به نام دادار هستی بخش



تمرین درس پی پیشرفته

دکتر حمید رضا رازقی

محمدرضا نبی زاده شهربابک ۹۳۶۸۵۱۴۲

سری شماره ۲

پاییز ۹۳

فهرست

I. فهرست اشکال ۳

II. مقدمه و تشریح مسئله ۴

III. حل مسئله و بدست آوردن ضریب اطمینان سطح لغزش بحرانی ۵

۱. حل مسئله به روش نمودار ها (Charts) ۵

۲. روش شیب بی نهایت (Infinite Slope) ۷

۳. روش لگاریتم حلزونی (Spiral Logarithmic) ۹

۴. روش دایره ی سوئدی (Swedish Circle) ۱۴

۵. روش قطعات متداول (Ordinary method of Slices) ۲۰

۶. روش بیشاپ ساده شده (Simplified Bishop Procedure) ۲۵

۷. روش اسپنسر (Spencer procedure) ۲۹

۸. روش Lowe and Karafiath ۳۶

IV. کنترل ظرفیت باربری بستر خاکریز ۴۰

V. مقایسه ی نتایج با نرم افزار GeoSlope ۴۱

VI. نتیجه گیری ۴۱

VII. منابع ..... Error! Bookmark not defined.

فهرست اشکال

عکس ۱- شکل خاکریز و ابعاد آن ٤
عکس ۲- نمودار های جهت بدست آوردن پارامتر های آب و سربار و ترک کششی ٥
عکس ۳- محاسبه ی ضریب فشار آب حفره ای در داخل بدنه خاکریز و ارتفاع معادل ٥
عکس ۴- تعیین ضریب Nd جهت تعیین ضریب اطمینان ٦
عکس ۵- دیاگرام جسم آزاد در روش شیب بی نهایت ٧
عکس ۶- نتایج برنامه اجرا شده در محیط فرترن برای روش شیب بی نهایت ٨
عکس ۷- ضریب اطمینان ۱,۳۵ محاسبه شده توسط نرم افزار ٩
عکس ۸ - محل تقریبی مرکز مارپیچ لگاریتمی ١٠
عکس ۹- نمودار ارائه شده در کتابTheoretical Foundation Engineering نوشته Braja M. Das جهت محاسبه سطح منحنی ١١
عکس ۱۰- پارامتر های استفاده شده برای محاسبه ی مرکز سطح مارپیچ لگاریتمی ١١
عکس ۱۱- نتایج محاسبه شده برا ضریب اطمینان توسط روش مارپیچ لگاریتمی ١٣
عکس ۱۲- سطوح لغزش پیشنهادی توسط V.N.S. Murthy برای روش دایره ١٤
عکس ۱۳- نمودار های پیشنهادی توسط V.N.S. Murthy برای روش دایره ١٤
عکس ۱۴- سطح لغزش ارائه شده توسط Manjriker Gunaratne ١٥
عکس ۱۵- مرکز جرم قطاع دایره ١٦
عکس ۱۶- ضریب اطمینان بحرانی برای روش بازوی لنگر ١٩
عکس ۱۷- ضریب اطمینان بحرانی توسط روش Foundation Engineering Handbook ١٩
عکس ۱۸- دیاگرام آزاد نیرو های وارده بر یک قطعه در روش قطعات متداول ٢٠
عکس ۱۹- دایره ی بحرانی بر اساس روش قطعات متداول در نرم افزار GeoSlope ٢١
عکس ۲۰- هندسه ی تعمیم داده شده ی خاکریز و محاسبه ی ضریب اطمینان ۱,۲۸ توسط نرم افزار ٢٢
عکس ۲۱- ضریب اطمینان بحرانی بدست آمده از روش کد نویسی با Fortran ٢٤
عکس ۲۲- دیاگرام آزاد نیرو های وارده بر یک قطعه در روش بیشاپ ٢٥
عکس ۲۳- سطح گسیختگی بحرانی برای روش بیشاپ ساده شده و جانبو ٢٦
عکس ۲۴- نتایج روش بیشاپ با برنامه نویسی به زبان Fortran ٢٨
عکس ۲۵- دیاگرام نیرو های وارده بر هر قطعه در روش اسپنسر ٢٩
عکس ۲۶ - نابرابری جزئی مقادیر ضریب اطمینان ناشی از برآورده شدن معادله ی ممان و نیروی بین قطعه ای ٣٠
عکس ۲۷- شکل سمت چپ نیروی بین قطعه ای Z و شکل سمت راست نیروی عمود بر هر قطعه N ٣١
عکس ۲۸- ضریب اطمینان ۲,۱۸ برای سطح لغزش دلخواه در روش اسپنسر ٣٥
عکس ۲۹- دیاگرام نیرو های وارد بر یک قطعه در روش Lowe ٣٦
عکس ۳۰- سطح لغزش بحرانی و ضریب اطمینان بحرانی ۱,۴۵ ارائه شده بر اساس روش Lowe ٣٧
عکس ۳۱- نتایج حاصله از روش Lowe در زبان Fortran ٤٠
عکس ۳۲- بهینه سازی سطح لغزش شکل سمت چپ حالت عادی و شکل سمت راست بهینه سازی شده ٤٢
عکس ۳۳- محدوده ی سطح گسیختگی های بحرانی در GeoSlope ٤٢

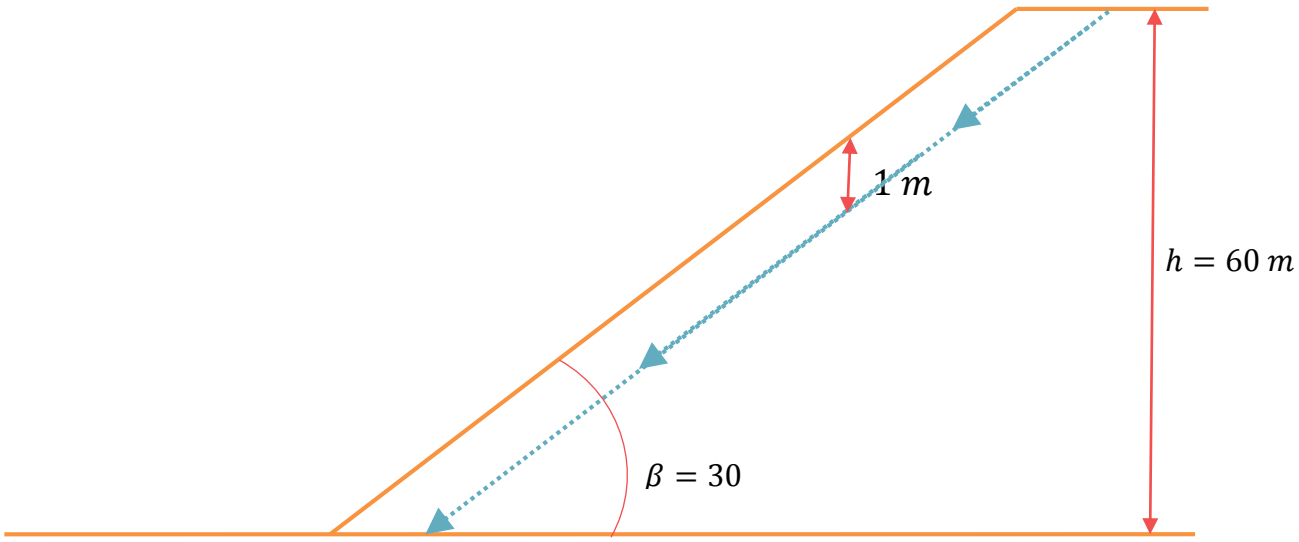
مقدمه و تشریح مسئله

در این مسئله با یک خاکریز رو برو هستیم که باید پایداری آن با روش های مختلف بررسی و ضرائب اطمینان آن محاسبه شود.

مشخصات خاک برابر خواهد بود با :

$$\gamma = 18\ {}^{kN}/_{m^3}\ , c = 70\ kPa\ , \varphi = 38°$$

هندسه ی خاکریز در شکل زیر نشان داده شده است.



عکس ۱- شکل خاکریز و ابعاد آن

این مسئله باید به روش های زیر حل شود :

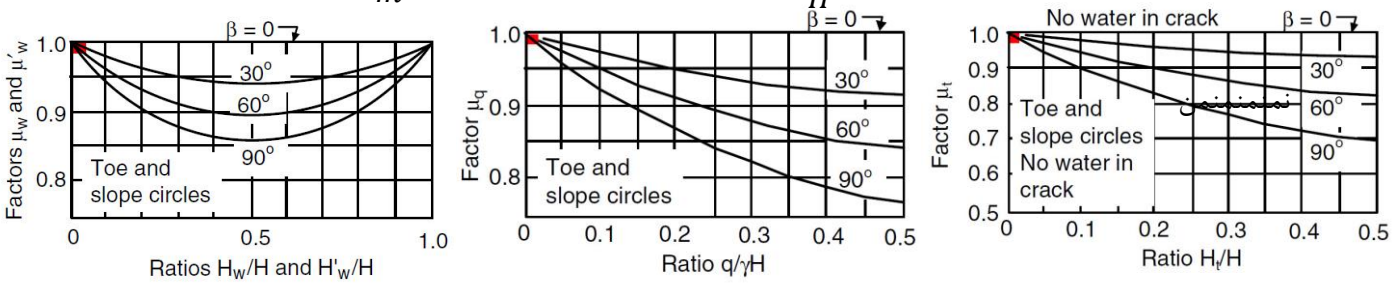
۱. نمودار ها (Charts)
۲. شیب بی نهایت (Infinite Slope)
۳. مارپیچ لگاریتمی (Spiral Logarithmic)
۴. دایره ی سوئدی (Swedish Circle)
۵. قطعات متداول (Ordinary)
۶. بیشاپ (Bishop)
۷. اسپنسر (Spencer)
۸. روش Lowe and Karafiath

۱. حل مسئله به روش نمودار ها (Charts)

در این روش ابتدا باید توسط نمودار هایی که در کتاب J. Michael) Soil Strength and Slope Stability Duncan, 2005)معرفی شده اند. به محاسبه یک سری ضرایب اقدام نمود و سپس با استفاده از رابطه ی بین ضرایب مقدار ضریب اطمینان را پیدا کرد. شرح این روش برای مسئله ی مذکور بدین شرح است.

Step 1 : Calculation of P_d :

$$\beta = 30^\circ , \gamma = 18\ {}^{kN}/_{m^3} , H = 60\ m, q = 0 , H_w = 0 , \frac{H_w}{H} = 0$$

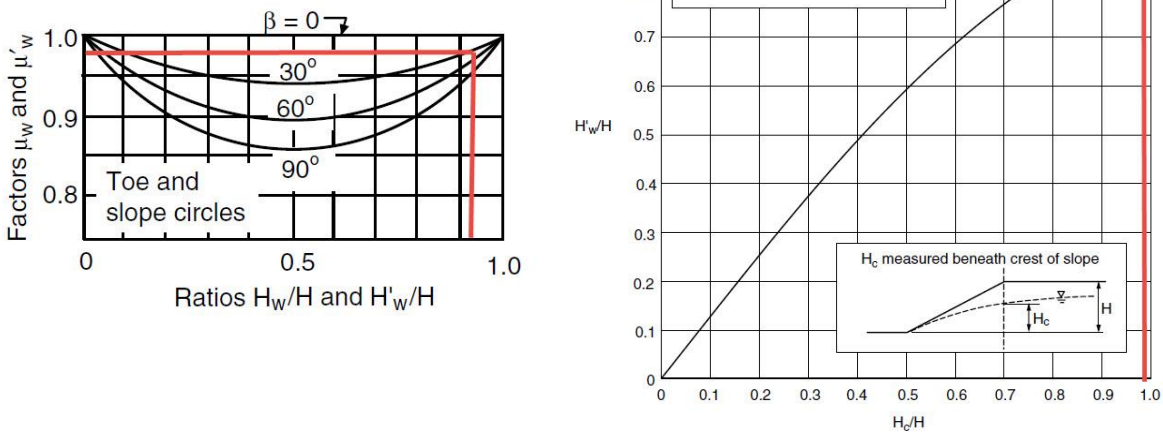


عکس ۲- نمودار های جهت بدست آوردن پارامتر های آب و سربار و ترک کششی

As figures shows :

$$\mu_q = \mu_t = \mu_w = 1.0 \ , \qquad P_d = \frac{\gamma H + q - \gamma_w H_w}{\mu_q \mu_t \ \mu_w} = \frac{100 * 48 - 0}{1 * 1 * 1} = 1080\ {}^{kN}/_{m^2}$$

Step 2 : Calculation of P_e :



عکس ۳- محاسبه ی ضریب فشار آب حفره ای در داخل بدنه خاکریز و ارتفاع معادل

$$H_c = 60 - 1 = 59\ m , \frac{H_c}{H} = 0.983 , \frac{H'_w}{H} = 0.925 , \mu'_w = 0.975 , H'_w = 0.975 * 60 = 58.5\ m$$

$$P_e = \frac{\gamma H + q - \gamma_w H'_w}{\mu_q \ \mu_w} = \frac{18 * 60 - 10 * 58.5}{1 * 1} = 495\ {}^{kN}/_{m^2}$$

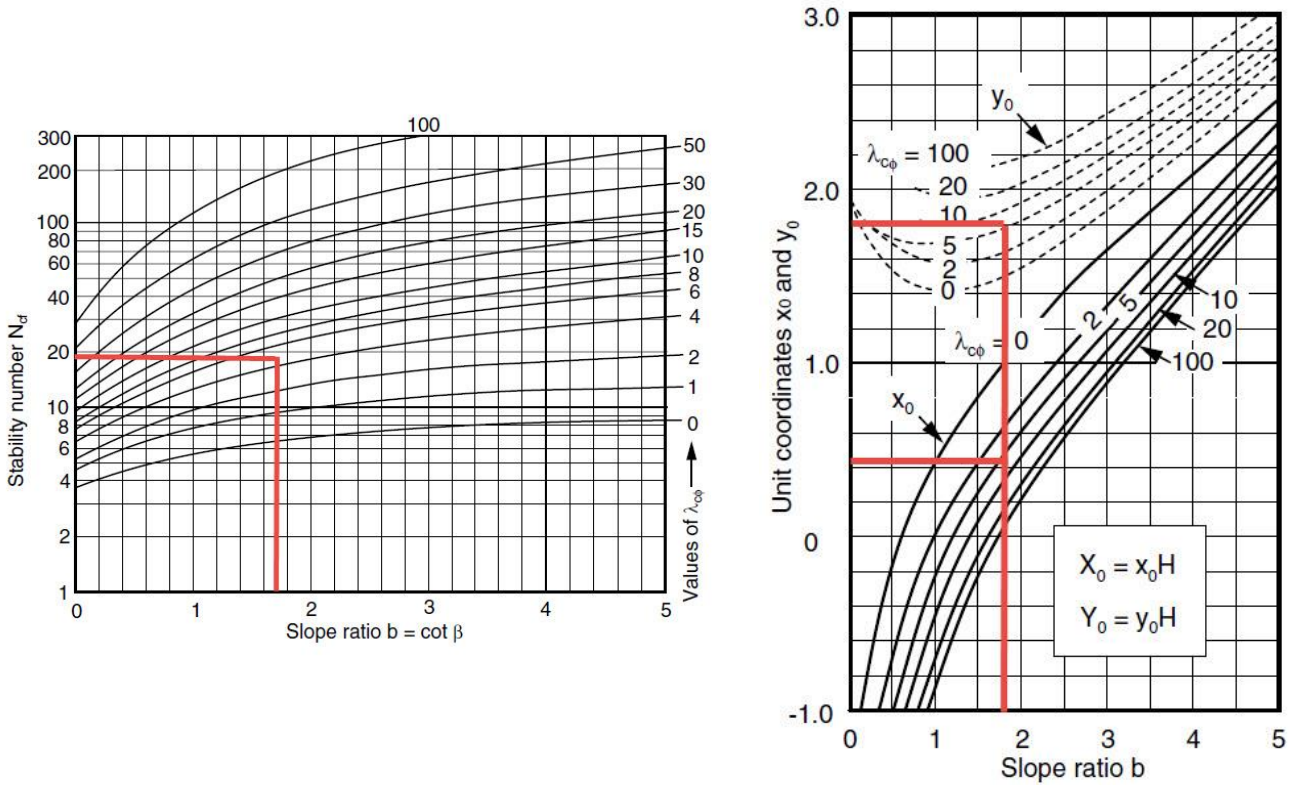
Step 3 : Calculation of $\lambda_{c\varphi}$:

$$\varphi = 38^\circ , c = 70\ kPa\ , \lambda_{c\varphi} = \frac{P_e \tan\varphi}{c} = \frac{495 * \tan 38}{70} = 5.52$$

Step 4 : Determination of N_{cf}:

$$b = \cot\beta = 1.73\,, \lambda_{c\varphi} = 5.52\,, N_{cf} = 19$$

Step 5 : Computation of F:



عکس ۴- تعیین ضریب Nd جهت تعیین ضریب اطمینان

$$F = N_{cf}\frac{c}{P_d} = 19 * \frac{70}{1080} = 1.23$$

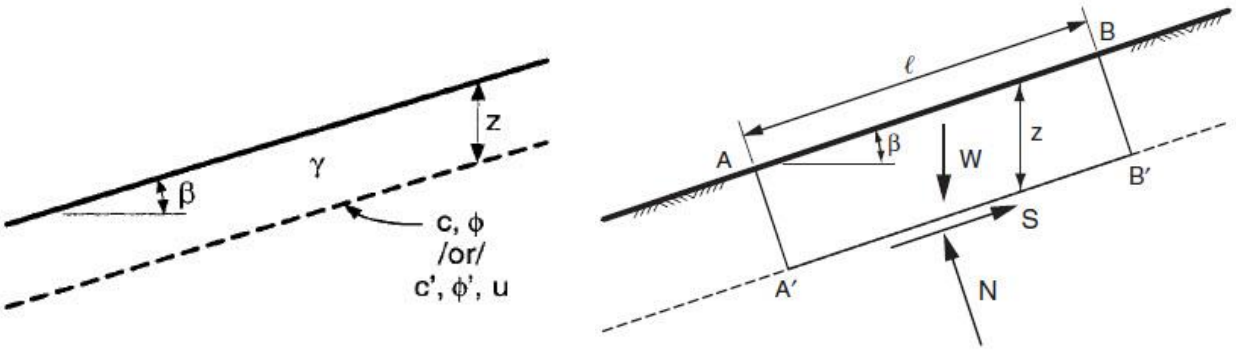
آنچه که در اینجا مهم است بدست آوردن محل تقریبی مرکز دایره منحنی گسیختگی می باشد پس با استفاده از نمودار روبرو می توانیم به طور تقریبی مرکز محل دایره گسیختگی را پیدا کنیم. که در روش های دیگر بتوانیم استفاده تفریبی کنیم از این نقطه :

$$X_o = x_0H = 0.28 * 60 = 16.8\,m$$
$$Y_o = y_0H = 1.8 * 60 = 108\,m$$

در اینجا استفاده از روش نمودار ها به پایان رسید. ضریب اطمینان ۱,۲۳ مقدار مناسبی جهت پایداری این خاکریز است.

۲. روش شیب بی نهایت (Infinite Slope)

در این روش تمام معادلات استاتیکی در دو جهت بر آورده می شود و از لحاظ کردن معادله لنگر خود داری می شود. شکل المان های مورد بررسی در این روش بدین گونه است :



عکس ۵- دیاگرام جسم آزاد در روش شیب بی نهایت

برای حل این مسئله به روش شیب بی نهایت از فرمول کلی این روش استفاده می کنیم که به شرح زیر است :

$$F = \frac{c' + (\gamma z \cos^2(\beta) - u)\tan\varphi'}{\gamma z \cos(\beta)\sin(\beta)}$$

$$u = \gamma_w h_w \cos^2(\beta)$$

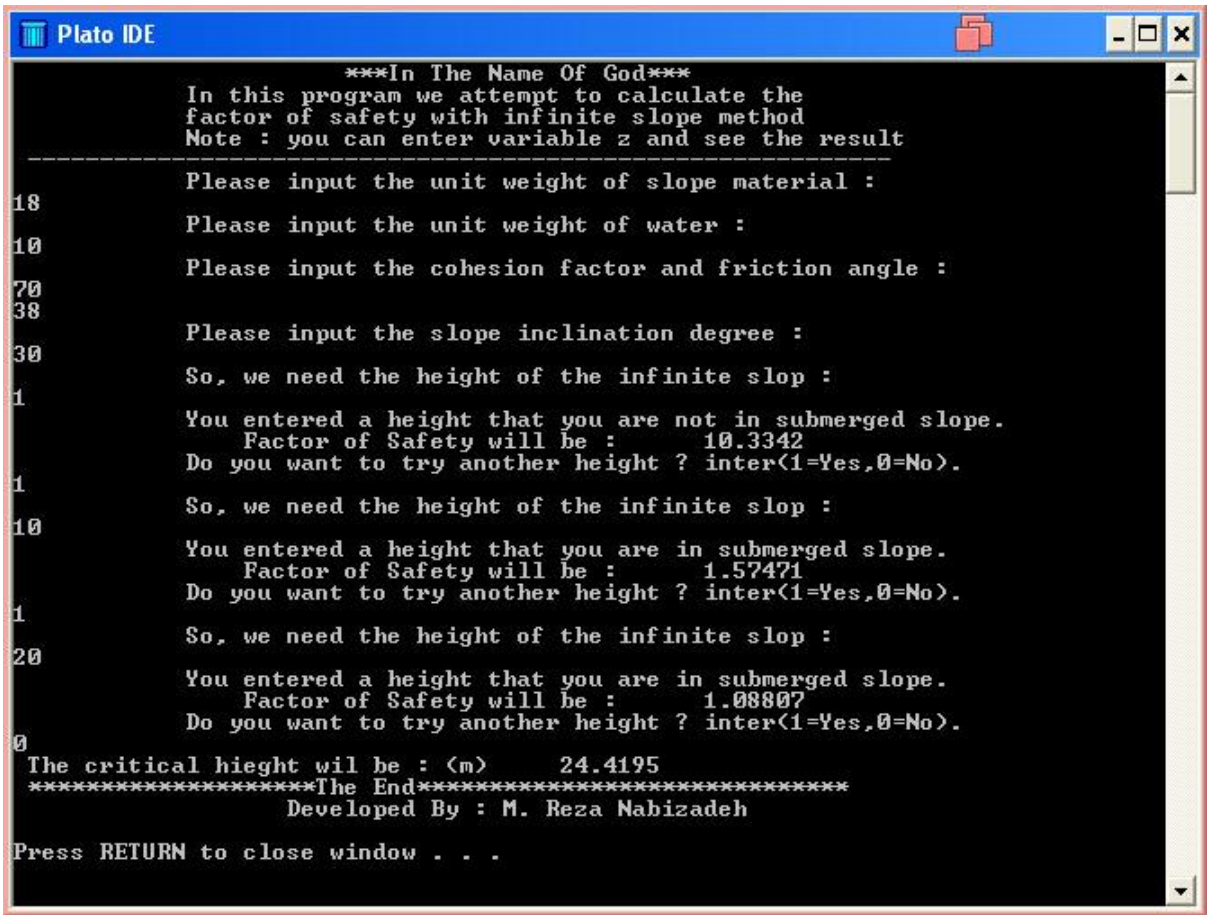
حال می توانیم با کد نویسی و بر حسب z های مختلف مقدار ضرایب اطمینان ها را بیابیم.

در این کد ابتدا برای هر ارتفاع دلخواه مقدار ضریب اطمینان محاسبه می شود سپس با قرار دادن مقدار ضریب اطمینان برابر ۱ و حل معادله به صورت معکوس ارتفاع بحرانی پیدا می شود.

کد برنامه ی نوشته شده به شرح زیر است :

```
program infiniteslopemethod
implicit none
real c,gams,gamw,z,b,fi ! Slope parameters
real pi,n,t,m,u,hw,cr,fir,br,FS,a,cz !Calculated parameters
        PRINT *,  ***In The Name Of God***'
  PRINT *, 'In this program we attempt to calculate the'
  PRINT *, ' factor of safety with infinite slope method'
  PRINT *, ' Note : you can enter variable z and see the
result'!Programs information
        print *, '-----------------------------------------------'
  print *, 'Please input the unit weight of slope material :'
  read *, gams
  print *, 'Please input the unit weight of water :'
  read *, gamw
  print *, 'Please input the cohesion factor and friction angle'
  read *, c,fi
  print *, ' Please input the slope inclination degree'
  read *, b
17  print *, ' So, we need the height of the infinite slop ?:'
  read *, z
  pi=asin(1.0)*2.0
  fir=(pi*fi)/180
  br=(pi*b)/180
  t=tan(fir)
  n=gams*z*((cos(br))**2)
  m=gams*z*cos(br)*sin(br)
                IF (z<=1) THEN
  print *, 'You entered a height that you are not in
submerged slope'
  FS=(c+n*t)/m
        print *, '    Factor of Safety will be : ' ,FS
                END IF
      IF (z>1) THEN
  print *, ' You entered a height that you are in submerged
slope'
 hw=z-1
 u=gamw*hw*((cos(br))**2)
FS=(c+(n-u)*t)/m
        print *, '    Factor of Safety will be : ' ,FS
                END IF
  print *, ' Do you want to try another height ? inter(1,0)'
  read *, a
        IF (a==1) THEN
goto 17
        END IF
 cz=(c+gamw*((cos(br))**2)*t)/(gams*cos(br)*sin(br)-
((gams*((cos(br))**2)-gamw*((cos(br))**2))*t))
  print *, 'The critical hieght wil be : (m)',cz
  print *, '********************The
End*****************************'
  print *, 'Developed By : M. Reza Nabizadeh'
 end program infiniteslopemethod
```

تصویری از نتایج محاسبه شده توسط کد نوشته شده در فرترن در این جا آورده شده است.

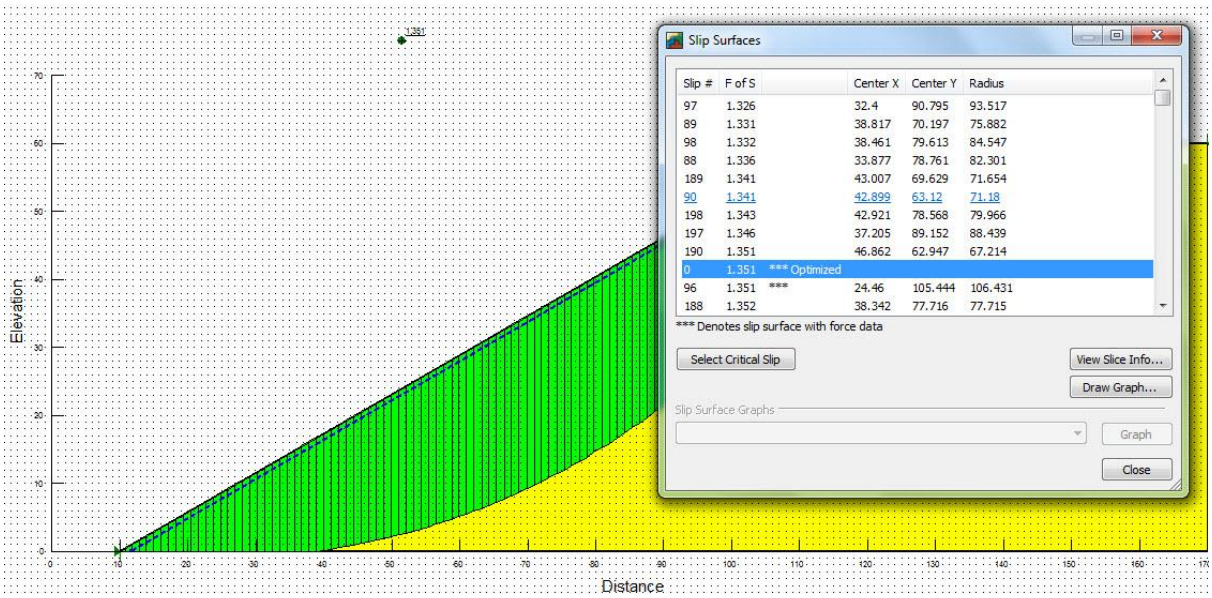


عکس ۶- نتایج برنامه اجرا شده در محیط فرترن برای روش شیب بی نهایت

مشاهده می شود که با افزایش ارتفاع لایه ی شیب بی نهایت ضریب اطمینان کاهش یافته و ارتفاع بحرانی که در آن ضریب اطمینان برابر یک خواهد بود نیز محاسبه شده است. در اینجا استفاده از روش شیب بی نهایت به پایان می رسد.

۳. روش لگاریتم حلزونی (Spiral Logarithmic)

این روش جزء روش های پیچیده در محاسبات ضریب اطمینان می باشد. ابتدا در این روش باید یک سطح و پس از آن یک مقدار ضریب اطمینان اولیه حدس زده شود. برای حدس مقدار ضریب اطمینان اولیه می توان از نرم افزار Geo-Slope استفاده نمود. کمترین مقدار ضریب اطمینان توسط روش قطعات برابر ۱,۳۵ می باشد. و این مقدار اولیه را برای حدس مقدار FS در روابط مارپیچ لگاریتمی استفاده می نمائیم.



عکس ۷- ضریب اطمینان ۱,۳۵ محاسبه شده توسط نرم افزار

روابط روش مارپیچ لگاریتمی به شرح زیر می باشد :

$$\tan(\varphi_d) = \frac{\tan(\varphi)}{FS}\ , r = r_0 e^{\theta \tan(\varphi_d)}\ , FS = \frac{\sum r_i c_i \Delta l_i}{Wa}$$

اما باید سیگمای بوجود آمده در بالا را به انتگرال تبدیل نمود. و از طرفی با یک انتگرال گیری مقدار W را نیز پیدا نمود.

$$\int r_i c_i \Delta l_i = \int_0^{\theta} c_i r_i^2 \, d\theta = c r_0^2 \int_0^{\theta} e^{2\theta \tan(\varphi_d)} \, d\theta = \frac{c r_0^2}{2\tan(\varphi_d)} e^{2\theta \tan(\varphi_d)} \Big|_0^{\theta}$$

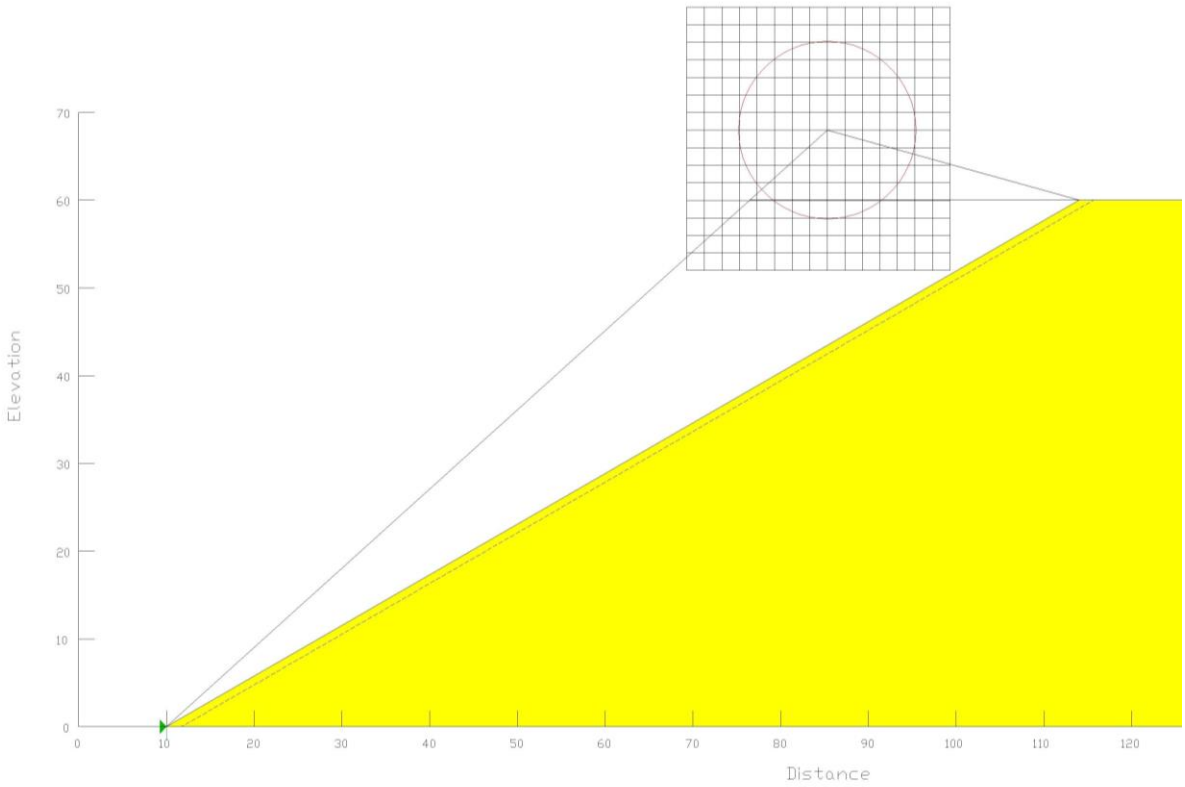
$$= \frac{c r_0^2}{2\tan(\varphi_d)} (e^{2\theta \tan(\varphi_d)} - 1)$$

حال نوبت به محاسبه ی سطح مارپیچ لگاریتمی می رسد.

$$\int_0^{\theta} \frac{r_i^2}{2} d\theta = \frac{r_0^2}{2} \int_0^{\theta} e^{2\theta \tan(\varphi_d)} \, d\theta = \frac{r_0^2}{4\tan(\varphi_d)} e^{2\theta \tan(\varphi_d)} \Big|_0^{\theta} = \frac{r_0^2}{4\tan(\varphi_d)} (e^{2\theta \tan(\varphi_d)} - 1)$$

البته فرمول بالا مساحت تمام سطح است و باید قسمت های غیر خاکریز از آن کم شود.

با سعی و خطای دستی با نرم افزار AutoCAD حدود نقطه ای که باید از آن سطح گسیختگی رسم شود را پیدا می کنیم. فرض می شود که سطح گسیختگی از پنجه ی خاکریز عبور می کند. محدوده ای را جهت امتحان سطوح گسیختگی توسط نرم افزار انتخاب می کنیم. این محدوده که رسم شده در شکل زیر آورده شده است.



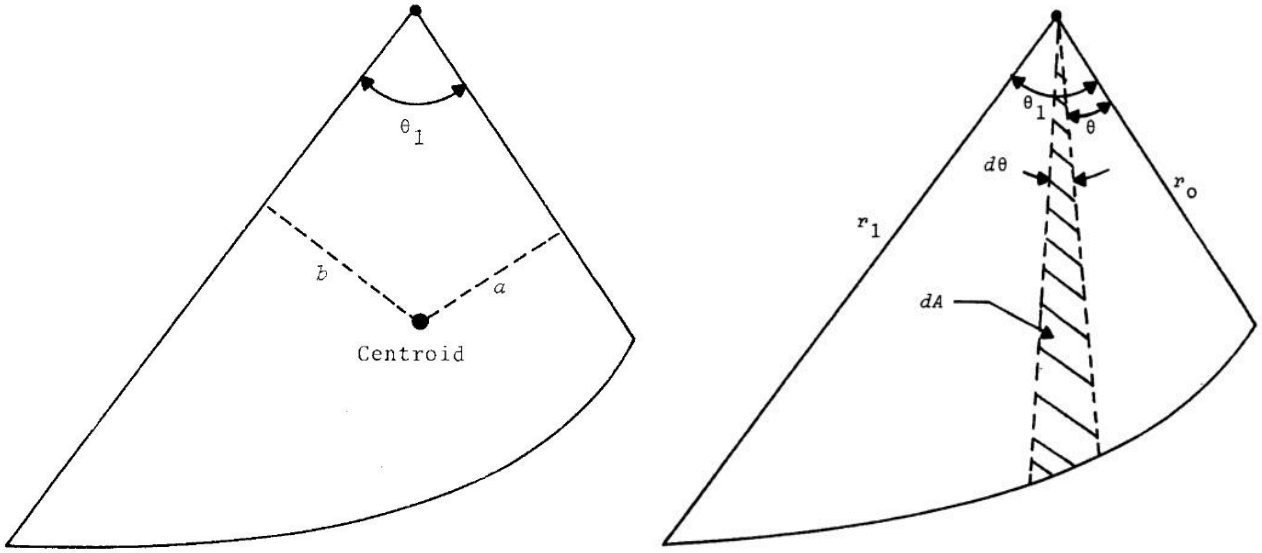
عکس ۸ - محل تقریبی مرکز مارپیچ لگاریتمی

نکته ی دیگر پیدا کردن مرکز جرم مارپیچ لگاریتمی جهت یافتن بازوی لنگر محرک می باشد. طبق رابطه ای که در کتاب Theoretical Foundation Engineering (Das, Theoretical Foundation Engineering, 2007) وجود دارد، مرکز جرم منحنی مارپیچ لگاریتمی برابر خواهد بود با :

$$a = \left[\frac{4r_0 \tan(\theta_1)}{3(9\tan^2(\varphi)+1)}\right]\left[\frac{\left(\frac{r_1}{r_0}\right)^3 (3\tan(\varphi)\sin(\theta_1) - \cos(\theta_1)) + 1}{\left(\frac{r_1}{r_0}\right)^2 + 1}\right]$$

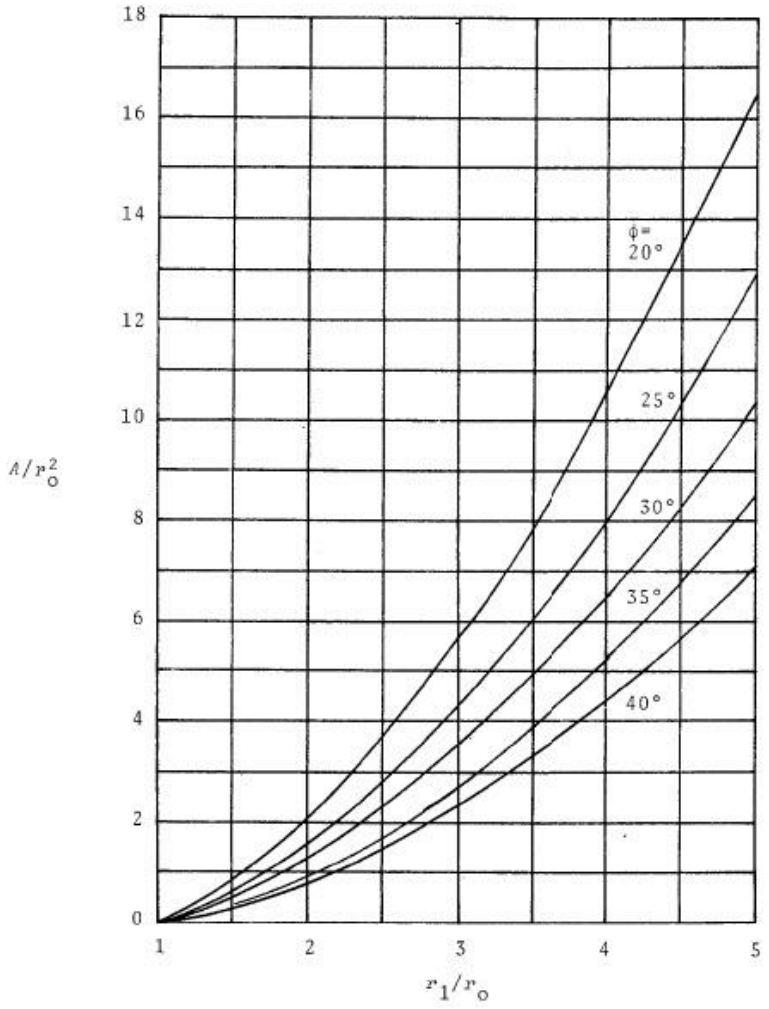
$$b = \left[\frac{4r_0 \tan(\varphi)}{3(9\tan^2(\varphi)+1)}\right]\left[\frac{\left(\frac{r_1}{r_0}\right)^2 - 3\tan(\varphi)\sin(\theta_1) - \cos(\theta_1)}{\left(\frac{r_1}{r_0}\right)^2 - 1}\right]$$

شکل صفحه بعد پارامتر های نامبرده در روابط بالا نشان می دهد. اما محاسبه ی این مقدار و سپس با استفاده کردن از قانون محاسبه مرکز جرم ها و بدست آوردن مرکز جرم توده ی خاک تشکیل شده توسط سطح گسیختگی، پیچیده است. پس از چند بار محاسبه مقدار بازوی لنگر را به صورت تقریبی تخمین زدیم. که مقدار آن برابر نصف فاصله ی افقی مرکز مارپیچ تا لبه ی تاج منهای پنج می باشد.



عکس ۱۰- پارامتر های استفاده شده برای محاسبه ی مرکز سطح مارپیچ لگاریتمی

در کتاب ذکر شده در بالا نموداری جهت محاسبه ی سطح مارپیچ لگاریتمی آورده شده است که ما در اینجا آن را نشان داده شده است. حال می توانیم با استفاده از فرض های ایجاد شده به سراغ نوشتن برنامه و اجرای آن برویم.

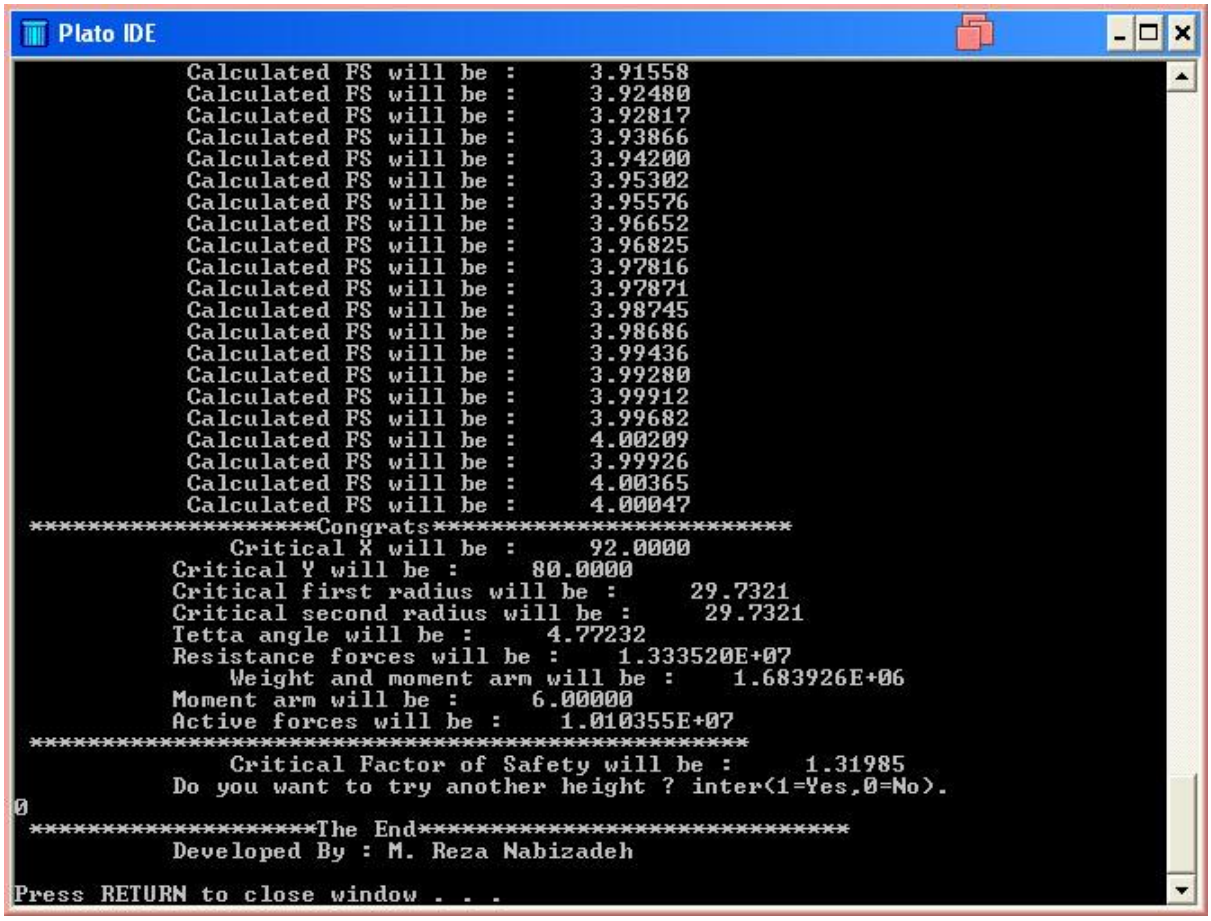


عکس ۹- نمودار ارائه شده در کتابTheoretical Foundation Engineering نوشته Braja M. Das جهت محاسبه سطح منحنی

کد برنامه ی نوشته شده به شرح زیر است :

```
program logmethod
implicit none

real c,gams,fi ! Slope parameters
real
pi,fir,FS,r1,r2,tet,FS1,k1,k2,e,res,aria,act,w,tfid,t,a !Calculated
parameters
real
al1,al2,ex,ex1,ex3,ex4,ex5,FScal,xcrit,ycrit,FScrit,r1crit,r2crit
 !Calculated parameters
real tetcrit,rescrit,wcrit,actcrit,r1crit,r2crit,acrit !Calculated
parameters
integer x1,y1

        PRINT *, '                    ***In The Name Of
God***'
   PRINT *, '                    In this program we attempt
to calculate the'
   PRINT *, '                    factor of safety with log
Spiral  method'
        print *, '-----------------------------------------------------
------'
   print *, '          Please input the unit weight of slope
material :'
   read *, gams
   print *, '          Please input the cohesion factor and
friction angle :'
   read *, c,fi
17  print *, '                    Please input the assumed
FS :'
   read *, FS

   pi=asin(1.0)*2.0
   fir=(pi*fi)/180
   t=tan(fir)
   tfid=t/FS
   e=2.718
   print *, '          First FS will be :',FS
                    DO x1=70,100
                        DO y1=60,80
                        r1=((114-x1)**2+(y1-
60)**2)**0.5
      r2=((114-x1)**2+(y1-60)**2)**0.5
      al1=atan((y1)/(x1-10.0))
      al2=atan((y1-60.0)/(114.0-x1))
      tet=(2*pi)-(al1+al2)
      k1=(r1**2)/(2*tfid)
      k2=e**(2*tet*tfid)
      res=c*k1*(k2-1)
                ex4=104*(y1-60)
      ex5=30*104
      ex1=(y1/2)*(x1-10)
      ex3=((114-x1)*(y1-60))/2
      ex=(ex4+ex5)-(ex1+ex3)
      a=((114-x1)/2)-5
      aria=0.5*k1*(k2-1)
      w=(aria-ex)*gams
      act=w*a
                FScal=(res/act)
                IF (FScal<=1) then
      print *, '                Calculated FS will be :
ERROR !'
      else
      print *, '                Calculated FS will be :',FScal
      END IF
      IF (FScal<=FS) then
      FScrit=FScal
      xcrit=x1
      ycrit=y1
      r1crit=r1
      r2crit=r2
      tetcrit=tet
      rescrit=res
      wcrit=w
      acrit=a
      actcrit=act
      END IF
                        END DO
      END DO
          print *,
'******************Congrats**************************'
          print *, '          Critical X will be :',xcrit
   print *, '          Critical Y will be :',ycrit
   print *, '          Critical first radius will be :',r1crit
   print *, '          Critical second radius will be :',r2crit
   print *, '          Tetta angle will be :',tetcrit
   print *, '          Resistance forces will be :',rescrit
          print *, '          Weight and moment arm
will be :',wcrit
   print *, '          Moment arm will be :',acrit
   print *, '          Active forces will be :',actcrit
          print *,
'************************************************'
          print *, '          Critical Factor of Safety will
be :',FScrit

   print *, '          Do you want to try another height ?
inter(1=Yes,0=No).'
   read *, a
                IF (a==1) THEN
                        goto 17
                        END IF
   print *, '******************The
End*****************************'
   print *, '                    Developed By : M. Reza
Nabizadeh'
 end program logmethod
```

تصویری از نتایج محاسبه شده توسط کد نوشته شده در فرترن در این جا آورده شده است.



عکس ۱۱- نتایج محاسبه شده برا ضریب اطمینان توسط روش مارپیچ لگاریتمی

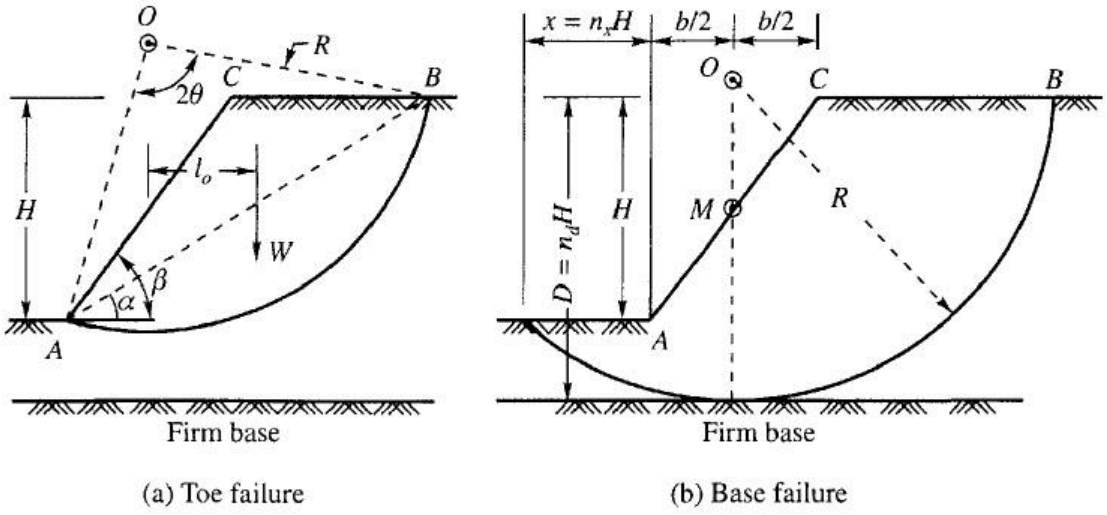
ضریب اطمینان محاسبه شده توسط نرم افزار با تقریب اولیه ۱,۳۵ برابر با ۱,۳۱ است. که به نسبت تقریب خوبیست.

۴. روش دایره ی سوئدی (Swedish Circle)

این روش برای خاک هایی که زاویه اصطکاک داخلی آن ها برابر با صفر باشد استفاده می شود. بدین ترتیب چون مصالح خاکریز در مسئله دارای زاویه اصطکاک داخلی ۳۸ درجه می باشند پس روش دایره سوئدی فقط تقریبی از ضریب اطمینان را محاسبه می کند.بر طبق نظر V.N.S. Murthy در کتاب :

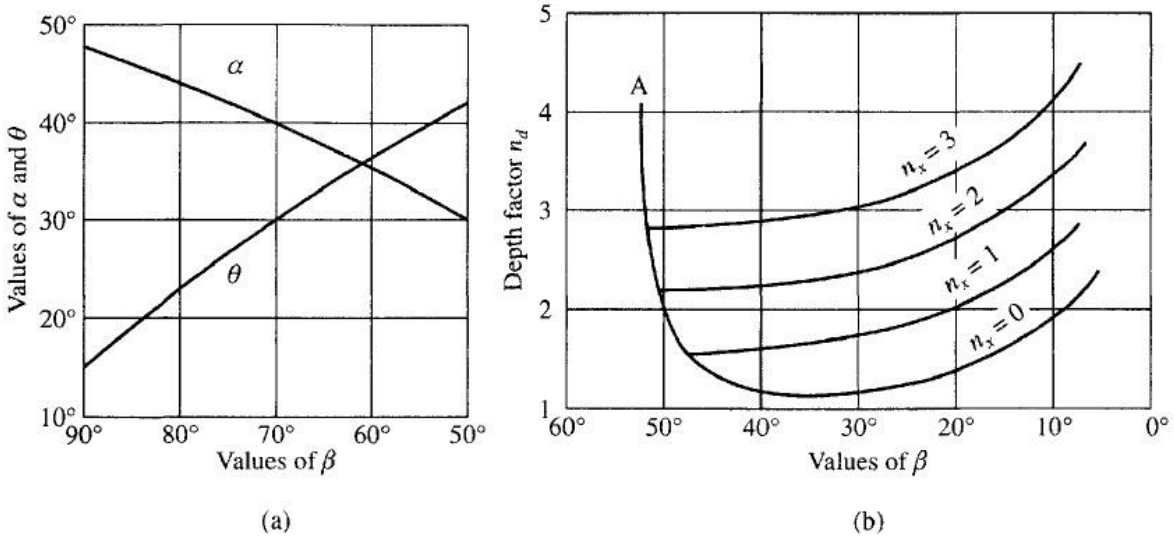
Geotechnical Engineering: Principles and Practices of Soil Mechanics and Foundation Engineering (Murthy, 2002)

در این حالت مسئله ما شاهد دو نوع گسیختگی و به طور کلی تشکیل دو نوع سطح گسیختگی هستیم دسته اول سطوحی اند که شکست پنجه دارند و دسته دوم گسیختگی های عمقی هستند که در اصطلاح به آن ها گسیختگی های قاشقی می گوئیم. این دو نوع سطح لغزش در شکل زیر نشان داده شده اند.



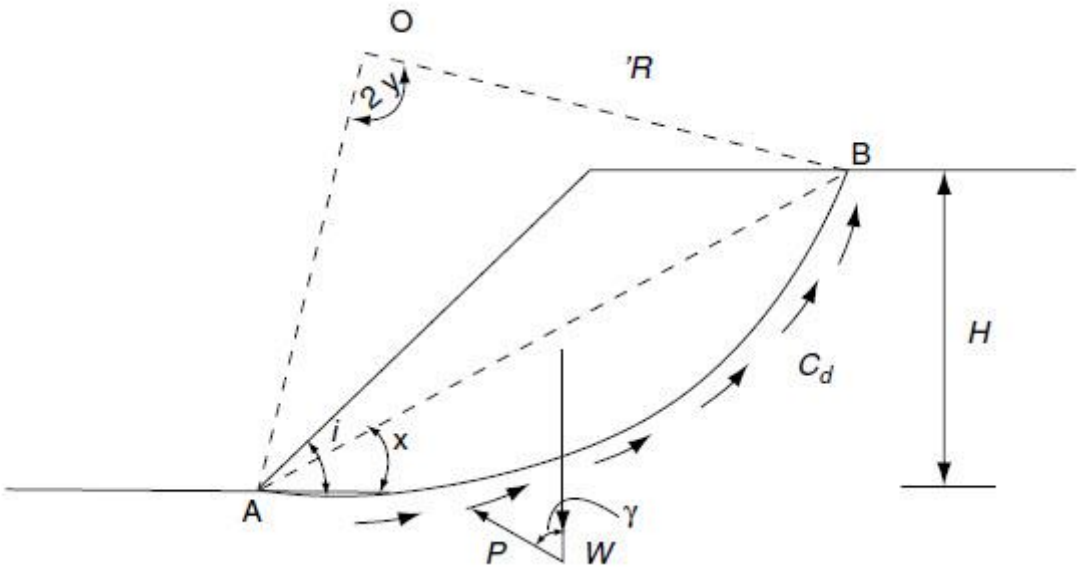
عکس ۱۲- سطوح لغزش پیشنهادی توسط V.N.S. Murthy برای روش دایره

او در ادامه با استفاده از نمودار هایی اقدام به محاسبات هندسی سطح لغزش می کند. این نمودار ها در زیر آورده شده اند.



عکس ۱۳- نمودار های پیشنهادی توسط V.N.S. Murthy برای روش دایره

نمودار های بالا طبق ادعای نویسنده توسط Fellenius در سال ۱۹۲۷ ارائه شده اند. Fellenius کسی است که در سال ۱۹۲۲ روش زاویه اصطکاک داخلی برابر صفر را معرفی می کند. اما استفاده از این نمودار ها در برنامه نویسی تقریبا غیر ممکن اند. برای استفاده از برنامه نویسی از فرمول های ارائه شده در کتاب The Foundation Engineering Handbook (Gunaratne, 2013)نوشته Manjriker Gunaratne استفاده می کنیم. بر طبق نظر این نویسنده می توان سطح لغزش بر طبق شکل زیر در نظر گرفت و آن را فرمول بندی کرد.



عکس ۱۴- سطح لغزش ارائه شده توسط Manjriker Gunaratne

اما Manjriker Gunaratne فرمول بندی زیر را ارائه می کند.

Toe failure :

$$\frac{c}{FS\gamma H} = \frac{\frac{1}{2}\csc^2(x)\,[y\csc^2(y) - \cot(y)] + \cot(x) - \cot(i)}{2[\cot(x)\cot(v) + 1]}$$

Base failure :

$$\frac{c}{FS\gamma H} = \frac{\frac{1}{2}\csc^2(x)\,[y\csc^2(y) - \cot(y)] + \cot(x) - \cot(i) - 2\eta}{2[\cot(x)\cot(v) + 1]}$$

که در آنها مقدار η برابر با نسبت فاصله ی پنجه ی شیب تا نقطه ی برخورد سطح لغزش به سطح افقی پنجه ی شیب بر ارتفاع خاکریز می باشد. و مقدار v در زیر نوشته شده است.

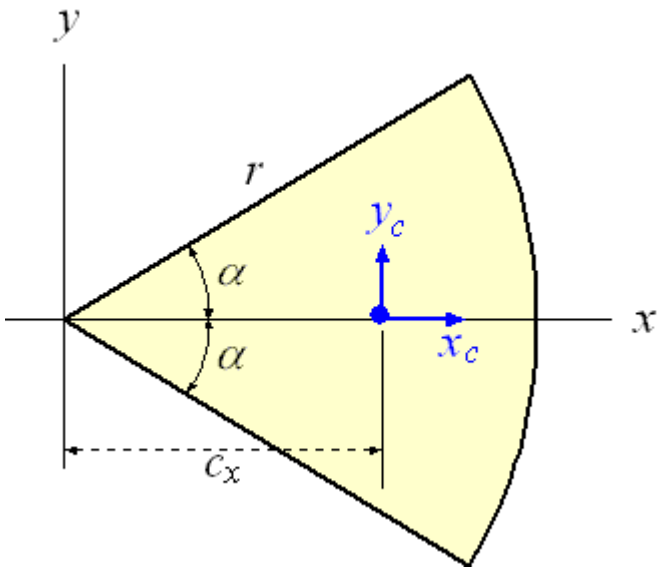
$$v = \frac{1}{2}(i + \varphi)$$

و روش دیگر استفاده از فرمول های اصلی دایره سوئدی است. راهکار اصلی جهت محاسبه ی مساحت و وزن توده ی خاک این است که مساحت قطاع دایره را حساب کنیم و مساحت باقی مانده را از آن کم کنیم. برای این روش استفاده از رابطه ی مساحت مختصاتی استفاده می کنیم که به صورت زیر است :

$$Aria = \frac{1}{2}\left|\left[(x_1 * y_2 + x_2 * y_3 + x_3 * y_4 + x_4 * y_5 + x_5 * y_1) - (x_2 * y_1 + x_3 * y_2 + x_4 * y_3 + x_5 * y_4 + x_1 * y_5)\right]\right|$$

مرکز جرم یک قطاع از دایره برابر است با :

$$x_c = \frac{4r\sin(\frac{\theta}{2})}{3\theta}$$



عکس ۱۵- مرکز جرم قطاع دایره

حال می توانیم سراغ نوشتن کد برنامه مورد نظر برویم.

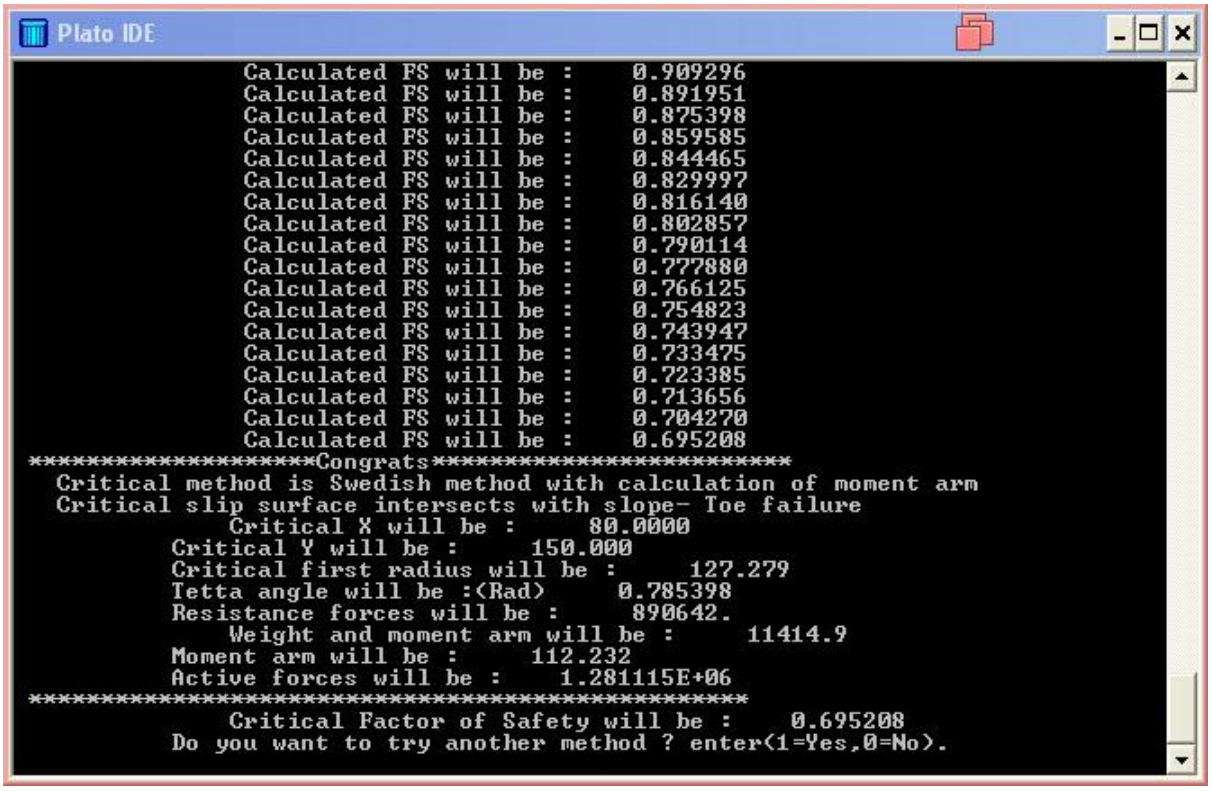
```
program swemethod
implicit none

real c,gams,fi ! Slope parameters
real adel,bdel,cdel,del,xc,yc !Calculated parameters
real r0,ra,ma,m1,m2,tet,area,ybar,len,res,w !Calculated parameters
real FScal,FSinp,FScrit,xcrit,ycrit,rcrit,tetcrit,rescrit,wcrit,acrit,actcrit !Calculated parameters
real totarea,ext,extbar,totareabar,i,h,noo,k1,x,k2,eta,a,act,mx,y!Calculated parameters
integer x1,y1,x3,meth,typ,anm

        PRINT *, '                    ***In The Name Of God***'
  PRINT *, '                    In this program we attempt to calculate the'
  PRINT *, '                    factor of safety with Swedish circle method' !Programs information
        print *, '-------------------------------------------------------'
  print *, '          Please input the unit weight of slope material :'
  read *, gams
  print *, '          Please input the cohesion factor and friction angle :'
  read *, c,fi
18 print *, '                    Please input the first assumption of FS :'
  read *, FSinp
25 print *, '                    Please select one of the methods :'
  print *, '          1) Swedish method with calculation of moment arm'
  print *, '          2) Swedish method with method of Foundation ENG handbook'
  read *, ma

    if (ma==1) then

                    DO x1=10,80
                    DO y1=80,150
     DO x3=114,170
                    r0=((x3-x1)**2+(y1-60)**2)**0.5
     ra=((x1-10)**2+(y1)**2)**0.5


        if(r0<=ra) then
        adel=1+((60/104)**2)
        bdel=0-(2*x1)-((2*600*60)/(104*104))-((2*y1*60)/104)
        cdel=(x1**2)+(y1**2)-(r0**2)+((600/104)**2)+((2*y1*600)/104)
        del=((bdel)**2)-4*cdel*adel
        xc=(-bdel-(del**0.5))/2*adel
        yc=y1-(((r0**2)-((xc-x1)**2))**0.5)
        m1=(60.0-y1)/(x3-x1)
        if (xc==x1) then
         tet=asin(1.0)-atan(abs(m1))
         else
        m2=(yc-y1)/(xc-x1)
        end if
        tet=atan(abs((m1-m2)/(1-m1*m2)))
        area=0.5*r0*r0*(tet-sin(tet))
        w=area*gams
        ybar=(4*r0*((sin(tet/2))**3))/(3*(tet-sin(tet)))
        a=ybar*cos(tet/2)
        len=r0*tet
        res=c*len*r0
```

```
            act=w*a
            FScal=res/act
                        print *, '                  Calculated FS will
be :',FScal


            IF (FScal<=FSinp) then
              meth=1
              typ=1
                                    FScrit=FScal
                                    xcrit=x1
                                    ycrit=y1
                                              rcrit=r0
                                    tetcrit=tet
                                    rescrit=res
                                    wcrit=w
                                    acrit=a
                                    actcrit=act
                                    END IF
            end if

            if(r0>=ra) then
            yc=0
            xc=x1-(r0**2-(yc-y1)**2)**0.5
            m1=(60.0-y1)/(x3-x1)
            if (xc==x1) then
             tet=asin(1.0)-atan(abs(m1))
             else
            m2=(yc-y1)/(xc-x1)
            end if
            tet=atan(abs((m1-m2)/(1-m1*m2)))
            totarea=(r0*r0*tet)/2
            ext=0.5*((60*x1+60*x3+xc*y1)-
(y1*x3+60*114+60*10))
            area=totarea-ext
            w=area*gams
            extbar=(x1+114+x3+10+xc)/5
            totareabar=(4*r0*sin(tet/2))/(3*tet)
            ybar=(totarea*totareabar-ext*extbar)/(totarea-
ext)
            a=ybar*cos(tet/2)
            len=r0*tet
            res=c*len*r0
            act=w*a
            FScal=res/act


                        print *, '                  Calculated FS will
be :',FScal
            IF (FScal<=FSinp) then
                                    meth=1
              typ=2
              FScrit=FScal
                                    xcrit=x1
                                    ycrit=y1
                                              rcrit=r0
                                    tetcrit=tet
                                    rescrit=res
                                    wcrit=w
                                    acrit=a
                                    actcrit=act
                                    END IF
            end if
             END DO
             END DO
             END DO
            end if
            if (ma==2) then

                                    DO x1=10,80
                                    DO y1=80,150
             DO x3=114,170
                                    r0=((x3-x1)**2+(y1-
60)**2)**0.5
             ra=((x1-10)**2+(y1)**2)**0.5


                  if(r0<=ra) then
                  i=0.66
                  h=60
                  noo=0.59
                              mx=60.0/(x3-10.0)
                  x=atan(abs((mx-0)/(1.0)))
                  adel=1+((60/104)**2)
                  bdel=0-(2*x1)-((2*600*60)/(104*104))-
((2*y1*60)/104)
                  cdel=(x1**2)+(y1**2)-
(r0**2)+((600/104)**2)+((2*y1*600)/104)
                  del=((bdel)**2)-4*cdel*adel
                  xc=(-bdel-(del**0.5))/2*adel
                  yc=y1-(((r0**2)-((xc-x1)**2))**0.5)
                  m1=(60.0-y1)/(x3-x1)
                   if (xc==x1) then
                    tet=asin(1.0)-atan(abs(m1))
                    else
                  m2=(yc-y1)/(xc-x1)
                  end if
                  tet=atan(abs((m1-m2)/(1-m1*m2)))
                  y=tet/2
                  k1=2*((1/tan(x))*(1.48)+1)
                  k2=0.5*((1/sin(x))**2)*(y*((1/sin(y))**2)-
(1/tan(y)))
                  FScal=(c*k1)/((gams*h)*(k2+(1/tan(x))-
(1/tan(i))))
                              print *, '                  Calculated FS will
be :',FScal
                  IF (FScal<=FSinp) then
                    meth=2
                    typ=1
                                    FScrit=FScal
                                    xcrit=x1
                                    ycrit=y1
                                              rcrit=r0
                                    tetcrit=tet
                                    END IF
                  end if

                  if(r0>=ra) then
                  i=0.66
                  h=60
                  noo=0.59
                              mx=60.0/(x3-10.0)
                  x=atan(abs(mx))
                  yc=0
                  xc=x1-(((r0**2)-((yc-y1)**2))**0.5)
```

```
m1=(60.0-y1)/(x3-x1)
if (xc==x1) then
 tet=asin(1.0)-atan(abs(m1))
 else
m2=(yc-y1)/(xc-x1)
end if
tet=atan(abs((m1-m2)/(1-m1*m2)))
y=tet/2
eta=abs(xc/60.0)
k1=2*((1/tan(x))*(1.48)+1)

k2=gams*60.0*(0.5*((1/sin(x))**2)*(y*((1/sin(y))**2)-
(1/tan(y))))
FScal=(c*k1)/((gams*h)*(k2+(1/tan(x))-
(1/tan(i))-2*eta))
print *, '                Calculated FS will
be :',FScal
IF (FScal<=FSinp) then
  meth=2
  typ=2
                FScrit=FScal
                xcrit=x1
                ycrit=y1
                        rcrit=r0
                tetcrit=tet
                END IF
end if
END DO
END DO
END DO
end if

print                                    *,
'*******************Congrats************************'
IF (meth==1) then
 print *, ' Critical method is Swedish method with
calculation of moment arm'
                END IF
IF (meth==2) then
 print *, ' Critical method is Swedish method with method of
Foundation ENG handbook'
                END IF
IF (typ==1) then
 print *, ' Critical slip surface intersects with slope- Toe
failure'
                END IF
IF (typ==2) then
 print *, ' Critical slip surface intersects with horizontal line-
Base failure'
                END IF
print *, '              Critical X will be :',xcrit
print *, '          Critical Y will be :',ycrit
print *, '          Critical first radius will be :',rcrit
print *, '          Tetta angle will be :(Rad) ',tetcrit
IF (meth==1) then
print *, '          Resistance forces will be :',rescrit
print *, '              Weight and moment arm will
be :',wcrit
print *, '          Moment arm will be :',acrit
print *, '          Active forces will be :',actcrit
End if
print                                    *,
'*************************************************'
print *, '              Critical Factor of Safety will
be :',FScrit
print *, '          Do you want to try another method ?
enter(1=Yes,0=No).'
read *, anm
                IF (anm==1) THEN
                        goto 25
                        END IF
print           *,          '*******************The
End**********************'
print *, '          Developed By : M. Reza Nabizadeh'
end program swemethod
```

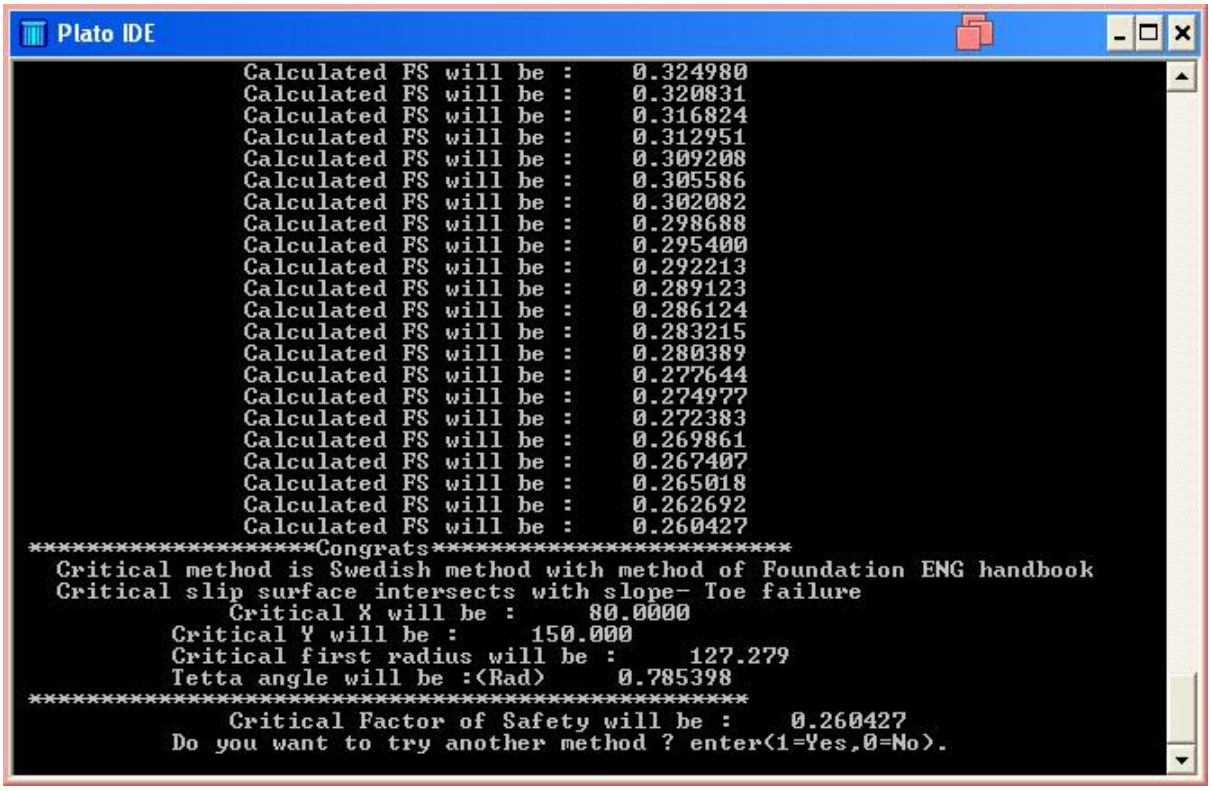
ضرایب اطمینان محاسبه توسط روش های بالا مقادیر نسبتا کمی دارند که دلیل این کم بودن در نظر نگرفتن اثر زاویه اصطکاک می باشد.

پس از اجرا کد می توان نتایج را مشاهده نمود. نتیجه ضرایب اطمینان به دو روش سوئدی به همراه بازوی لنگر و روش Foundation Engineering Handbook در زیر آورده شده است.



عکس ۱۶- ضریب اطمینان بحرانی برای روش بازوی لنگر

ضریب اطمینان پایین بدلیل صرف نظر از اثر اصطکاک می باشد.

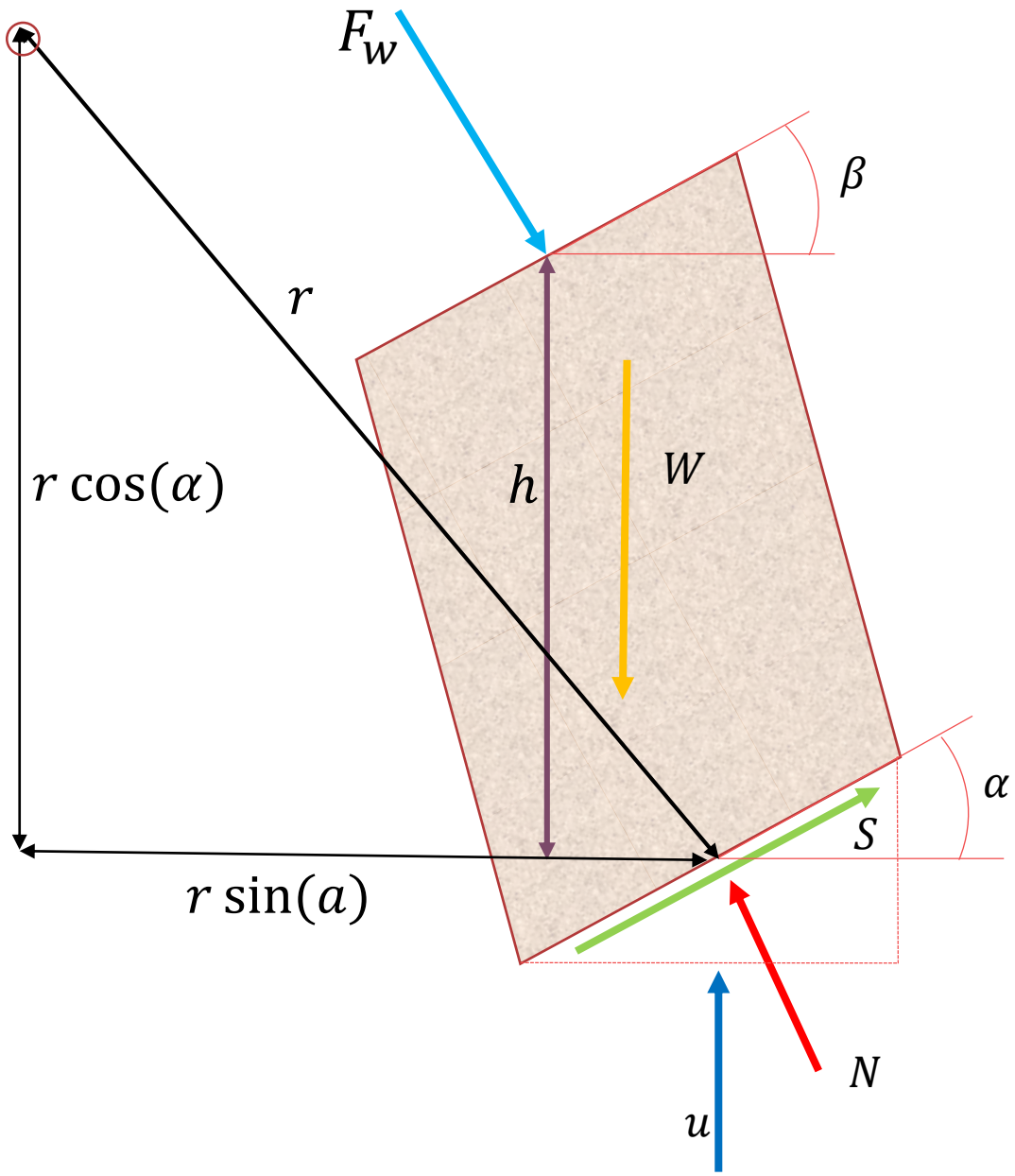


عکس ۱۷- ضریب اطمینان بحرانی توسط روش Foundation Engineering Handbook

۵. روش قطعات متداول (Ordinary method of Slices)

در این روش می بایست سطوح گسیختگی را اختیار نمود و با استفاده از قطعه بندی اقدام به محاسبه ی ضریب اطمینان نمود

در ابتدا می توان جهت تشریح پاسخ دیاگرام نیرو های وارد بر یک قطعه را رسم کرد.



عکس ۱۸- دیاگرام آزاد نیرو های وارده بر یک قطعه در روش قطعات متداول

در این مسئله ی بخصوص به علت عدم حضور آب در بالای سطح لغزش نیروی F_w برابر با صفر است.

با نوشتن معادله تعادل در راستای N داریم :

$$N = W\cos(\alpha) - u\Delta l\cos^2(\alpha)$$

با لنگرگیری نیرو های موجود حول نقطه ی شعاع خواهیم داشت :

$$\sum M_o = 0 \Rightarrow \tau * r - Wr\sin(\alpha) = 0$$

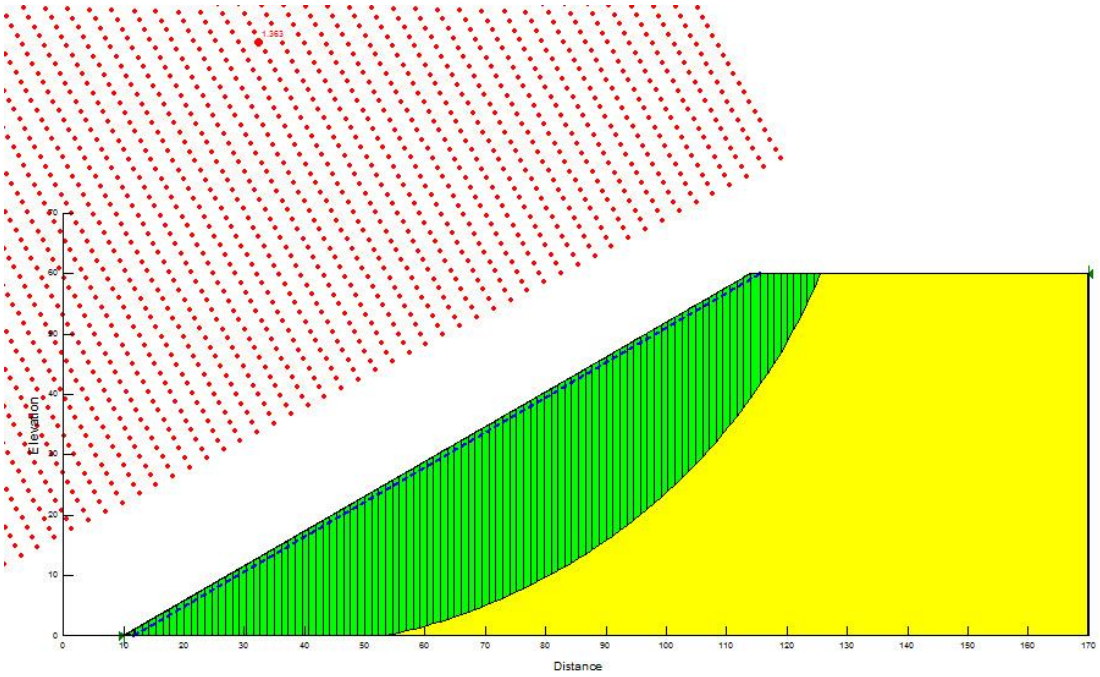
$$\tau * r = Wr\sin(\alpha) \Rightarrow \frac{S}{F} * r = Wr\sin(\alpha)$$

$$\frac{(c + N\tan(\varphi))}{F} * r = Wr\sin(\alpha)$$

در نهایت رابطه ی ضریب اطمینان برابر خواهد بود با :

$$F.S = \frac{\sum[(W\cos(\alpha) - u\Delta l\cos^2(\alpha))\tan(\varphi) + c\Delta l]r}{\sum[Wr\sin(\alpha)]}$$

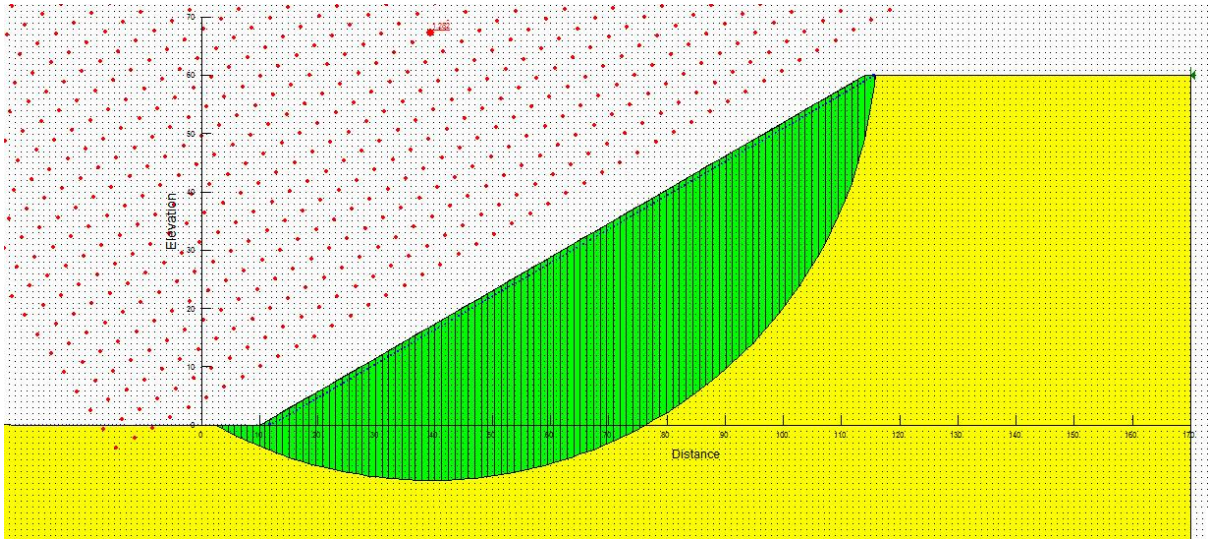
حال می توانیم به راحتی سراغ ایجاد سطوح لغزش های مورد نظر و محاسبات ضرائب اطمینان برویم. با استفاده از نرم افزار GeoSlope و استفاده از روش قطعات متداول جهت تحلیل سطح لغزش محل تقریبی سطح لغزش بحرانی را پیدا می کنیم. بدین ترتیب می توانیم در قسمت برنامه نویسی محدوده ای مناسب جهت جست و جو برای مرکز دایره ی سطح گسیختگی انتخاب کرد. ضریب اطمینان بحرانی بر اساس روش قطعات متداول ۱,۳۶ بدست آمد و دایره ی بحرانی در شکل زیر آورده شده است.



عکس ۱۹- دایره ی بحرانی بر اساس روش قطعات متداول در نرم افزار GeoSlope

البته باید توجه نمود که عدد ضریب اطمینان ۱,۳۲ با استفاده از تحلیل اجزا محدود تراوش و تعیین سطح آب بدست آمده است. بدین ترتیب باید این مدل را در عمق نیز گسترش دهیم تا بتوانیم گسیختگی های عمقی را نیز مدل سازی کنیم و ضریب اطمینان آن سطوح گسیختگی را مورد مطالعه قرار دهیم. با افزایش عمق خاکریز و ادامه دادن سطح تراوش آب در راستای افق در پنجه ی خاکریز مقدار ضریب اطمینان ۱,۲۸ بدست آمد. شکل صفحه ی بعد گویای این واقعیت است که می توان سطح لغزش هایی غیر دایره ای را با روش قطعات متداول یا بیشاپ حل کرد اما نباید انتظار داشت که محتمل ترین سطح لغزش سطوحی باشد که نرم افزار نشان می دهد. و مشاهده می شود که این ضرایب اطمینان ارائه شده توسط GeoSlope بسته به مکان اختیار شده ی مرکز و مماس ها می تواند متفاوت باشد، در این حین است که نقش قضاوت مهندسی و غلبه ی تصمیمات انسانی بر محاسبات کامپیوتری خود را بیش از پیش پر رنگ تر نشان می دهد.

شکل زیر نشان دهنده سطح لغزش با ضریب اطمینان ۱,۲۸ می باشد.



عکس ۲۰- هندسه ی تعمیم داده شده ی خاکریز و محاسبه ی ضریب اطمینان ۱,۲۸ توسط نرم افزار

حال می توانیم به کد نویسی در نرم افزار Fortran جهت پیدا کردن مقدار ضریب اطمینان بپردازیم.

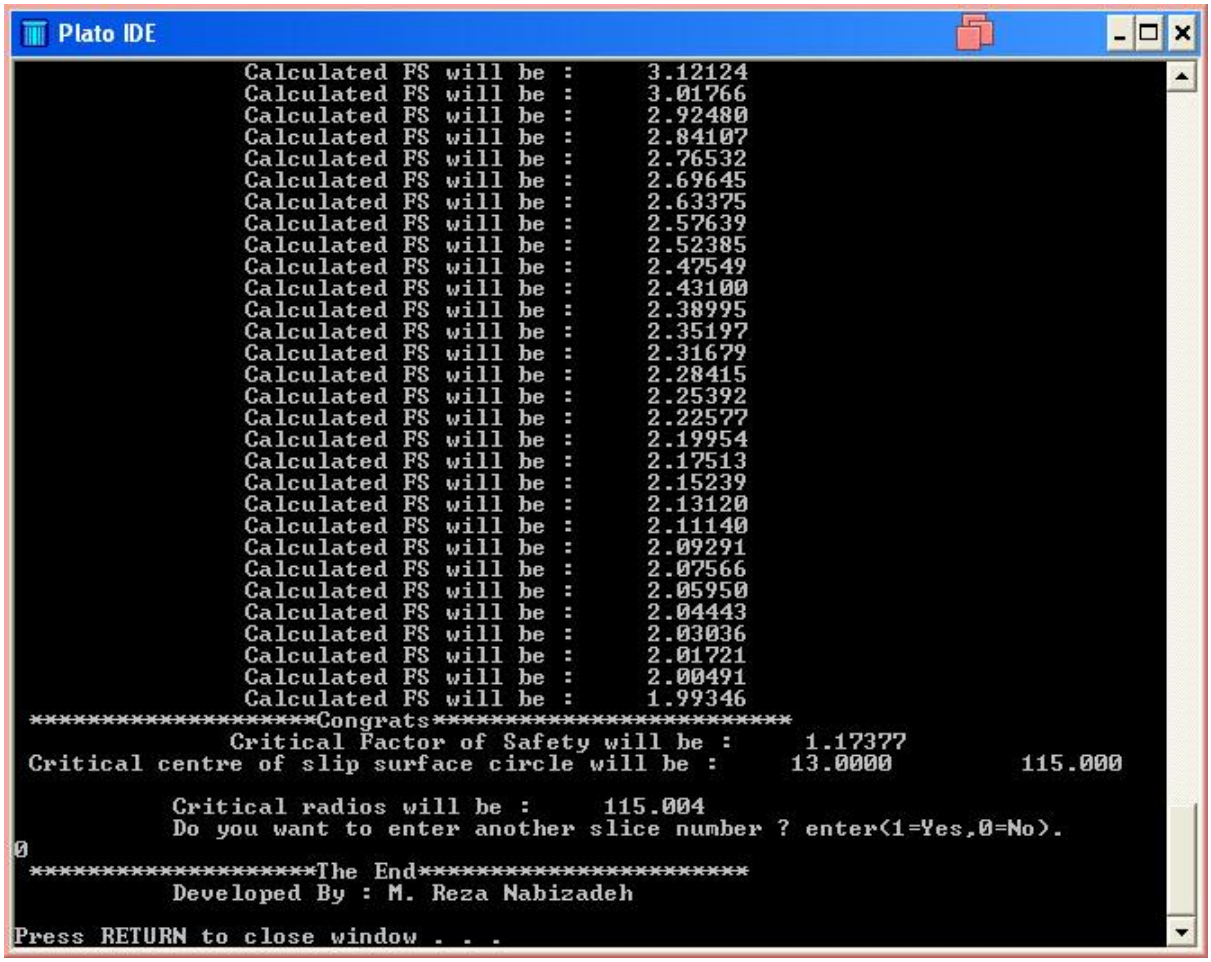
شرح کد نوشته شده در زیر آورده شده است :

```
program ordmethod
implicit none

real c,gams,fi,gamw ! Slope parameters
real pi,fir,r0,ra,adel,bdel,cdel,del,xc,yc,l,b,bb !Calculated
parameters
real
k1,k2,k3,xceni,ytopi,yboti,hsoili,wi,xlefti,xrighti !Calculated
parameters
real
ybotlefti,ybotrighti,delli,mi,alfai,ywatti,hwatti,ui,k1i,k2i,k3i !
Calculated parameters
real FSmin,Fsin,Fscal,xcrit,ycrit,rcrit !Calculated parameters
integer x1,y1,x3,asf,ns,i,snt

        PRINT *, '                    ***In The Name Of
God***'
  PRINT *, '                In this program we attempt
to calculate the'
  PRINT *, '                factor of safety with
Swedish circle method' !Programs information
        print *, '-----------------------------------------------------
------'
  print *, '            Please input the unit weight of slope
material and water :'
  read *, gams,gamw
  print *, '            Please input the cohesion factor and
friction angle :'
  read *, c,fi
19  print *, '            Please input number of the
slices  :'
  read *, ns
21  print *, '            Please input first
assumption of FS  :'
  read *, Fsin
23        pi=(asin(1.0))*2
  fir=(fi*pi)/180
  Fsmin=0
                        DO x1=10,80
                        DO y1=80,150
    DO x3=114,170
                        r0=((x3-x1)**2+(y1-
60)**2)**0.5
    ra=((x1-10)**2+(y1)**2)**0.5

                        IF(r0<=ra) then
        adel=1+((60/104)**2)
        bdel=0-(2*x1)-((2*600*60)/(104*104))-
((2*y1*60)/104)
        cdel=(x1**2)+(y1**2)-
(r0**2)+((600/104)**2)+((2*y1*600)/104)
        del=((bdel)**2)-4*cdel*adel
        xc=(-bdel-(del**0.5))/2*adel
        yc=y1-(((r0**2)-((xc-x1)**2))**0.5)
        l=x3-xc
        b=l/ns
        bb=b/2

    DO i=1,ns
                k1=0.0
        k2=0.0
        k3=0.0
                xceni=xc+(bb*i)
        ytopi=((60*xceni)/104)-600.0/104.0
        yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5)
        hsoili=ytopi-yboti
        wi=b*hsoili*gams
        xlefti=xc+bb*(i-1)
        xrighti=xc+bb*(i+1)
        ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5)
```

```
            ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5)
            delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
            mi=(ybotrighti-ybotlefti)/(xrighti-xlefti)
            alfai=tan(abs(mi))
                                    ywatti=((60*xceni)/104)-
720.0/104.0
            hwatti=ywatti-yboti
            ui=hwatti*gamw
            k1i=(wi*cos(alfai)-
ui*delli*((cos(alfai))**2))*tan(fir)
            k2i=c*delli
            k3i=wi*sin(alfai)
            If (mi<=0) then
              k3i=(0.0-1.0)*wi*sin(alfai)
            End If
            k1=k1+k1i
            k2=k2+k2i
            k3=k3+k3i
            IF (k3==0.0) THEN
            print *, '                    This Slip surface is Stable.'
              else
              Fscal=(k1+k2)/k3
              end if

        End DO
                        print *, '                    Calculated FS will
be :',Fscal
                                    End if

        IF(r0>ra) then
            yc=0
            xc=x1-(r0**2-(yc-y1)**2)**0.5
            l=x3-xc
            b=l/ns
            bb=b/2

        DO i=1,ns
                        k1=0.0
            k2=0.0
            k3=0.0
                        xceni=xc+(bb*i)
            if (xceni<=10) then
              ytopi=0
              else
            ytopi=((60*xceni)/104)-600.0/104.0
              end if
            yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5)
            hsoili=ytopi-yboti
            wi=b*hsoili*gams
            xlefti=xc+bb*(i-1)
            xrighti=xc+bb*(i+1)
            ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5)
            ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5)
            delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
            mi=(ybotrighti-ybotlefti)/(xrighti-xlefti)
            alfai=tan(abs(mi))
            if (xceni<=0) then
              ywatti=0
              else
            ywatti=((60*xceni)/104)-720.0/104.0
              end if
            hwatti=ywatti-yboti
            ui=hwatti*gamw
            k1i=(wi*cos(alfai)-
ui*delli*((cos(alfai))**2))*tan(fir)
            k2i=c*delli
            k3i=wi*sin(alfai)
            If (mi<=0) then
              k3i=(0.0-1.0)*wi*sin(alfai)
            End If
            k1=k1+k1i
            k2=k2+k2i
            k3=k3+k3i
            IF (k3==0.0) THEN
            print *, '                    This Slip surface is Stable.'
            else
            Fscal=(k1+k2)/k3
              end if
        End DO
                                                                print *, '
                        Calculated FS will be :',Fscal
                                    End IF
                                    IF (Fscal>=0 .and.
Fscal<Fsin) then
            FSmin=Fscal
              Fsin=Fscal
              xcrit=x1
              ycrit=y1
              rcrit=r0
            End If
        END DO
        END DO
        END DO
            if (Fsmin==0.0) then
  print *, '                You entered a low first assumption of
FS'
  print *, '                Do you want to enter another
assumption of FS ? enter(1=Yes,0=No).'
  print *, '                Enter 1 for manually input a new
assumption'
  print *, '                Enter 0 for let the program
automatically input a new assumption'
  read *, asf
                                                                IF
(asf==1) THEN

            goto 21
                                    end if
                                    if (asf==0) then
                                    fsin=fsin+0.1
                                    goto 23

            END IF
                                                end if
            if (Fsmin/=0.0)  then
            print *,
'********************Congrats************************'
            print *, '                Critical Factor of Safety will
be :',FSmin
```

```
            print *, 'Critical centre of slip surface circle will
be :',xcrit,ycrit
   print *, '              Critical radios will be :',rcrit
   print *, '              Do you want to enter another slice
number ? enter(1=Yes,0=No).'
   read *, snt
                        IF (snt==1) THEN
                                goto 19
                                END IF
                        end if
   print *, '**********The
End**********'
   print *, '              Developed By : M. Reza Nabizadeh'
end program ordmethod
```

در تصویر زیر مقدار ضریب اطمینان محاسبه شده مشاهده می شود که اختلاف این ضریب اطمینان و ضریب اطمینان ارائه شده توسط GeoSlope می تواند چگونگی محاسبات فشار آب حفره ای باشد.

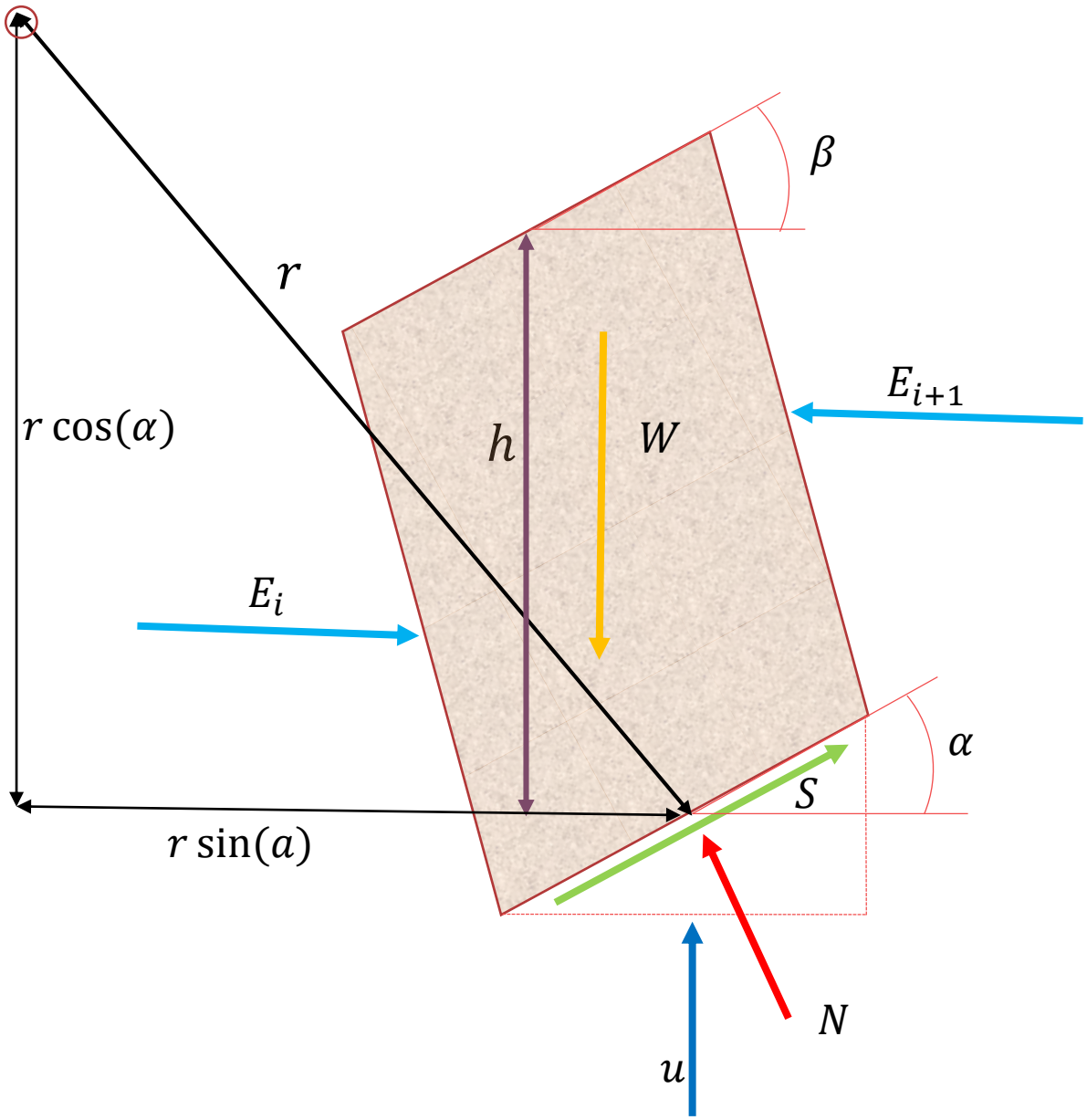


عکس ۲۱- ضریب اطمینان بحرانی بدست آمده از روش کد نویسی با Fortran

اگر در روش قطعات متداول مقدار زاویه اصطکاک داخلی خاک را برابر با صفر در نظر بگیریم ضریب اطمینان بدست آمده با ضریب اطمینان حاصل از روش سوئدی برابر است. با در نظر گرفتن شرایط زاویه اصطکاک داخلی برابر صفر مقدار ضریب اطمینان روش قطعات متداول برابر ۰،۲۹ خواهد بود که با جواب حاصل از فرمول کتاب The Foundation Engineering Handbook (Gunaratne, 2013) هم خوانی بیشتری دارد که مقدار ارائه شده برابر ۰،۲۶ می باشد. دلیل این همخوانی تقریبی بودن مقدار بازوی لنگر محرک در روش سوئدی با محاسبه بازوی لنگر است.

۶. روش بیشاپ ساده شده (Simplified Bishop Procedure)

در این روش نیرو های بین قطعه ای افقی فرض می شود و از نیرو های برشی نیز صرف نظر می شود. برای برآورده کردن تعادل معادلات آن در راستای عمودی نوشته می شود. دیگرام آزاد نیرو های وارده بر یک قطعه در زیر آورده شده است.

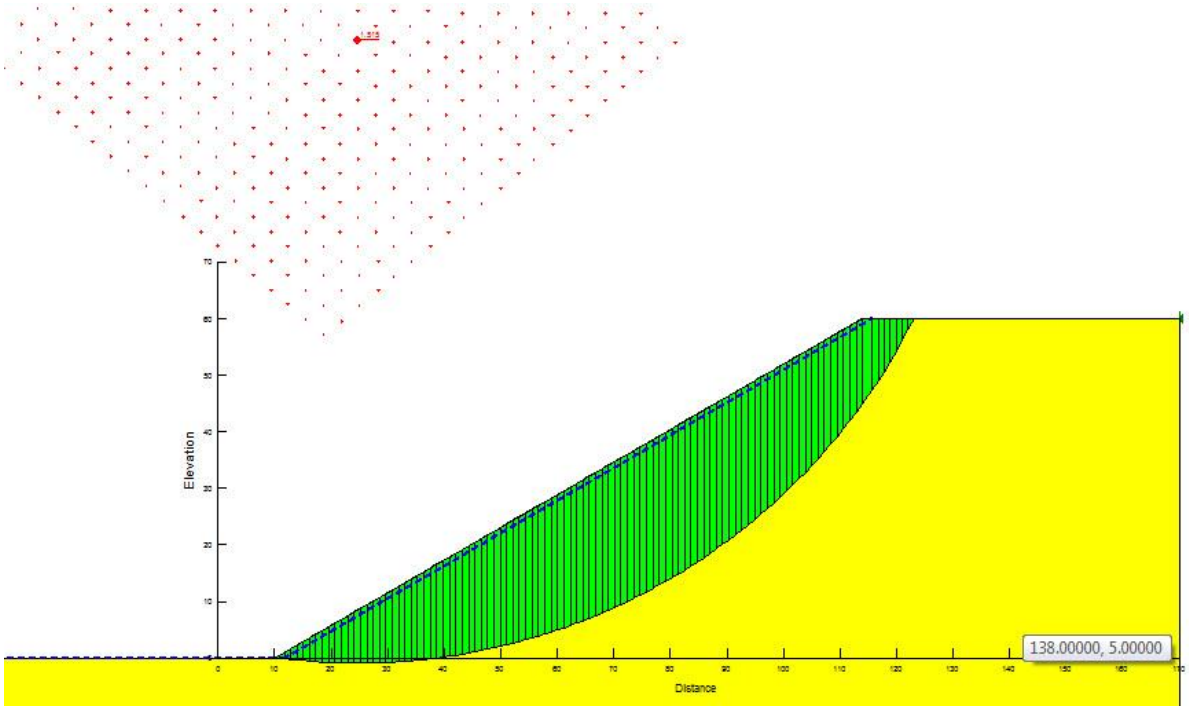


عکس ۲۲- دیاگرام آزاد نیرو های وارده بر یک قطعه در روش بیشاپ

فرمول نهایی این روش به شرح زیر است :

$$FS = \frac{\sum \left[\frac{c'\Delta l \cos(\alpha) + (W - u\Delta l \cos(\alpha)) \tan(\varphi')}{\cos(\alpha) + \frac{(\sin(\alpha)\tan(\varphi'))}{F}} \right]}{\sum W \sin(\alpha)}$$

می توان با استفاده از تقسیم بندی کردن به قطعات به حل این معادله پرداخت و ضریب اطمینان را بدست آورد. ضریب اطمینان بدست آمده توسط نرم افزار GeoSlope مقدار ۱,۵۱ است و مقدار ضریب اطمینان برای همان سطح لغزش بخصوص به روش جانبو مقدار ۱,۳۸ است که نشان دهنده منحصرا برآورده کردن معادلات لنگر در روش بیشاپ ساده شده است. که در شکل زیر سطح لغزش بحرانی قابل مشاهده است.



عکس ۲۳- سطح گسیختگی بحرانی برای روش بیشاپ ساده شده و جانبو

در زیر کد نوشته شده در نرم افزار Fortran آورده شده است.

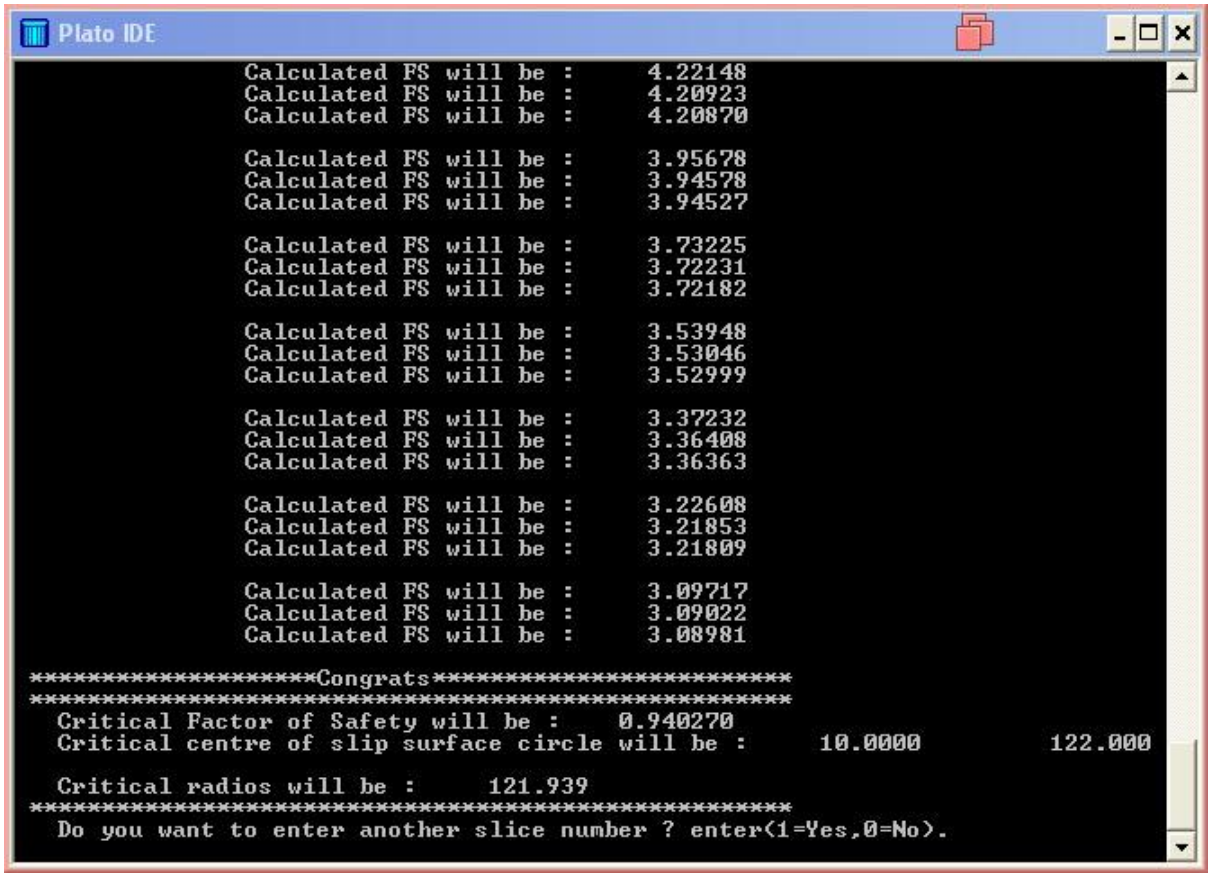
```
program bishopmethod
implicit none

real c,gams,fi,gamw ! Slope parameters
real
pi,fir,r0,ra,adel,bdel,cdel,del,xc,yc,l,b,bb,r0crit,y1crit,x1crit !C
alculated parameters
real
k5,k8,xceni,ytopi,yboti,hsoili,wi,xlefti,xrighti,fsfinal !Calculat
ed parameters
real
ybotlefti,ybotrighti,delli,mi,alfai,ywatti,hwatti,ui,k1i,k2i,k3i,k
4i,k5i !Calculated parameters
real k6i,k7i,k8i,FSmin,fsin,Fscal,fsreserv !Calculated
parameters
integer x1,y1,x3,ns,i,snt

        PRINT *, '                    ***In The Name Of
God***'
  PRINT *, '                    In this program we attempt
to calculate the'
  PRINT *, '                    factor of safety Simplified
Bishop method' !Programs information
        print *, '-----------------------------------------------------
------'
  print *, '          Please input the unit weight of slope
material and water :'
  read *, gams,gamw
  print *, '          Please input the cohesion factor and
friction angle :'
  read *, c,fi
19  print *, '                    Please input number of the
slices  :'
  read *, ns
  print *, '          Please input first assumption of FS  :'
  read *, Fsin
  fsreserv=fsin
        pi=(asin(1.0))*2
  fir=(fi*pi)/180
  Fsmin=0
                    DO x1=10,80
                    DO y1=80,150
    DO x3=114,140
                    r0=((x3-x1)**2+(y1-
60)**2)**0.5
    ra=((x1-10)**2+(y1)**2)**0.5

33                  IF(r0<=ra) then
        adel=1+((60/104)**2)
        bdel=0-(2*x1)-((2*600*60)/(104*104))-
((2*y1*60)/104)
        cdel=(x1**2)+(y1**2)-
(r0**2)+((600/104)**2)+(2*y1*600)/(104)
        del=((bdel)**2)-4*cdel*adel
        xc=(-bdel-(del**0.5))/2*adel
        yc=y1-(((r0**2)-((xc-x1)**2))**0.5)
        l=x3-xc
        b=l/ns
        bb=b/2
    DO i=1,ns
        k5=0.0
        k8=0.0
              xceni=xc+(bb*i)
        if (xceni>=114) then
        ytopi=60
          else
          ytopi=(xceni-10)*(60.0/104)
              end if
        yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5)
        hsoili=ytopi-yboti
        wi=b*hsoili*gams
        xlefti=xc+bb*(i-1)
```

```
            xrighti=xc+bb*(i+1)
            ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5)
            ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5)
            delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
            mi=(ybotrighti-ybotlefti)/(xrighti-xlefti)
            alfai=1.0*tan(1.0*abs(1.0*mi))
            if (xceni>=115.5) then
            ywatti=60
              else
              ywatti=(xceni-11.5)*(60.0/104)
                    end if
            hwatti=ywatti-yboti
            ui=hwatti*gamw
            k1i=c*delli*cos(alfai)
            k2i=(wi-ui*delli*cos(alfai))*tan(fir)
            k3i=cos(alfai)
            k4i=(abs(1.0*sin(alfai))*tan(fir))/(fsin)
            k5i=wi*sin(alfai)
            k6i=k1i+k2i
            k7i=k3i+k4i
            k8i=k6i/k7i
            If (mi<=0) then
              k5i=(0.0-1.0)*wi*sin(alfai)
            End If
            k5=k5+k5i
            k8=k8+k8i
            IF (k5==0) THEN
            print *, '                    This Slip surface is Stable.'
              else
              Fscal=k8/k5
              end if

        End DO
                    print *, '                    Calculated FS will
be :',Fscal
                                  End if

        IF(r0>ra) then
            yc=0
            xc=x1-(r0**2-(yc-y1)**2)**0.5
            l=x3-xc
            b=l/ns
            bb=b/2

        DO i=1,ns
                    k5=0.0
            k8=0.0
                    xceni=xc+(bb*i)
            if (xceni<=10) then
              ytopi=0
              else
                    if (xceni>=114) then
                    ytopi=60
                    else
                    ytopi=(xceni-10)*(60.0/104)
                    end if
              end if
            yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5)
            hsoili=ytopi-yboti
            wi=b*hsoili*gams
            xlefti=xc+bb*(i-1)
            xrighti=xc+bb*(i+1)
            ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5)
            ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5)
            delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
            mi=(ybotrighti-ybotlefti)/(xrighti-xlefti)
            alfai=1.0*tan(1.0*abs(1.0*mi))
            if (xceni<=10) then
              ywatti=0
              else
                    if (xceni>=115.5) then
                    ywatti=60
                    else
                    ywatti=(xceni-11.5)*(60.0/104)
                    end if
              end if
            hwatti=ywatti-yboti
            ui=hwatti*gamw
            k1i=c*delli*cos(alfai)
            k2i=(wi-ui*delli*cos(alfai))*tan(fir)
            k3i=cos(alfai)
            k4i=(abs(1.0*sin(alfai))*tan(fir))/(Fsin)
            k5i=wi*sin(alfai)
            k6i=k1i+k2i
            k7i=k3i+k4i
            k8i=k6i/k7i
            If (mi<=0) then
              k5i=(0.0-1.0)*wi*sin(alfai)
            End If
            k5=k5+k5i
            k8=k8+k8i
            IF (k5==0) THEN
            print *, '                    This Slip surface is Stable.'
              else
              Fscal=k8/k5
              end if

        End DO
                                                print *, '
                    Calculated FS will be :',Fscal
                                End IF


                    if (abs(1.0*(fscal-fsin))<=0.001) then

            Fsmin=Fscal
            go to 162
            else
            fsin=fscal
            go to 33
            end if
162                                         print *, ''
          IF (Fsmin>=0 .and. Fsmin<Fsreserv) then
            FSfinal=Fsmin
            Fsreserv=Fsmin
            x1crit=x1
            y1crit=y1
            r0crit=r0
            End If
        END DO
        END DO
        END DO
```

```
            print *,
'**************************Congrats************************'
            print *,
'**************************************************'
            print *, '  Critical Factor of Safety will be :',Fsfinal
            print *, '  Critical centre of slip surface circle will
be :',x1crit,y1crit
   print *, '  Critical radios will be :',r0crit
            print *,
'**************************************************'
```

```
   print *, '  Do you want to enter another slice number ?
enter(1=Yes,0=No).'
   read *, snt
                        IF (snt==1) THEN
                                goto 19
                                END IF

   print *, '**********************The
End**********************'
   print *, '                Developed By : M. Reza Nabizadeh'
 end program bishopmethod
```

که نتایج آن در زیر آورده شده است :

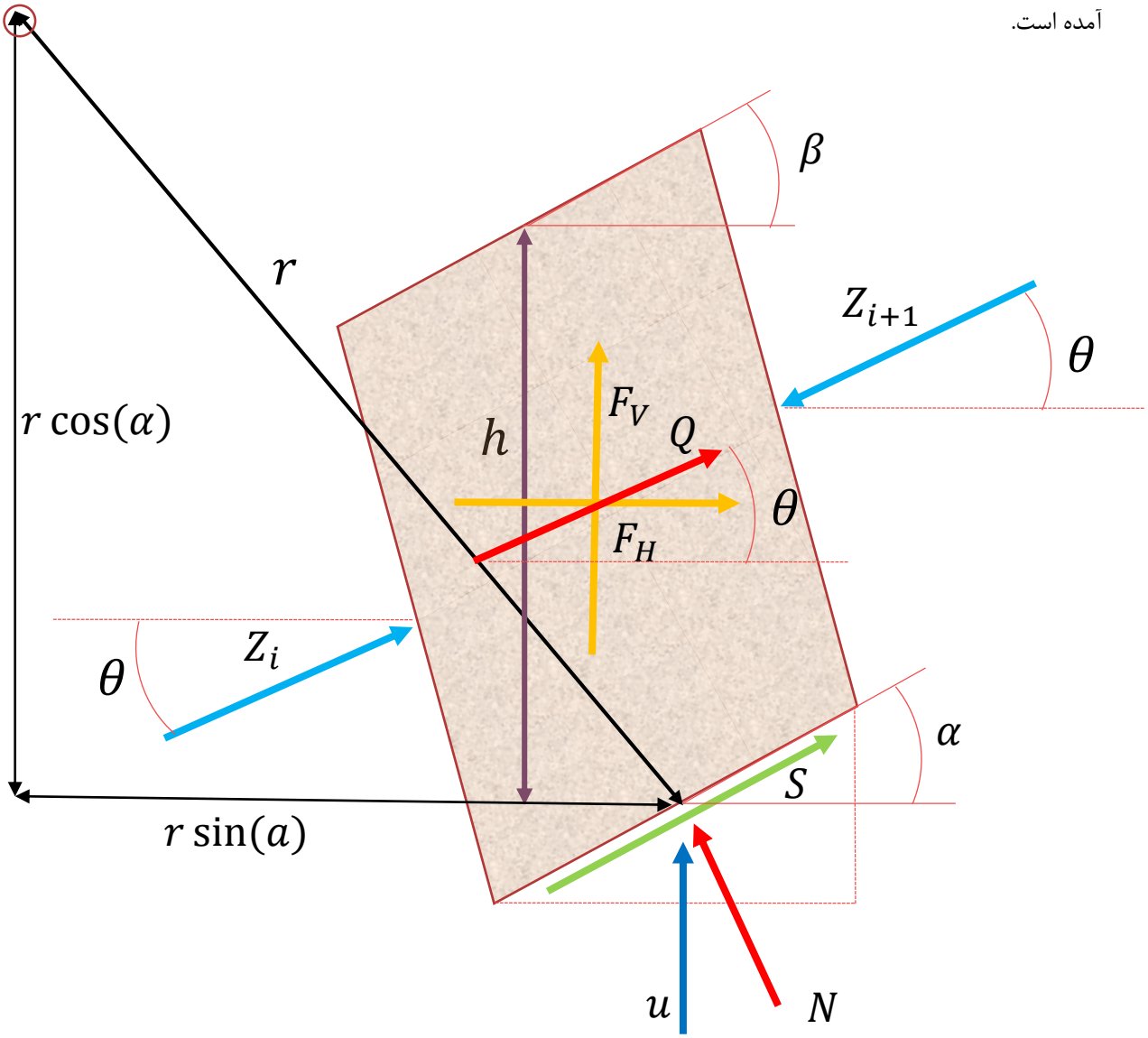


عکس ۲۴- نتایج روش بیشاپ با برنامه نویسی به زبان Fortran

اختلاف نتیجه ی اندک بالا و نرم افزار GeoSlope ریشه در اختلاف مقدار پارامتر های موثر و واقعی مقاومت برشی خاک دارد.

۷. روش اسپنسر (Spencer procedure)

این روش جز آندسته از روش هایی است که هم تعادل نیرو ها و هم تعادل لنگر ها را برآورده می کند. اساس این روش و معادلات ساده شده ی آن بر اصل موازی بودن نیرو های بین قطعه ای استوار است. بنابراین با ایجاد این فرض می توان اختلاف نیرو های بین قطعه ای طرفین یک قطعه را برآیند گیری عددی کرد. مقدار این برآیند از نوشتن معادلات تعادل نیرو های موجود در هر قطعه در راستای نیرو های عمود بر قطعه و راستای نیروی مقاومت برشی بدست می آید. و باید دو معادله ی مجموع نیرو های بین قطعه ای و گشتاور آن ها صفر شود. در شکل زیر دیاگرام نیرو های موجود در یک المان در روش اسپنسر آمده است.



عکس ۲۵- دیاگرام نیرو های وارده بر هر قطعه در روش اسپنسر

پس از برآیند گیری از نیرو های Z_{i+1} و Z_i و برابر قرار دادن مقدار آن با Q می توان معادلات تعادل در راستای N و S را برای این قطعه نوشت. در کتاب Soil Strength and Slope Stability نوشته ی J. Michael Duncan منظور از نیروی های افقی را نیرو های ناشی از زلزله و یا تسلیح خاک بیان کرده است. و جهت نیرو های عمودی که ناشی از وزن قطعه می باشد را رو به بالا فرض کرده است. با فرض نیروی وزن به سمت پایین و حذف نیرو های افقی معادلات تعادل به شرح زیر می باشد :

$$\sum F_N = N - W\cos(\alpha) - Q\sin(\alpha-\theta) = 0$$
$$\sum F_S = S - W\cos(\alpha) - Q\sin(\alpha-\theta) = 0$$

با در نظر گرفتن رابطه ی زیر که بین S و N نوشته می شود :

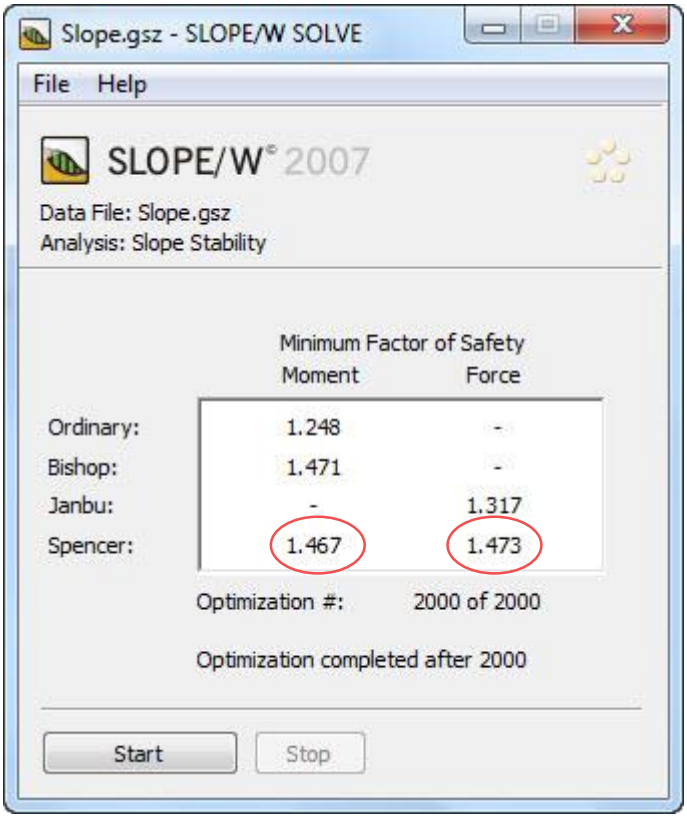
$$S = \frac{1}{F}[C'\Delta l + (N - u\Delta l)\tan(\varphi')]$$

که پس از ترکیب سه معادله ی بالا معادله ی نهایی جهت بدست آوردن مقدار Q به صورت زیر حاصل می شود :

$$Q = \frac{W\sin(\alpha) - \frac{C'\Delta l}{F} + \frac{\tan(\varphi')}{F}[u\Delta l - W\cos(\alpha)]}{\cos(\alpha - \theta) + \frac{\sin(\alpha - \theta)\tan(\varphi')}{F}}$$

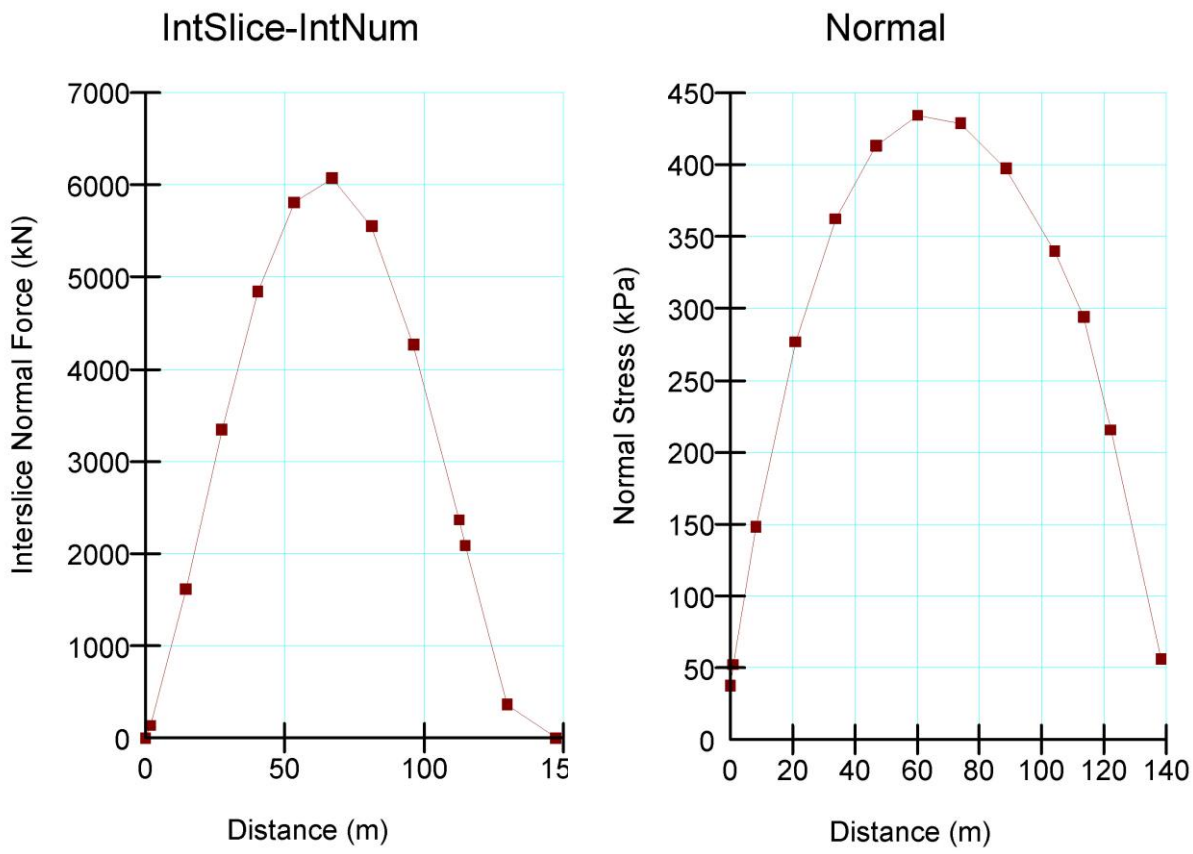
پروسه ی بدست آوردن ضریب اطمینان بوسیله ی این روش این گونه است که ابتدا مقداری برای زاویه نیرو های بین قطعه ای و ضریب اطمینان در نظر گرفته می شود. سپس مقدار Q تعیین شده و در رابطه های تعادل و لنگر نیرو های بین قطعه ای قرار داده می شود. به ازای یک زاویه و یک ضریب اطمینان مشخص هر دو معادله ی لنگر و نیرو های بین قطعه ای صفر می شوند و آن ضریب اطمینان همان ضریب اطمینان بحرانی می باشد.

نرم افزار GeoSlope نیز می تواند تحلیل یک خاکریز به روش اسپنسر را انجام دهد. با توجه به تصویر زیر به وضوح مشخص است که رسیدن به حالتی که هر دو معادله ی تعادل نیرو های بین قطعه ای و لنگر حول یک نقطه ی دلخواه دقیقا برآورده شوند تقریبا امری غیر ممکن و زمان بر است.



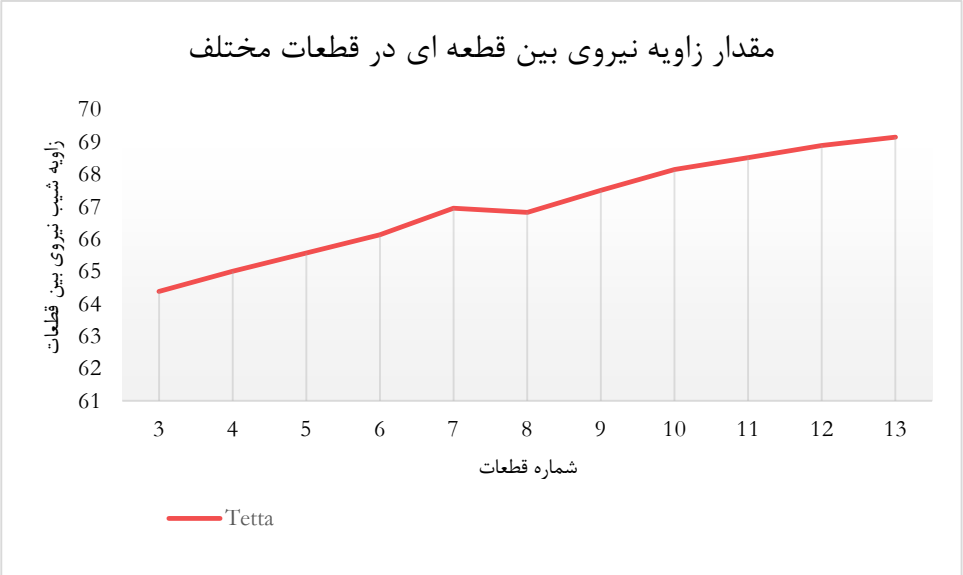
عکس ۲۶ - نابرابری جزئی مقادیر ضریب اطمینان ناشی از برآورده شدن معادله ی ممان و نیروی بین قطعه ای

برای جلوگیری از زمانبر بودن پروسه ی حل به این روش ما از کد نوشته شده در زبان Fortran می خواهیم که مقدار این دو معادله ی وابسته به نیروی بین قطعه ای را به یک تقریبی از صفر برساند. در اشکال زیر مقدار نیرو های عمودی N بر هر قطعه و وزن هر قطعه W آورده شده است. می توان با استفاده از اطلاعات استخراج شده از این دو نمودار مقدار زاویه شیب نیرو های بین قطعه ای را بدست آورد. با تشکیل یک جدول و محاسباتی ساده به ادامه تحلیل می پردازیم.



عکس ۲۷- شکل سمت چپ نیروی بین قطعه ای Z و شکل سمت راست نیروی عمود بر هر قطعه N

در نمودار زیر مقدار زاویه به طور تقریبی برابر با ۶۷ درجه می باشد.



برای جلوگیری از بالارفتن حجم محاسبات در برنامه نویسی ابتدا شرط می کنیم که معادله تعادل نیرو ها با تقریب خوبی به صفر برسد سپس معادله ی ممان ها کنترل شود. جهت درک بیشتر ماهیت روش اسپنسر قبل از نوشتن برنامه ی نهایی ابتدا یک برنامه ی آزمایش می نویسیم. در این برنامه سعی می شود با تغییر دستی مقادیر ضریب اطمینان و زاویه به تقریب خوبی جهت برآورده کردن شروط تعادل نیرو های بین قطعه ای برسیم. ابتدا یک سطح لغزش تقریبی را پیدا می کنیم. سپس به صورت دستی به پارامتر های F و θ مقدار دهی می کنیم که مقادیر بحرانی تقریبی آنها ۱،۰۶۲ و ۶،۵۱۷ می باشد.

نکات جالب و حائز توجهی از این آزمایش بدست آمد :

۱. تقریب ضریب اطمینان و زاویه ی نیرو های بین قطعه ای در روش اسپنسر بسیار بالا و بعضا بیهوده است. این نکته در عین حال که به دقیق بودن روش ارائه شده توسط اسپنسر منجر می شود، حجم محاسبات زیادی را ایجاد می کند.
۲. صفر شدن مطلق معادلات تعادل نیروهای بین قطعه ای و لنگر آن ها تقریبا غیر ممکن است. اما می توان با روش های آزمون و خطا تقریبی مناسب را تعیین کرد، این نکته به خوبی در اختلاف جزئی بوجود آمده در ضرائب اطمینان محاسبه شده به روش تعادل نیرو ها و لنگر ها در نرم افزار GeoSlope مشهود می باشد.
۳. افزایش هر کدام از پارامتر های F و θ باعث افزایش مقادیر معادلات می شود. به نظر می رسد که افزایش مقادیر اولیه ی F تاثیر بیشتری نسبت به θ در افزایش مقادیر محاسباتی دارد.

حال می توانیم وارد مرحله ی نوشتن کد در زبان برنامه نویسی Fortran بشویم.

فرض کنیم که مختصات نقطه ی مرکز دایره سطح لغزش را بین نقاط (۸۰و۱۰) و (۱۵۰و۸۰) ، مختصات نقاط برخورد دایره با خاکریز بین (۱۱۴و۶۰) و (۱۷۰و۶۰) ، تعداد قطعات هر سطح لغزش ۵۰ عدد، محدوده ی کنترل ضریب اطمینان های اولیه بین ۰ تا ۱۰ با دقت ۰،۰۰۱ و محدوده ی کنترل زوایای نیرو های بین قطعه ای بین ۳۰- تا ۳۰ درجه با دقت ۰،۰۰۱ باشد. تعداد حالاتی که نرم افزار باید به محاسبه بپردازد برابر خواهد بود با :

$$N = 70 * 70 * 56 * 50 * 10000 * 60000 = 8.232 * 10^{15}$$

با توجه به کد نرم افزاری نوشته شده تقریبا ۲۰۰ عملیات برای هر دوره ی گردش آن تدارک دیده شده است که تعداد نهایی محاسبات برابر خواهد بود با :

$$N_{total} = 16.464 * 10^{17}$$

این حجم عملیات بسیار زیاد است، پس چاره ای نداریم جز این که ابتدا کلیه نقاط محدوده بندی کنیم و هر محدوده را با دقت های ۰،۰۱ برای ضریب اطمینان و زاویه بررسی کنیم. سپس نقطه ی بحرانی هر محدوده را در اختیار داریم که می توانیم برای رسیدن به تقریبی از صفر برای معادلات تعادل و لنگر تلاش کنیم. این تلاش با زیاد تر شدن تعداد قطعات نیز حاصل می شود. بدین ترتیب می توانیم کد نرم افزار نوشته شده را بصورت زیر ارائه دهیم.

روال الگوریتم این برنامه بدین منوال است :

۱. ابتدا یک نقطه انتخاب می شود.
۲. سپس به F و θ مقدار دهی می شود.
۳. نوع لغزش (عمیق یا سطحی) مشخص می شود.
۴. محاسبات انجام می شود و اگر ترکیب F و θ موجود باعث کمتر شدن مقادیر q و m از ۰،۰۰۱ شد مقادیر q و m را چاپ می کند.
۵. اگر مقداری ضریب اطمینان کمتر از ۳ شد، مقدار ضریب اطمینان و زاویه بیرون کشیده می شود، همزمان مرز ضریب اطمینان کوچکتر می شود.
۶. از میان این نقاطی که مقادیر q و m آن ها از ۰،۰۰۱ کمتر شد آن نقطه ای که کوچکترین ضریب اطمینان را دارد پیدا می شود.

گام های بالا الگوریتم کلی روش اسپنسر می باشد. اما اجرای این الگوریتم برای تمامی نقاط کمی زمان بر است، بر همین اساس طبق پیشنهاد استاد محترم محدوده ی نقاط را کم در نظر می گیریم که از صحیح بودن الگوریتم مطمئن شویم.

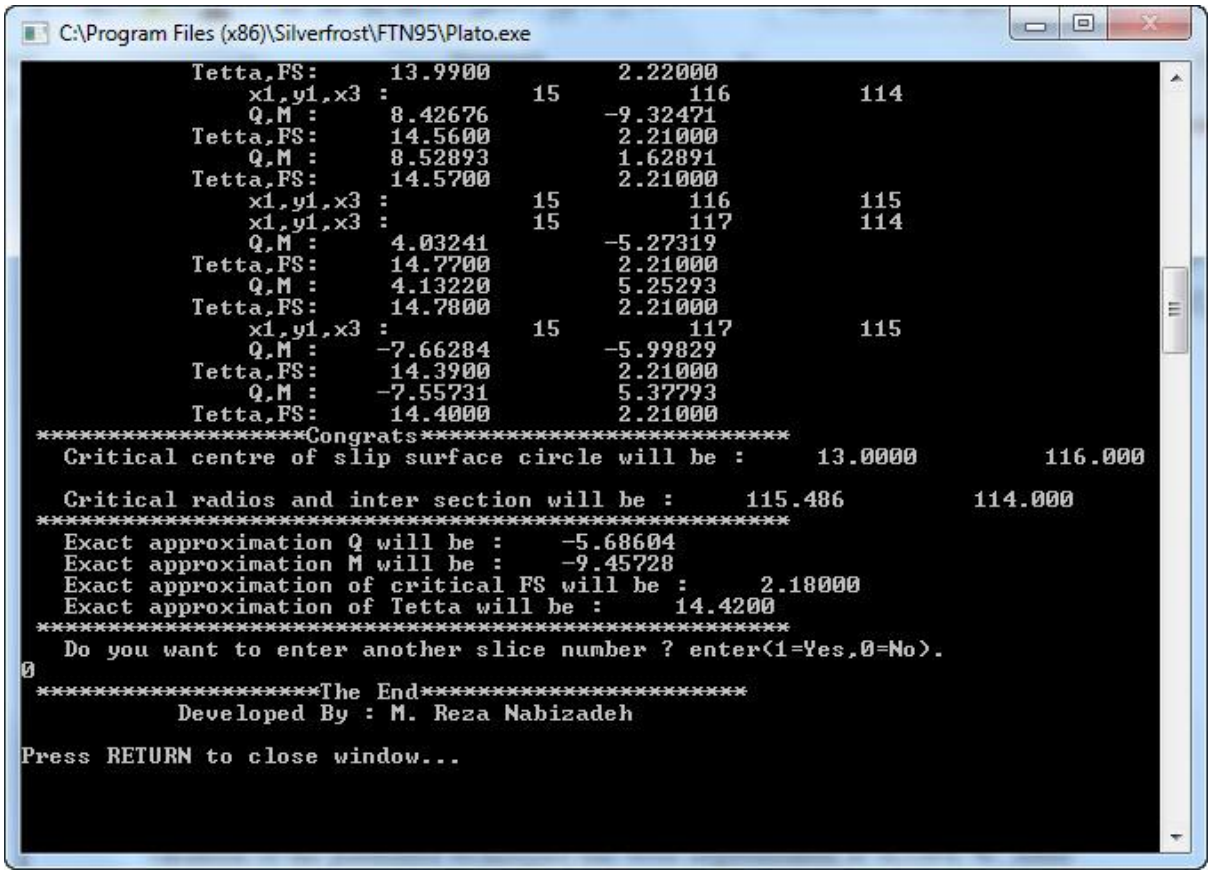
کد برنامه ی نوشته شده عبارت است از :

```
program spencerreservmethod
implicit none

real c,gams,fi,gamw ! Slope parameters
real pi,fir,r0,ra,adel,bdel,cdel,del,xc,yc,l,b,bb !Calculated
parameters
real
xceni,ytopi,yboti,hsoili,wi,xlefti,xrighti,qfinal,mfinal !Calculat
ed parameters
real
ybotlefti,ybotrighti,delli,mi,alfai,ywatti,hwatti,ui,k1i,k2i,k3i,k
4i,k5i,k6i,k7i !Calculated parameters
real qi,q,ml,m,Xcrit,Ycrit,x3crit,rcrit!Calculated parameters
real
fsreservi,fsreserv,fsmini,fsminiaf,tettaki,tettakf !Calculated
parameters
real tettak,tettakr,fsinj,fsfinal,qmini,mmini
integer ns,i,k,j,snt,x1,y1,x3

        PRINT *, '                    ***In The Name Of
God'***
   PRINT *, '                    In this program we attempt
to calculate the'
   PRINT *, '                    factor of safety with
Ordinary method' !Programs information
        print----------------------------------------------' ,*
'------------
   print *, '          Please input the unit weight of slope
material and water':
   read *, gams,gamw
   print *, '          Please input the cohesion factor and
friction angle':
   read *, c,fi
 ۲۰print *, '                    Please input number of the
slices':
   read *, ns
        pi=(asin(1.0))*2
   fir=(fi*pi)/180
        qmini=10
   mmini=10
        fsreservi=3.0
   fsreserv=3.0
   fsminiaf=0.0
    DO x1=10,12
         DO y1=120,122
    DO x3=115,117
        print *, '          x1,y1,x3 :',x1,y1,x3
                          r0=((x3-x1)**2+(y1-
60)**2)**0.5
      ra=((x1-10)**2+(y1)**2)**0.5
        DO j=1,300
        DO k=1,2000
    fsinj=1+(0.01*j(
    tettak=0.0+(0.01*k(
    tettakr=(tettak*pi)/180.0
        q=0.0
   m=0.0
                          IF(r0<=ra) then
        adel=1+((60/104)**2(
        bdel=0-(2*x1)-((2*600*60)/(104*104))-
((2*y1*60)/104(
        cdel=(x1**2)+(y1**2)-
(r0**2)+((600/104)**2)+((2*y1*600)/104(
        del=((bdel)**2)-4*cdel*adel
        xc=(-bdel-(del**0.5))/2*adel
        yc=y1-(((r0**2)-((xc-x1)**2))**0.5(
        l=x3-xc
        b=l/ns
        bb=b/2
      DO i=1,ns
                xceni=xc+(bb*i(
                if (xceni>=114) then
                ytopi=60
          else
          ytopi=((60*xceni)/104)-(600.0/104.0(
          end if
        yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5(
        hsoili=ytopi-yboti
        wi=b*hsoili*gams
        xlefti=xc+bb*(i-1(
        xrighti=xc+bb*(i+1(
        ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5(
        ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5(
        delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
        mi=(ybotrighti-ybotlefti)/(xrighti-
xlefti          (
        alfai=tan(abs(mi((
                if (xceni>=114) then
                ywatti=60
          else
          ywatti=((60*xceni)/104)-(720.0/104.0(
          end if
        hwatti=ywatti-yboti
        ui=hwatti*gamw
        k1i=wi*sin(alfai      (
        k2i=(c*delli)/(fsinj(
        k3i=(ui*delli-wi*cos(alfai))*((tan(fir))/(fsinj((
        k4i=cos(alfai-tettakr(
        k5i=abs(1.0*((sin(alfai-
tettakr))*(tan(fir)))/(fsinj((
                If (mi<=0) then
          k1i=(0.0-1.0)*wi*sin(alfai(
                End If
        qi=(k1i-k2i+k3i)/(k4i+k5i(
        k6i=xceni*sin(tettakr(
        k7i=yboti*cos(tettakr(
```

```
            ml=qi*(k6i-k7i(
            q=q+qi
            m=m+ml

        End DO
                                if (abs(1.0*q)<qmini) then
            if (abs(1.0*m)<mmini) then
                        print *, '                    Q,M :',q,m
            print *, '                    Tetta,FS:',tettak,fsinj
                        fsmini=fsinj
                                IF (fsmini<fsreservi) then
             fsminiaf=Fsmini
                        Fsreservi=Fsmini
              tettaki=tettak
                    end if

            end if
            end if


                                End IF

        IF(r0>ra) then
            yc=0
            xc=x1-(r0**2-(yc-y1)**2)**0.5
                        l=x3-xc
            b=l/ns
            bb=b/2

        DO i=1,ns
                        xceni=xc+(bb*i(
                        if (xceni<=10) then
              ytopi=0
              else
                        if (xceni>=114) then
                                ytopi=60
                        else
                        ytopi=((60*xceni)/104)-(600.0/104.0(
                        end if
              end if
            yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5(
            hsoili=ytopi-yboti
            wi=b*hsoili*gams
            xlefti=xc+bb*(i-1(
            xrighti=xc+bb*(i+1(
            ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5(
            ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5(
            delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
            mi=(ybotrighti-ybotlefti)/(xrighti-
xlefti          (
            alfai=tan(abs(mi   ((
                        if (xceni<=10) then
              ywatti=0
              else
                        if (xceni>=114) then
                                ywatti=60
                        else
                        ywatti=((60*xceni)/104)-
(720.0/104.0(
                        end if
                    end if
                hwatti=ywatti-yboti
                ui=hwatti*gamw
                k1i=wi*sin(alfai      (
                k2i=(c*delli)/(fsinj(
                k3i=(ui*delli-wi*cos(alfai))*((tan(fir))/(fsinj((
                k4i=cos(alfai-tettakr(
                k5i=abs(1.0*((sin(alfai-
tettakr))*(tan(fir)))/(fsinj((
                        If (mi<=0) then
                  k1i=(0.0-1.0)*wi*sin(alfai(
                        End If
                qi=(k1i-k2i+k3i)/(k4i+k5i(
                k6i=xceni*sin(tettakr(
                k7i=yboti*cos(tettakr(
                ml=qi*(k6i-k7i(
                q=q+qi
                m=m+ml
            End DO
                                if (abs(1.0*q)<qmini) then
                if (abs(1.0*m)<mmini) then
                        print *, '                    Q,M :',q,m
                print *, '                    Tetta,FS:',tettak,fsinj
                        fsmini=fsinj
                                IF (fsmini<fsreservi) then
                 fsminiaf=Fsmini
                        Fsreservi=Fsmini
                  tettaki=tettak
                      end if

                end if
                end if
                                End IF


                        End do
            End do
                                                        IF
(fsminiaf<fsreserv .and. fsminiaf/=0.0) then
                                        fsfinal=Fsminiaf


            Fsreserv=Fsminiaf
                    tettakf=tettaki
                    qfinal=q
                    mfinal=m
                    xcrit=x1
                    ycrit=y1
                    x3crit=x3
                    rcrit=r0
                    end if
      End DO
      End DO
      End DO

      print *,
'******************Congrats'**************************
      print *, '  Critical centre of slip surface circle will
be :',xcrit,ycrit
```

```
   print *, '  Critical radios and inter section will
be :',rcrit,x3crit
            print ,*
**************************************************'
'***
   print *, '  Exact approximation Q will be :',qfinal
   print *, '  Exact approximation M will be :',mfinal
   print *, '  Exact approximation of critical FS will be :',fsfinal
   print *, '  Exact approximation of Tetta will be :',tettakf
            print ,*
**************************************************'
'***
   print *, '  Do you want to enter another slice number ?
enter(1=Yes,0=No'.(
   read *, snt
                        IF (snt==1) THEN
                                goto 20
                                END IF

   print *, '********************The
End'************************
   print *, '              Developed By : M. Reza Nabizadeh'

 end program spencerreservmethod
```

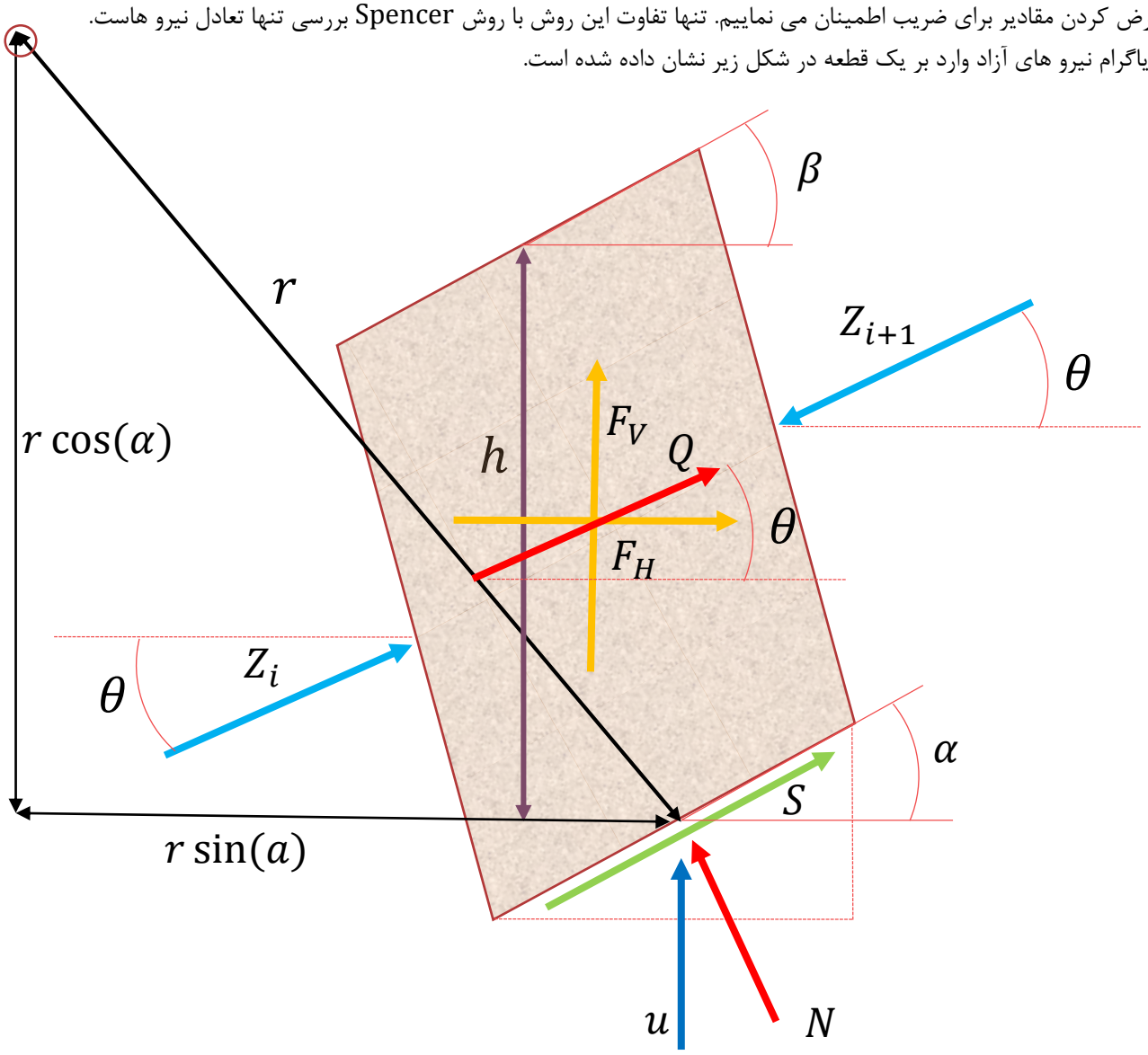
در این حالت می آییم نقاطی را که سایر روش ها به ما داده اند برای کنترل مقادیر در روش اسپنسر قرار می دهیم.
برای مثال مرکز دایره ای با مختصات ۱۳و ۱۱۵ و با محل برخورد ۱۱۴ مقدار ضریب اطمینان زیر را می دهد.



عکس ۲۸- ضریب اطمینان ۲،۱۸ برای سطح لغزش دلخواه در روش اسپنسر

۸. روش Lowe and Karafiath

این روش از دسته روش های تعادل نیرویی می باشد. در این روش ابتدا هر سطح لغزش قطعه بندی شده و محاسبات آن انجام می شود. با فرض این که زاویه نیرو های بین قطعه ای برابر است با میانگین زاویه شیب خاکریز و زاویه هر قطعه است اقدام به فرض کردن مقادیر برای ضریب اطمینان می نماییم. تنها تفاوت این روش با روش Spencer بررسی تنها تعادل نیرو هاست. دیاگرام نیرو های آزاد وارد بر یک قطعه در شکل زیر نشان داده شده است.



عکس ۲۹- دیاگرام نیرو های وارد بر یک قطعه در روش Lowe

پس از برآیند گیری از نیرو های Z_{i+1} و Z_i و برابر قرار دادن مقدار آن با Q می توان معادلات تعادل در راستای N و S را برای این قطعه نوشت. در کتاب Soil Strength and Slope Stability (J. Michael Duncan, 2005) منظور از نیروی های افقی را نیرو های ناشی از زلزله و یا تسلیح خاک بیان کرده است. و جهت نیرو های عمودی که ناشی از وزن قطعه می باشد را رو به بالا فرض کرده است. با فرض نیروی وزن به سمت پایین و حذف نیرو های افقی معادلات تعادل به شرح زیر می باشد :

$$\sum F_N = N - W\cos(\alpha) - Q\sin(\alpha - \theta) = 0$$

$$\sum F_S = S - W\cos(\alpha) - Q\sin(\alpha - \theta) = 0$$

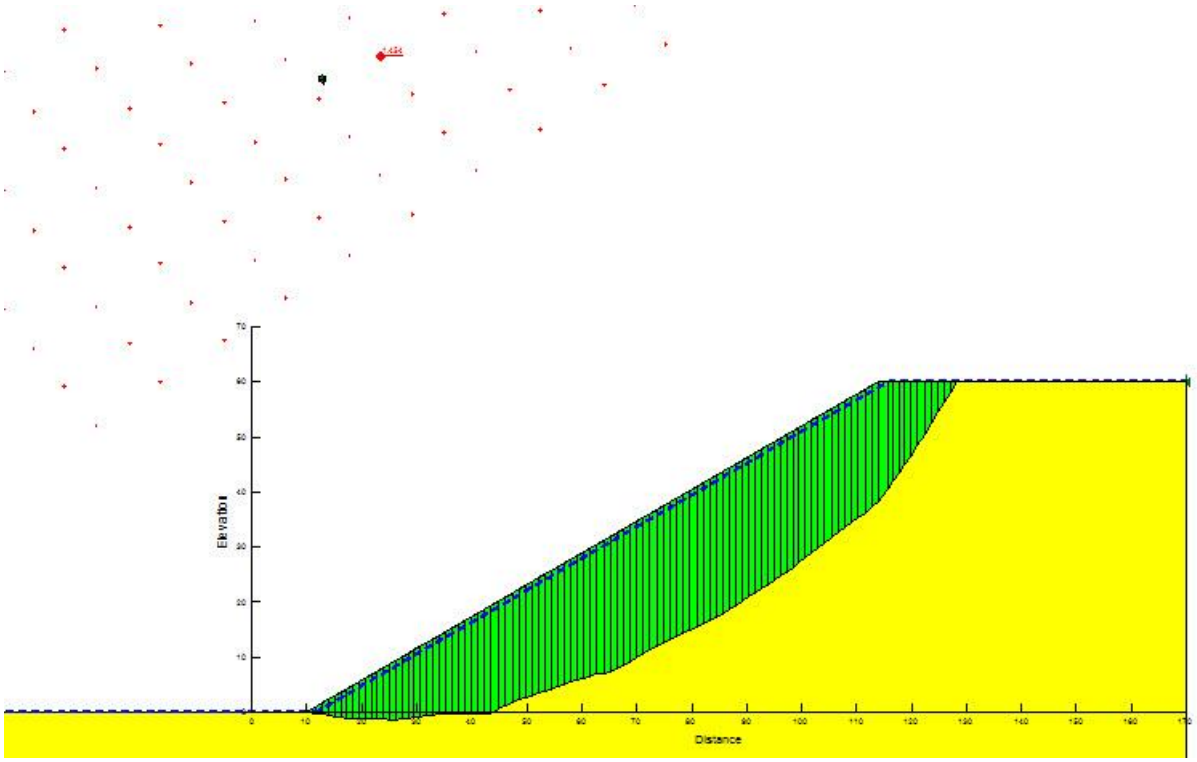
با در نظر گرفتن رابطه ی زیر که بین S و N نوشته می شود :

$$S = \frac{1}{F}[C'\Delta l + (N - u\Delta l)\tan(\varphi')]$$

که پس از ترکیب سه معادله ی بالا معادله ی نهایی جهت بدست آوردن مقدار Q به صورت زیر حاصل می شود :

$$Q = \frac{W\sin(\alpha) - \frac{C'\Delta l}{F} + \frac{\tan(\varphi')}{F}[u\Delta l - W\cos(\alpha)]}{\cos(\alpha-\theta) + \frac{\sin(\alpha-\theta)\tan(\varphi')}{F}}$$

تحلیل نرم افزار GeoSlope از پیدا کردن سطح لغزش بحرانی و مقدار ضریب اطمینان به روش Lowe در شکل زیر نشان داده شده است. مقدار ضریب اطمینان بحرانی ارائه شده برابر ۱,۴۵ می باشد.



عکس ۳۰- سطح لغزش بحرانی و ضریب اطمینان بحرانی ۱,۴۵ ارائه شده بر اساس روش Lowe

حال می توانیم به نوشتن کد در زبان برنامه نویسی Fortran بپردازیم. برنام ی نوشته شده به شرح زیر است :

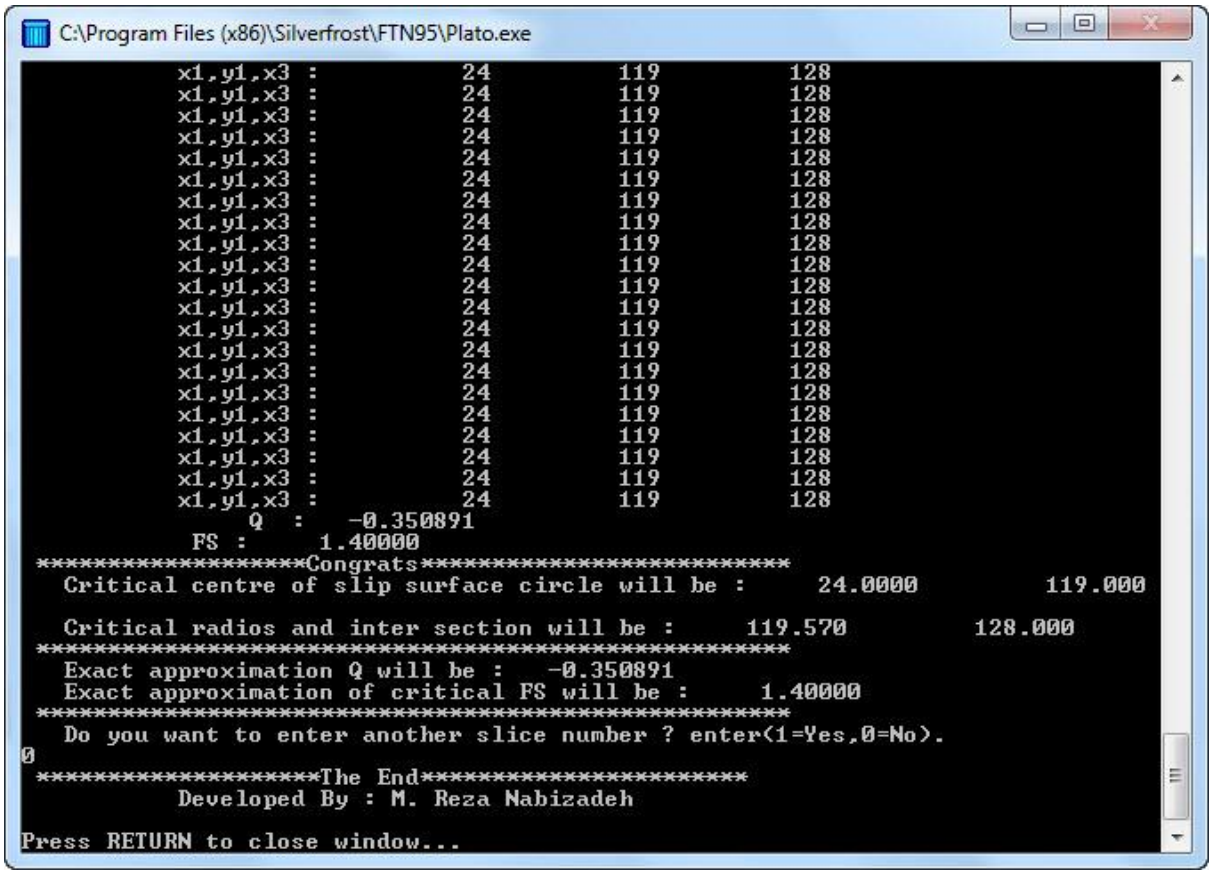
```
program lowemethod
implicit none

real c,gams,fi,gamw ! Slope parameters
real pi,fir,r0,ra,adel,bdel,cdel,del,xc,yc,l,b,bb !Calculated parameters
real xceni,ytopi,yboti,hsoili,wi,xlefti,xrighti,qfinal !Calculated parameters
real ybotlefti,ybotrighti,delli,mi,alfai,ywatti,hwatti,ui,k1i,k2i,k3i,k4i,k5i !Calculated parameters
real qi,q,Xcrit,Ycrit,x3crit,rcrit!Calculated parameters
real fsreservi,fsreserv,fsmini,fsminiaf,qpre !Calculated parameters
real fsinj,fsfinal,br,tettai
integer ns,i,snt,x1,y1,x3

        PRINT *, '                    ***In The Name Of God***'
  PRINT *, '                    In this program we attempt to calculate the'
  PRINT *, '                    factor of safety with Ordinary method' !Programs information
        print *, '-------------------------------------------------------'
  print *, '          Please input the unit weight of slope material and water :'
  read *, gams,gamw
  print *, '          Please input the cohesion factor and friction angle :'
  read *, c,fi
20 print *, '                    Please input number of the slices :'
  read *, ns
  print *, '          Please x1,y1,x3 :'
  read *, x1,y1,x3
        pi=(asin(1.0))*2
  fir=(fi*pi)/180
  br=     (38.0*pi)/180
        fsreservi=3.0
  fsreserv=3.0
  fsminiaf=0.0
```

```
                                    r0=((x3-x1)**2+(y1-
60)**2)**0.5
        ra=((x1-10)**2+(y1)**2)**0.5

    fsinj=1.5
    q=0.0
40   print *, '                          x1,y1,x3 :',x1,y1,x3
                                    IF(r0<=ra) then
              adel=1+((60/104)**2)
              bdel=0-(2*x1)-((2*600*60)/(104*104))-
((2*y1*60)/104)
              cdel=(x1**2)+(y1**2)-
(r0**2)+((600/104)**2)+((2*y1*600)/104)
              del=((bdel)**2)-4*cdel*adel
              xc=(-bdel-(del**0.5))/2*adel
              yc=y1-(((r0**2)-((xc-x1)**2))**0.5)
              l=x3-xc
              b=l/ns
              bb=b/2
         DO i=1,ns
                        xceni=xc+(bb*i)
              if (xceni>=114) then
              ytopi=60
                else
                ytopi=((60*xceni)/104)-(600.0/104.0)
                end if
              yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5)
              hsoili=ytopi-yboti
              wi=b*hsoili*gams
              xlefti=xc+bb*(i-1)
              xrighti=xc+bb*(i+1)
              ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5)
              ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5)
              delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
              mi=(ybotrighti-ybotlefti)/(xrighti-xlefti)
              alfai=tan(abs(mi))
              tettai=(alfai+br)/2.0
              if (xceni>=114) then
              ywatti=60
                else
                ywatti=((60*xceni)/104)-(720.0/104.0)
                end if
              hwatti=ywatti-yboti
              ui=hwatti*gamw
              k1i=wi*sin(alfai)
              k2i=(c*delli)/(fsinj)
              k3i=(ui*delli-wi*cos(alfai))*((tan(fir))/(fsinj))
              k4i=cos(alfai-tettai)
              k5i=abs(1.0*((sin(alfai-tettai))*(tan(fir)))/(fsinj))
              If (mi<=0) then
                k1i=(0.0-1.0)*wi*sin(alfai)
              End If
              qi=(k1i-k2i+k3i)/(k4i+k5i)
              q=q+qi
         End DO
                                    if (q<1 .and. q>-1) then
                        print *, '                Q  :',q
              print *, '                FS :',fsinj
              fsmini=fsinj
                                    IF (fsmini<fsreservi) then
                fsminiaf=Fsmini
                        Fsreservi=Fsmini
              qpre=q
                end if
                                          else
              if(q>1) then
              fsinj=fsinj-0.1
              go to 40
              end if
              if(q<-1) then
              fsinj=fsinj+0.1
              go to 40
              end if
            end if

                                    End IF

        IF(r0>ra) then
            yc=0
            xc=x1-(r0**2-(yc-y1)**2)**0.5
            l=x3-xc
            b=l/ns
            bb=b/2

        DO i=1,ns
                        xceni=xc+(bb*i)
            if (xceni<=10) then
              ytopi=0
              else
                        if (xceni>=114) then
                        ytopi=60
                        else
                        ytopi=((60*xceni)/104)-(600.0/104.0)
                        end if
              end if
            yboti=y1-(((r0**2)-((xceni-x1)**2))**0.5)
            hsoili=ytopi-yboti
            wi=b*hsoili*gams
            xlefti=xc+bb*(i-1)
            xrighti=xc+bb*(i+1)
            ybotlefti=y1-(((r0**2)-((xlefti-x1)**2))**0.5)
            ybotrighti=y1-(((r0**2)-((xrighti-x1)**2))**0.5)
            delli=((xrighti-xlefti)**2+(ybotrighti-
ybotlefti)**2)**0.5
            mi=(ybotrighti-ybotlefti)/(xrighti-xlefti)
            alfai=tan(abs(mi))
            tettai=(alfai+br)/(2.0)
            if (xceni<=10) then
              ywatti=0
              else
                        if (xceni>=114) then
                        ywatti=60
                        else
                        ywatti=((60*xceni)/104)-
(720.0/104.0)
                        end if
              end if
            hwatti=ywatti-yboti
            ui=hwatti*gamw
            k1i=wi*sin(alfai)
            k2i=(c*delli)/(fsinj)
            k3i=(ui*delli-wi*cos(alfai))*((tan(fir))/(fsinj))
            k4i=cos(alfai-tettai)
```

```
k5i=abs(1.0*((sin(alfai-tettai))*(tan(fir)))/(fsinj))
If (mi<=0) then
  k1i=(0.0-1.0)*wi*sin(alfai)
End If
qi=(k1i-k2i+k3i)/(k4i+k5i)
q=q+qi
End DO
                    if (q<1 .and. q>-1) then
          print *, '                Q  :',q
print *, '                FS :',fsinj
fsmini=fsinj
          IF (fsmini<fsreservi) then
  fsminiaf=Fsmini
          Fsreservi=Fsmini
  qpre=q
          end if
                                        else
          if(q>1) then
          fsinj=fsinj-0.1
          go to 40
          end if
          if(q<-1) then
    fsinj=fsinj+0.1
            go to 40
            end if
end if

End If

                                        IF
(fsminiaf<fsreserv .and. fsminiaf/=0.0) then
                                fsfinal=Fsminiaf

Fsreserv=Fsminiaf
      qfinal=qpre
```

```
xcrit=x1
ycrit=y1
x3crit=x3
rcrit=r0
end if

                                IF
(abs(1.0*fsminiaf)<0.00001) then
  print *, 'With this kind of limits the is no resonable FS exists'
                                        else
  print                                        *,
'******************Congrats*************************'
  print *, '   Critical centre of slip surface circle will
be :',xcrit,ycrit
  print *, '   Critical radios and inter section will
be :',rcrit,x3crit
        print                                  *,
'*****************************************************'
  print *, '  Exact approximation Q will be :',qfinal
  print *, '  Exact approximation of critical FS will be :',fsfinal
        print                                  *,
'*****************************************************'
                                        end if
  print *, '  Do you want to enter another slice number ?
enter(1=Yes,0=No).'
  read *, snt
                IF (snt==1) THEN
                        goto 20
                        END IF

  print          *,          '*******************The
End**********************'
  print *, '          Developed By : M. Reza Nabizadeh'

end program lowemethod
```

نتایج بدست آمده برای سطح لغزش بحرانی در شکل زیر نشان داده شده است.



عکس ۳۱- نتایج حاصله از روش Lowe در زبان Fortran

همانگونه که ملاحظه می شود اضریب اطمینان ارائه شده تقریب خوبی از ضریب اطمینان ارائه شده توسط نرم افزار GeoSlopeاست.

کنترل ظرفیت باربری بستر خاکریز

بر اساس نظریه ترزاقی (Das, Principles of Foundation Engineering, 2015) ظرفیت باربری برابر خواهد بود با :

$$q_u = cN_c + qN_q + \frac{1}{2}\gamma B N_\gamma$$

چون خاکر کف خاکریز را در نظر می گیریم که مدفون نمی باشد از ترم دوم رابطه ی ظرفیت باربری صرف نظر خواهیم کرد.
با مراجعه به جدول ها مقادیر N_c و N_γ به ترتیب برابر ۹۰ و ۷۰ می باشند. اگر عرض پی را ۱۰۰ متر در نظر بگیریم خواهیم داشت :

$$q_u = cN_c + qN_q + \frac{1}{2}\gamma B N_\gamma = 70 * 90 + 0.5 * 18 * 100 * 70 = 69300 \ {}^{kN}/_{m^2}$$

اگر مقدار تنش وارده برابر با :

$$Q = 18 * 60 = 1080 \ {}^{kN}/_{m^2}$$

باشد. مقدار ضریب اطمینان بیشتر از ۳ برای ما قابل قبول است پس :

$$FS = \frac{69300}{1080} = 64.16$$

این مقدار قابل قبول است.

مقایسه ی نتایج با نرم افزار GeoSlope

در جدول زیر در سطوح لغزشی اجباری مقادیر ضرایب اطمینان محاسبه شده توسط نرم افزار GeoSlope از مجموعه نرم افزاری GeoStudio با ضرایب اطمینان محاسبه شده توسط کد های نوشته شده در زبان Fortran مقایسه شده است.

روش	ضریب اطمینان	
	GeoSlope	زبان Fortran
Ordinary	۱,۳۲	۱,۳۶
Simplified Bishop	۱,۴۸	۱,۱۹
Spencer	۱,۴۶	۱,۶۵
Lowe	۱,۴۵۴	۱,۴

تقریب های بوجود آمده می تواند دلایلی همچون تفاوت بین اعداد حقیقی و صحیح، مقدار نیروی فشار آب حفره ای و ... باشد.

نتیجه گیری

در هر حال انجام این تمرین به پایان رسید و نکات بسیار آموزنده ای از آن بوجود آمد. مهم ترین نکته ی موجود در بررسی نتایج و اعداد حاصله ی این تمرین مقایسه ی نتایج با نرم افزار GeoStudio می باشد. به کرات مشاهده شد که ضریب اطمینان های محاسبه شده به مقدار ضریب اطمینان ها ی ارائه شده توسط نرم افزار همخوانی مطلوبی دارند اما آنچه که مهم است نزدیک بودن این اعداد مهم نیست. چرا که ماهیت پروسه ی انتخاب ضریب اطمینان بحرانی همانند یک جعبه ی سیاه نیست که از یک طرف ورودی و از طرف دیگر خروجی داشته باشیم. آنچه که حائز اهمیت است آن است که در سطح لغزش های یکسان بین نرم افزار و روش های حل ارائه شده همخوانی مطلوبی وجود داشته باشد.

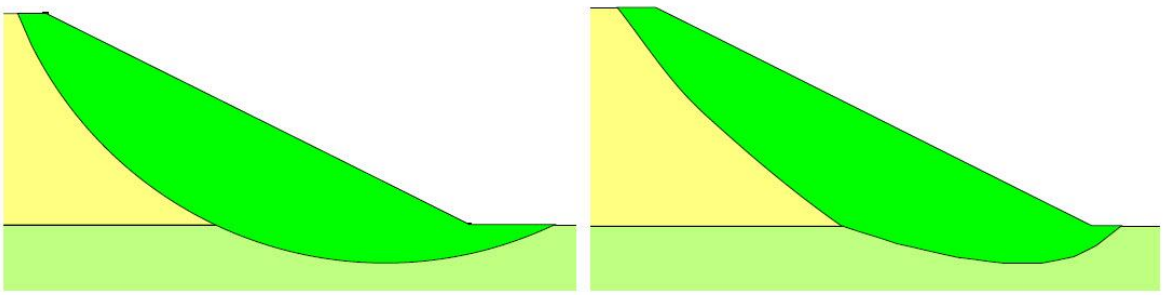
در این موضوع که سطح گسیختگی بحرانی ارائه شده این دو رویکرد با هم مختلف است باید تامل کرد در راهنمای نرم افزاری GeoSlope (GEO-SLOPE International, 2008)آورده شده است :

Recent research has shown that assumed user-specified slip surface shapes do not necessarily give the lowest possible factor of safety. Algorithms that alter the slip surface shape in some systematic manner tend to give lower safety factors than for a predetermined slip surface shape. For example, say a minimum factor of safety has been found for a circular slip surface. Further refinement of the shape can lead to a non-circular slip surface with a lower factor of safety. In SLOPE/W this is called optimization, auto-search or auto location of the slip surface shape and position. Finding the critical slip surface shape and position still remains one of the key issues in any limit equilibrium stability analysis, in spite of all the advanced techniques available. It is not as yet a black box solution. Guidance from the user still remains an essential ingredient. (GEO-SLOPE International, 2008)

به هر حال اهمیت قضاوت مهندسی جهت انتخاب سطح لغزش بحرانی در این گونه مسائل بسیار مورد توجه است. چرا که انتخاب یا عدم انتخاب یک سطح لغزش می تواند هزینه های گزاف اجرائی یا از آن بدتر خسارت های جانی را به بار آورد. نا غافل نمایم که یک از عوامل تاثیرگذار در انتخاب سطح لغزش توسط نرم افزار GeoStudio مبحث بهینه سازی سطح لغزش است که در ادامه در توضیح این اصطلاح داریم :

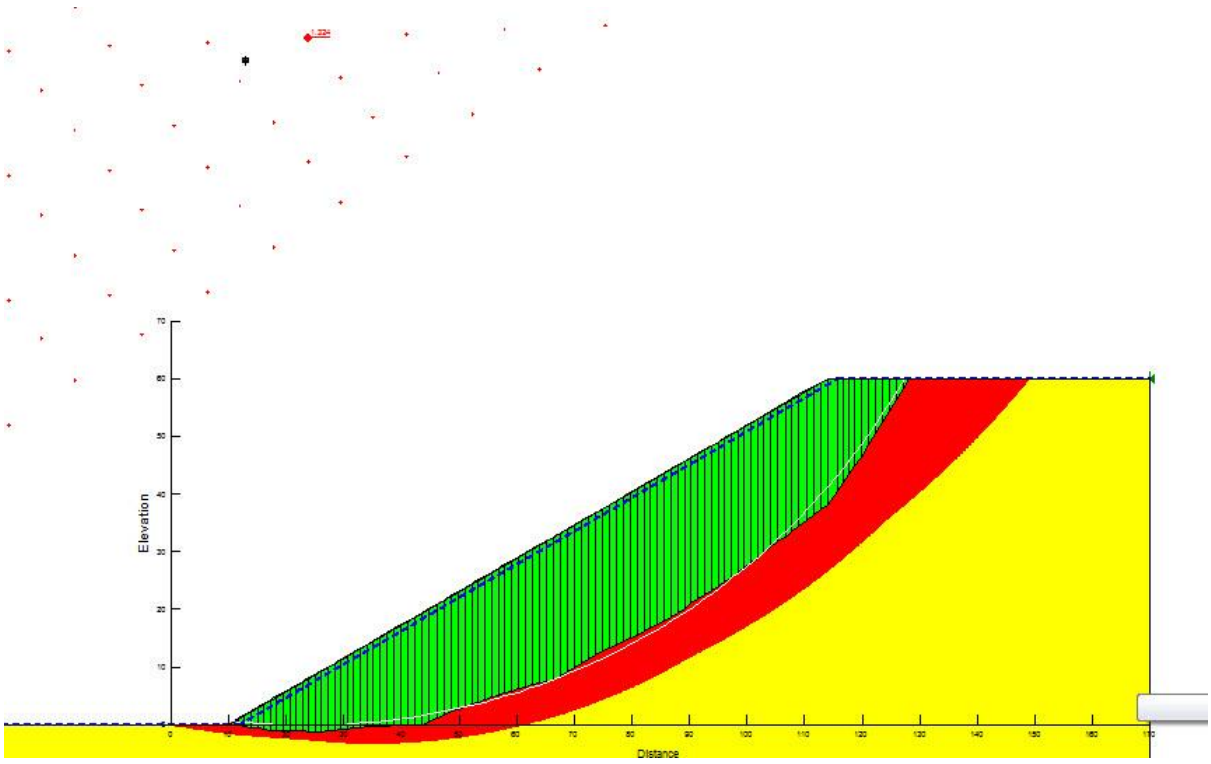
Optimization :

All the traditional methods of forming trial slip surfaces change the entire slip surface. Recent research has explored the possibility of incrementally altering only portions of the slip surface (Greco, 1996; Malkawi, Hassan and Sarma, 2001). A variation of the published techniques has been implemented in SLOPE/W. After finding the critical slip surface by one of the more traditional methods, the new segmental technique is applied to optimize the solution. The first step in the optimization process is to divide the slip surface into a number of straight line segments. The slip surface in essence becomes just like a Fully Specified slip surface. Next, the end points of the line segments are moved to probe the possibility of a lower safety factor. The process starts with the point
where the slip surface enters ground surface. This point is moved backward and forward randomly along the ground surface until the lowest safety factor is found. Next, adjustments are made to the next point along the slip surface until again the lowest safety factor is found. This process is repeated for all the points along the slip surface. Next, the longest slip surface line segment is subdivided into two parts and a new point is inserted into the middle. This process is repeated over and over
until changes in the computed safety factor are within a specified tolerance or until the process reaches the specified limits (e.g., the maximum number of optimization trials). Figure 4-30 presents an optimized slip surface relative to a traditional circular slip surface of a simple slope. The material above the toe is somewhat stronger than the underlying soil below the toe elevation in this example. The factor of safety for this circular slip surface is 1.280, while for the optimized case the factor of safety is 1.240. Of interest is the observation that there is another slip surface that leads to a lower factor of safety than the one obtained for an assumed circular slip surface. A key element in the optimization procedure is the technique used to move the end points of the line segments. SLOPE/W moves the points within an elliptical search area using a statistical random walk procedure based on the Monte Carlo method.



عکس ۳۲- بهینه سازی سطح لغزش شکل سمت چپ حالت عادی و شکل سمت راست بهینه سازی شده

یکی از امکانات بسیار جذاب نرم افزار GeoStudio امکان مشاهده ی محدوده ی سطوح لغزش بحرانی به صورت همزمان است. در شکل زیر محدوده ی قرمز رنگ محل قرار گیری سطوح لغزش بحرانی و سطح سبز رنگ همان سطح لغزش بحرانی است. یکی از نکات جالب در عین حال مهم و تاثیرگذار در قضاوت های این نرم افزار عبور اجباری سطح لغزش از پنجه ی خاکریز می باشد که این موضوع خود بر عدم برابری سطوح لغزش بحرانی ارائه شده توسط دو رویکرد نرم افزار GeoSlope و حل های صحه می گذارد.



عکس ۳۳- محدوده ی سطح گسیختگی های بحرانی در GeoSlope

و به عنوان نکته کلیدی آخر باید بیان شود که :

قضاوت مهندسی و درک اولیه ی عمیق از صورت مسئله می تواند عاملی بسیار تعیین کننده و مهم در چگونگی طراحی و اجرای یک پروژه باشد.

منابع

Bibliography

Das, B. M. (2007). *Theoretical Foundation Engineering.*

Das, B. M. (2015). *Principles of Foundation Engineering.*

GEO-SLOPE International, L. (2008). *Stability Modeling with Slope/W 2007.*

Gunaratne, M. (2013). *The Foundation Engineering Handbook, Second Edition.*

J. Michael Duncan, S. G. (2005). *Soil Strength and Slope Stability.*

Murthy, V. (2002). *Geotechnical Engineering: Principles and Practices of Soil Mechanics and Foundation Engineering (Civil and Environmental Engineering).*