تمرین

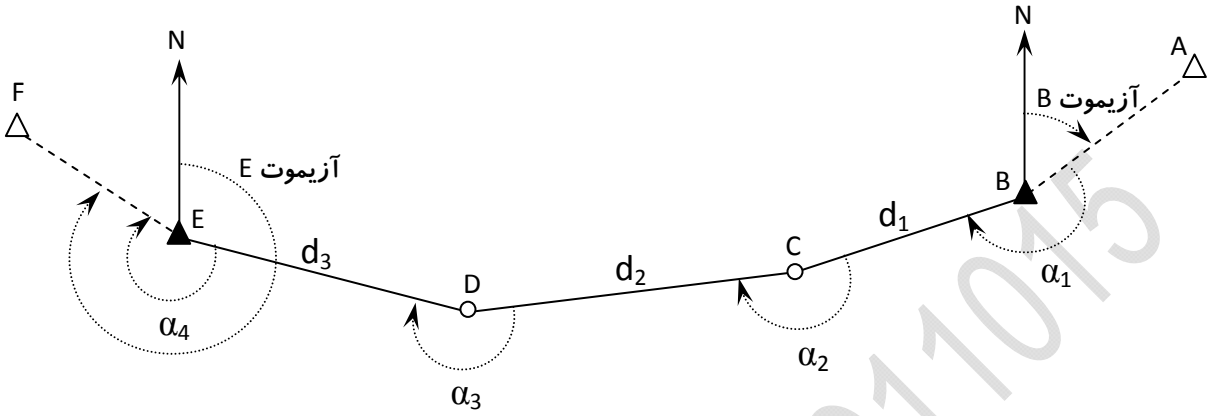
دروس

تعدیل و سرشکنی و نقشه‌برداری ژئودتیک

مهندس سعادت

دی ماه ۱۳۹۱

مساله: در پیمایش زیر زوایا و طول‌های اشاره‌شده به همراه انحراف معیار مشاهدات داده شده است.



زاویه	σ_1	طول	σ_2
α_1=172˚ 53′ 34″	10″	d_1=281.832 m	0.016 m
α_2=185˚ 22′ 14″	10″	d_2=271.300 m	0.016 m
α_3=208˚ 26′ 19″	10″	d_3=274.100 m	0.016 m
α_4=205˚ 13′ 51″	10″		

نقاط	مختصات		
	X (m)	Y (m)	آزیموت
B	8478.139	2483.826	68˚ 15′ 20.7″
E	7709.336	2263.411	300˚ 11′ 30.5″

مطلوب است تعیین مختصات نقاط مجهول و برآورد دقت محاسبات؟

الف) حالت مینیمم کانسترینت (Minimum Constraints): اگر مختصات نقاط B و E ثابت درنظر گرفته شود.

۱) بردار مشاهدات:

$$l = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ d_1 \ d_2 \ d_3]^T$$

$$C_l = diag([\sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_2^2 \ \sigma_2^2 \ \sigma_2^2]^T) \quad ; \quad P = \sigma_0^2 C_l^{-1}$$

۲) بردار مجهولات:

$$x = [x_C \ y_C \ x_D \ y_D]^T$$

$$x^o = [x_C^o \ y_C^o \ x_D^o \ y_D^o]^T$$

۳) درجه آزادی:

$$n = 7 \quad ; \quad u = 4 \quad ; \quad Constraint = 0 \quad ; \quad Defect = 0$$

$$r = n - u + Constraint + Defect = 3$$

۴) مدل ریاضی (روش پارامتریک):

$$l_{n\times 1} = f(x_{u\times 1})$$

$$\alpha_1 = tg^{-1}\frac{x_C - x_B}{y_C - y_B} + \pi - Az_B$$

$$\alpha_2 = tg^{-1}\frac{x_D - x_C}{y_D - y_C} + \pi - tg^{-1}\frac{x_B - x_C}{y_B - y_C}$$

$$\alpha_3 = 2\pi - tg^{-1}\frac{x_E - x_D}{y_E - y_D} - tg^{-1}\frac{x_C - x_D}{y_C - y_D}$$

$$\alpha_4 = Az_B - \pi + tg^{-1}\frac{x_D - x_E}{y_D - y_E}$$

$$d_1 = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}$$

$$d_2 = \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2}$$

$$d_3 = \sqrt{(x_E - x_D)^2 + (y_E - y_D)^2}$$

جواب مساله:

$$A\delta\hat{x} = \hat{v} + \delta l \quad ; \quad A = \frac{\partial f}{\partial x} \quad ; \quad \delta l = l^o - f(x^o)$$

$$N = A^T P A \quad ; \quad U = A^T P \delta l$$

$$\delta\hat{x} = N^{-1}U \quad ; \quad \hat{v} = A\delta\hat{x} - \delta l$$

$$\hat{x} = x^o + \delta\hat{x} \quad ; \quad \hat{l} = l^o + \hat{v}$$

$$\hat{\sigma}_0^2 = \frac{\hat{v}^T P \hat{v}}{r}$$

$$C_{\hat{x}} = A^T C_l^{-1} A = \sigma_0^2 N^{-1} \quad ; \quad \hat{C}_{\hat{X}} = \hat{\sigma}_0^2 N^{-1}$$

$$C_{\hat{l}} = A(A^T C_l^{-1} A)^{-1} A^T = \sigma_0^2 A N^{-1} A^T \quad ; \quad \hat{C}_{\hat{l}} = \hat{\sigma}_0^2 A N^{-1} A^T$$

$$C_{\hat{v}} = C_l - C_{\hat{l}} \quad ; \quad \hat{C}_{\hat{v}} = \hat{\sigma}_0^2 (P^{-1} - A N^{-1} A^T)$$

ماتریس آزادی:

$$R = C_{\hat{v}} C_l^{-1}$$

اعداد آزادی:

$$Trace(R) = \sum_{i=1}^{n} d_i$$

تست فاکتور وریانس:

$$\frac{r\hat{\sigma}_0^2}{\sigma_0^2} \rightsquigarrow \chi_r^2$$

ابعاد بیضی خطای مطلق:

$$a_0^2 = \frac{1}{2}\left(\sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2}\right) \quad \rightarrow \quad a = \sqrt{K}a_0$$

$$b_0^2 = \frac{1}{2}\left(\sigma_x^2 + \sigma_y^2 - \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2}\right) \quad \rightarrow \quad b = \sqrt{K}b_0$$

$$\theta = \frac{1}{2} tg^{-1}\left(\frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2}\right) \quad ; \quad K = \chi_{r,1-\alpha}^2$$

برنامه مربوط به حالت Minimum Constraints:

```
clc
clear
format short g

%Minimum constrain adjustment program
%Constant value
MapScale=500;
D2R=pi/180;
Epsilon=0.01;
Sigma02=1;
n=7;
u=4;
Defect=0;
Constrain=0;

%Degree of freedom
r=n-u+Defect+Constrain;

%Confidence level
Alfa=0.01;
Alfa1=Alfa/2;
Alfa2=1-Alfa1;

%Fixed Coordinates and Azimuth
XB=8478.139; YB=2483.826;
XE=7709.336; YE=2263.411;

Az_B=(68+15/60+20.7/3600)*D2R;
Az_E=(300+11/60+30.5/3600)*D2R;

%Observations
a1=(172+53/60+34/3600)*D2R;
a2=(185+22/60+14/3600)*D2R;
a3=(208+26/60+19/3600)*D2R;
a4=(205+13/60+51/3600)*D2R;
d1=281.832;
d2=271.300;
d3=274.100;

lo=[a1;a2;a3;a4;d1;d2;d3];

%Standard diviations of observations
sd_a=(10/3600)*D2R;
sd_d=0.016;

Cl=diag([sd_a^2 sd_a^2 sd_a^2 sd_a^2 sd_d^2 sd_d^2 sd_d^2]);
P=Sigma02*inv(Cl);

%Unknown parameters
%X=[XC,YC,XD,YD]'

%Approximate value of coordinates
Az_BC=Az_B+a1;
XCo=XB+d1*sin(Az_BC);
YCo=YB+d1*cos(Az_BC);
```

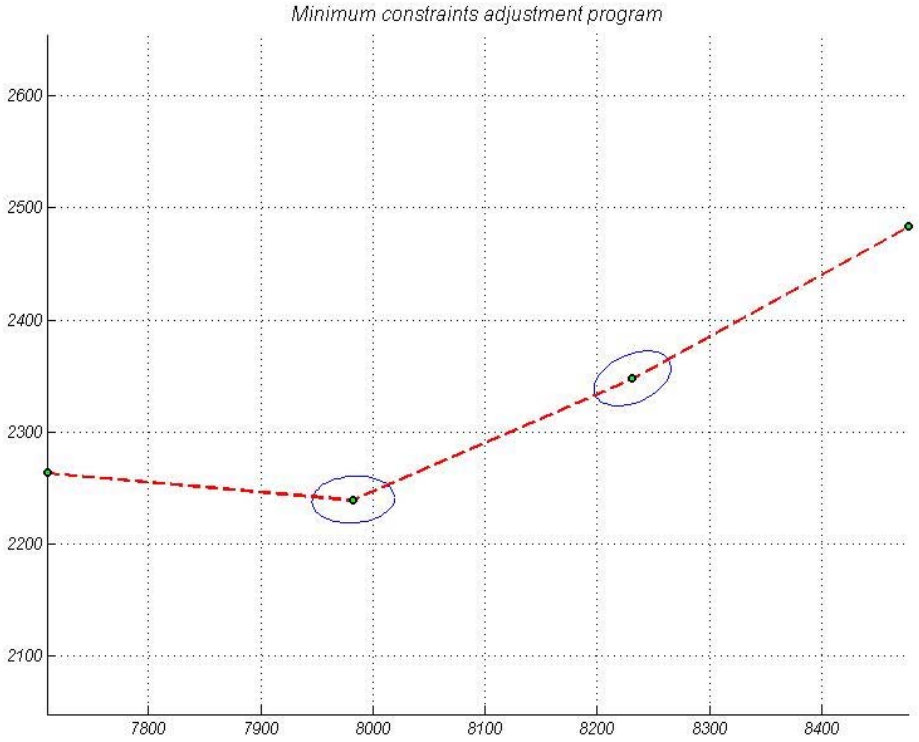
```
Az_CB=Az_BC-pi;
Az_CD=Az_CB+a2;
XDo=XCo+d2*sin(Az_CD);
YDo=YCo+d2*cos(Az_CD);

Xo=[XCo,YCo,XDo,YDo]';

%Calculations
dx=Epsilon;
Iteration=0;
while norm(dx)>=Epsilon
    XCo=Xo(1); YCo=Xo(2); XDo=Xo(3); YDo=Xo(4);

    L_BC=sqrt((XCo-XB)^2+(YCo-YB)^2);
    L_CD=sqrt((XDo-XCo)^2+(YDo-YCo)^2);
    L_DE=sqrt((XE-XDo)^2+(YE-YDo)^2);

    A=[(YCo-YB)/L_BC^2     -(XCo-XB)/L_BC^2     0     0
       (YB-YCo)/L_BC^2-(YDo-YCo)/L_CD^2  -(XB-XCo)/L_BC^2+(XDo-
       XCo)/L_CD^2 (YDo-YCo)/L_CD^2      -(XDo-XCo)/L_CD^2
       -(YCo-YDo)/L_CD^2                  (XCo-XDo)/L_CD^2
       (YCo-YDo)/L_CD^2-(YE-YDo)/L_DE^2  -(XCo-XDo)/L_CD^2+(XE-
       XDo)/L_DE^2
       0      0            (YDo-YE)/L_DE^2   -(XDo-XE)/L_DE^2
       (XCo-XB)/L_BC       (YCo-YB)/L_BC     0        0
       -(XDo-XCo)/L_CD    -(YDo-YCo)/L_CD  (XDo-XCo)/L_CD   (YDo-
       YCo)/L_CD
       0      0           -(XE-XDo)/L_DE      -(YE-YDo)/L_DE];

    dl=[a1-atan(abs((XCo-XB)/(YCo-YB)))-pi+Az_B
        a2-atan(abs((XDo-XCo)/(YDo-YCo)))-pi+atan(abs((XB-XCo)/(YB-YCo)))
        a3-2*pi+atan(abs((XE-XDo)/(YE-YDo)))+atan(abs((XCo-XDo)/(YCo-
        YDo)))
        a4-Az_E+pi-atan(abs((XDo-XE)/(YDo-YE)))
        d1-L_BC
        d2-L_CD
        d3-L_DE];

    N=A'*P*A;
    U=A'*P*dl;

    dx=inv(N)*U;
    X_cap=Xo+dx;
    Xo=X_cap;
    Iteration=Iteration+1;
end
 %Variance-Covariance Matrix
Cx_cap=Sigma02*inv(N);
for i=1:u
    sd_Xcap(i)=sqrt(Cx_cap(i,i));
end

v_cap=A*dx-dl;
Sigma02_cap=(v_cap'*P*v_cap)/r;

%Chi-square test
Test_chi=r*Sigma02_cap/Sigma02;
```

```
Chi_square1=chi2inv(Alfa1,r);
Chi_square2=chi2inv(Alfa2,r);
if (Chi_square1<=Test_chi) & (Test_chi<=Chi_square2)
    TestResult='< Test passed >';
else
    TestResult='< Test failed >';
end

Ccap_xcap=Sigma02_cap*inv(N);
for i=1:u
    sd_cap_Xcap(i)=sqrt(Ccap_xcap(i,i));
end

l_cap=lo+v_cap;
C_vcap=Cl-A*inv(A'*inv(Cl)*A)*A';
C_lcap=Cl-C_vcap;
R=C_vcap*inv(Cl);
TracR=trace(R);
for i=1:n
    sd_lcap(i)=sqrt(C_lcap(i,i));
    di(i)=R(i,i);
end

%Print resulte
fprintf('%s %g\n','Degree of freedom (Trace R): ',TracR);
fprintf('\n')
fprintf('%s %g\n','Number of iteration: ',Iteration);
fprintf('\n')
fprintf('%s %0.3f %s\n','Sigma0 cap: ',Sigma02_cap,TestResult);
fprintf('\n')
disp('   X_cap      sd_Xcap  sd_cap_Xcap')
disp('---------------------------------')
for i=1:u
    fprintf('%10.3f%10.3f%10.3f\n',X_cap(i),sd_Xcap(i),sd_cap_Xcap(i));
end
fprintf('\n')
disp('      lo       v_cap    l_cap    sd_lcap     di  ')
disp('-------------------------------------------------')
for i=1:n

fprintf('%10.3f%10.3f%10.3f%10.3f%10.2f\n',lo(i),v_cap(i),l_cap(i),sd_lca
p(i),di(i));
end

%Plot
t=0:0.1:2*pi;
Xp(1)=XB; Yp(1)=YB;
k=2;
for i=1:2:u
    Xp(k)=X_cap(i);
    Yp(k)=X_cap(i+1);

    hold on

    sx2=Ccap_xcap(i,i);
    sy2=Ccap_xcap(i+1,i+1);
    sxy=Ccap_xcap(i,i+1);
    K=sqrt(chi2inv(1-Alfa,2))*MapScale;
```

```
    a=K*sqrt((sx2+sy2+sqrt((sx2-sy2)^2+4*sxy^2))/2);
    b=K*sqrt((sx2+sy2-sqrt((sx2-sy2)^2+4*sxy^2))/2);
    teta=atan(2*sxy/(sx2-sy2))/2;

    u=a*cos(t);
    v=b*sin(t);
    x=Xp(k)+u.*cos(teta)-v.*sin(teta);
    y=Yp(k)+u.*sin(teta)+v.*cos(teta);
    plot(x,y);
    k=k+1;
end
Xp(k)=XE; Yp(k)=YE;
plot(Xp,Yp,'--ro','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',5)
grid on
axis equal;
title('Minimum constraints adjustment program','fontsize',12)
```

نتایج:

```
Degree of freedom (Trace R):  3
Number of iteration:  2
Sigma0 cap:  3.421 < Test passed >
   X_cap      sd_Xcap  sd_cap_Xcap
---------------------------------
  8231.277     0.012     0.023
  2347.835     0.009     0.016
  7982.431     0.013     0.024
  2239.737     0.007     0.014

      lo       v_cap     l_cap    sd_lcap     di
-----------------------------------------------
     3.018     0.000     3.018     0.000     0.69
     3.235    -0.000     3.235     0.000     0.30
     3.638    -0.000     3.638     0.000     0.31
     3.582     0.000     3.582     0.000     0.67
   281.832     0.009   281.841     0.013     0.35
   271.300     0.011   271.311     0.013     0.35
   274.100     0.020   274.120     0.013     0.34
>>
```



ب) حالت Weighted Constraints**: اگر مختصات نقطه** B **و** E **با دقت** 1 cm **داده شده باشد.**

۱) بردار مشاهدات:

$$l = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ d_1 \ d_2 \ d_3]^T$$

$$C_l = diag([\sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_2^2 \ \sigma_2^2 \ \sigma_2^2]^T) \quad ; \quad P = \sigma_0^2 C_l^{-1}$$

۲) بردار مجهولات:

$$x = [x_B \ y_B \ x_C \ y_C \ x_D \ y_D \ x_E \ y_E]^T$$

$$x^o = [x_B^o \ y_B^o \ x_C^o \ y_C^o \ x_D^o \ y_D^o \ x_E^o \ y_E^o]^T$$

۳) مقادیر اولیه:

$$x' = [x_B \ y_B \ 0 \ 0 \ 0 \ 0 \ x_E \ y_E]^T$$

$$P_{x'} = diag\left(\left[\frac{1}{\sigma_{x_B}^2} \ \frac{1}{\sigma_{x_B}^2} \ 0 \ 0 \ 0 \ 0 \ \frac{1}{\sigma_{x_E}^2} \ \frac{1}{\sigma_{x_E}^2}\right]^T\right)$$

۴) درجه آزادی:

$$n = 7 \quad ; \quad u = 8 \quad ; \quad Constraint = 4 \quad ; \quad Defect = 0$$

$$r = n - u + Constraint + Defect = 3$$

۵) مدل ریاضی (روش پارامتریک):

$$l_{n\times 1} = f(x_{u\times 1})$$

جواب مساله:

$$A\delta\hat{x} = \hat{v} + \delta l \quad ; \quad A = \frac{\partial f}{\partial x} \quad ; \quad \delta l = l^o - f(x^o)$$

$$\delta\hat{x} = \hat{v}_{x'} - f_x \quad ; \quad f_x = x^o - x'$$

$$\begin{bmatrix} A \\ I \end{bmatrix}\delta\hat{x} = \begin{bmatrix} \hat{v} \\ \hat{v}_{x'} \end{bmatrix} + \begin{bmatrix} \delta l \\ -f_x \end{bmatrix} \quad \Longrightarrow \quad \mathbb{A}\delta\hat{x} = \hat{\mathbb{v}} + \delta\mathbb{l}$$

$$\mathbb{A} = \begin{bmatrix} A_{n\times u} \\ I_{u\times u} \end{bmatrix} \quad ; \quad \mathbb{P} = \begin{bmatrix} P & 0 \\ 0 & P_{x'} \end{bmatrix} \quad ; \quad \hat{\mathbb{v}} = \begin{bmatrix} \hat{v}_{n\times 1} \\ \hat{v}_{x'_{u\times 1}} \end{bmatrix}$$

$$\mathbb{N} = A^T PA + P_{x'} \quad ; \quad \mathbb{U} = A^T P\delta l - P_{x'} f_x \quad ; \quad \delta\hat{x} = \mathbb{N}^{-1}\mathbb{U}$$

$$\hat{v} = A\delta\hat{x} - \delta l \quad ; \quad \hat{v}_{x'} = \delta\hat{x} + f_x$$

$$\hat{\sigma}_0^2 = \frac{\hat{\mathbb{v}}^T\mathbb{P}\hat{\mathbb{v}}}{r} = \frac{\hat{v}^T P\hat{v} + \hat{v}_{x'}^T P_{x'}\hat{v}_{x'}}{r}$$

$$C_{\hat{x}} = \sigma_0^2(A^T PA + P_{x'})^{-1}N\,(A^T PA + P_{x'})^{-1} = \sigma_0^2\mathbb{N}^{-1}\,N\mathbb{N}^{-1} \quad ; \quad \hat{C}_{\hat{X}} = \hat{\sigma}_0^2\mathbb{N}^{-1}\,N\mathbb{N}^{-1}$$

$$C_{\hat{l}} = \sigma_0^2\mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T \quad ; \quad \hat{C}_{\hat{l}} = \hat{\sigma}_0^2\mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T$$

$$C_{\hat{\mathbb{v}}} = \sigma_0^2(\mathbb{P}^{-1} - \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T) \quad ; \quad \hat{C}_{\hat{\mathbb{v}}} = \hat{\sigma}_0^2(\mathbb{P}^{-1} - \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T)$$

ماتریس آزادی:

$$R_1 = C_{\hat{v}}C_l^{-1} \quad ; \quad R_2 = C_{\hat{v}_{x'}}P_{x'}$$

اعداد آزادی:

$$Trace(R) = Trace(R_1) + Trace(R_2) = \sum_{i=1}^{n+u} d_i$$

برنامه مربوط به حالت Weighted Constraints:

```
clc
clear
format short g

%Weighted constrain adjustment
%Constant value
MapScale=500;
D2R=pi/180;
Epsilon=0.01;
Sigma02=1;
n=7;
u=8;
Defect=0;
Constrain=4;

%Degree of freedom
r=n-u+Defect+Constrain;

%Confidence level
Alfa=0.01;
Alfa1=Alfa/2;
Alfa2=1-Alfa1;

%Initial Coordinates and Azimuth
XB=8478.139; YB=2483.826;
XE=7709.336; YE=2263.411;

Az_B=(68+15/60+20.7/3600)*D2R;
Az_E=(300+11/60+30.5/3600)*D2R;

X=[XB;YB;0;0;0;0;XE;YE];

%Standard diviations of Initial Coordinates
sd_xy=0.01;

Px=diag([1/0.01^2 1/0.01^2 0 0 0 0 1/0.01^2 1/0.01^2]);

%Observations
a1=(172+53/60+34/3600)*D2R;
a2=(185+22/60+14/3600)*D2R;
a3=(208+26/60+19/3600)*D2R;
a4=(205+13/60+51/3600)*D2R;
d1=281.832;
d2=271.300;
d3=274.100;

lo=[a1;a2;a3;a4;d1;d2;d3];

%Standard diviations of observations
sd_a=(10/3600)*D2R;
sd_d=0.016;

Cl=diag([sd_a^2 sd_a^2 sd_a^2 sd_a^2 sd_d^2 sd_d^2 sd_d^2]);
P=Sigma02*inv(Cl);
```

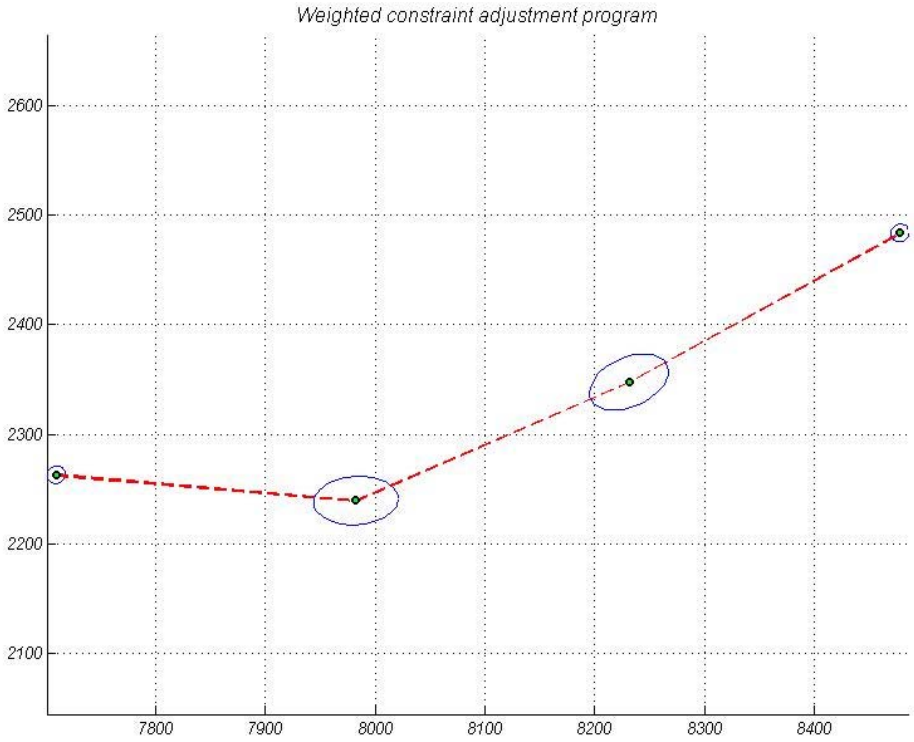
```
%Unknown parameters
%X=[XB,YB,XC,YC,XD,YD,XE,YE]'

%Approximate value of coordinates
Az_BC=Az_B+a1;
XCo=XB+d1*sin(Az_BC);
YCo=YB+d1*cos(Az_BC);

Az_CB=Az_BC-pi;
Az_CD=Az_CB+a2;
XDo=XCo+d2*sin(Az_CD);
YDo=YCo+d2*cos(Az_CD);

Xo=[XB,YB,XCo,YCo,XDo,YDo,XE,YE]';

fx=Xo-X;
%Calculations
dx=Epsilon;
Iteration=0;
while dx>=Epsilon
    XBo=Xo(1); YBo=Xo(2); XCo=Xo(3); YCo=Xo(4); XDo=Xo(5);
YDo=Xo(6);XEo=Xo(7); YEo=Xo(8);

    L_BC=sqrt((XCo-XBo)^2+(YCo-YBo)^2);
    L_CD=sqrt((XDo-XCo)^2+(YDo-YCo)^2);
    L_DE=sqrt((XEo-XDo)^2+(YEo-YDo)^2);

    A=[-(YCo-YBo)/L_BC^2 (XCo-XBo)/L_BC^2 (YCo-YBo)/L_BC^2
       -(XCo-XBo)/L_BC^2                  0
       0                                  0                  0
        -(YBo-YCo)/L_BC^2 (XBo-XCo)/L_BC^2 (YBo-YCo)/L_BC^2-(YDo-
       YCo)/L_CD^2 -(XBo-XCo)/L_BC^2+(XDo-XCo)/L_CD^2 (YDo-YCo)/L_CD^2
       -(XDo-XCo)/L_CD^2                  0                  0
        0                  0              -(YCo-YDo)/L_CD^2
       (XCo-XDo)/L_CD^2                   (YCo-YDo)/L_CD^2-(YEo-
       YDo)/L_DE^2 -(XCo-XDo)/L_CD^2+(XEo-XDo)/L_DE^2 (YEo-YDo)/L_DE^2
       -(XEo-XDo)/L_DE^2
        0                  0              0
       0                                   (YDo-YEo)/L_DE^2
       -(XDo-XEo)/L_DE^2                    -(YDo-YEo)/L_DE^2  (XDo-
       XEo)/L_DE^2
        -(XCo-XBo)/L_BC   -(YCo-YBo)/L_BC  (XCo-XBo)/L_BC
       (YCo-YBo)/L_BC                       0
       0                                   0                  0
        0                  0              -(XDo-XCo)/L_CD
       -(YDo-YCo)/L_CD                     (XDo-XCo)/L_CD
       (YDo-YCo)/L_CD                      0                  0
        0                  0              0
       0                                    -(XEo-XDo)/L_DE
       -(YEo-YDo)/L_DE                      (XEo-XDo)/L_DE     (YEo-
       YDo)/L_DE];

   dl=[a1-atan(abs((XCo-XB)/(YCo-YB)))-pi+Az_B
        a2-atan(abs((XDo-XCo)/(YDo-YCo)))-pi+atan(abs((XB-XCo)/(YB-YCo)))
        a3-2*pi+atan(abs((XE-XDo)/(YE-YDo)))+atan(abs((XCo-XDo)/(YCo-
YDo)))
        a4-Az_E+pi-atan(abs((XDo-XE)/(YDo-YE)))
        d1-L_BC
```

```
        d2-L_CD
        d3-L_DE];

    N=A'*P*A+Px;
    U=A'*P*dl-Px*fx;

    dx=inv(N)*U;
    X_cap=Xo+dx;
    Xo=X_cap;
    fx=Xo-X;
    Iteration=Iteration+1;
end

%Variance-Covariance Matrix
N1=A'*P*A;
Cx_cap=Sigma02*inv(N)*N1*inv(N);
for i=1:u
    sd_Xcap(i)=sqrt(Cx_cap(i,i));
end

v_cap=A*dx-dl;
v_cap_x=dx+fx;
Sigma02_cap=(v_cap'*P*v_cap+v_cap_x'*Px*v_cap_x)/r;

Test_chi=r*Sigma02_cap/Sigma02;
Chi_square1=chi2inv(Alfa1,r);
Chi_square2=chi2inv(Alfa2,r);
if (Test_chi>=Chi_square1)&(Test_chi<=Chi_square2)
    TestResult='< Test passed >';
else
    TestResult='< Test failed >';
end

Ccap_xcap=Sigma02_cap*inv(N)*N1*inv(N);
for i=1:u
    sd_cap_Xcap(i)=sqrt(Ccap_xcap(i,i));
end

l_cap=lo+v_cap;
A2=[A;eye(u)];

C_lcap=Sigma02*A2*inv(N)*A2';

C_lcap1=C_lcap(1:n,1:n);
C_vcap1=Cl-C_lcap1;

Clx=diag([0.01^2 0.01^2 0 0 0 0 0.01^2 0.01^2]);
C_lcap2=C_lcap(n+1:n+u,n+1:n+u);
C_vcap2=Clx-C_lcap2;

R1=C_vcap1*inv(Cl);
R2=C_vcap2*Px;

TracR1=trace(R1);
TracR2=trace(R2);
TracR=TracR1+TracR2;
for i=1:n
    sd_lcap(i)=sqrt(C_lcap(i,i));
```

```
    di(i)=R1(i,i);
end

%Print resulte
fprintf('%s %g\n','Degree of freedom (Trace R): ',TracR);
fprintf('\n')
fprintf('%s %g\n','Number of iteration: ',Iteration);
fprintf('\n')
fprintf('%s %0.3f %s\n','Sigma0 cap: ',Sigma02_cap,TestResult);
fprintf('\n')
disp('   X_cap      sd_Xcap  sd_cap_Xcap')
disp('-----------------------------------')
for i=1:u
    fprintf('%10.3f%10.3f%10.3f\n',X_cap(i),sd_Xcap(i),sd_cap_Xcap(i));
end
fprintf('\n')
disp('      lo       v_cap     l_cap    sd_lcap     di  ')
disp('-------------------------------------------------')
for i=1:n

fprintf('%10.3f%10.3f%10.3f%10.3f%10.2f\n',lo(i),v_cap(i),l_cap(i),sd_lca
p(i),di(i));
end

%Plot
t=0:0.1:2*pi;
k=1;
for i=1:2:u
    Xp(k)=X_cap(i);
    Yp(k)=X_cap(i+1);

    hold on

    sx2=Ccap_xcap(i,i);
    sy2=Ccap_xcap(i+1,i+1);
    sxy=Ccap_xcap(i,i+1);
    K=sqrt(chi2inv(1-Alfa,2))*MapScale;
    a=K*sqrt((sx2+sy2+sqrt((sx2-sy2)^2+4*sxy^2))/2);
    b=K*sqrt((sx2+sy2-sqrt((sx2-sy2)^2+4*sxy^2))/2);
    teta=atan(2*sxy/(sx2-sy2))/2;

    u=a*cos(t);
    v=b*sin(t);
    x=Xp(k)+u.*cos(teta)-v.*sin(teta);
    y=Yp(k)+u.*sin(teta)+v.*cos(teta);
    plot(x,y);
    k=k+1;
end
plot(Xp,Yp,'--ro','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',5)
grid on
axis equal;
title('Weighted constrain adjustment program','fontsize',12)
```

نتایج:

```
Degree of freedom (Trace R):  3
Number of iteration:  1
Sigma0 cap:  3.714 < Test passed >
   X_cap      sd_Xcap  sd_cap_Xcap
---------------------------------
  8478.133     0.003     0.006
  2483.831     0.003     0.005
  8231.275     0.012     0.024
  2347.837     0.009     0.017
  7982.433     0.013     0.025
  2239.735     0.008     0.015
  7709.342     0.003     0.006
  2263.406     0.003     0.005

      lo       v_cap     l_cap    sd_lcap     di
-----------------------------------------------
     3.018     0.000     3.018     0.000      0.61
     3.235    -0.000     3.235     0.000      0.29
     3.638    -0.000     3.638     0.000      0.30
     3.582     0.000     3.582     0.000      0.59
   281.832     0.007   281.839     0.014      0.27
   271.300     0.009   271.309     0.014      0.27
   274.100     0.015   274.115     0.014      0.27
>>
```



ج) حالت Inner Constraints: اگر در شبکه نقطه ثابتی وجود نداشته باشد.

۱) بردار مشاهدات:

$$l = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ d_1 \ d_2 \ d_3]^T$$

$$C_l = diag([\sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_2^2 \ \sigma_2^2 \ \sigma_2^2]^T) \quad ; \quad P = \sigma_0^2 C_l^{-1}$$

۲) بردار مجهولات:

$$x = [x_B \ y_B \ x_C \ y_C \ x_D \ y_D \ x_E \ y_E]^T$$

$$x^o = [x_B^o \ y_B^o \ x_C^o \ y_C^o \ x_D^o \ y_D^o \ x_E^o \ y_E^o]^T$$

۳) ماتریس اینر:

$$C = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

۴) درجه آزادی:

$$n = 7 \quad ; \quad u = 8 \quad ; \quad Constraint = 0 \quad ; \quad Defect = 2$$

$$r = n - u + Constraint + Defect = 1$$

۵) مدل ریاضی (روش پارامتریک):

$$l_{n\times 1} = f(x_{u\times 1})$$

جواب مساله:

$$A\delta\hat{x} = \hat{v} + \delta l \quad ; \quad A = \frac{\partial f}{\partial x} \quad ; \quad \delta l = l^o - f(x^o)$$

$$C\delta\hat{x} = 0$$

$$\begin{bmatrix} A \\ C \end{bmatrix}\delta\hat{x} = \hat{v} + \delta l \quad \Longrightarrow \quad \mathbb{A}\delta\hat{x} = \hat{v} + \delta l$$

$$\mathbb{A} = \begin{bmatrix} A_{n\times u} \\ C_{2\times u} \end{bmatrix} \quad ; \quad \mathbb{P} = \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix}$$

$$\mathbb{N} = A^T PA + C^T C \quad ; \quad U = A^T P\delta l \quad ; \quad \delta\hat{x} = \mathbb{N}^{-1}U$$

$$\hat{v} = A\delta\hat{x} - \delta l$$

$$\hat{\sigma}_0^2 = \frac{\hat{v}^T P\hat{v}}{r}$$

$$C_{\hat{x}} = \sigma_0^2(A^T PA + C^T C)^{-1}N\,(A^T PA + C^T C)^{-1} = \sigma_0^2\mathbb{N}^{-1}\,N\mathbb{N}^{-1}$$

$$\hat{C}_{\hat{x}} = \hat{\sigma}_0^2\mathbb{N}^{-1}\,N\mathbb{N}^{-1}$$

$$C_{\hat{l}} = \sigma_0^2\mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T \quad ; \quad \hat{C}_{\hat{l}} = \hat{\sigma}_0^2\mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T$$

$$C_{\hat{v}} = \sigma_0^2(\mathbb{P}^{-1} - \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T) \quad ; \quad \hat{C}_{\hat{v}} = \hat{\sigma}_0^2(\mathbb{P}^{-1} - \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T)$$

ماتریس آزادی:

$$C_{\hat{v}} = C_l - A(A^T PA + C^T C)^{-1}A^T \quad ; \quad R = C_{\hat{v}}C_l^{-1}$$

اعداد آزادی:

$$Trace(R) = \sum_{i=1}^{n} d_i$$

برنامه مربوط به حالت Inner Constraints:

```
clc
clear
format short g

%Inner constrain adjustment
%Constant value
MapScale=500;
D2R=pi/180;
Epsilon=0.05;
Sigma02=1;
n=7;
u=8;
Defect=2;  %Only without fixed coordinates
Constrain=0;

%Degree of freedom
r=n-u+Defect+Constrain;

%Confidence level
Alfa=0.01;
Alfa1=Alfa/2;
Alfa2=1-Alfa1;

%Approximate Coordinates and Azimuth
XB=8478.139; YB=2483.826;
XE=7709.336; YE=2263.411;

Az_B=(68+15/60+20.7/3600)*D2R;
Az_E=(300+11/60+30.5/3600)*D2R;

%Observations
a1=(172+53/60+34/3600)*D2R;
a2=(185+22/60+14/3600)*D2R;
a3=(208+26/60+19/3600)*D2R;
a4=(205+13/60+51/3600)*D2R;
d1=281.832;
d2=271.300;
d3=274.100;

lo=[a1;a2;a3;a4;d1;d2;d3];

%Standard diviations of observations
sd_a=(10/3600)*D2R;
sd_d=0.016;

Cl=diag([sd_a^2 sd_a^2 sd_a^2 sd_a^2 sd_d^2 sd_d^2 sd_d^2]);
P=Sigma02*inv(Cl);

%Unknown parameters
%X=[XB,YB,XC,YC,XD,YD,XE,YE]'

%Approximate value of coordinates
Az_BC=Az_B+a1;
XCo=XB+d1*sin(Az_BC);
YCo=YB+d1*cos(Az_BC);
```

```
Az_CB=Az_BC-pi;
Az_CD=Az_CB+a2;
XDo=XCo+d2*sin(Az_CD);
YDo=YCo+d2*cos(Az_CD);

Xo=[XB,YB,XCo,YCo,XDo,YDo,XE,YE]';

%Calculations
dx=Epsilon;
Iteration=0;
while norm(dx)>=Epsilon
    XBo=Xo(1); YBo=Xo(2); XCo=Xo(3); YCo=Xo(4); XDo=Xo(5);
YDo=Xo(6);XEo=Xo(7); YEo=Xo(8);

    L_BC=sqrt((XCo-XBo)^2+(YCo-YBo)^2);
    L_CD=sqrt((XDo-XCo)^2+(YDo-YCo)^2);
    L_DE=sqrt((XEo-XDo)^2+(YEo-YDo)^2);

    A=[-(YCo-YBo)/L_BC^2 (XCo-XBo)/L_BC^2 (YCo-YBo)/L_BC^2
     -(XCo-XBo)/L_BC^2                    0
     0                                    0                    0
       -(YBo-YCo)/L_BC^2 (XBo-XCo)/L_BC^2 (YBo-YCo)/L_BC^2-(YDo-
     YCo)/L_CD^2 -(XBo-XCo)/L_BC^2+(XDo-XCo)/L_CD^2 (YDo-YCo)/L_CD^2
     -(XDo-XCo)/L_CD^2                    0                    0
       0                  0                -(YCo-YDo)/L_CD^2
     (XCo-XDo)/L_CD^2                     (YCo-YDo)/L_CD^2-(YEo-YDo)/L_DE^2
     -(XCo-XDo)/L_CD^2+(XEo-XDo)/L_DE^2 (YEo-YDo)/L_DE^2   -(XEo-
     XDo)/L_DE^2
       0                  0                  0
     0                                      (YDo-YEo)/L_DE^2
     -(XDo-XEo)/L_DE^2                     -(YDo-YEo)/L_DE^2  (XDo-
     XEo)/L_DE^2
       -(XCo-XBo)/L_BC   -(YCo-YBo)/L_BC  (XCo-XBo)/L_BC
     (YCo-YBo)/L_BC                       0
     0                                    0                    0
       0                  0                -(XDo-XCo)/L_CD
     -(YDo-YCo)/L_CD                      (XDo-XCo)/L_CD
     (YDo-YCo)/L_CD                       0                    0
       0                  0                0
     0                                    -(XEo-XDo)/L_DE
     -(YEo-YDo)/L_DE                       (XEo-XDo)/L_DE     (YEo-
     YDo)/L_DE];

   dl=[a1-atan(abs((XCo-XB)/(YCo-YB)))-pi+Az_B
       a2-atan(abs((XDo-XCo)/(YDo-YCo)))-pi+atan(abs((XB-XCo)/(YB-YCo)))
       a3-2*pi+atan(abs((XE-XDo)/(YE-YDo)))+atan(abs((XCo-XDo)/(YCo-
      YDo)))
       a4-Az_E+pi-atan(abs((XDo-XE)/(YDo-YE)))
       d1-L_BC
       d2-L_CD
       d3-L_DE];

    C=[1 0 1 0 1 0 1 0
       0 1 0 1 0 1 0 1];
    N=A'*P*A;
    U=A'*P*dl;
```

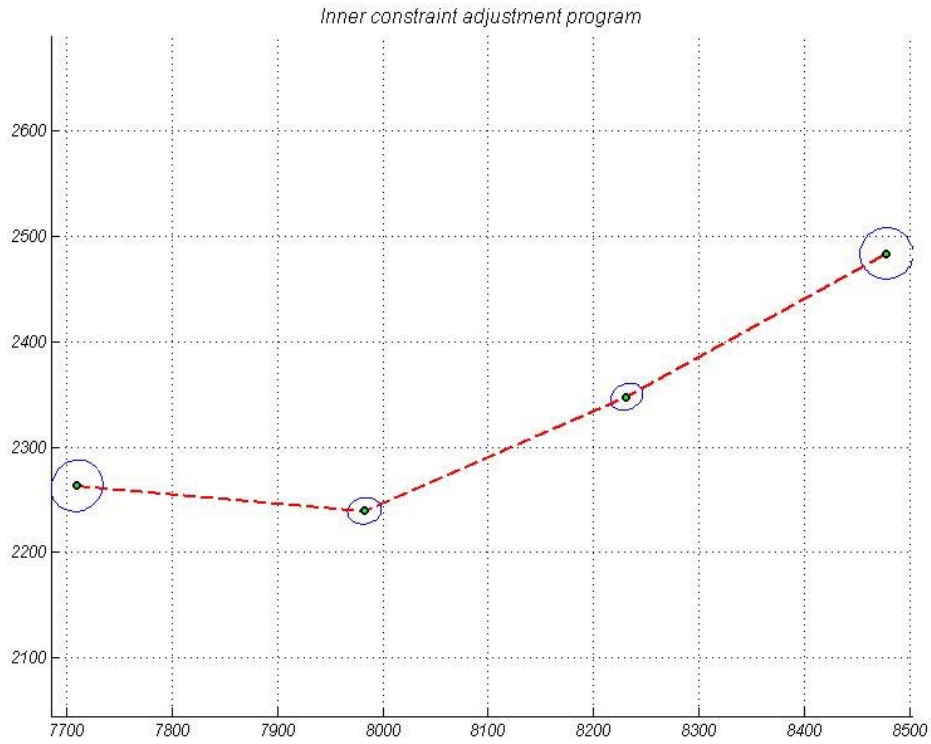
```
    dx=inv(N+C'*C)*U;
    X_cap=Xo+dx;
    Xo=X_cap;
    Iteration=Iteration+1;
end

%Variance-Covariance Matrix
Cx_cap=Sigma02*inv(N+C'*C)*N*(inv(N+C'*C))';
for i=1:u
    sd_Xcap(i)=sqrt(Cx_cap(i,i));
end

v_cap=A*dx-dl;
Sigma02_cap=(v_cap'*P*v_cap)/r;

Test_chi=r*Sigma02_cap/Sigma02;
Chi_square1=chi2inv(Alfa1,r);
Chi_square2=chi2inv(Alfa2,r);
if (Chi_square1<=Test_chi)&(Test_chi<=Chi_square2)
    TestResult='< Test passed >';
else
    TestResult='< Test failed >';
end

Ccap_xcap=Sigma02_cap*inv(N+C'*C)*N*(inv(N+C'*C))';
for i=1:u
    sd_cap_Xcap(i)=sqrt(Ccap_xcap(i,i));
end

l_cap=lo+v_cap;
C_vcap=Cl-A*inv(A'*inv(Cl)*A+C'*C)*A';
C_lcap=Cl-C_vcap;
R=C_vcap*inv(Cl);
TracR=trace(R);
for i=1:n
    sd_lcap(i)=sqrt(C_lcap(i,i));
    di(i)=R(i,i);
end

%Print resulte
fprintf('%s %g\n','Degree of freedom (Trace R): ',TracR);
fprintf('\n')
fprintf('%s %g\n','Number of iteration: ',Iteration);
fprintf('\n')
fprintf('%s %0.3f %s\n','Sigma0 cap: ',Sigma02_cap,TestResult);
fprintf('\n')
disp('    X_cap      sd_Xcap  sd_cap_Xcap')
disp('-----------------------------------')
for i=1:u
    fprintf('%10.3f%10.3f%10.3f\n',X_cap(i),sd_Xcap(i),sd_cap_Xcap(i));
end
fprintf('\n')
disp('      lo       v_cap     l_cap    sd_lcap     di  ')
disp('-------------------------------------------------')
for i=1:n

fprintf('%10.3f%10.3f%10.3f%10.3f%10.2f\n',lo(i),v_cap(i),l_cap(i),sd_lca
p(i),di(i));
```

```
end

%Plot
t=0:0.1:2*pi;
k=1;
for i=1:2:u
    Xp(k)=X_cap(i);
    Yp(k)=X_cap(i+1);

    hold on

    sx2=Ccap_xcap(i,i);
    sy2=Ccap_xcap(i+1,i+1);
    sxy=Ccap_xcap(i,i+1);
    K=sqrt(chi2inv(1-Alfa,2))*MapScale;
    a=K*sqrt((sx2+sy2+sqrt((sx2-sy2)^2+4*sxy^2))/2);
    b=K*sqrt((sx2+sy2-sqrt((sx2-sy2)^2+4*sxy^2))/2);
    teta=atan(2*sxy/(sx2-sy2))/2;

    u=a*cos(t);
    v=b*sin(t);
    x=Xp(k)+u.*cos(teta)-v.*sin(teta);
    y=Yp(k)+u.*sin(teta)+v.*cos(teta);
    plot(x,y);
    k=k+1;
end
plot(Xp,Yp,'--ro','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',5)
grid on
axis equal;
title('Inner constraint adjustment program','fontsize',12)
```

نتایج:

```
Degree of freedom (Trace R):  1

Number of iteration:  2

Sigma0 cap:  1.169 < Test passed >

   X_cap      sd_Xcap  sd_cap_Xcap
---------------------------------
  8478.113     0.015     0.016
  2483.865     0.015     0.017
  8231.281     0.009     0.010
  2347.839     0.008     0.009
  7982.452     0.010     0.011
  2239.728     0.008     0.008
  7709.374     0.015     0.016
  2263.369     0.015     0.016
```

```
      lo       v_cap      l_cap     sd_lcap      di
-------------------------------------------------------
     3.018     -0.000      3.018      0.000      0.25
     3.235     -0.000      3.235      0.000      0.25
     3.638     -0.000      3.638      0.000      0.25
     3.582      0.000      3.582      0.000      0.25
   281.832     -0.000    281.832      0.016      0.00
   271.300     -0.000    271.300      0.016     -0.00
   274.100      0.000    274.100      0.016     -0.00
>>
```



د) حالت قیود تابعی (Functional Constraints)**: اگر مختصات نقاط** C **بر روی دایره‌ای به شعاع** 8559.5 m **قرار داشته باشد.**

ا) بردار مشاهدات:

$$l = [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ d_1 \ d_2 \ d_3]^T$$

$$C_l = diag([\sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_1^2 \ \sigma_2^2 \ \sigma_2^2 \ \sigma_2^2]^T) \quad ; \quad P = \sigma_0^2 C_l^{-1}$$

۲) بردار مجهولات:

$$x = [x_C \;\; y_C \;\; x_D \;\; y_D]^T$$

$$x^o = [x_C^o \;\; y_C^o \;\; x_D^o \;\; y_D^o]^T$$

۲) معادله قید تابعی:

$$x_C^2 + y_C^2 - 8559.5^2 = 0$$

۳) درجه آزادی:

$$n = 7 \quad ; \quad u = 4 \quad ; \quad Constraint = 1 \quad ; \quad Defect = 0$$

$$r = n - u + Constraint + Defect = 4$$

۴) مدل ریاضی (روش پارامتریک):

$$l_{n\times 1} = f(x_{u\times 1})$$

جواب مساله:

$$\delta\hat{x} = \hat{v} + \delta l \quad ; \quad A = \frac{\partial f}{\partial x} \quad ; \quad \delta l = l^o - f(x^o)$$

$$N = A^T P A \quad ; \quad U = A^T P \delta l \quad ; \quad \delta\hat{x}^{old} = N^{-1}U$$

$$A_c \delta\hat{x} + W_c = 0$$

$$\delta\hat{x} = \delta\hat{x}^{old} + \Delta\delta\hat{x}$$

$$\Delta\delta\hat{x} = -N^{-1}{A_c}^T K_c$$

$$K_c = \left({A_c}^T N^{-1} A_c\right)^{-1}(W_c + A_c\delta\hat{x}^{old})$$

روش حل دیگر:

$$\begin{bmatrix} A \\ A_c \end{bmatrix}\delta\hat{x} = \hat{v} + \begin{bmatrix} \delta l \\ -W_c \end{bmatrix} \quad \Longrightarrow \quad \mathbb{A}\delta\hat{x} = \hat{v} + \delta\mathbb{l}$$

$$\mathbb{A} = \begin{bmatrix} A_{n\times u} \\ A_{c_{1\times u}} \end{bmatrix} \quad ; \quad \mathbb{P} = \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix}$$

$$\mathbb{N} = A^T PA + {A_c}^T A_c \quad ; \quad \mathbb{U} = A^T P\delta l - {A_c}^T W_c \quad ; \quad \delta\hat{x} = \mathbb{N}^{-1}\mathbb{U}$$

$$\hat{v} = A\delta\hat{x} - \delta l$$

$$\hat{\sigma}_0^2 = \frac{\hat{v}^T P\hat{v}}{r}$$

$$C_{\hat{x}} = \sigma_0^2\left(A^T PA + {A_c}^T A_c\right)^{-1} N \left(A^T PA + {A_c}^T A_c\right)^{-1} = \sigma_0^2 \mathbb{N}^{-1} N\mathbb{N}^{-1}$$

$$\hat{C}_{\hat{x}} = \hat{\sigma}_0^2 \mathbb{N}^{-1} N\mathbb{N}^{-1}$$

$$C_{\hat{l}} = \sigma_0^2 \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T \quad ; \quad \hat{C}_{\hat{l}} = \hat{\sigma}_0^2 \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T$$

$$C_{\hat{v}} = \sigma_0^2(\mathbb{P}^{-1} - \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T) \quad ; \quad \hat{C}_{\hat{v}} = \hat{\sigma}_0^2(\mathbb{P}^{-1} - \mathbb{A}\mathbb{N}^{-1}\mathbb{A}^T)$$

ماتریس آزادی:

$$C_{\hat{v}} = C_l - A\left(A^T PA + {A_c}^T A_c\right)^{-1} A^T \quad ; \quad R = C_{\hat{v}} C_l^{-1}$$

اعداد آزادی:

$$Trace(R) = \sum_{i=1}^{n} d_i$$

برنامه مربوط به حالت Functional Constraints:

```
clc
clear
format short g

%Minimum constrain adjustment program
%Constant value
MapScale=500;
D2R=pi/180;
Epsilon=0.01;
Sigma02=1;
n=7;
u=4;
Defect=0;
Constrain=1;
Radius=8559.5;

%Degree of freedom
r=n-u+Defect+Constrain;

%Confidence level
Alfa=0.01;
```

```
Alfa1=Alfa/2;
Alfa2=1-Alfa1;

%Fixed Coordinates and Azimuth
XB=8478.139; YB=2483.826;
XE=7709.336; YE=2263.411;

Az_B=(68+15/60+20.7/3600)*D2R;
Az_E=(300+11/60+30.5/3600)*D2R;

%Observations
a1=(172+53/60+34/3600)*D2R;
a2=(185+22/60+14/3600)*D2R;
a3=(208+26/60+19/3600)*D2R;
a4=(205+13/60+51/3600)*D2R;
d1=281.832;
d2=271.300;
d3=274.100;

lo=[a1;a2;a3;a4;d1;d2;d3];

%Standard diviations of observations
sd_a=(10/3600)*D2R;
sd_d=0.016;

Cl=diag([sd_a^2 sd_a^2 sd_a^2 sd_a^2 sd_d^2 sd_d^2 sd_d^2]);
P=Sigma02*inv(Cl);

%Unknown parameters
%X=[XC,YC,XD,YD]'

%Approximate value of coordinates
Az_BC=Az_B+a1;
XCo=XB+d1*sin(Az_BC);
YCo=YB+d1*cos(Az_BC);

Az_CB=Az_BC-pi;
Az_CD=Az_CB+a2;
XDo=XCo+d2*sin(Az_CD);
YDo=YCo+d2*cos(Az_CD);

Xo=[XCo,YCo,XDo,YDo]';

%Calculations
dx=Epsilon;
Iteration=0;
while norm(dx)>=Epsilon
    XCo=Xo(1); YCo=Xo(2); XDo=Xo(3); YDo=Xo(4);

    L_BC=sqrt((XCo-XB)^2+(YCo-YB)^2);
    L_CD=sqrt((XDo-XCo)^2+(YDo-YCo)^2);
    L_DE=sqrt((XE-XDo)^2+(YE-YDo)^2);

    A=[(YCo-YB)/L_BC^2                    -(XCo-XB)/L_BC^2
0                              0
       (YB-YCo)/L_BC^2-(YDo-YCo)/L_CD^2  -(XB-XCo)/L_BC^2+(XDo-
XCo)/L_CD^2 (YDo-YCo)/L_CD^2                    -(XDo-XCo)/L_CD^2
```

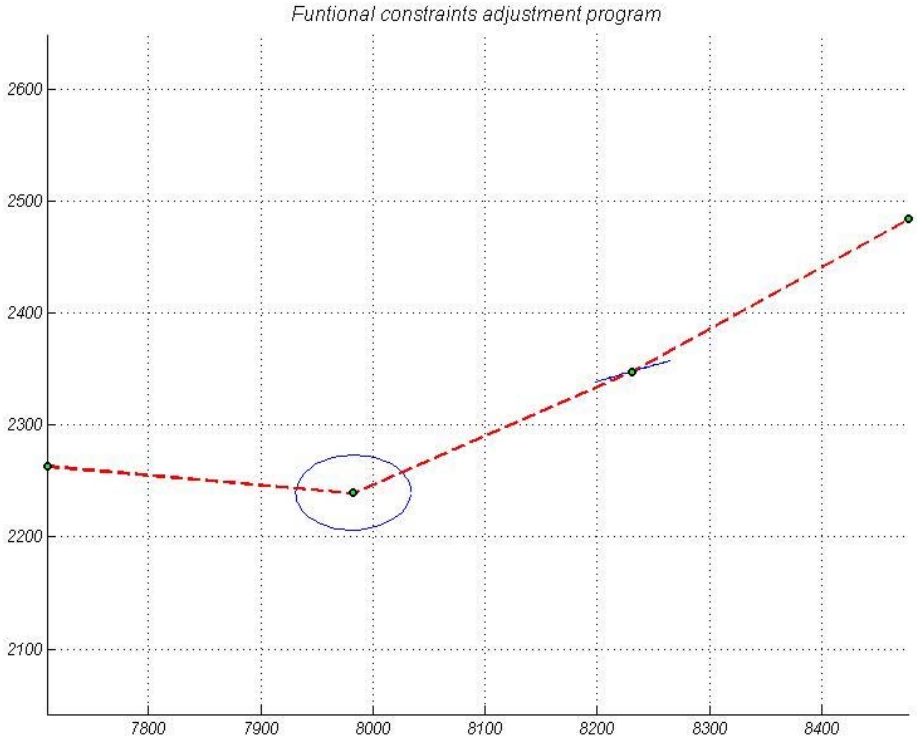
```
       -(YCo-YDo)/L_CD^2                    (XCo-XDo)/L_CD^2
(YCo-YDo)/L_CD^2-(YE-YDo)/L_DE^2   -(XCo-XDo)/L_CD^2+(XE-XDo)/L_DE^2
       0                                    0
(YDo-YE)/L_DE^2                     -(XDo-XE)/L_DE^2
       (XCo-XB)/L_BC                        (YCo-YB)/L_BC
0                                   0
       -(XDo-XCo)/L_CD                      -(YDo-YCo)/L_CD
(XDo-XCo)/L_CD                      (YDo-YCo)/L_CD
       0                                    0
-(XE-XDo)/L_DE                      -(YE-YDo)/L_DE];
    dl=[a1-atan(abs((XCo-XB)/(YCo-YB)))-pi+Az_B
        a2-atan(abs((XDo-XCo)/(YDo-YCo)))-pi+atan(abs((XB-XCo)/(YB-YCo)))
        a3-2*pi+atan(abs((XE-XDo)/(YE-YDo)))+atan(abs((XCo-XDo)/(YCo-
YDo)))
        a4-Az_E+pi-atan(abs((XDo-XE)/(YDo-YE)))
        d1-L_BC
        d2-L_CD
        d3-L_DE];

    N=A'*P*A;
    U=A'*P*dl;

    dx_old=inv(N)*U;
    %Functional constraint
    Ac=[2*XCo 2*YCo 0 0];
    Wc=[XCo^2+YCo^2-Radius^2];
    Kc=inv(Ac*inv(N)*Ac')*(Wc+Ac*dx_old);
    Ddx=-inv(N)*Ac'*Kc;

    dx=dx_old+Ddx;

    X_cap=Xo+dx;
    Xo=X_cap;
    Iteration=Iteration+1;
end

%Variance-Covariance Matrix
Cx_cap=Sigma02*inv(N+Ac'*Ac)*N*inv(N+Ac'*Ac);
for i=1:u
    sd_Xcap(i)=sqrt(Cx_cap(i,i));
end

v_cap=A*dx-dl;
Sigma02_cap=(v_cap'*P*v_cap)/r;

%Chi-square test
Test_chi=r*Sigma02_cap/Sigma02;
Chi_square1=chi2inv(Alfa1,r);
Chi_square2=chi2inv(Alfa2,r);
if (Chi_square1<=Test_chi) & (Test_chi<=Chi_square2)
    TestResult='< Test passed >';
else
    TestResult='< Test failed >';
end

Ccap_xcap=Sigma02_cap*inv(N+Ac'*Ac)*N*inv(N+Ac'*Ac);
for i=1:u
    sd_cap_Xcap(i)=sqrt(Ccap_xcap(i,i));
```

```
end

l_cap=lo+v_cap;
C_vcap=Cl-A*inv(A'*inv(Cl)*A+Ac'*Ac)*A';
C_lcap=Cl-C_vcap;
R=C_vcap*inv(Cl);
TracR=round(trace(R));
for i=1:n
    sd_lcap(i)=sqrt(C_lcap(i,i));
    di(i)=R(i,i);
end

%Print resulte
fprintf('%s %g\n','Degree of freedom (Trace R): ',TracR);
fprintf('\n')
fprintf('%s %g\n','Number of iteration: ',Iteration);
fprintf('\n')
fprintf('%s %0.3f %s\n','Sigma0 cap: ',Sigma02_cap,TestResult);
fprintf('\n')
disp('   X_cap      sd_Xcap  sd_cap_Xcap')
disp('---------------------------------')
for i=1:u
    fprintf('%10.3f%10.3f%10.3f\n',X_cap(i),sd_Xcap(i),sd_cap_Xcap(i));
end
fprintf('\n')
disp('      lo       v_cap     l_cap    sd_lcap     di  ')
disp('-------------------------------------------------')
for i=1:n

fprintf('%10.3f%10.3f%10.3f%10.3f%10.2f\n',lo(i),v_cap(i),l_cap(i),sd_lca
p(i),di(i));
end

%Plot
t=0:0.1:2*pi;
Xp(1)=XB; Yp(1)=YB;
k=2;
for i=1:2:u
    Xp(k)=X_cap(i);
    Yp(k)=X_cap(i+1);

    hold on

    sx2=Ccap_xcap(i,i);
    sy2=Ccap_xcap(i+1,i+1);
    sxy=Ccap_xcap(i,i+1);
    K=sqrt(chi2inv(1-Alfa,2))*MapScale;
    a=K*sqrt((sx2+sy2+sqrt((sx2-sy2)^2+4*sxy^2))/2);
    b=K*sqrt((sx2+sy2-sqrt((sx2-sy2)^2+4*sxy^2))/2);
    teta=atan(2*sxy/(sx2-sy2))/2;

    u=a*cos(t);
    v=b*sin(t);
    x=Xp(k)+u.*cos(teta)-v.*sin(teta);
    y=Yp(k)+u.*sin(teta)+v.*cos(teta);
    plot(x,y);
    k=k+1;
end
```

```
Xp(k)=XE; Yp(k)=YE;
plot(Xp,Yp,'--ro','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',5)
grid on
axis equal;
title('Funtional constraints adjustment program','fontsize',12)
```

نتایج:

```
Degree of freedom (Trace R):  4
Number of iteration:  2
Sigma0 cap:  8.736 < Test failed >
   X_cap       sd_Xcap   sd_cap_Xcap
---------------------------------
  8231.212      0.002      0.006
  2347.807      0.007      0.022
  7982.395      0.011      0.034
  2239.727      0.007      0.022

      lo        v_cap      l_cap    sd_lcap      di
-------------------------------------------------
     3.018      0.000      3.018      0.000      0.70
     3.235     -0.000      3.235      0.000      0.30
     3.638     -0.000      3.638      0.000      0.31
     3.582      0.000      3.582      0.000      0.68
   281.832      0.080    281.912      0.002      0.99
   271.300     -0.024    271.276      0.011      0.52
   274.100     -0.016    274.084      0.011      0.49
>>
```



تست آماری رد می‌شود که با تغییر دقت طول از 16mm به 35mm نتایج برابر است با:

```
Degree of freedom (Trace R):   4
Number of iteration:  2
Sigma0 cap:  3.355 < Test passed >
   X_cap       sd_Xcap   sd_cap_Xcap
-------------------------------------
  8231.213      0.002      0.004
  2347.803      0.008      0.015
  7982.413      0.023      0.043
  2239.729      0.008      0.014

    lo        v_cap      l_cap     sd_lcap      di
-------------------------------------------------------
     3.018      0.000      3.018      0.000      0.65
     3.235     -0.000      3.235      0.000      0.29
     3.638     -0.000      3.638      0.000      0.30
     3.582      0.000      3.582      0.000      0.63
   281.832      0.081    281.913      0.002      1.00
```

```
   271.300     -0.042    271.258      0.023      0.58
   274.100      0.002    274.102      0.023      0.56
>>
```

مدل جامع سرشکنی (Unified approach and parameter constraints):

$$\begin{cases} F(x,l)=0 \\ F_c(x,l')=0 \\ \hat{x}=x^o+\delta\hat{x}=x+\hat{v}_x \end{cases} \Longrightarrow \begin{cases} A\delta\hat{x}+B\hat{v}+W=0 \\ A_c\delta\hat{x}+B_c\hat{v}_c+W_c=0 \\ \delta\hat{x}-\hat{v}_x+f_x=0 \end{cases}$$

$$\begin{bmatrix} A \\ A_c \\ I \end{bmatrix}\delta\hat{x}+\begin{bmatrix} B & 0 & 0 \\ 0 & B_c & 0 \\ 0 & 0 & -I \end{bmatrix}\begin{bmatrix} \hat{v} \\ \hat{v}_c \\ \hat{v}_x \end{bmatrix}+\begin{bmatrix} W \\ W_c \\ f_x \end{bmatrix}=0$$

$$\mathbb{A}\delta\hat{x}+\mathbb{B}\hat{v}+\mathbb{W}=0$$

$$\delta\hat{x}=-\mathbb{N}^{-1}\mathbb{U}=(N+N_c+P_x)^{-1}(U+U_c+P_xf_x)$$