دانشگاه اصفهان

دانشکده ی فنی و مهندسی

گروه نقشه برداری

پروژه شماره 3 درس محاسبات ژئودزی

سرشکنی یک شبکه ژئودزی

استاد مربوطه : جناب آقای دکتر موسوی الکاظم

تهیه کننده : مهران قندهاری 851921326

m.ghandehary@gmail.com

تیر 88

مقدمه :

در این پروژه هدف سرشکنی یک شبکه ی ژئودزی بر روی بیضوی (در دو بعد) می باشد. مساله ی حل یک شبکه ی ژئودزی با استفاده از کمترین مربعات (که معمولا سرشکنی کمترین مربعات نامیده می شود) در رابطه با ترکیب مشاهدات به بهترین نحو ممکن برای رسیدن به مطمئن ترین مختصات برای نقاط شبکه میباشد.حل دو بعدی در رابطه با مشاهداتی که بر روی سطح بیضوی تصحیح شده اند انجام میگیرد.بنابراین شامل:

1)امتدادها و زوایای ژئودتیک

2)فواصل ژئودتیک

3)آزیموت ژئودتیک

4)مختصات ژئودتیک

اما قبل از سرشکنی لازم است شبکه دارای شکل،موقعیت،مقیاس و جهت باشد.برای اینکه شکل شبکه فراهم شود لازم است موقعیت هر ایستگاه مرتبط و متصل به ایستگاهای دیگر که دارای موقعیت معلوم هستند باشد.برای اینکه موقعیت شبکه ایجاد شود لازم است مختصات یک نقطه مشخص باشد.برای اینکه مقیاس شبکه فراهم گردد بایستی فاصله ی بین دو نقطه مشاهده شود و یا اینکه معلوم باشد.و برای توجیه شبکه لازم است مختصات بیش از یک نقطه معلوم باشد و یا حداقل یک آزیموت مشاهده گردد.

زمینه ی نظری :

مراحل سرشکنی کمترین مربعات (دو بعدی بر روی بیضوی) به صورت زیر میبا شد :

1) محاصبه ی مقادیر تقریبی
2) تعیین انحراف از معیار مشاهدات
3) آزمون های قبل از سرشکنی
   - نرمال بودن مشاهدات
   - آزمون out lier
   - آزمون واریانس دستگاه
4) انتقال مشاهدات به فضای محاصبات(بیضوی یا صفحه ی تصویر)
5) انجام سرشکنی کمترین مربعات با حداقل قید
6) آزمون های بعد از سرشکنی
   - آزمون نرمال بودن باقیمانده های استاندارد شده
   - آزمون کلی (واریانس ثانویه)
   - آزمون موضعی مشاهدات (تست باردا)
7) اضافه کردن همه ی قیود موجود به شبکه و تکرار سرشکنی
8) تکرار مرحله ی 6
   - حذف خطای سیستماتیک
   - کاهش وزن نقاط کنترل
   - حذف نقاط کنترل دارای اشکال
9) بررسی دقت مجهولات

10)ارائه ی نتایج

-مختصات و مشاهدات سرشکن شده

-دقت مختصات و مشاهدات سرشکن شده

- رسم بیضی خطا

-ارائه ی اعداد آزادی

- ارائه ی اعتماد پذیری داخلی وخارجی

- هیستوگرام باقیمانده ها

حال به سایر جزئیات مراحل بالا می پردازیم:

سرشکنی به روش پارامترهای وزن دار:

در بعضی از مسائل سرشکنی تمام یا قسمتی از مجهولات مدل ریاضی ،با دقت مشخص در دست است و ما علاقه مندیم از این اطلاعات به عنوان مشاهدات کمکی در سرشکنی شبکه استفاده کنیم.

فرمول های مورد استفاده در این روش از قرار زیر است :

$$l = f(x) \; --\rightarrow \; A * \delta x + W = V$$

$$l_x = f_x(x) --\rightarrow \; A_x * \delta x + W_x = \mathrm{V_x}$$

$$\delta x = -(A^t * P * A + A_x^t * P_x * A_x)^{-1}(A^t * P * W + A_x^t * P_x)$$

$$\mathrm{X_{HAT}} = x_0 + \delta x$$

$$\sum X_{HAT} = \sigma_0^2 * (A^t * P * A + A_x^t * P_x * A_x)^{-1}$$

$$\sigma_0^2 = (v^t * P * v + v_x^t * P_x * v_x)/df$$

معادلات مشاهدات برای شبکه های مسطحاتی(روی بیضوی) :

معادله ی مشاهده ی طول :

$$v_{sij} = a_{ij} * \mathrm{d\varphi_i} + \mathrm{b_{ij}} * \mathrm{d\lambda_i} + \mathrm{c_{ij}} * \mathrm{d\varphi_j} + \mathrm{d_{ij}} * \mathrm{d\lambda_j} + \mathrm{S_{ij}^0} - \mathrm{S_{ij}}$$

معادله ی مشاهده ی آزیموت :

$$v_{aij} = e_{ij} * \mathrm{d\varphi_i} + \mathrm{f_{ij}} * \mathrm{d\lambda_i} + \mathrm{g_{ij}} * \mathrm{d\varphi_j} + (-\mathrm{f})_{\mathrm{ij}} * \mathrm{d\lambda_j} + \mathrm{a_{ij}^0} - \mathrm{a_{ij}}$$

معادله ی مشاهده ی زاویه :

$$v_{wijk} = \left(e_{jk} - e_{ji}\right) * \mathrm{d\varphi_j} + \left(\mathrm{f_{jk}} - \mathrm{f_{ji}}\right) * \mathrm{d\lambda_j} + \mathrm{g_{ik}} * \mathrm{d\varphi_k} +$$

$$(-\mathrm{f})_{\mathrm{jk}} * \mathrm{d\lambda_k} - \mathrm{g_{ji}} * \mathrm{d\varphi_{i+}} \mathrm{f_{ji}} * \mathrm{d\lambda_i} + \mathrm{w_{ij}^0} - \mathrm{w_{ij}}$$

ضرایب معادلات بالا :

$$a_{ij} = -M_i * \mathrm{cos\alpha_{ij}}$$

$$b_{ij} = N_j * \mathrm{sin\alpha_{ji}} * \mathrm{cos\varphi_j}$$

$$c_{ij} = -M_j * \mathrm{cos\alpha_{ji}}$$

$$d_{ij} = -N_j * \mathrm{sin\alpha_{ji}} * \mathrm{cos\varphi_j}$$

$$e_{ij} = \frac{M_i * \mathrm{sin\alpha_{ij}}}{S_{ij}}$$

$$f_{ij} = \frac{N_j * \mathrm{sin\alpha_{ji}} * \mathrm{cos\varphi_j}}{S_{ij}}$$

$$e_{ij} = \frac{M_j * \mathrm{sin\alpha_{ji}}}{S_{ij}}$$

آزمون آماری جهت کشف اشتباهات :

یکی از آزمون های آماری که بعد از سرشکنی انجام می پذیرد،آزمون فاکتور واریانس ثانویه میباشد :

$$\frac{\mathrm{df} * \sigma_0^2\_hat}{\aleph^2_{\mathrm{df},1-\alpha/2}} < \sigma_0^2 < \frac{\mathrm{df} * \sigma_0^2\_hat}{\aleph^2_{\mathrm{df},\alpha/2}}$$

تابع توزیع خی دو $---- >$ $\aleph$

یکی از علل رد این آزمون وجود مشاهدات لشتباه می باشد.جهت کشف مشاهدات اشتباه میتوان از آزمون باقیمانده های استاندارد شده استفاده نمود:

فرض صفر به صورت زیر است :

$$H_0:\ W_i = \frac{v_i}{\sigma_{vi}}$$

در حالتی که σ_0^2 معلوم نباشد باقیمانده های استاندارد شده دارای تابع توزیع تاو هستند و برای سطح اطمینان (1-α)‰ داریم:

$$-\tau_{df,1-\alpha/2} * \mathrm{\sigma_{vi}} < v_i < \tau_{df,1-\alpha/2} * \mathrm{\sigma_{vi}}$$

تابع توزیع خی تاو $---- >$ τ

ارتباط بین متغیر t و τ به صورت زیر است :

$$\tau_{df} = \frac{df^{1/2} * t_{df-1}}{(df - 1 + t^2_{df-1})^{1/2}}$$

در این حالت به روش زیر عمل می کنیم :

1)مشاهده ای که بزگترین باقیمانده ی استاندارد را دارد از بین مشاهدات حذف می کنیم.

2)مجددا سرشکنی را انجام می دهیم و باقیمانده ها ی استاندارد شده را تست می کنیم.

3)در صورتیکه چند باقیمانده ی استاندارد در تست رد گردند،دوباره مشاهده ای که بزگترین باقیمانده ی استاندارد را دارد را از بین مشاهدات حذف و سرشکنی را تکرار می کنیم.

4)مراحل فوق را به قدری تکرار می کنیم که هیچ باقیمانده ای رد نگردد.

5)اولین مشاهده ی رد شده را وارد لیست ساخته و سرشکنی را انجام می دهیم.

6)در صورتی که وارد ساختن مشاهده ی جدید باعث رد هیچ باقیمانده ای نگردد،آن مشاهده درست خواهد بود.

7)این عمل را برای تک تک مشاهدات حذف شده تکرار می کنیم.

8)هر مشاهده ای که وارد ساختن آن در لیست مشاهدات موجب رد باقیمانده ها گردد، اشتباه بوده و می بایستی رد گردد.

روش هلمرت در برآورد مولفه های ماتریس واریانس :

اگر مشاهدات از یک جنس نباشند باید ماتریس وزنرا تصحیح نمود.فرمول های به کار رفته از قرار زیر است:

معادله ی هلمرت $H * S = C$

$$C_i = V_i^t * P_i * V_i$$

$$S_i = \sigma_i^2$$

$$h_{ii} = n_i - 2 * trace(N^{-1} * N_i) + trace(N^{-1} * N_i * N^{-1} * N_i)$$

$$h_{ij} = trace(N^{-1} * N_j * N^{-1} * N_j)$$

$$N_i = (A_i^t * P_i * A_i)^{-1}$$

$$N = N1 + N2 + \cdots + Nn$$

$$V_i = A_i * \hat{x} - l_i$$

مراحل کار:

1)برآورد وزن های P_i (اولین بار $P_i = I$)

2)محاسبه ی N_i و N

3)سرشکنی و محاسبه ی $\hat{x}$ و V_i

4)تشکیل معادل ی هلمرت و محاسبه ی S (مولفه های واریانس)

5)محاسبه ی وزن های جدید $P_i = \frac{P_i}{S_i}$

6)تکرار محاصبات بالا تا هنگامی که $\mathrm{S_i} = 1$ شود.

مشاهدات و معلومات :

در شبکه نقاط 1001 و 1004 معلوم هستند.کمیت های مشاهده شده پس از انتقال بر روی بیضوی در زیر آورده شده است :

From	To	Azimuth	St.Dev.[sec]
1003	1001	178 15 8.06	±1.0
1003	1004	205 46 47.50	±1.0
1005	1004	175 51 33.33	±1.0

From	To	Distance	St.Dev.[sec]
1001	1002	20204.068	10mm+3ppm
1001	1003	47442.343	10mm+3ppm
1002	1004	29617.434	10mm+3ppm
1003	1002	27454.890	10mm+3ppm
1006	1005	22011.770	10mm+3ppm

From	**Occupied**	To	Distance	St.Dev.[sec]
1002	1001	1004	273 39 55.80	±1
1001	1002	1003	168 56 24.80	±1
1001	1004	1005	257 41 06.40	±1
1003	1006	1005	94 35 13.50	±1
1001	1005	1002	321 32 4.11	±1
1006	1005	1002	97 44 27.96	±1

Point	**Lattitude**	**StDev.[m]**	**Longitude**	**StDev.[m]**
1001	N 30 00 0.00019	±0.005	W 90 00 0.00013	±0.005
1004	N 30 01 47.00550	±0.006	W 90 14 9.00022	±0.006

point	**Approx. Latitude**	**Approx. Longitude**
1005	*N* 3011 47	*W* 90 14 58
1006	*N* 30 23 40	*W* 90 15 57
1002	*N* 30 10 54	*W* 89 58 58
1003	*N* 30 25 40	*W* 9000 54

محاصبات و مراحل کار :

این پروژه با برنامه نویسی در محیط متلب انجام گرفت که برنامه ها و توابع آن در ضمیمه ی گزارش آورده شده است. مراحل و نتایج انجام پروژه از قرار زیر است:

1)با توجه به 14 مشاهده ی موجود (3 آزیموت،5 طول و 6 زاویه)،14 معادله ی مشاهده تشکیل و از روی آن ها ماتریس A(14,12) (12 مجهول) تشکیل شد.

2)با دو نقطه ی ثابت نیز 4 معادله تشکیل و ماتریس Ax(4,12) تشکیل شد.

البته در مرحله فقط از یک نقطه ی ثابت استفاده میکنیم)

3)ماتریس های وزن P(14,14) و Px(4,4) با توجه به انحراف معیار مشاهدات و نقاط معلوم تشکیل شد.

به علت اینکه مقادیر انحراف معیار نقاط معلوم بر حسب متر آورده شده ،مقادیر آن بر شعاع زمین تقسیم گردید.

4)برای محاصبه ی طول و آزیموت بین دو نقطه با مختصات معلوم بر روی بیضوی از فرمول های Robbins استفاده شد.(فرمول ها و تابع آن در ضمیمه آورده شده است.)

در جدول زیر مقادیر مشاهداتی و محاصبا تی مقایسه می گردد:

	Observed	Compute
S12	20204.068(m)	20206.5959506511
S13	47442.343	47442.6977049288
S24	29617.434	29641.3909710709
S32	27454.89	27459.1180052106
S65	22011.77	22012.6856260654
A31	3.11108846766259(rad)	3.11108222555291
A34	3.59153611068553	3.59151042326993
A54	3.06932307803194	3.07065068926187
α214	4.77636402411654	4.77506536201333
α123	2.94856311682867	2.94628807493619
α145	4.49745362225631	4.49877103678294
α365	1.65085603402492	1.65228516487451
α152	5.61183524742139	5.61076325797223
α652	1.70590400952103	1.70470401331821

5)تشکیل ماتریس W= Compute- Observed

(در این مرحله می توان پیش بینی کرد که طول 2 به 4 دارای خطاست.)

6)حال سرشکنی را به روش پارامترهای وزن دار انجام می دهیم.مقدار واریانس فاکتور ثانویه نیز محاصبه میگردد:

$$\widehat{\sigma_0^2} = 34.6920$$

واریانس نقاط مجهول نیز از ماتریس $\hat{C}_{\hat{x}}$ بدست می آید که در بازه ی 10^{-16} تا 10^{-19} می باشد.

(البته ابتدا مقدار واریانس فاکتور ثانویه عدد بزرگی بود که پس از بررسی اصلاحاتی در ماتریس A صورت گرفت.)

7)انجام تست فاکتور وار یانس ثانویه با(α=0.05) که مقدار فاکتور وار یانس در این تست رد شد.

8)تست out lier انجام شد و طول دوم و سوم حذف گردید.

$$\widehat{\sigma_0^2} = 0.1925$$

(روش هلمرت در برآورد مولفه های ماتریس واریانس نیز انجام شد و مقدار فاکتور واریانس کاهش یافت ولی بدون انجام این کار تست فاکتور واریانس قبول شد و در ضمن روش هلمرت باعث کاهش دقت مجهولات برآورد شده شد.)

9)سرشکنی با دو نقطه ی ثابت انجام گرفت و شرط حلقه ی تکرار نیز به صورت زیر بود.زیرا این شرط معادل 1 میلیمتر بر روی زمین است:

$$NORM(\delta_{\hat{x}}) < 10^{-11}$$

نتایج بدست آمده از قرار زیر است:

point	**Ajusted Latitude**	**Ajusted Longitude**
1001	*N* 30 00 0.000218	*W* 90 00 0.000001
1002	*N* 30 10 53.987749	*W* 89 58 58.987710
1003	*N* 30 25 39.949019	*W* 90 00 53.979427
1004	*N* 30 01 47.005458	*W* 90 14 9.000227
1005	*N* 30 11 48.023655	*W* 90 14 59.002660
1006	*N* 30 23 40.995481	*W* 90 15 57.969548

point	Variance.**Latitude**.[rad]	Variance.**Longitude**.[rad]
1001	*4.42e-019*	*4.33e-019*
1002	*8.88e-017*	*1.09e-016*
1003	*2.13e-016*	*5.13e-016*
1004	*6.36e-019*	*6.38e-019*
1005	*1.95e-015*	*9.79e-017*
1006	*2.27e-015*	*1.71e-015*

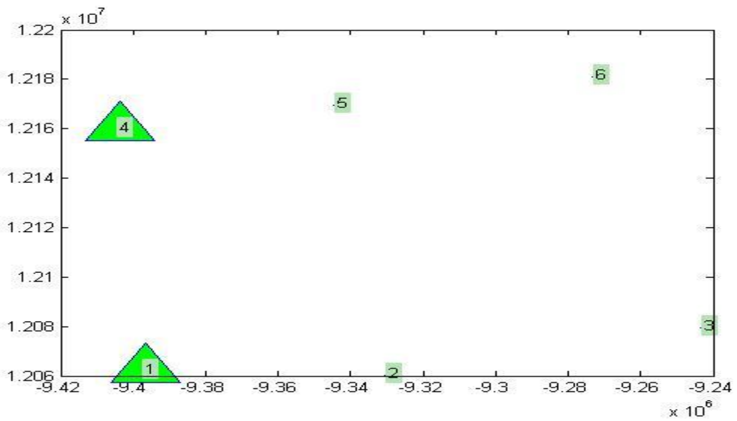
$$\widehat{\sigma_0^2} = 0.7584$$

10)مختصات نقاط در سیستم تصویر UTM محاسبه و دقت نقاط نیز توسط قانون انتشار واریانس- کواریانس بدست آمد:

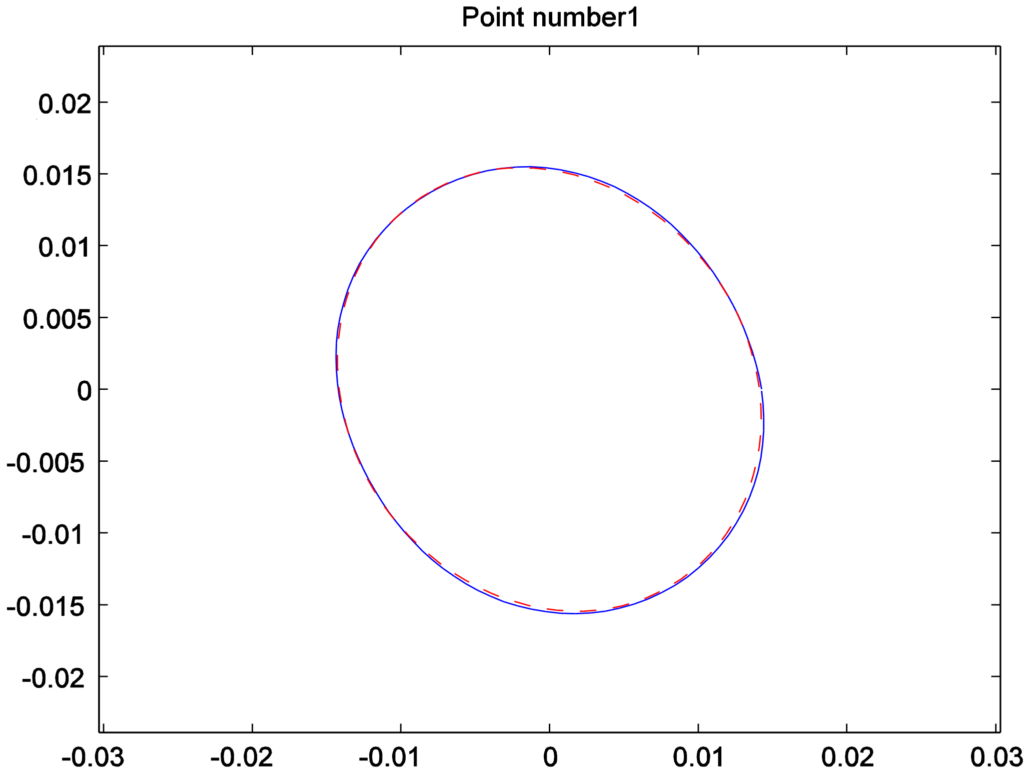
point	X(m)	Y(m)
1001	*-9396700.5716*	*12062829.0242*
1002	*-9329234.8889*	*12061147.4902*
1003	*-9242461.2258*	*12080156.7801*
1004	*-9403780.1886*	*12160406.0444*
1005	*-9343419.7597*	*12170444.0799*
1006	*-9272138.4329*	*12181581.4294*

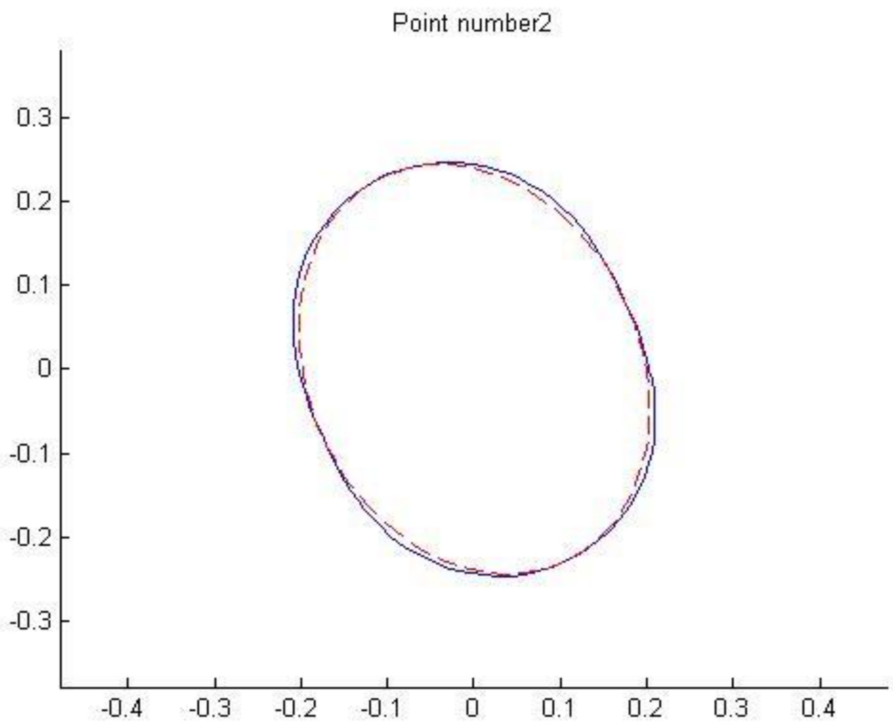
point	Variance.X.[m]	Variance.Y.[m]
1001	*0.0002*	*0.0002*
1002	*0.0412*	*0.0593*
1003	*0.1107*	*0.2719*
1004	*0.0002*	*0.0003*
1005	*0.8582*	*0.0627*
1006	*1.3366*	*0.8501*

11)شکل شبکه

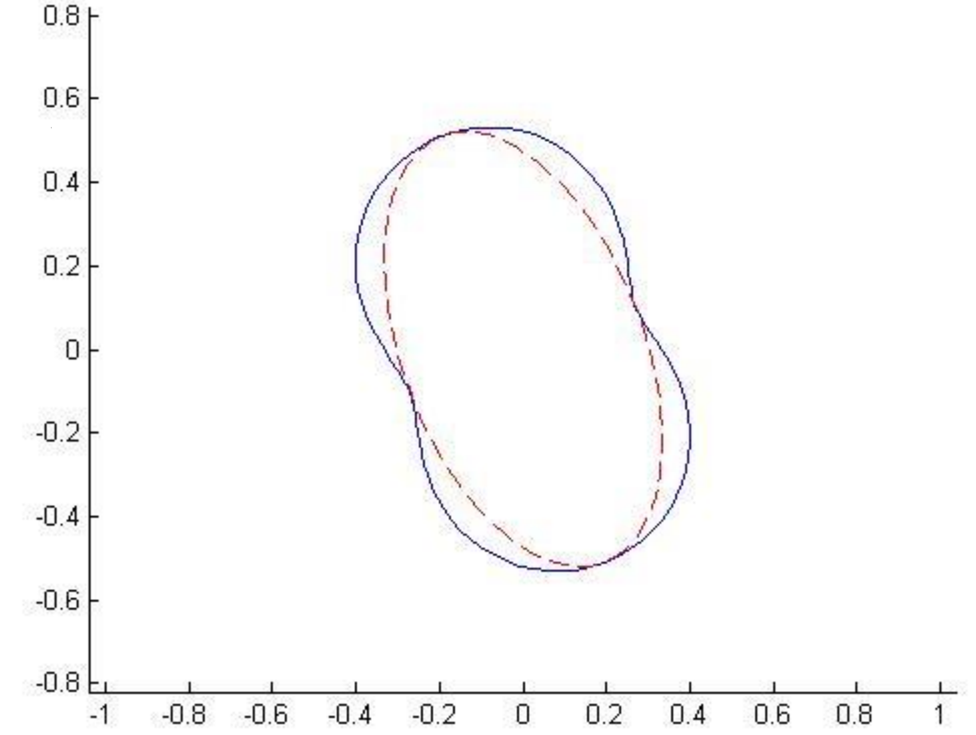


12)رسم بیضی خطای مطلق و منحنی پدال:

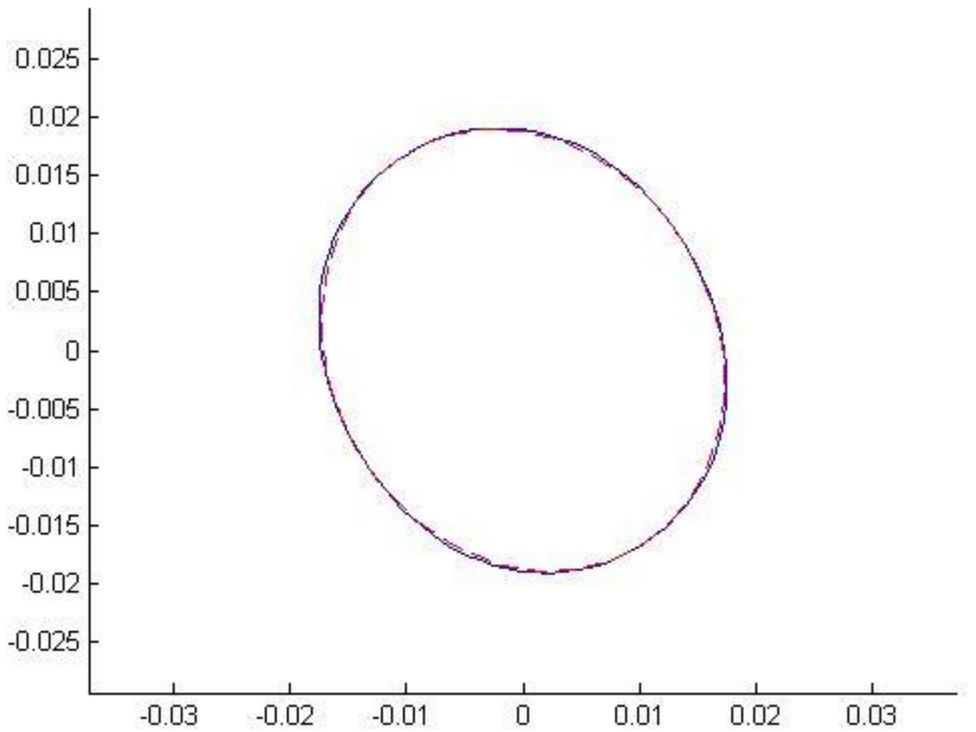


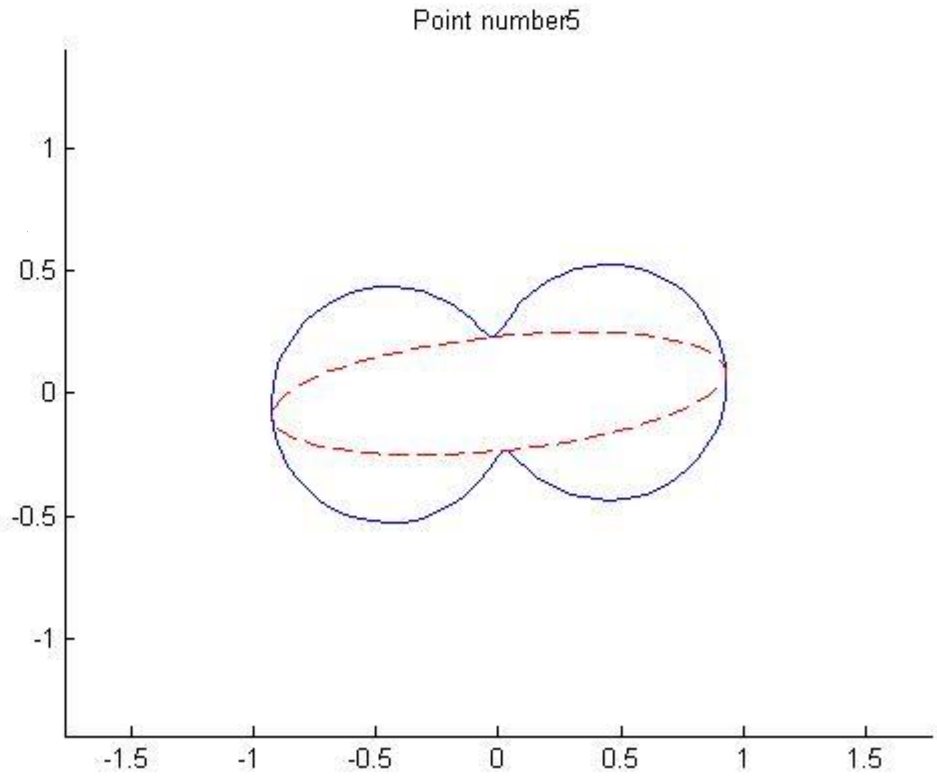


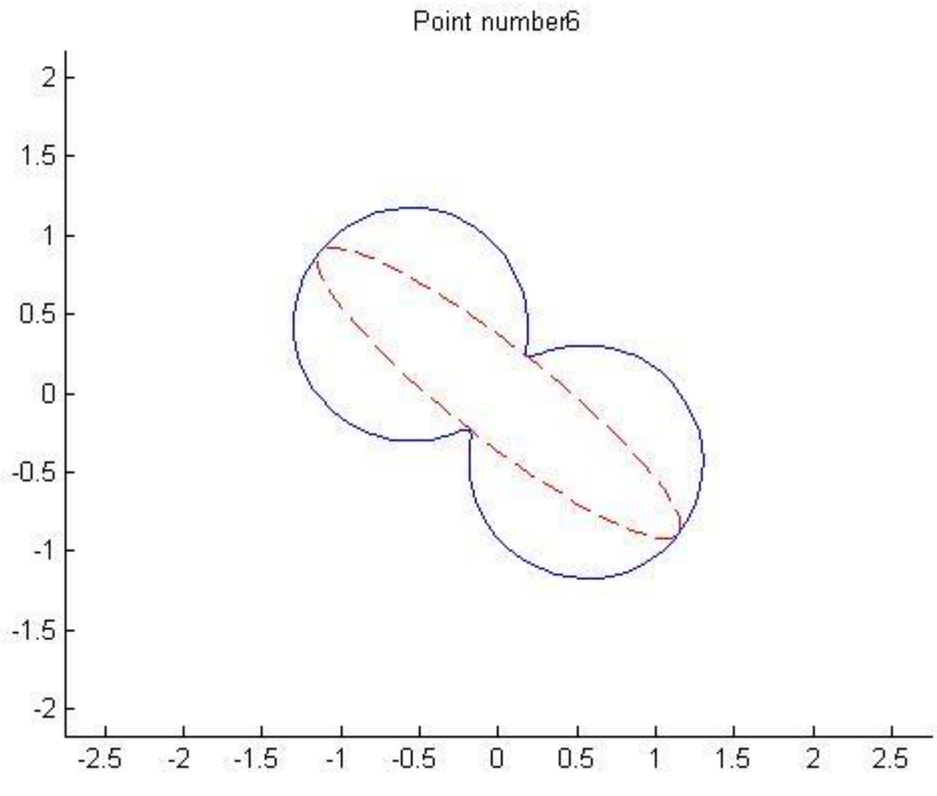
Point number3



Point number4




Point number5
1
0.5
0
-0.5
-1
-1.5
-1
-0.5
0
0.5
1
1.5


Point number6
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

نتایج :

1)در سرشکنی شبکه های ژئودزی می توان از نقاط شبکه های دیگر به عنوان نقطه ی معلوم استفاده نمود.

2)اگر قیودی که به مساله اعمال می شود دارای وزن باشد،شبکه را به روش پارامترهای وزن دار سرشکن می کنیم.

3) در صورتی که واریانس فاکتور ثانویه در تست رد شود،عوامل زیر را بررسی می کنیم:

-خطای سیستماتیک

- اشتباهات بزرگ

- مدل ریاضی

- محاصبات ریاضی

- مقادیر تقریبی

- وزن نسبی نقاط

4) نقاطی که به عنوان نقطه ی ثابت وارد مساله می شود پس از سرشکنی با بالاترین دقت برآورد می شود.

ضمایم

برنامه های نوشته شده در متلب:

```
%IN THE NAME OF GOD
clc;clear all;format short g
disp('-------------------------------------------------------------------
')
disp('        |                                       |
')
disp('        |project by :      mehran ghandehary    |
')
disp('        |no : 851921326                         |
')
disp('        |********Surveying Engineering**********|_
')
disp('        |********Isfahan University*************|
')
%-------------------Information-----------------------
%Geodesy 2 computatation
%Third project-------->Adjustment a Geodetic network on the Ellipsoid
% WGS84    a = 6378137.000000 ; b = 6356752.314245;
%There are six point in the network that 2 point are known
%coordinate of known point
%Point   Lattitude     StDev.[m]   Longitude       StDev.[m]
%1    N 30 00 0.00019  ±0.005      W 90 00 0.00013 ±0.005
%4    N 30 01 47.00550 ±0.006      W 90 14 9.00022 ±0.006
%Approximate coordinate of unknown point
% Point Approx. Latitude Approx. Longitude
% 5     N 30 11 47       W 90 14 58
% 6     N 30 23 40       W 90 15 57
% 2     N 30 10 54       W 89 58 58
% 3     N 30 25 40       W 90 00 54
Fi=[30 00 0.00019;30 10 54;30 25 40 ;30 01 47.00550;30 11 47; 30 23 40];
Lambda=[90 00 0.00013;89 58 58;90 00 54;90 14 9.00022;90 14 58;90 15 57];
Fi=deg2rad(dms2degrees(Fi));Lambda=-deg2rad(dms2degrees(Lambda));
Fi1=Fi(1);Fi2=Fi(2);Fi3=Fi(3);Fi4=Fi(4);Fi5=Fi(5);Fi6=Fi(6);
Lambda1=Lambda(1);Lambda2=Lambda(2);Lambda3=Lambda(3);Lambda4=Lambda(4);Lam
bda5=Lambda(5);Lambda6=Lambda(6);
j=1;
for i=1:2:12
    x(i)=Fi(j);
    x(i+1)=Lambda(j);
    j=j+1;
end
x=x';
x1=x;
%There are fourteen observation(five distance,3 Azimutes,6 Angle)
% From To Azimuth      St.Dev.[sec]
%  3   1  178 15 8.06  ±1.0
%  3   4  205 46 47.50 ±1.0
%  5   4  175 51 33.33 ±1.0
Az=[178 15 8.06;205 46 47.50;175 51 33.33];
Az_o=deg2rad(dms2degrees(Az));
% From To Distance[m] StDev.
%  1   2  20204.068   10mm+3ppm
%  1   3  47442.343   10mm+3ppm
%  2   4  29617.434   10mm+3ppm
%  3   2  27454.890   10mm+3ppm
%  6   5  22011.770   10mm+3ppm
s_o=[20204.068;27454.890;22011.770];
% From Occupied To Angle           St.Dev.[sec]
%  2      1       4  273 39 55.80   ±1
%  1      2       3  168 56 24.80   ±1
%  1      4       5  257 41 06.40   ±1
```

```
%  3     6      5  94 35 13.50    ±1
%  1     5      2  321 32 4.11    ±1
%  6     5      2  97 44 27.96    ±1
Angle=[273 39 55.80;168 56 24.80;257 41 06.40;94 35 13.50;321 32 4.11;97 44
27.96];
Angle_o=deg2rad(dms2degrees(Angle));
%--------------------First step--------------------
%Adjustment network with weighted parameters method
%Organizing matrix A(14,12)
e=0.081819191;
deltax_hat=5;
while norm(deltax_hat)>10^-9
Fi1=x(1);Fi2=x(3);Fi3=x(5);Fi4=x(7);Fi5=x(9);Fi6=x(11);
Lambda1=x(2);Lambda2=x(4);Lambda3=x(6);Lambda4=x(8);Lambda5=x(10);Lambda6=x
(12);
A(12,12)=0;
k=1;
% %line 1
A(k,1)=cofficient('a',Fi1,Lambda1,Fi2,Lambda2);
A(k,2)=cofficient('b',Fi1,Lambda1,Fi2,Lambda2);
A(k,3)=cofficient('c',Fi1,Lambda1,Fi2,Lambda2);
A(k,4)=cofficient('d',Fi1,Lambda1,Fi2,Lambda2);
k=k+1;
%line 2
% A(k,1)=cofficient('a',Fi1,Lambda1,Fi3,Lambda3);
% A(k,2)=cofficient('b',Fi1,Lambda1,Fi3,Lambda3);
% A(k,5)=cofficient('c',Fi1,Lambda1,Fi3,Lambda3);
% A(k,6)=cofficient('d',Fi1,Lambda1,Fi3,Lambda3);
% k=k+1;
%line 3
% A(k,3)=cofficient('a',Fi2,Lambda2,Fi4,Lambda4);
% A(k,4)=cofficient('b',Fi2,Lambda2,Fi4,Lambda4);
% A(k,7)=cofficient('c',Fi2,Lambda2,Fi4,Lambda4);
% A(k,8)=cofficient('d',Fi2,Lambda2,Fi4,Lambda4);
% k=k+1;
%line 4
A(k,3)=cofficient('c',Fi3,Lambda3,Fi2,Lambda2);
A(k,4)=cofficient('d',Fi3,Lambda3,Fi2,Lambda2);
A(k,5)=cofficient('a',Fi3,Lambda3,Fi2,Lambda2);
A(k,6)=cofficient('b',Fi3,Lambda3,Fi2,Lambda2);
k=k+1;
% % line 5
A(k,9)=cofficient('c',Fi6,Lambda6,Fi5,Lambda5);
A(k,10)=cofficient('d',Fi6,Lambda6,Fi5,Lambda5);
A(k,11)=cofficient('a',Fi6,Lambda6,Fi5,Lambda5);
A(k,12)=cofficient('b',Fi6,Lambda6,Fi5,Lambda5);
k=k+1;
%line 6
A(k,1)=cofficient('g',Fi3,Lambda3,Fi1,Lambda1);
A(k,2)=-cofficient('f',Fi3,Lambda3,Fi1,Lambda1);
A(k,5)=cofficient('e',Fi3,Lambda3,Fi1,Lambda1);
A(k,6)=cofficient('f',Fi3,Lambda3,Fi1,Lambda1);
k=k+1;
%line 7
A(k,5)=cofficient('e',Fi3,Lambda3,Fi4,Lambda4);
A(k,6)=cofficient('f',Fi3,Lambda3,Fi4,Lambda4);
A(k,7)=cofficient('g',Fi3,Lambda3,Fi4,Lambda4);
A(k,8)=-cofficient('f',Fi3,Lambda3,Fi4,Lambda4);
k=k+1;
%line 8
A(k,7)=cofficient('g',Fi5,Lambda5,Fi4,Lambda4);
```

```
A(k,8)=-cofficient('f',Fi5,Lambda5,Fi4,Lambda4);
A(k,9)=cofficient('e',Fi5,Lambda5,Fi4,Lambda4);
A(k,10)=cofficient('f',Fi5,Lambda5,Fi4,Lambda4);
k=k+1;
%line 9
A(k,1)=cofficient('e',Fi1,Lambda1,Fi4,Lambda4)-
cofficient('e',Fi1,Lambda1,Fi2,Lambda2);
A(k,2)=cofficient('f',Fi1,Lambda1,Fi4,Lambda4)-
cofficient('f',Fi1,Lambda1,Fi2,Lambda2);
A(k,3)=-cofficient('g',Fi1,Lambda1,Fi2,Lambda2);
A(k,4)=cofficient('f',Fi1,Lambda1,Fi2,Lambda2);
A(k,7)=cofficient('g',Fi1,Lambda1,Fi4,Lambda4);
A(k,8)=-cofficient('f',Fi1,Lambda1,Fi4,Lambda4);
k=k+1;
%line 10
A(k,1)=-cofficient('g',Fi2,Lambda2,Fi1,Lambda1);
A(k,2)=cofficient('f',Fi2,Lambda2,Fi1,Lambda1);
A(k,3)=cofficient('e',Fi2,Lambda2,Fi3,Lambda3)-
cofficient('e',Fi2,Lambda2,Fi1,Lambda1);
A(k,4)=cofficient('f',Fi2,Lambda2,Fi3,Lambda3)-
cofficient('f',Fi2,Lambda2,Fi1,Lambda1);
A(k,5)=cofficient('g',Fi2,Lambda2,Fi3,Lambda3);
A(k,6)=-cofficient('f',Fi2,Lambda2,Fi3,Lambda3);
k=k+1;
%line 11
A(k,1)=-cofficient('g',Fi4,Lambda4,Fi1,Lambda1);
A(k,2)=cofficient('f',Fi4,Lambda2,Fi1,Lambda1);
A(k,7)=cofficient('e',Fi4,Lambda4,Fi5,Lambda5)-
cofficient('e',Fi4,Lambda4,Fi1,Lambda1);
A(k,8)=cofficient('f',Fi4,Lambda4,Fi5,Lambda5)-
cofficient('f',Fi4,Lambda4,Fi1,Lambda1);
A(k,9)=cofficient('g',Fi4,Lambda4,Fi5,Lambda5);
A(k,10)=-cofficient('f',Fi4,Lambda4,Fi5,Lambda5);
k=k+1;
%line 12
A(k,5)=-cofficient('g',Fi6,Lambda6,Fi3,Lambda3);
A(k,6)=cofficient('f',Fi6,Lambda6,Fi3,Lambda3);
A(k,9)=cofficient('g',Fi6,Lambda6,Fi5,Lambda5);
A(k,10)=-cofficient('f',Fi6,Lambda6,Fi5,Lambda5);
A(k,11)=cofficient('e',Fi6,Lambda6,Fi5,Lambda5)-
cofficient('e',Fi6,Lambda6,Fi3,Lambda3);
A(k,12)=cofficient('f',Fi6,Lambda6,Fi5,Lambda5)-
cofficient('f',Fi6,Lambda6,Fi3,Lambda3);
k=k+1;
%line 13
A(k,1)=-cofficient('g',Fi5,Lambda5,Fi1,Lambda1);
A(k,2)=cofficient('f',Fi5,Lambda5,Fi1,Lambda1);
A(k,3)=cofficient('g',Fi5,Lambda5,Fi2,Lambda2);
A(k,4)=-cofficient('f',Fi5,Lambda5,Fi2,Lambda2);
A(k,9)=cofficient('e',Fi5,Lambda5,Fi2,Lambda2)-
cofficient('e',Fi5,Lambda5,Fi1,Lambda1);
A(k,10)=cofficient('f',Fi5,Lambda5,Fi2,Lambda2)-
cofficient('f',Fi5,Lambda5,Fi1,Lambda1);
k=k+1;
%line 14
A(k,3)=cofficient('g',Fi5,Lambda5,Fi2,Lambda2);
A(k,4)=-cofficient('f',Fi5,Lambda5,Fi2,Lambda2);
A(k,9)=cofficient('e',Fi5,Lambda5,Fi2,Lambda2)-
cofficient('e',Fi5,Lambda5,Fi6,Lambda6);
A(k,10)=cofficient('f',Fi5,Lambda5,Fi2,Lambda2)-
cofficient('f',Fi5,Lambda5,Fi6,Lambda6);
```

```
A(k,11)=-cofficient('g',Fi5,Lambda5,Fi6,Lambda6);
A(k,12)=cofficient('f',Fi5,Lambda5,Fi6,Lambda6);
%Organizing matrix Ax(4,12)
% Ax(4,12)=0;
% Ax(1,1)=1;Ax(2,2)=1;
% % clx=[.005,.005];
% Ax(3,7)=1;Ax(4,8)=1;
% clx=[.005/6400000,.005/6400000,.006/6400000,.006/6400000];
% clx=diag(clx);
% px=inv(clx^2);
Ax(4,12)=0;
Ax(1,1)=1;Ax(2,2)=1;
% clx=[.005/6400000,.005/6400000];
Ax(3,7)=1;Ax(4,8)=1;
clx=[.005/6400000,.005/6400000,.006/6400000,.006/6400000];
clx=diag(clx);
px=inv(clx^2);
[alfa12,s_cc1]=Robbins(Fi1,Lambda1,Fi2,Lambda2);s_c(1)=s_cc1;
% [alfa12,s_cc2]=Robbins(Fi1,Lambda1,Fi3,Lambda3);s_c(2)=s_cc2;
% [alfa12,s_cc3]=Robbins(Fi2,Lambda2,Fi4,Lambda4);s_c(3)=s_cc3;
[alfa12,s_cc4]=Robbins(Fi3,Lambda3,Fi2,Lambda2);s_c(2)=s_cc4;
[alfa12,s_cc5]=Robbins(Fi6,Lambda6,Fi5,Lambda5);s_c(3)=s_cc5;
[alfa12,s_cc6]=Robbins(Fi3,Lambda3,Fi1,Lambda1);Az_c(1)=alfa12;
[alfa12,s_cc7]=Robbins(Fi3,Lambda3,Fi4,Lambda4);Az_c(2)=alfa12;
[alfa12,s_cc8]=Robbins(Fi5,Lambda5,Fi4,Lambda4);Az_c(3)=alfa12;
Angle_c(1)=Robbins(Fi1,Lambda1,Fi4,Lambda4)-
Robbins(Fi1,Lambda1,Fi2,Lambda2);
Angle_c(2)=Robbins(Fi2,Lambda2,Fi3,Lambda3)-
Robbins(Fi2,Lambda2,Fi1,Lambda1);
Angle_c(3)=Robbins(Fi4,Lambda4,Fi5,Lambda5)-
Robbins(Fi4,Lambda4,Fi1,Lambda1);
Angle_c(4)=Robbins(Fi6,Lambda6,Fi5,Lambda5)-
Robbins(Fi6,Lambda6,Fi3,Lambda3);
Angle_c(5)=(Robbins(Fi5,Lambda5,Fi2,Lambda2)-
Robbins(Fi5,Lambda5,Fi1,Lambda1))+2*pi;
Angle_c(6)=(Robbins(Fi5,Lambda5,Fi2,Lambda2)+(2*pi-
Robbins(Fi5,Lambda5,Fi6,Lambda6)));
obs=[s_o;Az_o;Angle_o];
comp=[s_c,Az_c,Angle_c]';
W=(comp-obs);
%Organizing matrix p(weigt)
% cl=[70.6122,152.327,98.8522,92.3646,76.0353,4.848*10^(-6),4.848*10^(-
6),4.848*10^(-6),4.848*10^(-6),4.848*10^(-6),4.848*10^(-6),4.848*10^(-
6),4.848*10^(-6),4.848*10^(-6)];
cl=[70.6122*.001,92.3646*.001,76.0353*.001,1/206265,1/206265,1/206265,1/206
265,1/206265,1/206265,1/206265,1/206265,1/206265];
cl=diag(cl);
% p1=[14.9422508727314,8.73301892655232,12.8867968226999,651288130237599,
651288130237599];
%
p2=[651288130237599,651288130237599,651288130237599,651288130237599,6512881
30237599,651288130237599,651288130237599];
% p=[p1,p2];
% p=diag(p);
p=inv(cl^2);
A1=A'*p*A+Ax'*px*Ax;
deltax_hat=-inv(A1)*(A'*p*W);
norm(deltax_hat);
x_hat=x+deltax_hat
x=x_hat;
v_hat=A*deltax_hat+W
```

```
v_hatx=Ax*deltax_hat;
sigma02_hat=((v_hat'*p*v_hat)+(v_hatx'*px*v_hatx))/4
end
Cx_hat=sigma02_hat*inv(A1)
Cl_hat=A*Cx_hat*A';
Cv_hat=(cl-Cl_hat);
khi2_1=11.14;
khi2_2=.48;
k=0;
%teste sigma02_hat
if 6*sigma02_hat/khi2_1<=1&&6*sigma02_hat/khi2_2>=1
    k=1
end
%teste Barda
if k==0
    m=outliers(v_hat,Cv_hat);
end
 j=1;
%utm
for i=1:6
    [X(i),Y(i)]=utm(x(j),x(j+1));
    j=j+2;
end
    B(12,12)=0;
for j=1:2:12
    Fi=x(j);Lambda=x(j+1);
    B1=utm2(Fi,Lambda);
    B(j,j)=B1(1,1);B(j,j+1)=B1(1,2);B(j+1,j)=B1(2,1);B(j+1,j+1)=B1(2,2);
end
Cx_hat_utm=B*Cx_hat*B'

 plot(X,Y,'^','MarkerFaceColor','g','MarkerSize',30);        hold on;
for i=1:6
    text(X(i),Y(i),num2str(i),'BackgroundColor',[.7 .9 .7]);
end
N=Cx_hat_utm;j=1;temp=1;
for i=1:2:11
    A=[N(i,i) N(i,i+1);N(i+1,i) N(i+1,i+1)];
%    SUPPORT Plot of support function and pertinent confidence
delete support.eps
[m,n] = size(A);
if m ~= 2 | n ~= 2
   error('Wrong dimension of matrix');
end
[v,d] = eig(A);
if d(1,1) <= 0 | d(2,2) <= 0
   error('The input matrix is no covariance matrix');
end;

% Calculations for confidence ellipse
[lambda,k] = sort(diag(d));
v = v(:,k);
if any(any(v)) == 1
   alpha = atan2(v(2,2),v(1,2))+pi/2;
else
   alpha = 0;
end
rot = [cos(alpha) sin(alpha);-sin(alpha) cos(alpha)];
t = linspace(0,2*pi,100);
a = sqrt(lambda(2))
b = sqrt(lambda(1));
```

```
pl = [a*cos(t);b*sin(t)];
for t = 1:100
   current = rot*pl(:,t); curve(1:2,t) = current;
end

% Calculations for support function
phi = linspace(0,2*pi,100);
support = sqrt(A(1,1)*(cos(phi)).^2 + A(2,1)*sin(2*phi)...
   + A(2,2)*(sin(phi)).^2);

% The 1-axis is oriented upwards and the 2-axis towards the right.
% In the polar plot we add pi/2 and in the cartesian plot
% interchanged the 1 and 2 columns of curve
h = figure(temp);
hold on
axis([-1.5*a 1.5*a -1.5*a 1.5*a])
axis equal

polar(phi,support,'b-')
axis(axis)
plot(curve(2,1:100),curve(1,1:100),'r--')

% hold off
% print support -deps
%%%%%%%% end support.m %%%%%%%%%%%%%%%%%%%

title(['Point number',num2str(temp)])
temp=temp+1;
end
```

```
function [alfa12,s]=Robbins(Fi1,Lambda1,Fi2,Lambda2)
%Robbins' formulae [Cooper, 1987]:
deltaLambda=Lambda2-Lambda1;
deltaFi=Fi2-Fi1;
a = 6378137.000000;e=0.081819191;
N1=a/(1-(e^2)*((sin(Fi1))^2))^.5;
N2=a/(1-(e^2)*((sin(Fi2))^2))^.5;
t=((1-e^2)*tan(Fi2)+(e^2)*(N1*sin(Fi1))/(N2*cos(Fi2)));
alfa12 = (abs(acot((cos(Fi1)*t -
sin(Fi1)*cos(deltaLambda))/sin(deltaLambda))))*180/pi;
if deltaFi>=0&& deltaLambda<0
    alfa12=2*180-alfa12;
end
if deltaFi<0&& deltaLambda<0
    alfa12=180+alfa12;
end
if deltaFi<0&& deltaLambda>0
    alfa12=180-alfa12;
end
alfa12=alfa12*pi/180;
sigma=asin(sin(deltaLambda)*cos(Fi2)/sin(alfa12));
z = e^2 /(1- e^2 );
```

```
h = sqrt(z*(cos(Fi1))^2* (cos(alfa12))^2);
g=sqrt(z*(sin(Fi1)^2));
s=N1*sigma*(1-(sigma^2)/6*(h^2)*(1-h^2)+(sigma^3)/8*g*h*(1-
2*h^2)+(sigma^4)/120*(h^2*(4-7*h^2)-3*g^2*(1-7*h^2))-(sigma^5)/48*g*h);
```

```
function set=cofficient(str,Fi1,Lambda1,Fi2,Lambda2)
a = 6378137.000000;e=0.081819191;
N1=a/(1-(e^2)*(sin(Fi1)^2))^.5;
N2=a/(1-(e^2)*(sin(Fi2)^2))^.5;
M1=a*(1-e^2)/(1-(e^2)*(sin(Fi1)^2))^(3/2);
M2=a*(1-e^2)/(1-(e^2)*(sin(Fi2)^2))^(3/2);
[alfa21,s12]=Robbins(Fi2,Lambda2,Fi1,Lambda1);
[alfa12,s12]=Robbins(Fi1,Lambda1,Fi2,Lambda2);
if str=='a'
    set=-M1*cos(alfa12);%/206265
end
if str=='b'
    set=N2*sin(alfa21)*cos(Fi2);
end
if str=='c'
    set=-M2*cos(alfa21);
end
if str=='d'
    set=-N2*sin(alfa21)*cos(Fi2);
end
if str=='e'
    set=M1*sin(alfa12)/s12;
end
if str=='f'
    set=N2*cos(alfa21)*cos(Fi2)/s12;
end
if str=='g'
    set=M2*sin(alfa21)/s12;
end
```

```
function [x,y]=utm(Fi,Lambda)
a = 6378137.000000;b = 6356752.314245;e=0.081819191;
e1=sqrt((a^2-b^2)/b^2);
N=a/(1-(e^2)*(sin(Fi)^2))^.5;
A0=1-(1/4*e^2)-(3/64*e^4)-(5/256*e^6)-(175/16384*e^8);
A2=3/8*((e^2)+(1/4*e^4)+(15/128*e^6)-(455/4096*e^8));
A4=15/256*((e^4)+(3/4*e^6)-(77/128*e^8));
A6=35/3072*((e^6)-(41/32*e^8));
A8=-315/131072*e^8;
% s=a*((Fi+3/4*Fi*e^2-Fi*e^2+45/64*Fi*e^4-3/4*Fi*e^4-45/64*Fi*e^6)+(-
3/8*e^2*sin(2*Fi)-
15/32*e^4*sin(2*Fi)+3/8*e^4*sin(2*Fi)+15/32*e^6*sin(2*Fi))+(15/256*e^4*sin(
4*Fi)-15/256*e^6*sin(4*Fi)));
s=a*(A0*Fi-A2*sin(2*Fi)+A4*sin(4*Fi)-A6*sin(6*Fi)+A8*sin(8*Fi));
t=tan(Fi);
k=sqrt((e1^2)/(cos(Fi))^2);
x=N*(Lambda*cos(Fi)+(Lambda^3*(cos(Fi))^3/6)*(1-
t^2+k^2)+(Lambda^5*(cos(Fi))^5/120)*(5-18*t^2+t^4+14*k^2-
58*t^2*k^2+13*k^4+4*k^6-64*k^4*t^2-24*k^6*t^2)...
    +(Lambda^7*(cos(Fi))^7/5040)*(61-479*t^2+179*t^4-t^6));
```

```
y=N*((s/N)+(Lambda^2/2*sin(Fi)*cos(Fi))+(Lambda^4/24*sin(Fi)*(cos(Fi))^3*(5
-t^2+9*k^2+4*k^4))...
    +(Lambda^6/720*sin(Fi)*(cos(Fi))^5*(61-58*t^2+t^4+270*k^2-
330*t^2*k^2+445*k^4+324*k^6-680*k^4*t^2+88*k^8-600*k^6*t^2-192*k^8*t^2))...
    +(Lambda^8/40320*sin(Fi)*(cos(Fi))^7*(1385-311*t^2+543*t^4-t^6)));
x=.9996*x;y=.9996*y;
```

```
function g=outliers(v_hat,Cv_hat)
T=2.77;
g=0;
s=1;
for i=1:size(v_hat)
    w(i)=v_hat(i)/sqrt(Cv_hat(i,i));
    if v_hat(i)<T*sqrt(Cv_hat(i,i))&&v_hat(i)>-T*sqrt(Cv_hat(i,i))
        g(s)=i;
        s=s+1;
        w(i)
    end
end
```

مراجع:

1)جزوه ی ژئودزی 2 آقای موسوی الکاظم

2)جزوه ی ژئودزی 2 آقای محمد کریم

3)جزوه ی ژئودتیک دکتر امیری

4)جزوه ی سرشکنی دکتر عسگری،مهندس نفیسی

5)UNB lecture notes

- map projection in geodesy

- geodetic position computation

-mathematical model for the horizontal geodetic network

6)Geodesy the concepts

7)Control surveys in civil Engineering(Cooper, M. A. R. (1987))