.5*b*s2+0.25*c*s4-d*s6/6.0+e*s8/8.0-f*s10/10.
      sss=sss*(1.0-e2)*ar
      dramda= (dlon-ramd0)* radian
      cn=ar/dsqrt(1.0-e2*dlrs**2)
c
      cr=cn* dlrc
      cl=cn* dlrc / dlrs
      teta=dramda*dlrs
      x=cl*dsin(teta)
      y=sss+x*dtan(0.5*teta)
      x=x*rrate
      y=y*rrate
      return
      end

```