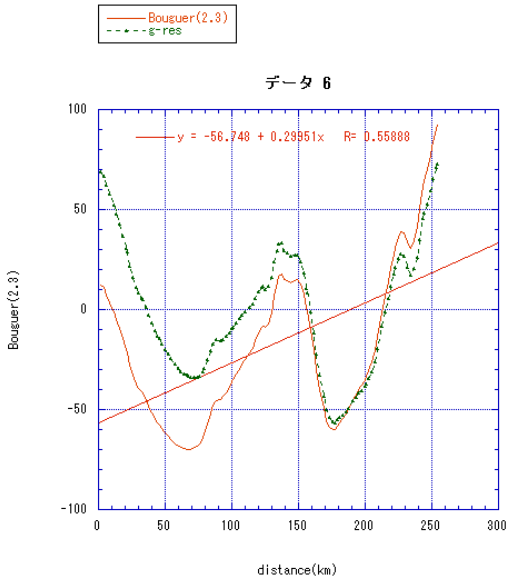
重力モデル計算メモ（２）

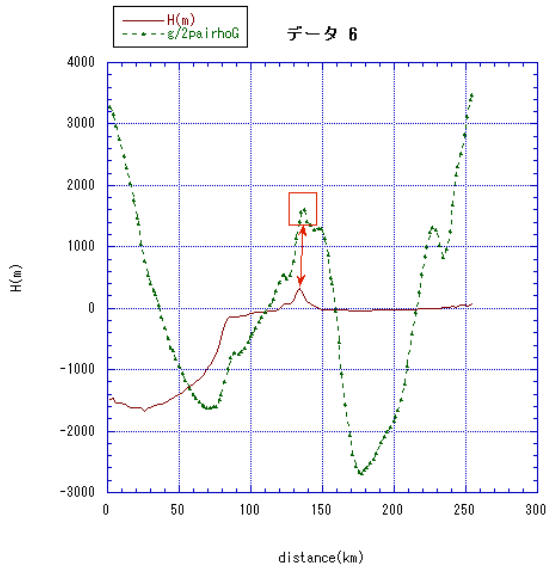
抜き出した重力データよりリニアトレンドを計算（カレイダグラフを利用）し、除去する。



重力値を２π Δ ρ Ｇで割って起伏を計算する。

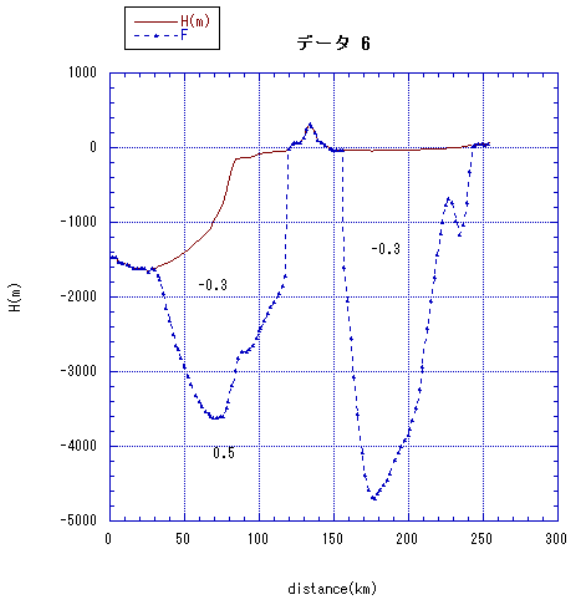
H(m)=g(mgal)/(2*3.14*6.67*0.001*0.5)

ここでΔ ρ=0.5g/cc

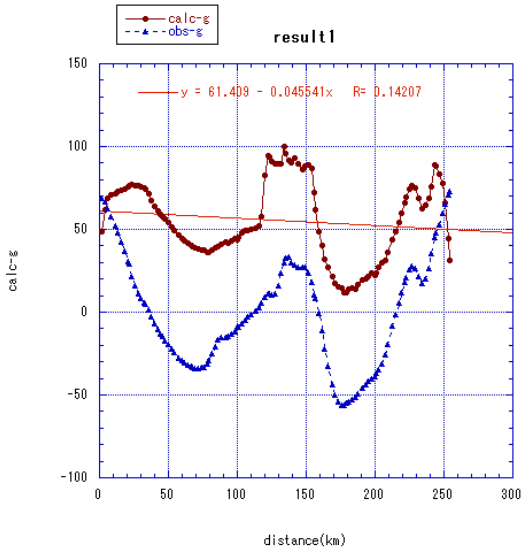


同様に抜き出した地形データと比べて、基盤は地表よりも深くならなくてはいけないから、山頂(319m)と基盤層（1416m）を合わせる（赤い矢印のところ)。まずは全体を 2000m 下げる。

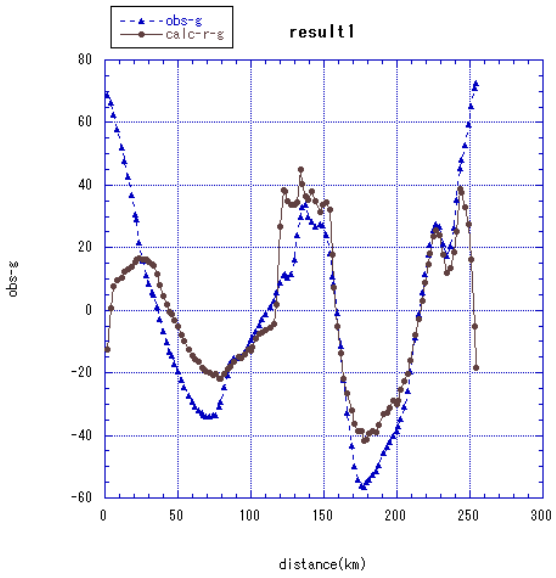
次に水深より上の部分を水深にそろえる。



上のモデル計算は下図。



ここで、計算された重力異常のリニアトレンドを計算し、差し引く。



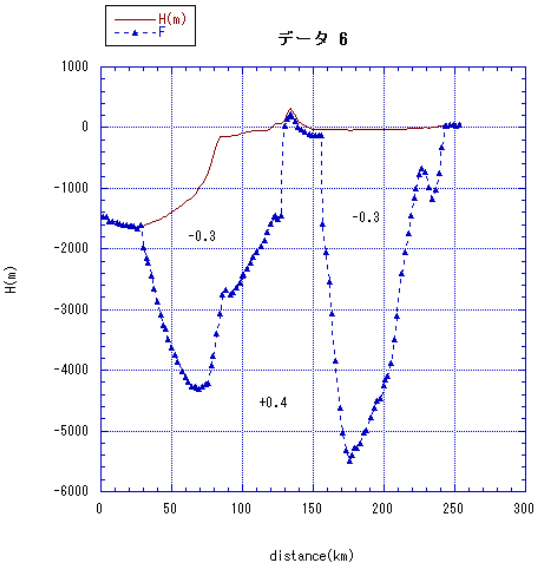
モデルを微修正する。

ΔH(m)=(calcg-obsg) (mgal)/(2*3.14*6.67*0.001*0.5)

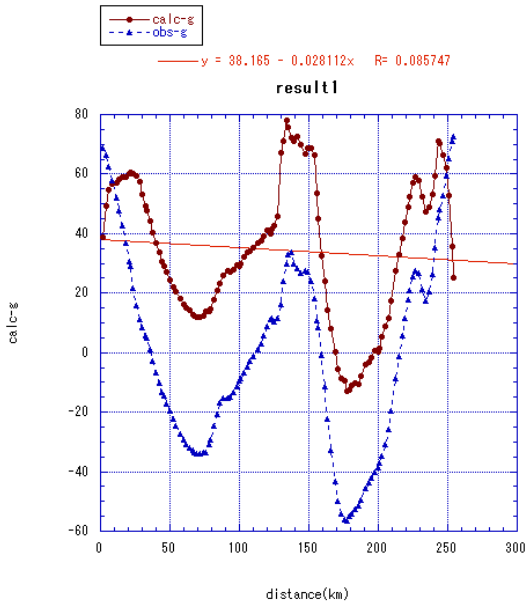
を計算し、修正したい箇所に加える。

何回か計算し、最終的に密度差を 0.4g/cm3 にした。

修正したモデル

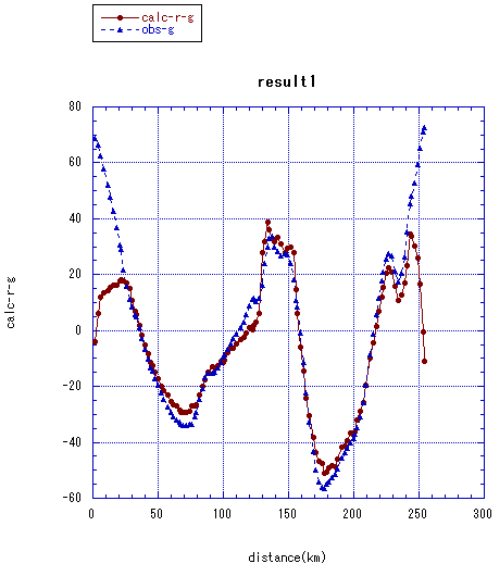


計算結果



計算値からトレンドを引く

y=38.165-0.028112*x



両端の値はモデルの打切り誤差の影響なので、ここでは無視する。

更に修正を加える場合もあるし、この辺で良しとする場合もある。

このモデルの修正を自動化するやり方もあるが、そのとき、どこで妥協するのか、数式で表現することが必要になる。

プログラム

```
c       cal-grv1
```

```
c                    BLAKELY(1995)
c
c     set prisms
c
c     observed point (x0,y0,z0)
c     prism x1-x2, y1-y2, z1-z2 (km)
c     external einc,edec : positive below horizontal,
c                          positive east of true North
c     theta azimth of X-axis positive east of true North
c
c     density rho(kg/(m**3))
c     distance parameters in units of km
c
      dimension x1s(500), x2s(500), y1s(500), y2s(500)
      dimension z1s(500), z2s(500), den(500)
      dimension xc(500),ggb(500),ggh(500)
c
      open(10,file='I:¥grav-DB¥datgr.txt')
      open(3,file='I:¥grav-DB¥parag.txt')
      open(2,file='I:¥grav-DB¥gresult.txt')
      data d2rad/0.017453293/
c--- data
      nq=0
    1 continue
      read(10,*,end=9) xp0,gb
      nq=nq+1
      xc(nq)=xp0
      ggb(nq)=gb
      go to 1
    9 continue
      nd=nq
      write(6,*) nd
c--- position(m)
      np=0
    2 continue
      read(3,*,end=99) xp1,xp2,gh,ghd,denp
```

```
      np=np+1
      x1s(np)=xp1
      x2s(np)=xp2
      z1s(np)=-gh/1000
      z2s(np)=-ghd/1000
      den(np)=denp
      go to 2
   99 continue
      nnp=np
      write(6,*) nnp
c
      do 23 i=1,nnp
       y1s(i)=-10
       y2s(i)=10
   23 continue
c
      do 10 mm=1,nd
      x0=xc(mm)
      y0=0
      z0=-0.3
      gg=0
       do 20 j=1,nnp
        x1=x1s(j)
        x2=x2s(j)
        y1=y1s(j)
        y2=y2s(j)
        z1=z1s(j)
        z2=z2s(j)
c
        rho=den(j)
c
      call gbox(x0,y0,z0,x1,y1,z1,x2,y2,z2,rho,g)
      gg=gg+g
   20 continue
         write(2,*) x0,gg,ggb(mm)
         write(6,*) x0,gg,ggb(mm)
```

```
   10 continue
      close(2)
      stop
      end
c
      subroutine gbox(x0,y0,z0,x1,y1,z1,x2,y2,z2,rho,g)
      dimension x(2),y(2),z(2),isign(2)
      real km2m
      data isign/-1,1/
c
      km2m=1000.
      si2mg=100000.
      gamma=6.670e-11
      twopi=6.2831853
c
      x(1)=x0-x1
      x(2)=x0-x2
      y(1)=y0-y1
      y(2)=y0-y2
      z(1)=z0-z1
      z(2)=z0-z2
      sum=0.
      do 1 i=1,2
       do 1 j=1,2
        do 1 k=1,2
         rijk=sqrt(x(i)**2+y(j)**2+z(k)**2)
         ijk=isign(i)*isign(j)*isign(k)
         arg1=atan2((x(i)*y(j)),(z(k)*rijk))
         if(arg1.lt.0) arg1=arg1+twopi
         arg2=rijk+y(j)
         arg3=rijk+x(i)
         if(arg2.le.0) write(6,*) 'bad point'
         if(arg3.le.0) write(6,*) 'bad point'
         arg2=alog(arg2)
         arg3=alog(arg3)
         sum=sum+ijk*(z(k)*arg1-x(i)*arg2-y(j)*arg3)
```

```
    1 continue
      g=rho*gamma*sum*si2mg*km2m
      return
      end
```

入力モデル x1(km),x2(km),z1(m),z2(m),den(kg/m3)

0.62523　2.85025　-1473.8　-6473.8　400

2.848605　5.076915　-1468.3　-6468.3　400

出力データ（x(km)、calcg(mgal)、obsg(mgal)）

1.73774	38.9541	68.6535
3.96276	49.2713	66.4221
6.19107	54.9227	62.3787
8.42010	56.4976	57.7191
11.5540	56.9493	52.0785