بسم الله الرحمن الرحیم

گزارش کار پروژه تئوری خطا

تهیه کنندگان:

سپیده آبادپور(810388090)

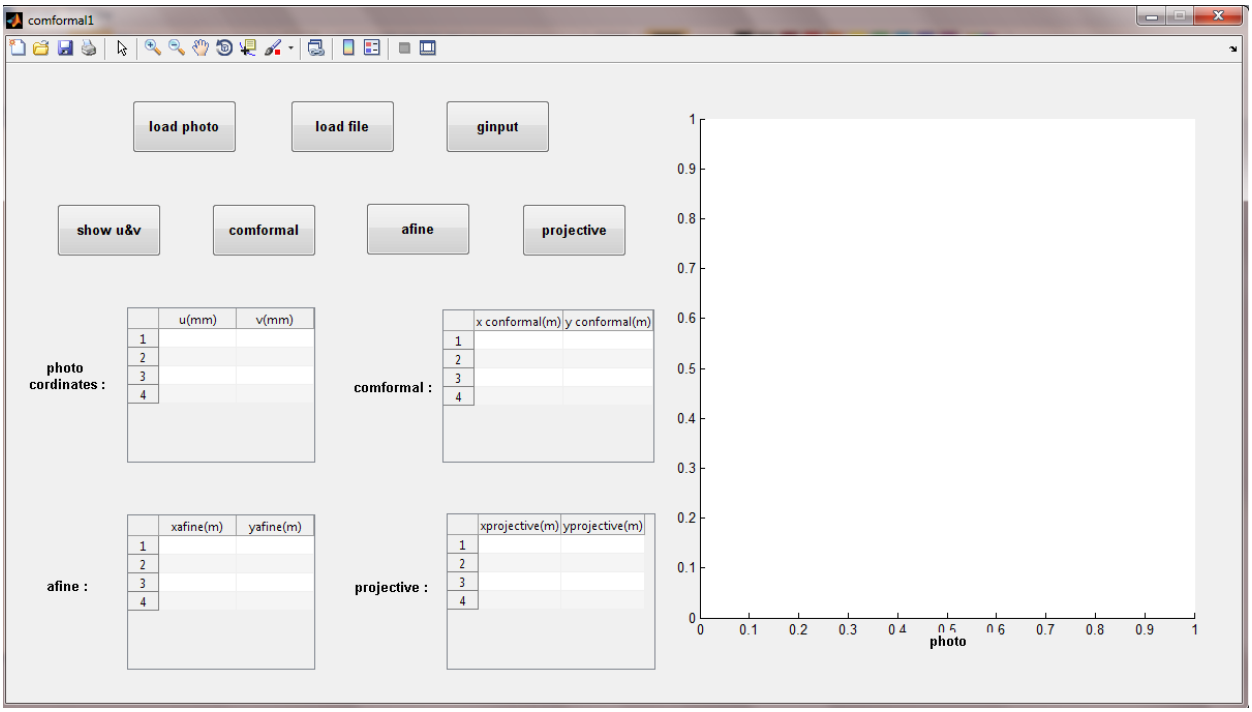
سیمین سادات میروهابی(810388065)

استاد راهنما:جناب آقای دکتر معتق

فهرست

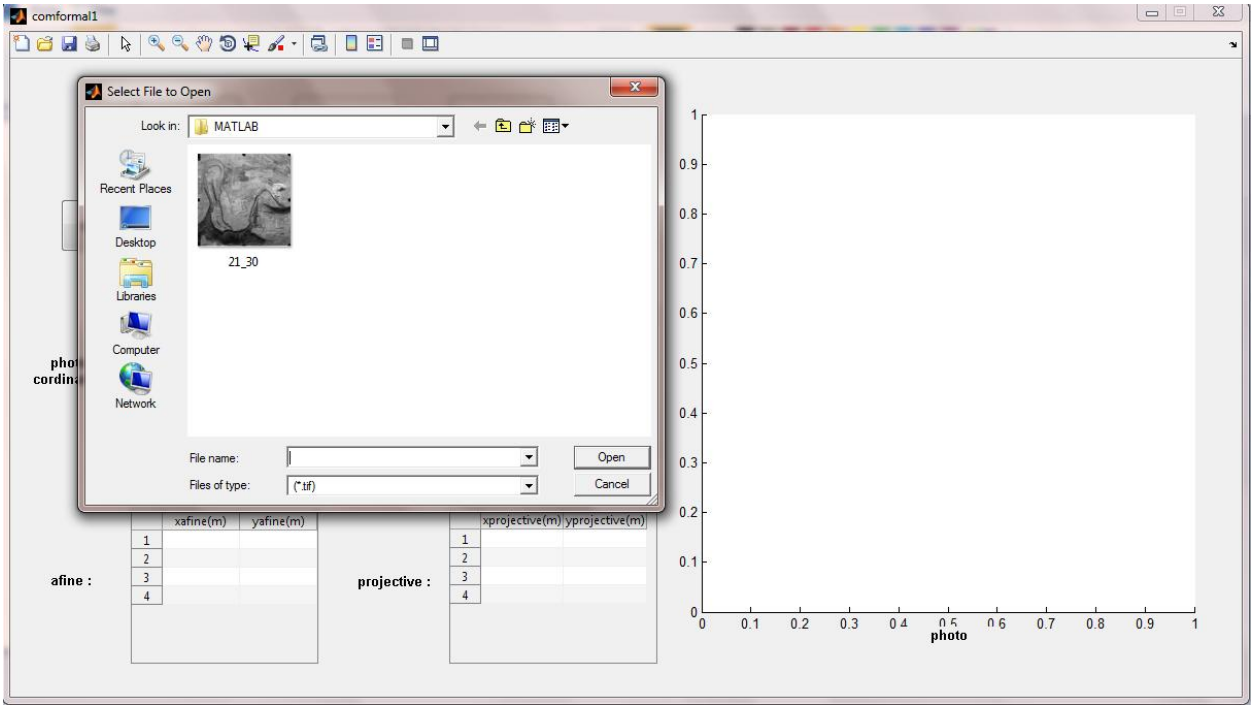
برنامه transformation

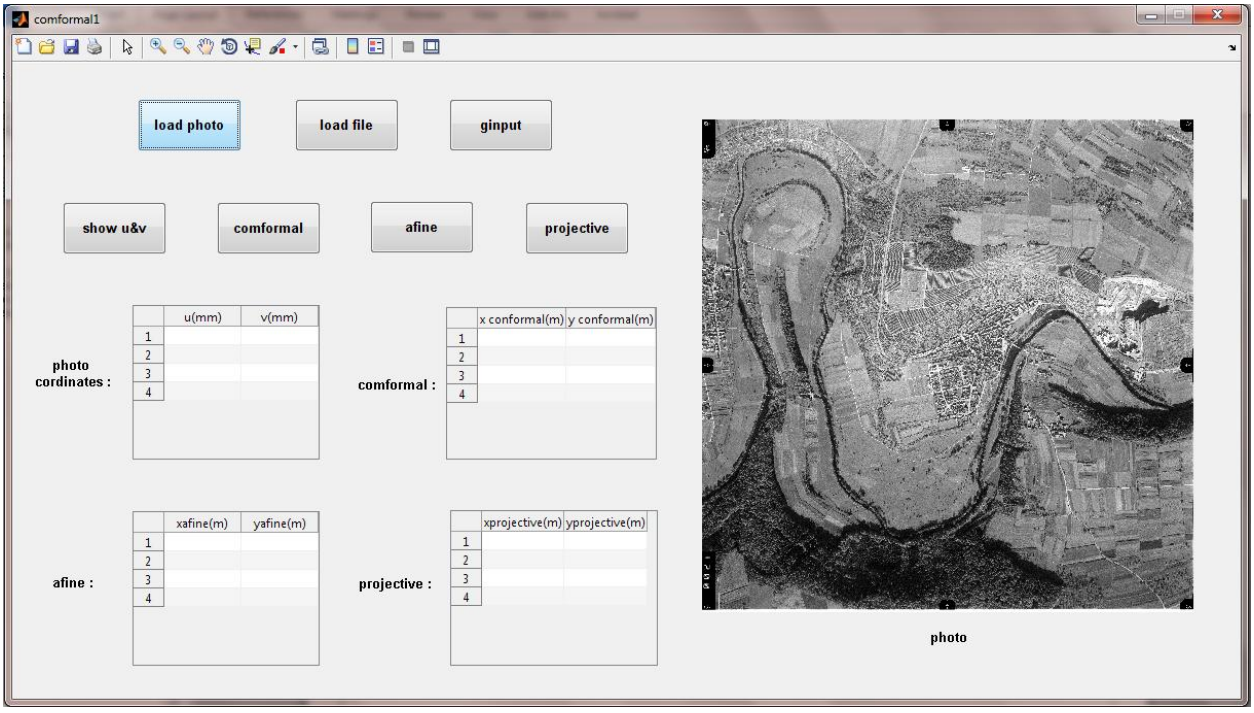
تصویر زیر شمای کلی برنامه یtransformation (تبدیلات) می باشد.



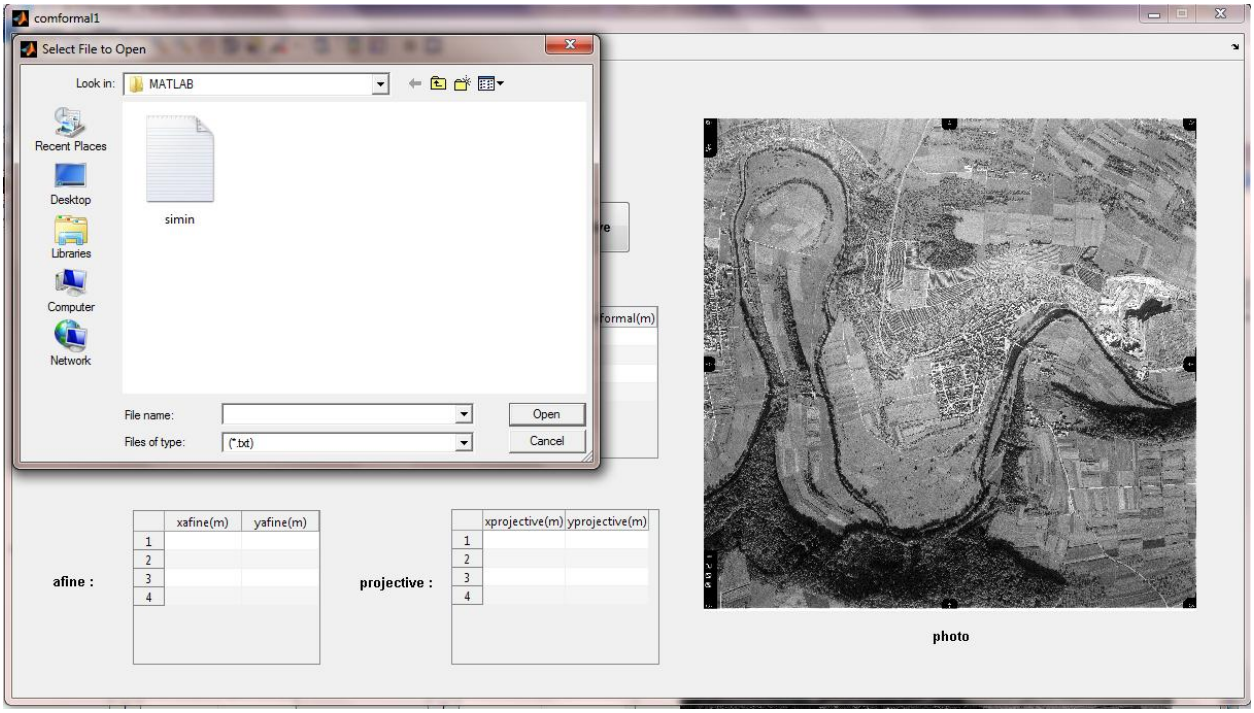
.

در مرحله ی اول باید ابتدا از دکمه ی load photo عکس کورد نظر را باز می کنیم

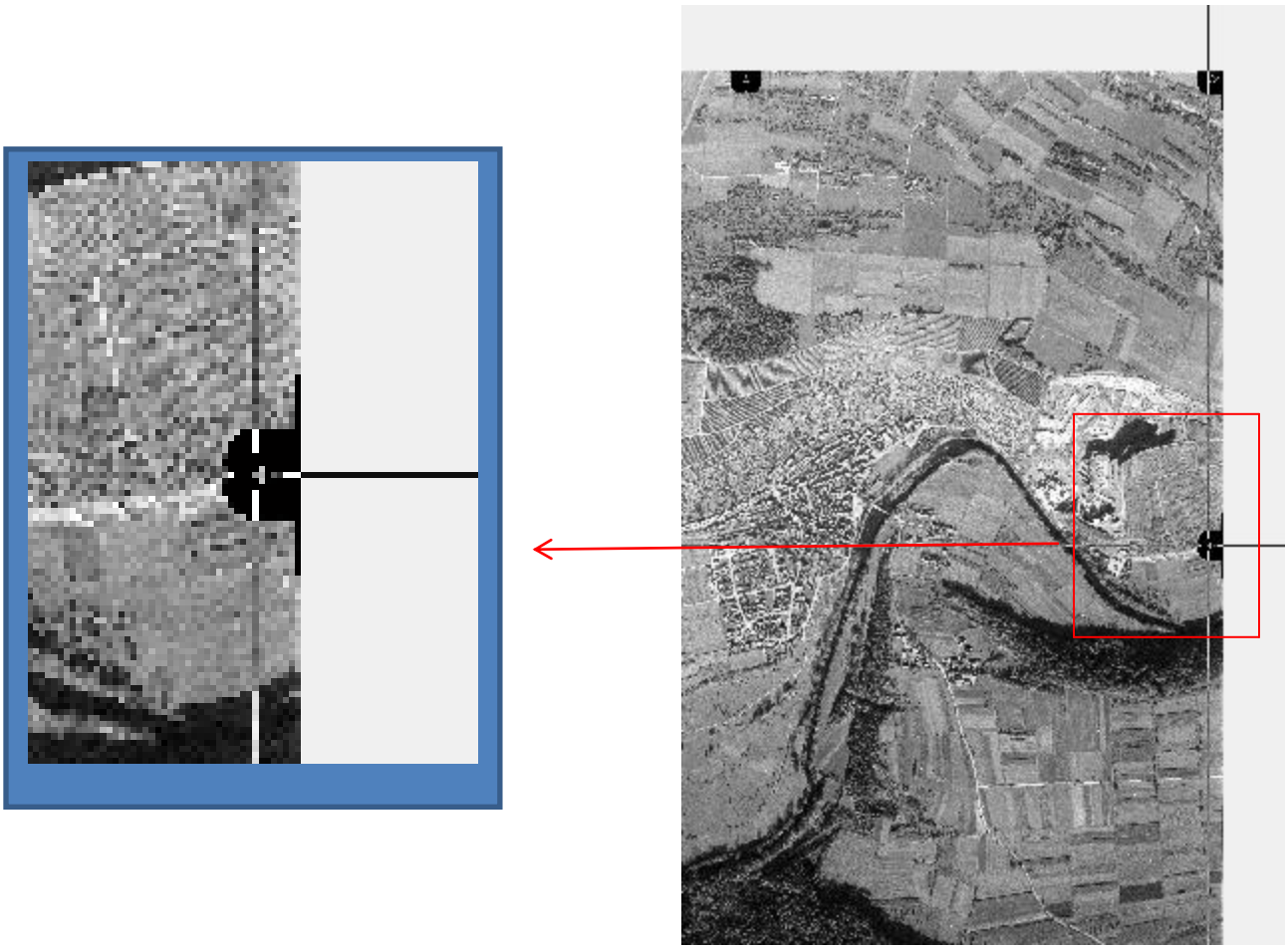




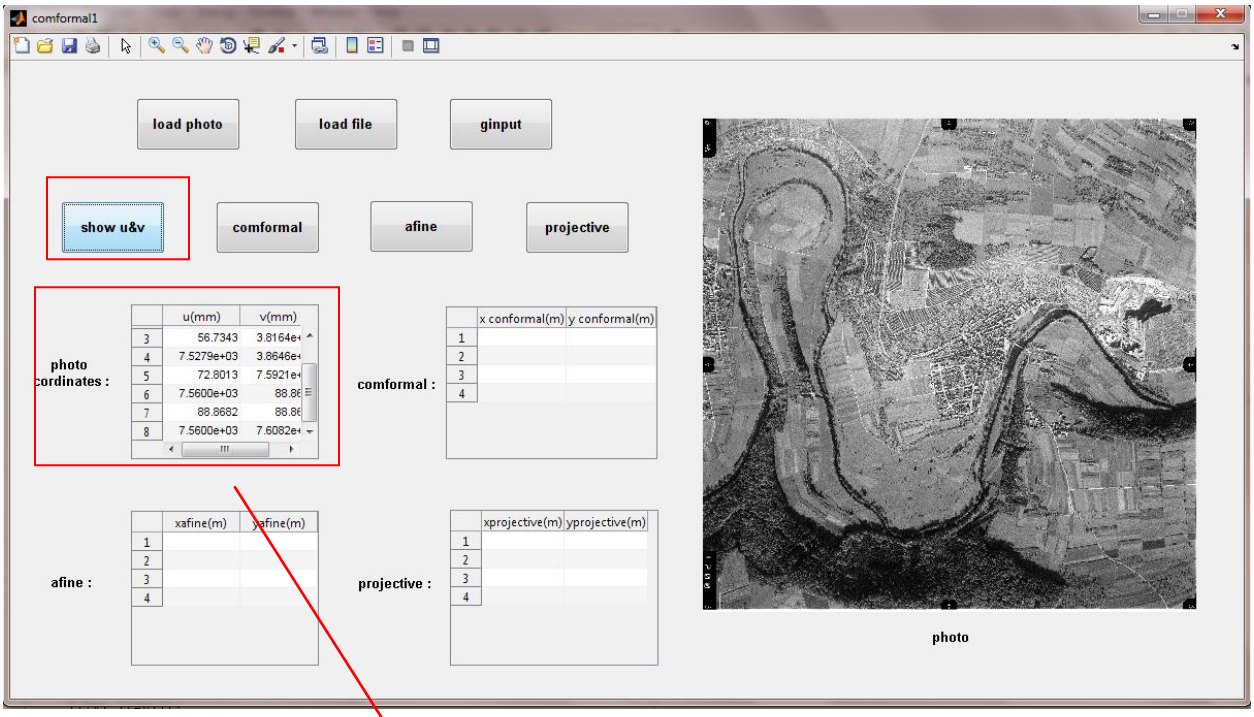
سپس روی دکمه ی load file کلیک می کنیم.

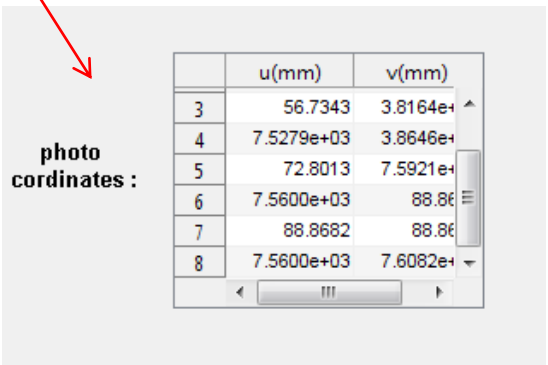


از منوی باز شده فایل نقاط رو انتخاب می کنیم . به این ترتیب مختصات نقاط فیدوشیال مارک در سیستم مختصات principal point در ماتریس مربوطه در برنامه وارد می شود. حال با استفاده از دکمه ی ginput رو عکس مختصات نقاط فیدوشیال مارک را وارد می کنیم.



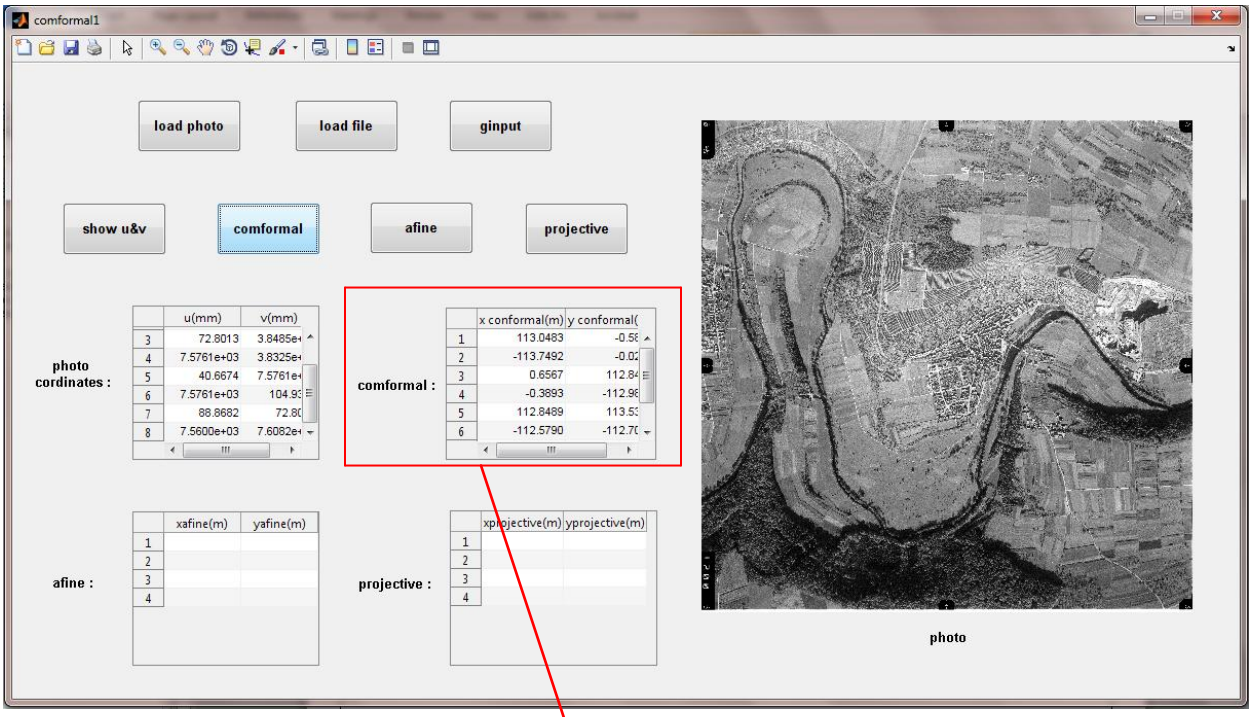
بعد از انتخاب فیدوشیال مارک ها با دستور ginput می توانید با استفاده از دکمه show V&U مختصات پیکسلی عکس ها را در جدول photo cordnate ببینید

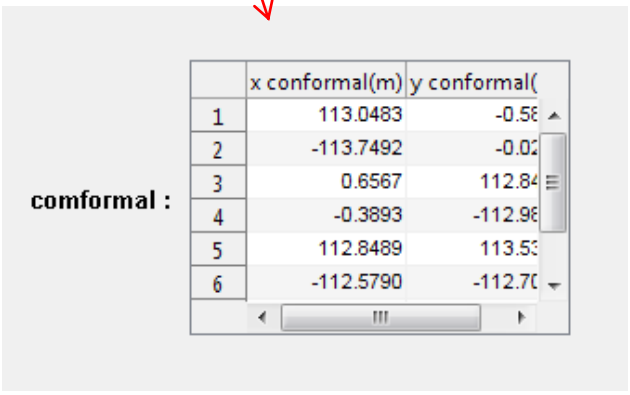




سپس به محاسبه ی مقادیر مورد نظر با استفاده از سه دکمه comformal و afine و projective می پردازیم. پس از آن نتایج مربوطه در سه جدول زیر نمایش داده می شود.

```
function pushbutton1_Callback(hObject, eventdata, handles)


global matrix counter txt u v
[n,p]=uigetfile('*.tif');
a=[p,n];
imread(a);
imshow(a);
```



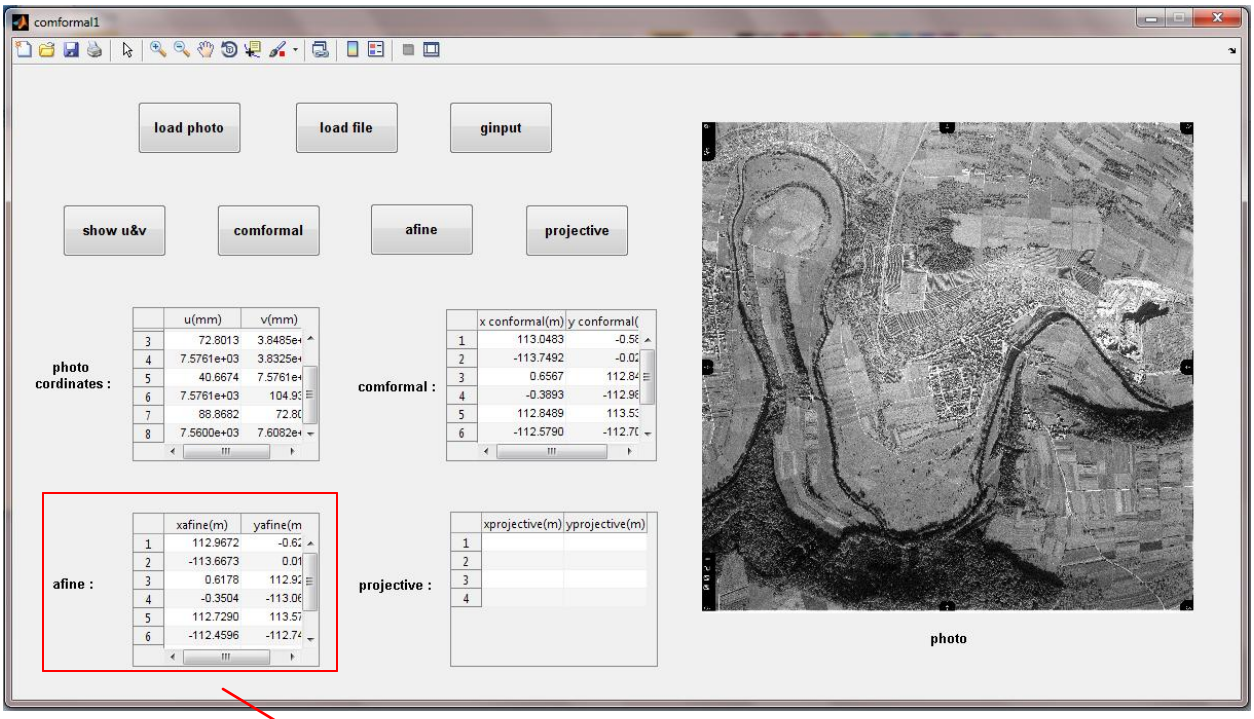


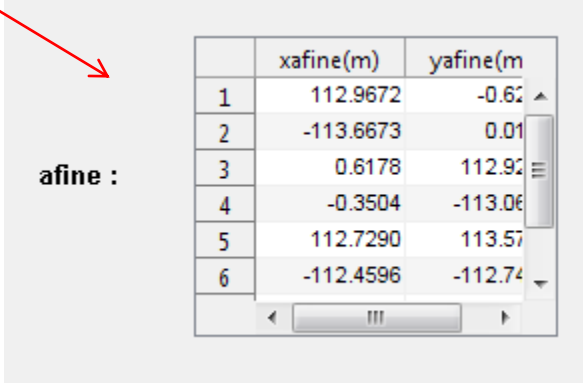
$$\begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

$$\begin{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_8 \\ y_8 \end{pmatrix} \end{bmatrix} = \begin{pmatrix} u_1 & -v_1 & 1 & 0 \\ v_1 & u_1 & 0 & 1 \\ \vdots & \ddots & & \vdots \\ u_8 & -v_8 & 1 & 0 \\ v_8 & u_8 & 0 & 1 \end{pmatrix} \begin{bmatrix} a_1 \\ b_1 \\ a_0 \\ b_0 \end{bmatrix}$$

$L = A.\ X$ $\qquad\qquad$ $X=(A^TA)^{-1}A^TL$

```
function pushbutton4_Callback(hObject, eventdata, handles)
global matrix counter txt u v
format long;
for i=1:length(matrix)
    a(2*i-1,1:4)=[matrix(i,1)/1000,-matrix(i,2)/1000,1,0];
    a(2*i,1:4)=[matrix(i,2)/1000,matrix(i,1)/1000,0,1];
end
x=txt(:,1);
y=txt(:,2);
for i=1:length(x)
    l(2*i-1,1)=x(i);
    l(2*i,1)=y(i);
end

q=inv(a'*a)*a'*l
for i=1:length(x)
   xcom(i)=[matrix(i,1)/1000,-matrix(i,2)/1000,1,0]*q;
   ycom(i)=[matrix(i,2)/1000,matrix(i,1)/1000,0,1]*q;
end
m=[xcom',ycom']
set(handles.uitable2,'data',m)
```





X1=a1*u1-b1*v1+a0

Y1=b1*u1+a1*v1+b0

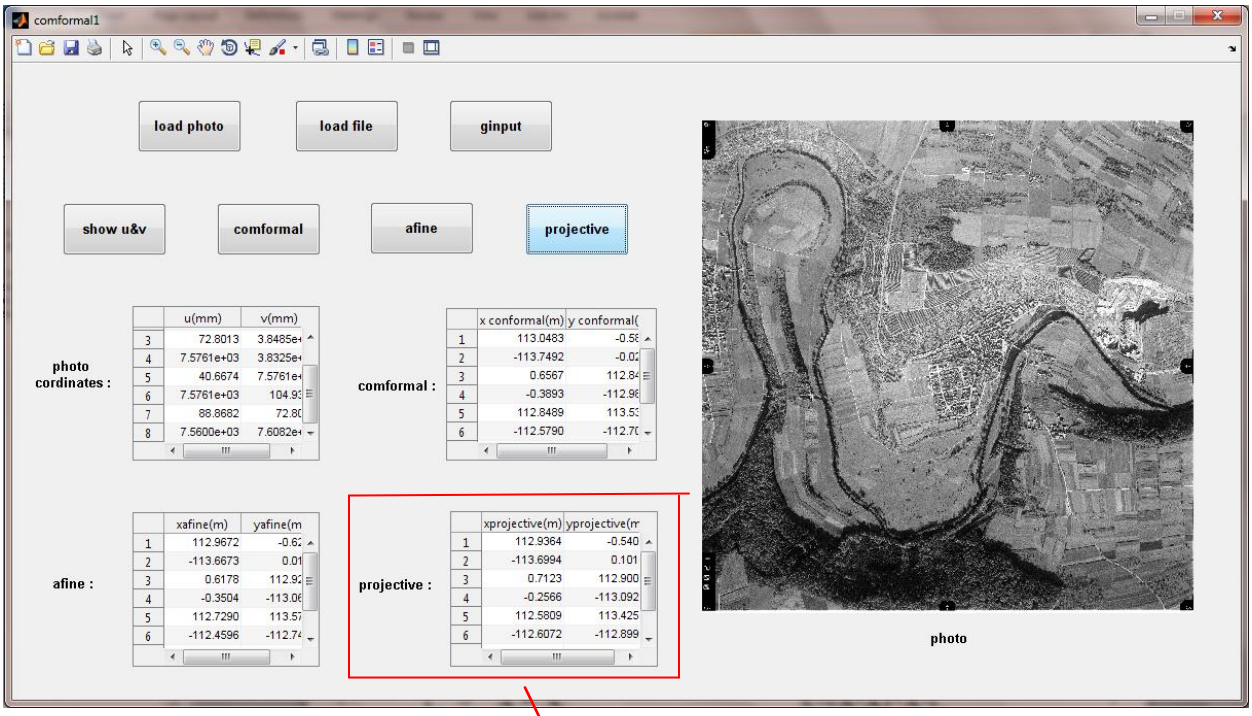
X2=a1*u2-b1*|v2+a0

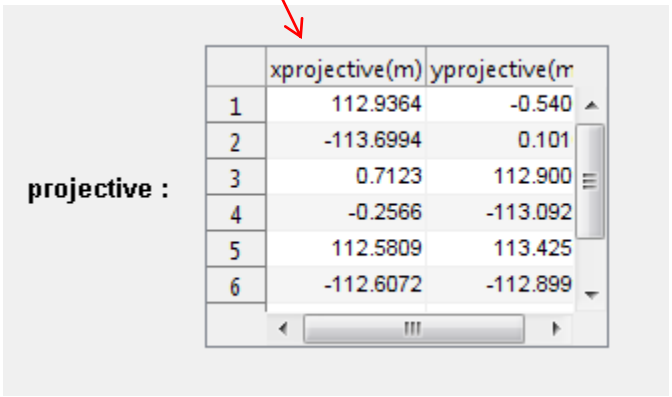
Y2=b1*u2+a1*v2+b0

$$\begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_8 \\ y_8 \end{pmatrix} = \begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_2 & v_2 & 1 \\ \vdots & & \vdots & & \vdots & \\ u_8 & v_8 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_8 & v_8 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_0 \\ b_1 \\ b_2 \\ b_0 \end{pmatrix}$$

$$L = A * X \qquad X=(A^{T}A)^{-1}A^{T}L$$

```
function pushbutton5_Callback(hObject, eventdata, handles)

global matrix counter txt u v
format long;
for i=1:length(matrix)
    a(2*i-1,1:6)=[matrix(i,1)/1000,-matrix(i,2)/1000,1,0,0,0];
    a(2*i,1:6)=[0,0,0,matrix(i,2)/1000,matrix(i,1)/1000,1];
end
x=txt(:,1);
y=txt(:,2);
for i=1:length(x)
    l(2*i-1,1)=x(i);
    l(2*i,1)=y(i);
end

q=inv(a'*a)*a'*l
for i=1:length(x)
   xcom(i)=[matrix(i,1)/1000,-matrix(i,2)/1000,1,0,0,0]*q;
   ycom(i)=[0,0,0,matrix(i,2)/1000,matrix(i,1)/1000,1]*q;
end
m=[xcom',ycom']
set(handles.uitable3,'data',m)
```





X=a1*u+a2*v+a3/a7*u+a8*v+1

Y=a4*u+a5*v+a6/a7*u+a8*v+1

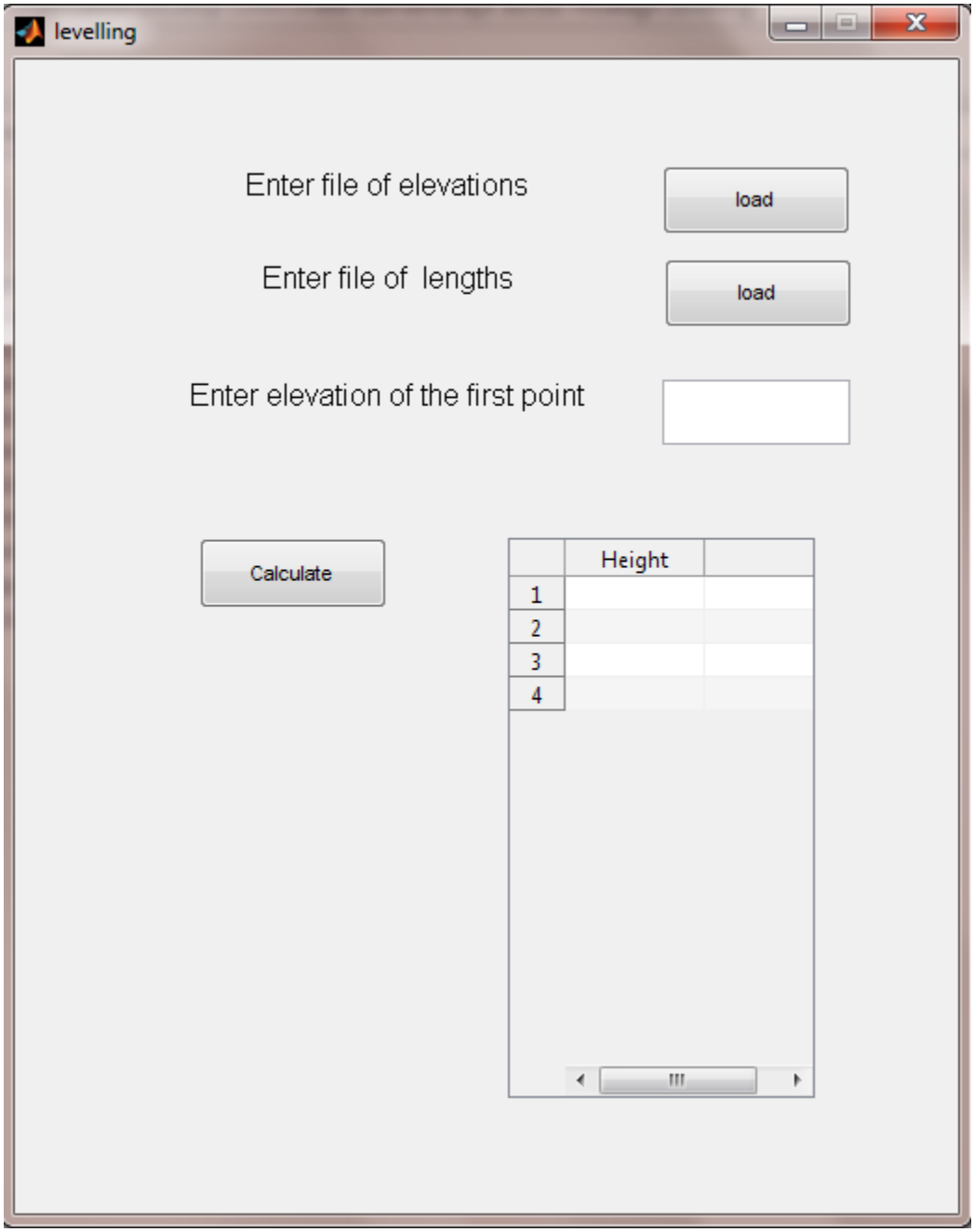
$$\begin{pmatrix} x_1 \\ y_2 \\ x_1 \\ y_2 \\ \vdots \\ x_8 \\ y_8 \end{pmatrix} = \begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -x_1v_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -y_1u_1 & -y_1v_1 \\ & \vdots & & & & \ddots & & \vdots \\ u_8 & v_8 & 1 & 0 & 0 & 0 & -x_8u_8 & -x_8v_8 \\ 0 & 0 & 0 & u_8 & v_8 & 1 & -y_8u_8 & -y_8v_8 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix}$$

$$L = A * X \qquad X=(A^TA)^{-1}A^TL$$

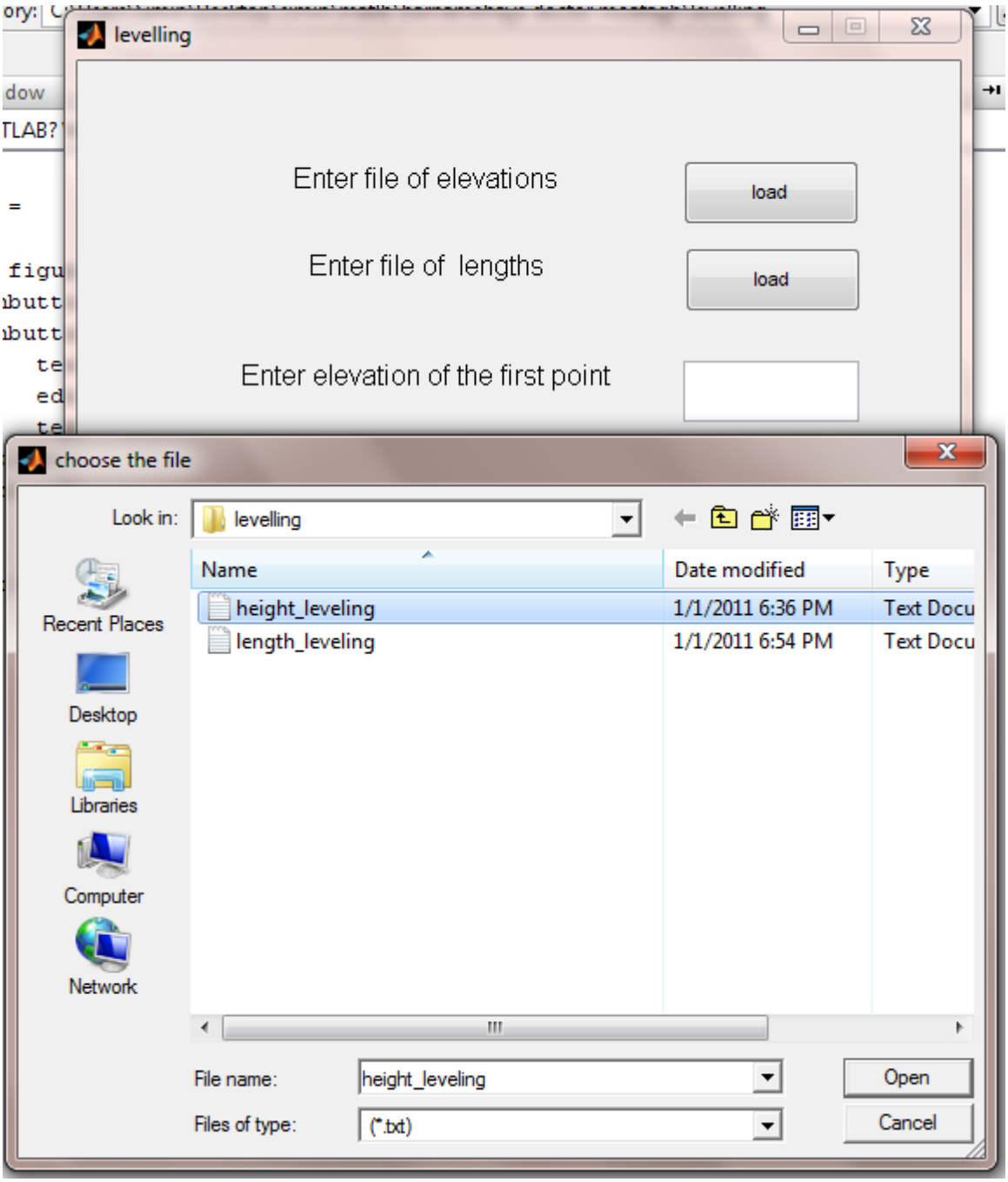
```
function pushbutton6_Callback(hObject, eventdata, handles)

global matrix counter txt u v
x=txt(:,1);
y=txt(:,2);
for i=1:length(x)
    l(2*i-1,1)=x(i);
    l(2*i,1)=y(i);
end
for i=1:length(matrix)
    a(2*i-1,1:8)=[matrix(i,1)/1000,matrix(i,2)/1000,1,0,0,0,-
x(i)*matrix(i,1)/1000,-x(i)*matrix(i,2)/1000];
    a(2*i,1:8)=[0,0,0,matrix(i,2)/1000,matrix(i,1)/1000,1,-
y(i)*matrix(i,1)/1000,-y(i)*matrix(i,2)/1000];
end

q=inv(a'*a)*a'*l
for i=1:length(x)
   xcom(i)=[matrix(i,1)/1000,matrix(i,2)/1000,1,0,0,0,-
x(i)*matrix(i,1)/1000,-x(i)*matrix(i,2)/1000]*q;
   ycom(i)=[0,0,0,matrix(i,2)/1000,matrix(i,1)/1000,1,-
y(i)*matrix(i,1)/1000,-y(i)*matrix(i,2)/1000]*q;
end
u=[xcom',ycom']
set(handles.uitable4,'data',u)ک
```

برنامه levelling(ترازیابی)

تصویر زیر شمای کلی برنامه ترازیابی را نشان می دهد:

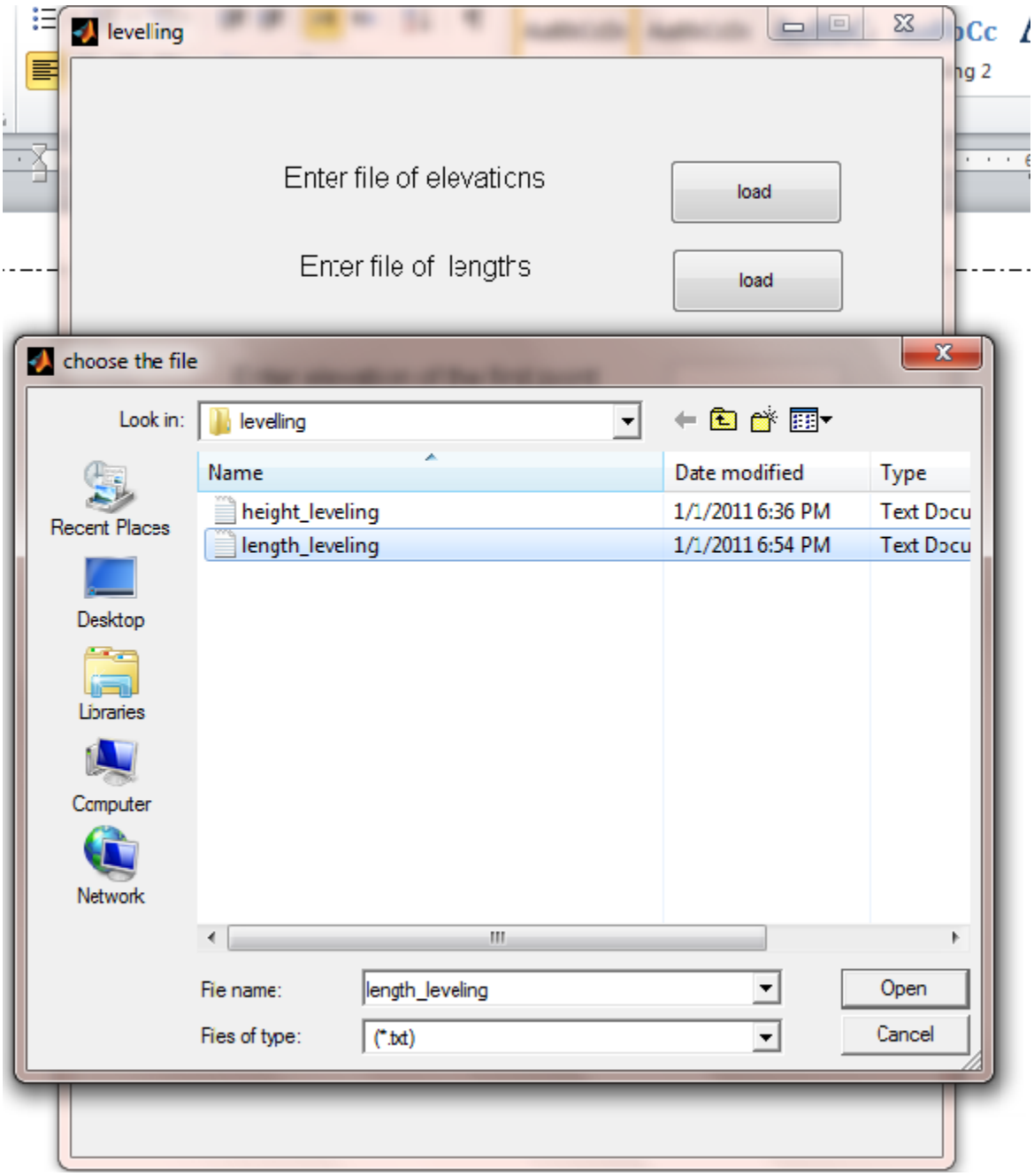


ابتدا روی اولین دکمه daol کلیک می کنیم و از منوی باز شده فایل ارتفاعات را وارد می کنیم تا اطلاعات اتفاعی مورد نظر وارد برنامه شود.

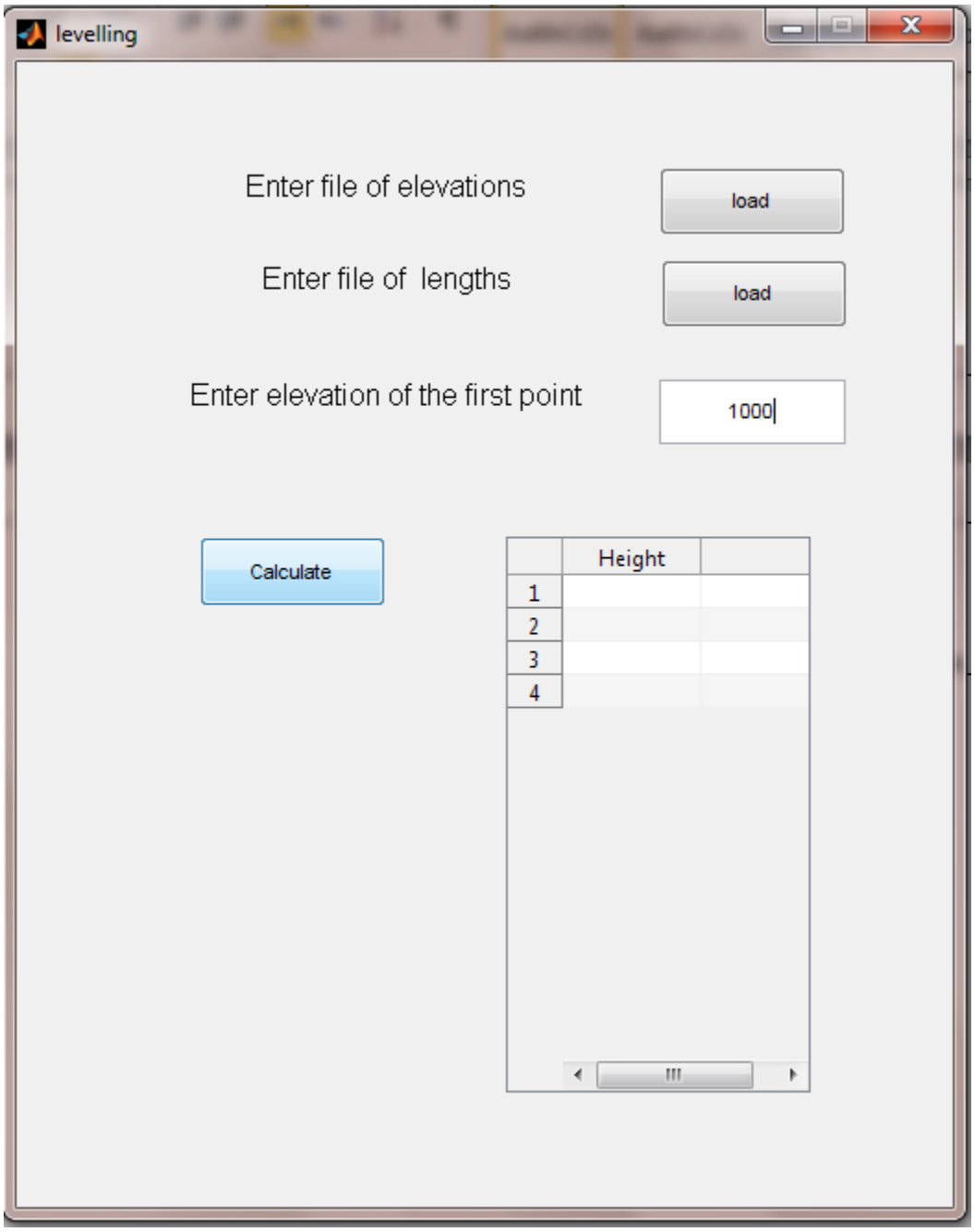


سپس با استفاده از دومین دکمه daol فایل مربوط به فاصله بین ایستگاه ها را وارد می کنیم.

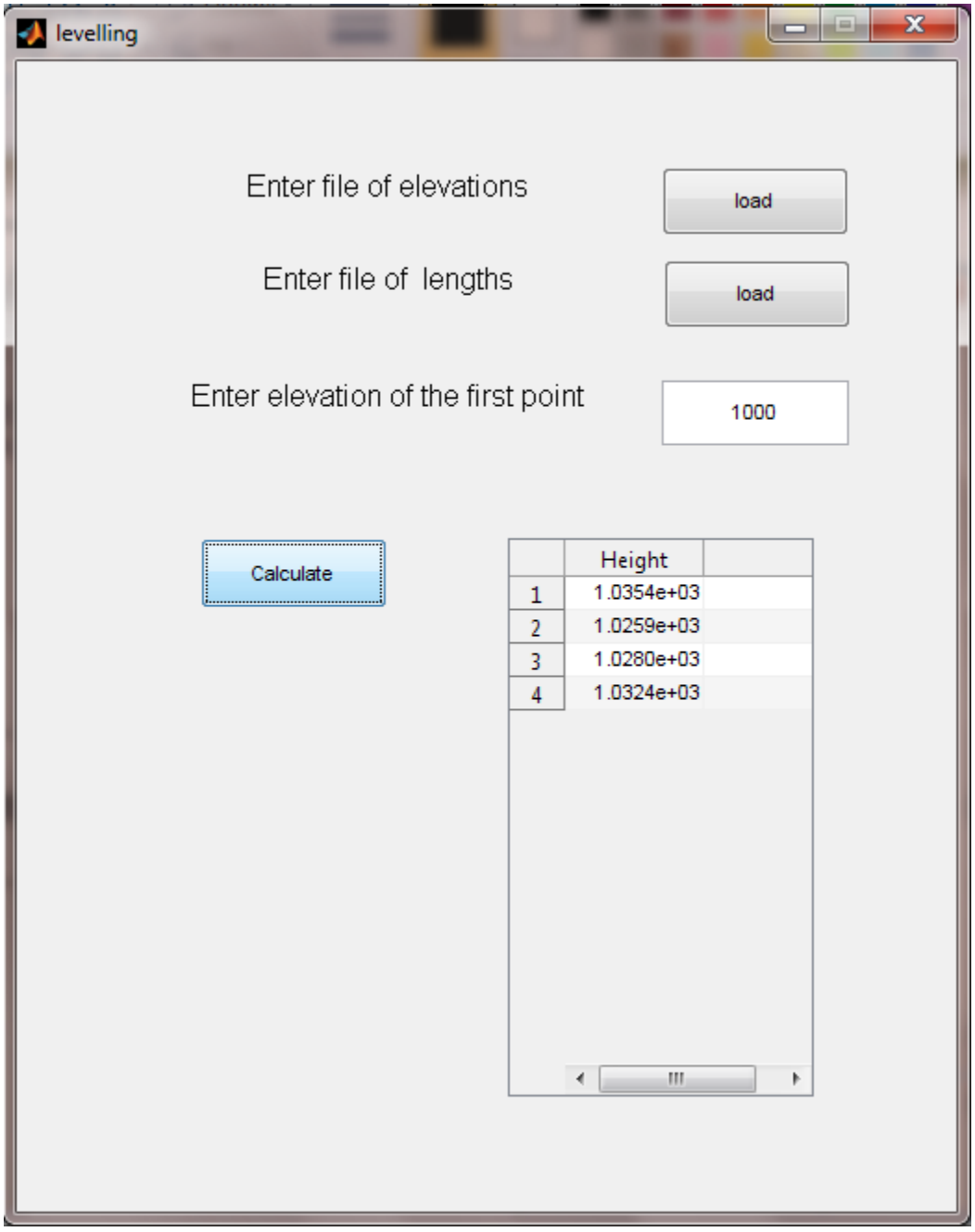
از طول ها برای تشکیل ماریس وزن استفاده می شود به طوریکه عکس طول و یا عکس طول به توان دو به عنوان وزن هر داده استفاده می کنیم.

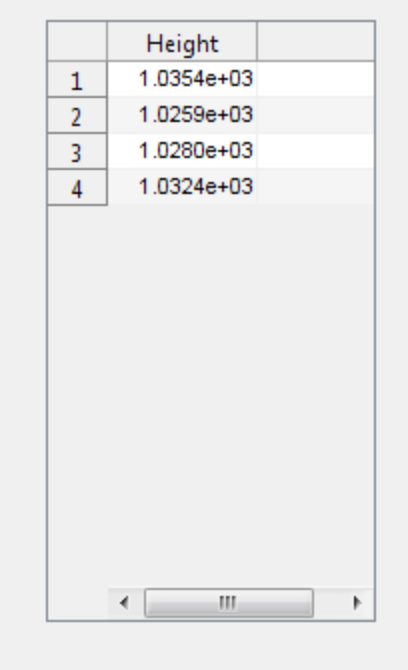


سپس ارتفاع مربوط به اولین نقطه یا همان نقطه کنترل را ارد می کنیم و کلید اینتر را میزنیم.



در آخر هم با زدن دکمه ی etaluclac ارتفاع های تصحیح شده به روش کمترین مربعات در جدول نشان داده می شود.






```
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

height=get(handles.edit1,'string')
height=str2num(height)
for i=1:length(handles.elevation)
    if handles.elevation(1,i)~=0
        handles.elevation(1,i)=handles.elevation(1,i)+height
    end
end
for i=1:length(handles.elevation)
    if handles.elevation(i,1)~=0
        handles.elevation(i,1)=handles.elevation(i,1)-height
    end
end
counter=0
for i=1:length(handles.elevation)
    for j=1:length(handles.elevation)
        if handles.elevation(i,j)~=0
            counter=counter+1
            if i==1
                A(counter,j-1)=1
                L(counter,1)=handles.elevation(i,j)
            elseif j==1
                A(counter,i-1)=-1
                L(counter,1)=handles.elevation(i,j)
            else
                A(counter,i-1)=-1
                A(counter,j-1)=1
                L(counter,1)=handles.elevation(i,j)
            end
```

```
        end
    end
end
counter=1
for i=1:length(handles.length)
    for j=1:length(handles.length)
        if handles.length(i,j)~=0
            p=1/handles.length(i,j)
            W(counter,counter)=p
            counter=counter+1

        end
    end
end
X=((A'*W*A)^(-1))*A'*W*L
set(handles.uitable1,'data',X)

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[x,y]=ginput(1)
plot(x,y,'*')
hold on

m=handles.plotting
m=m+1
handles.plotting=num2str(handles.plotting)
text(x,y,handles.plotting)
handles.plotting=str2num(handles.plotting)
```

handles =

figure1: 170.0016
pushbutton4: 6.0017
pushbutton3: 5.0017
text3: 4.0017
edit1: 3.0017
text2: 2.0017
uitable1: 1.0017
pushbutton1: 0.0017
text1: 171.0016

output: 170.0016
plotting: 0

filename =

height_leveling.txt

pathname =

C:\Users\Simin\Desktop\simin\matlb\barnamehaye doctor moatagh\levelling\

file =

C:\Users\Simin\Desktop\simin\matlb\barnamehaye doctor moatagh\levelling\height_leveling.txt

handles =

figure1: 170.0016
pushbutton4: 6.0017
pushbutton3: 5.0017
text3: 4.0017
edit1: 3.0017
text2: 2.0017
uitable1: 1.0017
pushbutton1: 0.0017
text1: 171.0016

```
        output: 170.0016
      plotting: 0
     elevation: [5x5 double]


filename =

length_leveling.txt


pathname =

C:\Users\Simin\Desktop\simin\matlb\barnamehaye doctor moatagh\levelling\


file =

C:\Users\Simin\Desktop\simin\matlb\barnamehaye doctor
moatagh\levelling\length_leveling.txt


handles =

       figure1: 170.0016
   pushbutton4: 6.0017
   pushbutton3: 5.0017
         text3: 4.0017
         edit1: 3.0017
         text2: 2.0017
     uitable1: 1.0017
   pushbutton1: 0.0017
```

```
          text1: 171.0016
         output: 170.0016
       plotting: 0
     elevation: [5x5 double]
         length: [5x5 double]


height =

1000


height =

        1000


handles =

        figure1: 170.0016
    pushbutton4: 6.0017
    pushbutton3: 5.0017
          text3: 4.0017
          edit1: 3.0017
          text2: 2.0017
       uitable1: 1.0017
    pushbutton1: 0.0017
          text1: 171.0016
         output: 170.0016
       plotting: 0
     elevation: [5x5 double]
```

```
        length: [5x5 double]


handles =

       figure1: 170.0016
   pushbutton4: 6.0017
   pushbutton3: 5.0017
         text3: 4.0017
         edit1: 3.0017
         text2: 2.0017
      uitable1: 1.0017
   pushbutton1: 0.0017
         text1: 171.0016
        output: 170.0016
      plotting: 0
     elevation: [5x5 double]
        length: [5x5 double]


handles =

       figure1: 170.0016
   pushbutton4: 6.0017
   pushbutton3: 5.0017
         text3: 4.0017
         edit1: 3.0017
         text2: 2.0017
      uitable1: 1.0017
   pushbutton1: 0.0017
         text1: 171.0016
```

output: 170.0016
plotting: 0
elevation: [5x5 double]
length: [5x5 double]

counter =

0

counter =

1

A =

1

L =

1.0254e+003

counter =

2

```
A =

     1
    -1


A =

     1     0     0
    -1     0     1


L =

  1.0e+003 *

    1.0254
   -0.0155


counter =

     3


A =

     1     0     0
    -1     0     1
     0    -1     0
```

```
L =

  1.0e+003 *

    1.0254
   -0.0155
   -1.0352


counter =

     4


A =

     1     0     0
    -1     0     1
     0    -1     0
     0    -1     0


A =

     1     0     0
    -1     0     1
     0    -1     0
     1    -1     0
```

```
L =

  1.0e+003 *

    1.0254
   -0.0155
   -1.0352
    0.0103


counter =

     5


A =

     1     0     0
    -1     0     1
     0    -1     0
     1    -1     0
     0     0    -1


A =

     1     0     0
    -1     0     1
     0    -1     0
     1    -1     0
     0     1    -1
```

```
L =

  1.0e+003 *

    1.0254
   -0.0155
   -1.0352
    0.0103
   -0.0261


counter =

     6


A =

     1     0     0
    -1     0     1
     0    -1     0
     1    -1     0
     0     1    -1
     0     0    -1


A =

     1     0     0     0
```

```
    -1     0     1     0
     0    -1     0     0
     1    -1     0     0
     0     1    -1     0
     0     0    -1     1


L =

  1.0e+003 *

     1.0254
    -0.0155
    -1.0352
     0.0103
    -0.0261
     0.0213


counter =

     7


A =

     1     0     0     0
    -1     0     1     0
     0    -1     0     0
     1    -1     0     0
     0     1    -1     0
```

```
     0     0    -1     1
     0     0     0    -1


L =

  1.0e+003 *

     1.0254
    -0.0155
    -1.0352
     0.0103
    -0.0261
     0.0213
    -1.0310


counter =

     8


A =

     1     0     0     0
    -1     0     1     0
     0    -1     0     0
     1    -1     0     0
     0     1    -1     0
     0     0    -1     1
     0     0     0    -1
```

```
     0     0     0    -1
```

A =

```
     1     0     0     0
    -1     0     1     0
     0    -1     0     0
     1    -1     0     0
     0     1    -1     0
     0     0    -1     1
     0     0     0    -1
     0     1     0    -1
```

L =

```
  1.0e+003 *

    1.0254
   -0.0155
   -1.0352
    0.0103
   -0.0261
    0.0213
   -1.0310
    0.0048
```

counter =

```
     1


p =

    0.0552


W =

    0.0552


counter =

     2


p =

    0.0568


W =

    0.0552         0
         0    0.0568


counter =
```

```
     3


p =

    0.0704


W =

    0.0552         0         0
         0    0.0568         0
         0         0    0.0704


counter =

     4


p =

    0.1064


W =

    0.0552         0         0         0
         0    0.0568         0         0
         0         0    0.0704         0
         0         0         0    0.1064
```

```
counter =

     5


p =

    0.0714


W =

    0.0552         0         0         0         0
         0    0.0568         0         0         0
         0         0    0.0704         0         0
         0         0         0    0.1064         0
         0         0         0         0    0.0714


counter =

     6


p =

    0.0741
```

W =

```
   0.0552         0         0         0         0         0
        0    0.0568         0         0         0         0
        0         0    0.0704         0         0         0
        0         0         0    0.1064         0         0
        0         0         0         0    0.0714         0
        0         0         0         0         0    0.0741
```

counter =

```
     7
```

p =

```
   0.0725
```

W =

```
   0.0552         0         0         0         0         0         0
        0    0.0568         0         0         0         0         0
        0         0    0.0704         0         0         0         0
        0         0         0    0.1064         0         0         0
        0         0         0         0    0.0714         0         0
        0         0         0         0         0    0.0741         0
        0         0         0         0         0         0    0.0725
```

counter =

8

p =

0.1010

W =

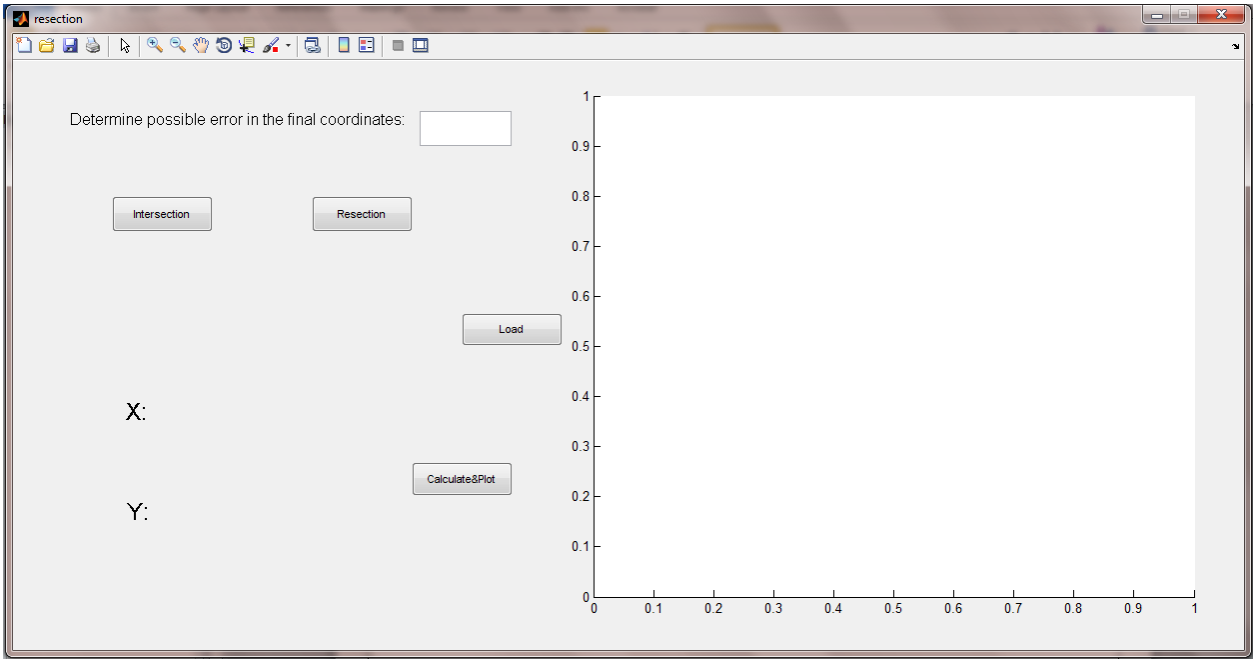
0.0552	0	0	0	0	0	0	0
0	0.0568	0	0	0	0	0	0
0	0	0.0704	0	0	0	0	0
0	0	0	0.1064	0	0	0	0
0	0	0	0	0.0714	0	0	0
0	0	0	0	0	0.0741	0	0
0	0	0	0	0	0	0.0725	0
0	0	0	0	0	0	0	0.1010

counter =

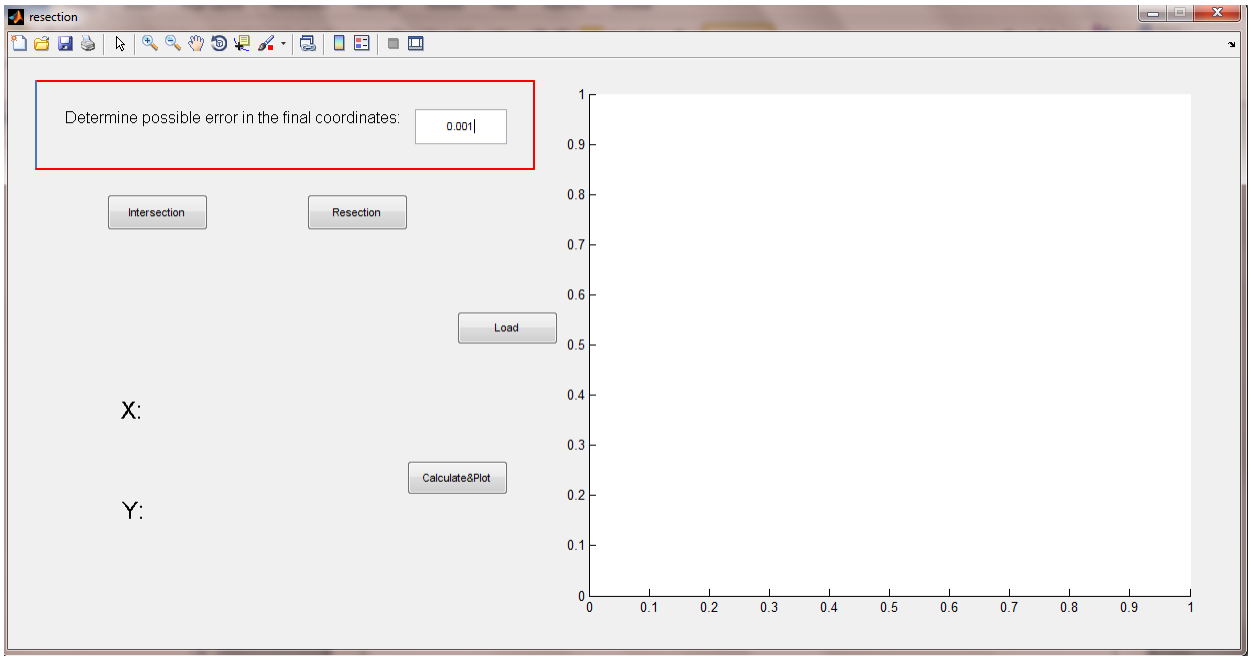
9

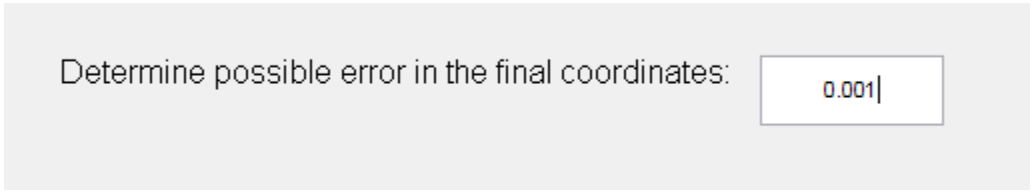
برنامه intersection & resection

شمای کلی برنامه ترفیع و تقاطع در شکل زیر نشان داده شده است:



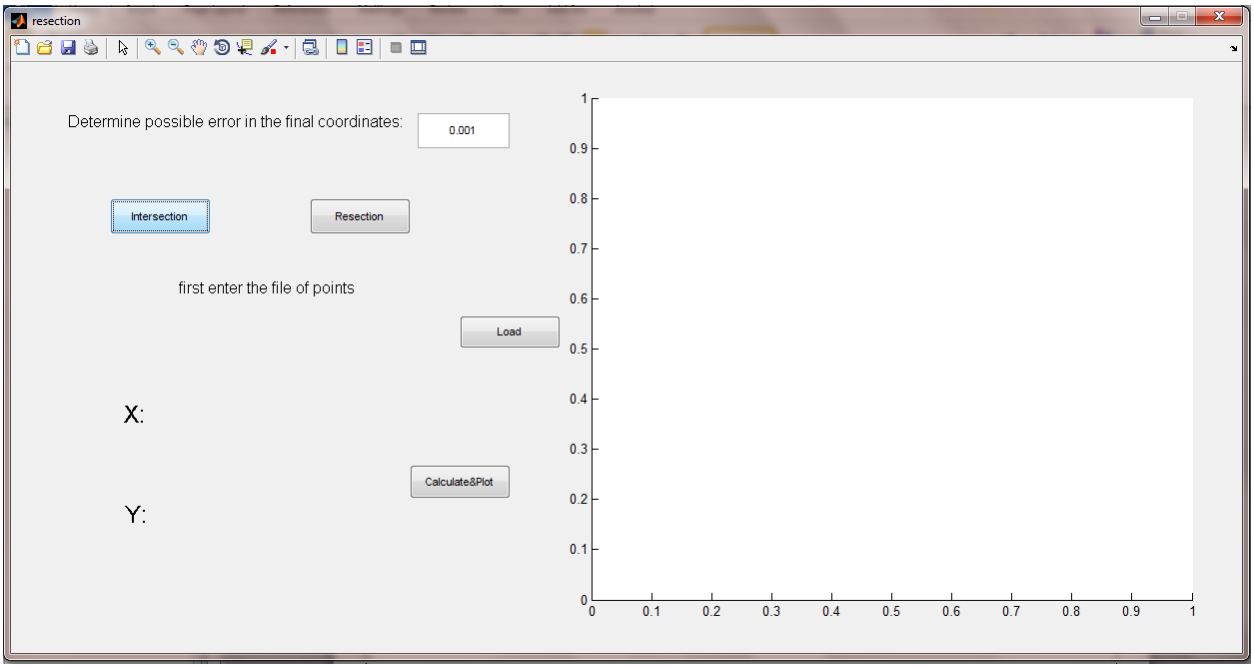
در این برنامه ابتدا در کادر سفید خطای کلی مربوط به کل مختصات ها را وارد می کنیم وسپس اینتر می زنیم.



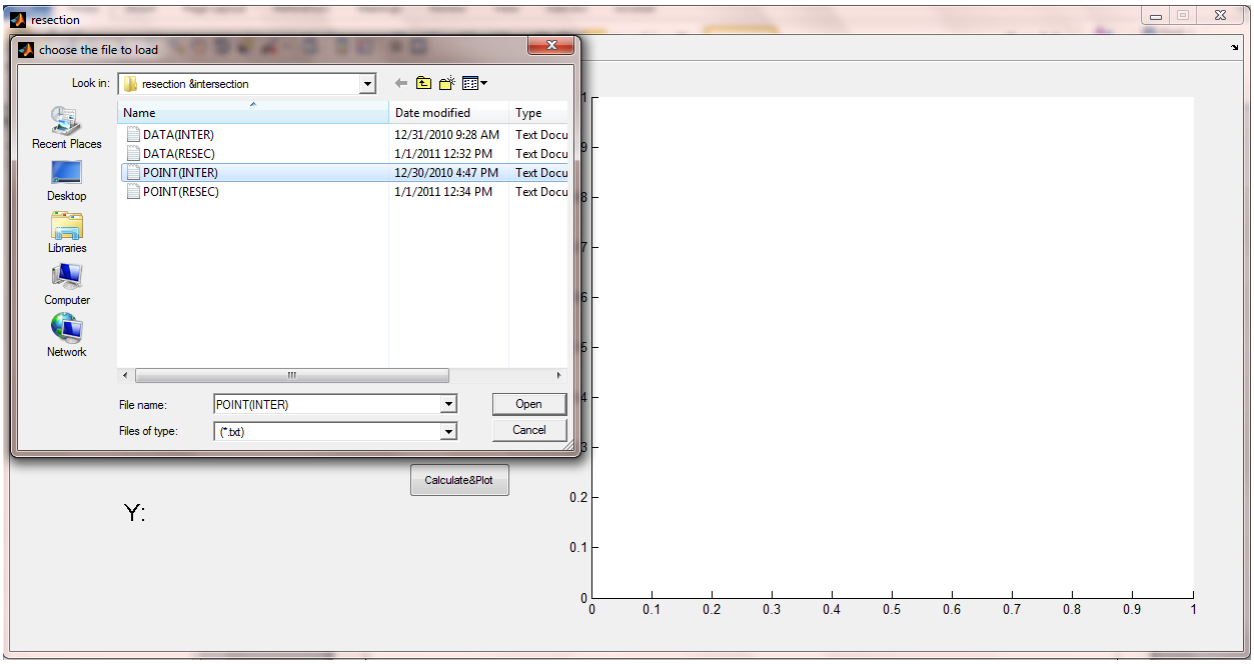


سپس نوع عملیات خود را که ترفیع است یا تقاطع انتخاب می کنیم.

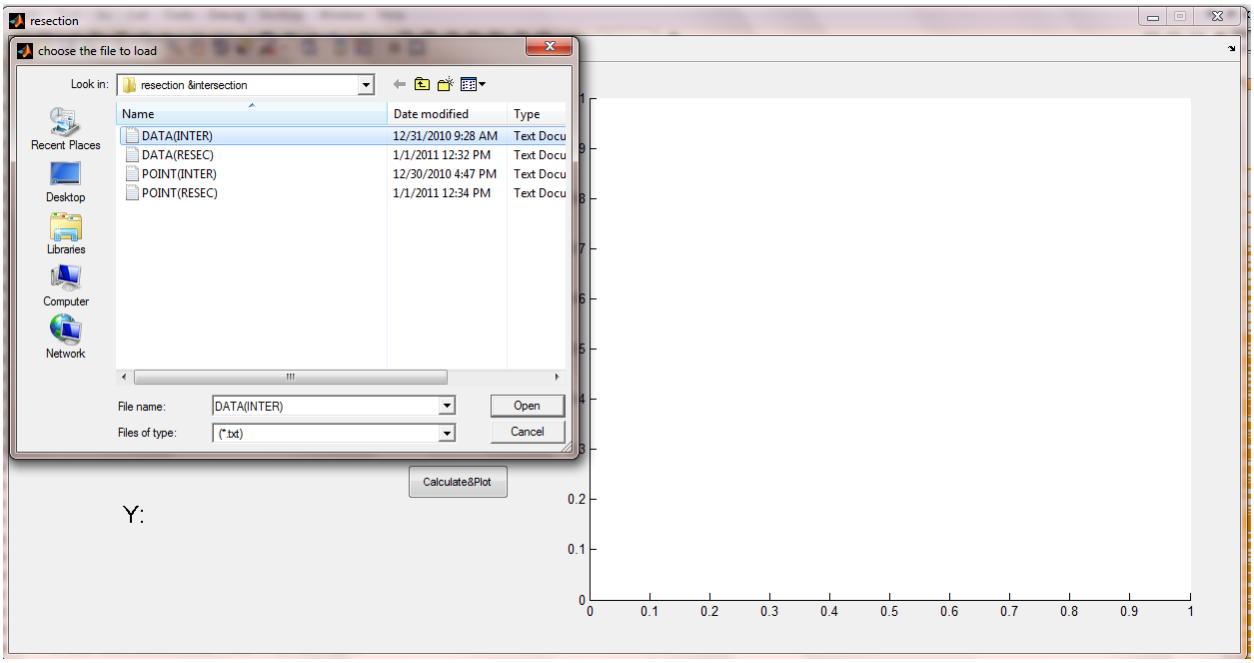
ابتدا ما از عملیات etnisrnoitce آغاز می کنیم.



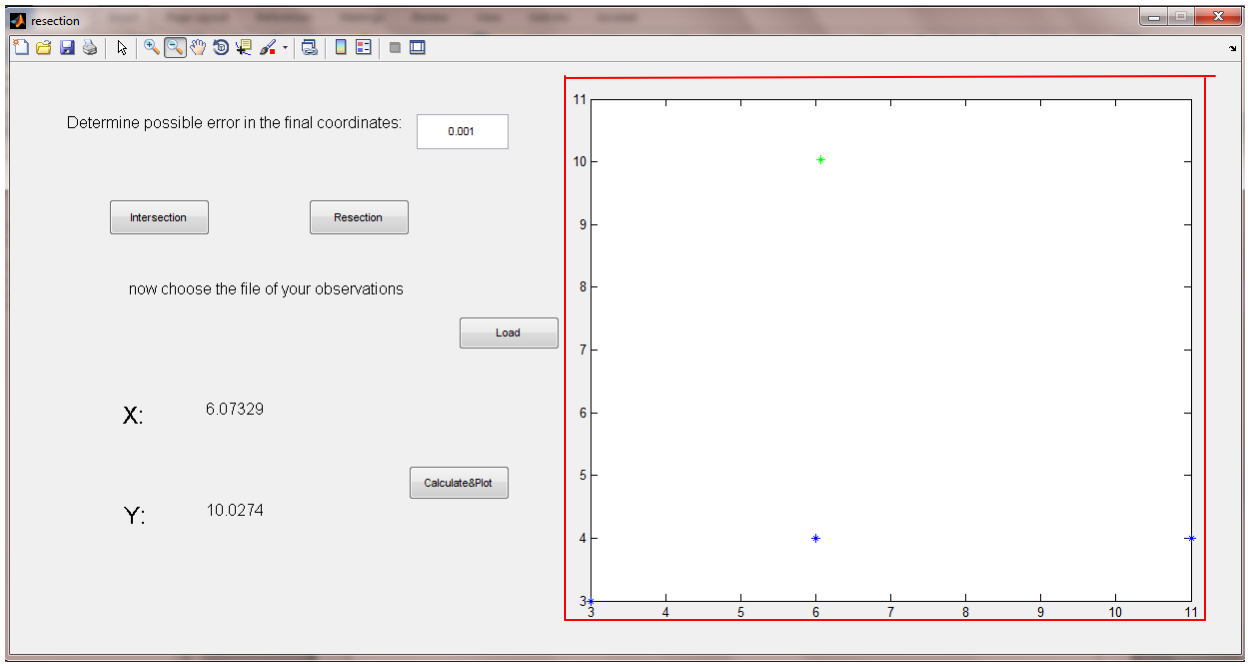
سپس دکمه ی daol را می زنیم و از منوی باز شده فایل tniop(retni) را انتخاب می کنیم.

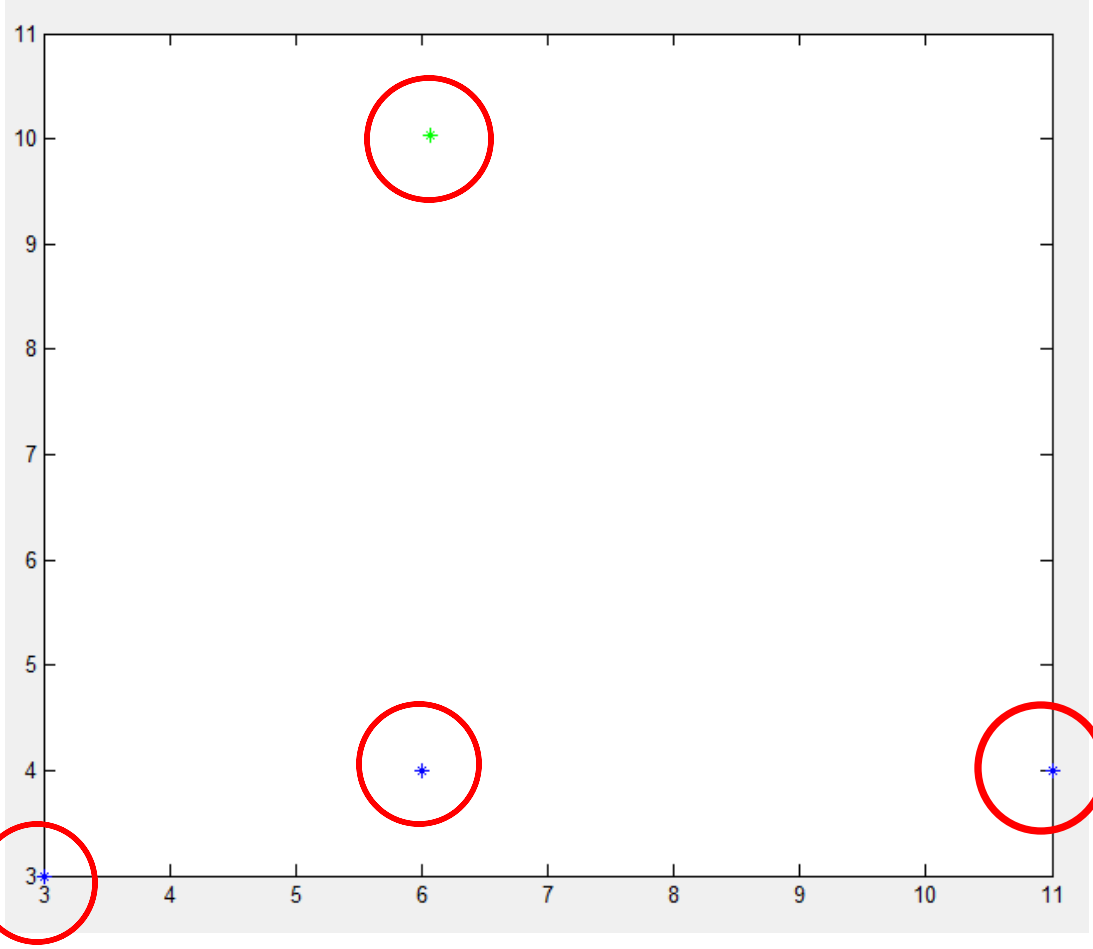


سپس دوباره دکمه ی daol را می زنیم و از منوی باز شده فایل ataD(retni) را انتخاب می کنیم.

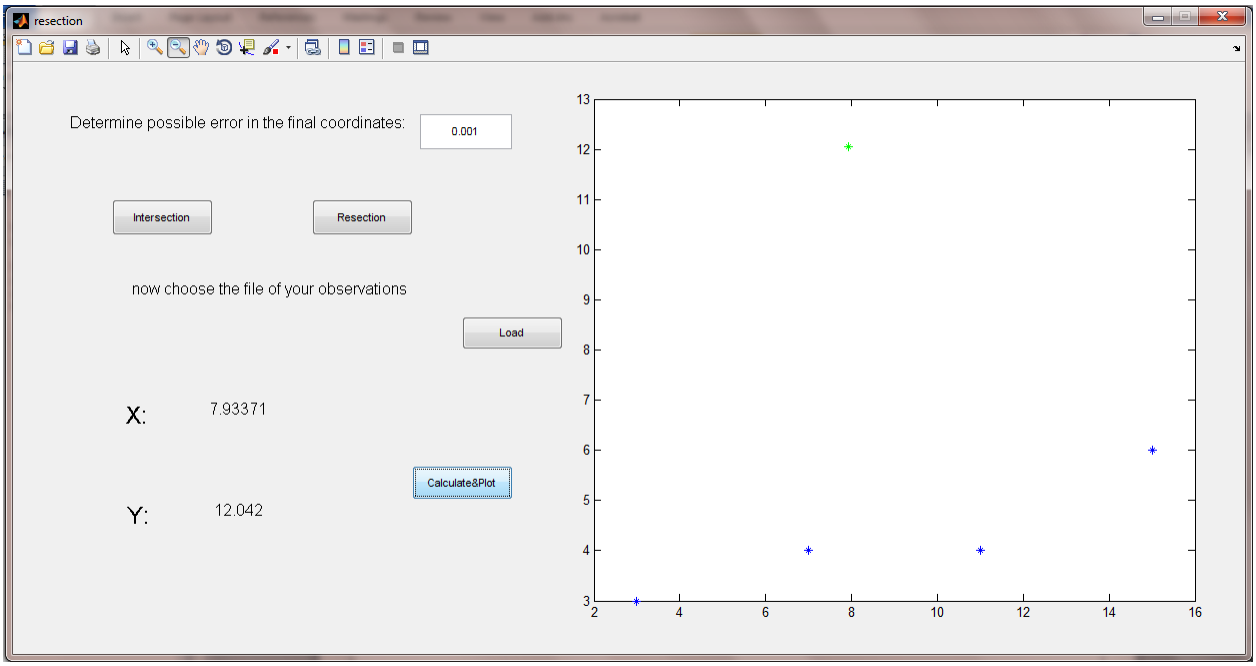


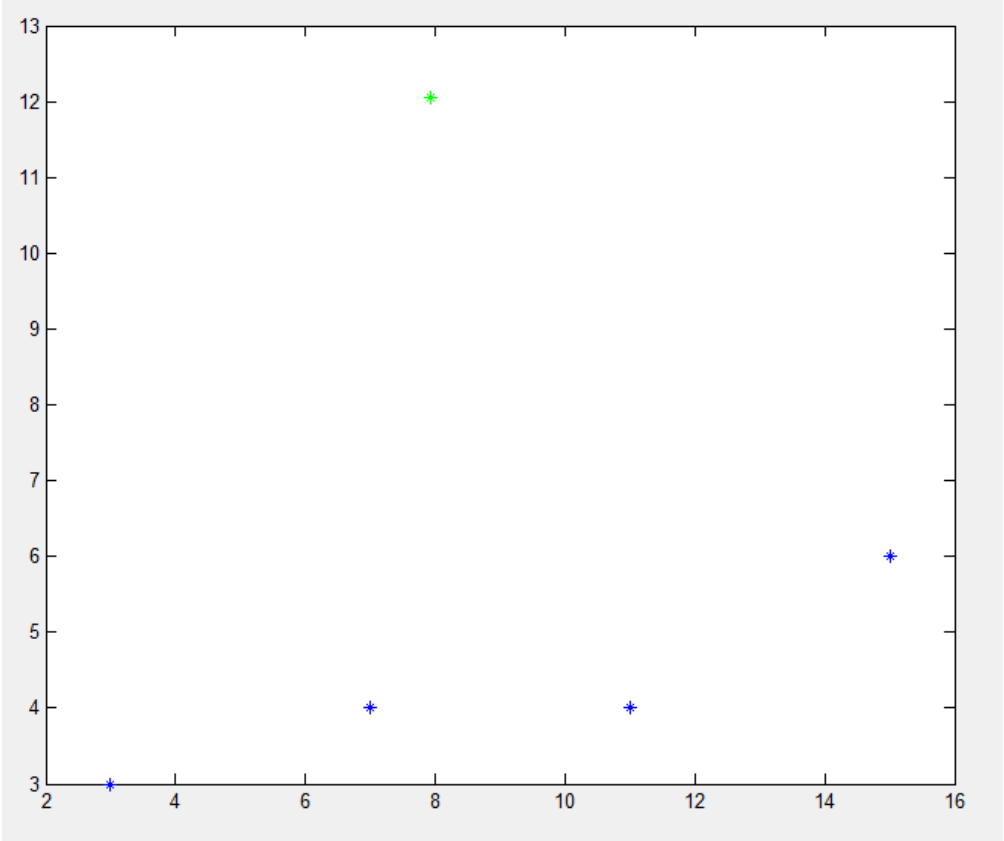
با زدن دکمه etaluclac & tolp مسئله حل می شود و نمودار مربوط به آن رسم می شود.





برای noitceserهم طبق بالا عمل می کنیم.و در نتیجه داریم:





```
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
for i=1:length(handles.coor)
   for j=1:length(handles.coor)
    if i~=j
        handles.obser(i,j)=pi*handles.obser(i,j)/180
        guidata(hObject,handles)
    end
   end
end
if handles.choosing==0
    shomarandeh=0
for i=1:(length(handles.coor)-1)
    if handles.obser(i,i)~=0&&handles.obser(i+1,i+1)~=0
        distance=sqrt((handles.coor(i+1,1)-
handles.coor(i,1))^2+(handles.coor(i+1,2)-handles.coor(i,2))^2)
        cosine=(handles.obser(i,i)^2+handles.obser(i+1,i+1)^2-
distance^2)/(2*handles.obser(i,i)*handles.obser(i+1,i+1))
        teta=acos(cosine)
        sinus=handles.obser(i+1,i+1)*sin(teta)/distance
        sigma=asin(sinus)
        if handles.obser(i+1,i+1)^2>handles.obser(i,i)^2+distance^2
            if sigma>0
                sigma=pi-sigma
            end
            if sigma<0
                sigma=-pi-sigma
            end
        end
        if sigma<0
            sigma=2*pi+sigma
        end
        g=gizman(handles.coor(i+1,1)-handles.coor(i,1),handles.coor(i+1,2)-
handles.coor(i,2))
        gizz=g-sigma
        x0(i,1)=handles.coor(i,1)+handles.obser(i,i)*sin(gizz)
        y0(i,1)=handles.coor(i,2)+handles.obser(i,i)*cos(gizz)
    end
end
shomarandeh=shomarandeh+1
x_avvalieh(shomarandeh,1)=sum(x0)/length(x0)
y_avvalieh(shomarandeh,1)=sum(y0)/length(y0)
    if handles.obser(i,i+1)~=0&&handles.obser(i+1,i)~=0
        if handles.obser(i,i+1)>pi
            handles.obser(i,i+1)=2*pi-handles.obser(i,i+1)
            parcham=0
        end
        if handles.obser(i+1,i)>pi
            handles.obser(i+1,i)=2*pi-handles.obser(i+1,i)
            parcham=1
        end
        handles.obser
        teta=pi-(handles.obser(i,i+1)+handles.obser(i+1,i))
        distance=sqrt((handles.coor(i+1,1)-
handles.coor(i,1))^2+(handles.coor(i+1,2)-handles.coor(i,2))^2)
```

```
        l=distance*sin(handles.obser(i+1,i))/sin(teta)
        if parcham==0
            handles.obser(i,i+1)==2*pi-handles.obser(i,i+1)
        end
        if parcham==1
            handles.obser(i+1,i)=2*pi-handles.obser(i+1,i)
        end
        g=gizman(handles.coor(i+1,1)-handles.coor(i,1),handles.coor(i+1,2)-
handles.coor(i,2))
        gizz=g-handles.obser(i,i+1)
        x0(i,1)=handles.coor(i,1)+l*sin(gizz)
        y0(i,1)=handles.coor(i,2)+l*cos(gizz)
    end
    shomarandeh=shomarandeh+1
    x_avvalieh(shomarandeh,1)=sum(x0)/length(x0)
    y_avvalieh(shomarandeh,1)=sum(y0)/length(y0)
    x0=sum(x_avvalieh)/length(x_avvalieh)
    y0=sum(y_avvalieh)/length(y_avvalieh)

    k=1
for i=1:length(handles.coor)
    for j=1:length(handles.coor)
    if i==j&&handles.obser(i,j)~=0
        l(k,1)=sqrt((x0-handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
        dl(k,1)=handles.obser(i,j)-l(k,1)
        moshtag(k,1)=(x0-handles.coor(i,1))/sqrt((x0-
handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
        moshtag(k,2)=(y0-handles.coor(i,2))/sqrt((x0-
handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
        k=k+1
    end
    if i~=j&&handles.obser(i,j)~=0
        l(k,1)=gizman(handles.coor(j,1)-handles.coor(i,1),handles.coor(j,2)-
handles.coor(i,2))-gizman(x0-handles.coor(i,1),y0-handles.coor(i,2))
       if l(k,1)<0
           l(k,1)=2*pi+l(k,1)
       end
        dl(k,1)=handles.obser(i,j)-l(k,1)
        moshtag(k,1)=-(y0-handles.coor(i,2))/((x0-handles.coor(i,1))^2+(y0-
handles.coor(i,2))^2)
        moshtag(k,2)=(x0-handles.coor(i,1))/((x0-handles.coor(i,1))^2+(y0-
handles.coor(i,2))^2)
        k=k+1
    end
    end
end
DX=((moshtag'*moshtag)^(-1))*moshtag'*dl
x0=x0+DX(1,1)
y0=y0+DX(2,1)
e=get(handles.edit1,'string')
e=str2double(e)
E=[e;e]
while(abs(DX)>=E)

k=1
for i=1:length(handles.coor)
```

```
    for j=1:length(handles.coor)
    if i==j&&handles.obser(i,j)~=0
        l(k,1)=sqrt((x0-handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
        dl(k,1)=handles.obser(i,j)-l(k,1)
        moshtag(k,1)=(x0-handles.coor(i,1))/l(k,1)
        moshtag(k,2)=(y0-handles.coor(i,2))/l(k,1)
        k=k+1
    end
    if i~=j&&handles.obser(i,j)~=0
        l(k,1)=gizman(handles.coor(j,1)-handles.coor(i,1),handles.coor(j,2)-
handles.coor(i,2))-gizman(x0-handles.coor(i,1),y0-handles.coor(i,2))
        if l(k,1)<0
            l(k,1)=2*pi+l(k,1)
        end
        dl(k,1)=handles.obser(i,j)-l(k,1)
        moshtag(k,1)=-(y0-handles.coor(i,2))/(l(k,1)^2)
        moshtag(k,2)=(x0-handles.coor(i,1))/(l(k,1)^2)
        k=k+1
    end
    end
end

DX=((moshtag'*moshtag)^(-1))*moshtag'*dl
x0=x0+DX(1,1)
y0=y0+DX(2,1)
end

end
if handles.choosing==1
    shomarandeh=0
     for i=1:(length(handles.coor)-2)
         if handles.obser(i,i+1)~=0&&handles.obser(i+1,i+2)~=0
             zavieh=gizman(handles.coor(i+2,1)-
handles.coor(i+1,1),handles.coor(i+2,2)-handles.coor(i+1,2))-
gizman(handles.coor(i,1)-handles.coor(i+1,1),handles.coor(i,2)-
handles.coor(i+1,2))
             if zavieh<0
                 zavieh=2*pi+zavieh
             end
             G=2*pi-(handles.obser(i,i+1)+handles.obser(i+1,i+2)+zavieh)
             distance1=sqrt((handles.coor(i+1,1)-
handles.coor(i,1))^2+(handles.coor(i+1,2)-handles.coor(i,2))^2)
             distance2=sqrt((handles.coor(i+2,1)-
handles.coor(i+1,1))^2+(handles.coor(i+2,2)-handles.coor(i+1,2))^2)

h=(distance2*sin(handles.obser(i,i+1)))/(distance1*sin(handles.obser(i+1,i+2)
))
             angle=atan(sin(G)/(h+cos(G)))
             otherangle=pi-(handles.obser(i+1,i+2)+angle)
             distance3=distance2*sin(otherangle)/sin(handles.obser(i+1,i+2))
             az=gizman(handles.coor(i+1,1)-
handles.coor(i+2,1),handles.coor(i+1,2)-handles.coor(i+2,2))
             azimuth=az+angle
             x0(i,1)=handles.coor(i+2,1)+distance3*sin(azimuth)
             y0(i,1)=handles.coor(i+2,2)+distance3*cos(azimuth)
         end
```

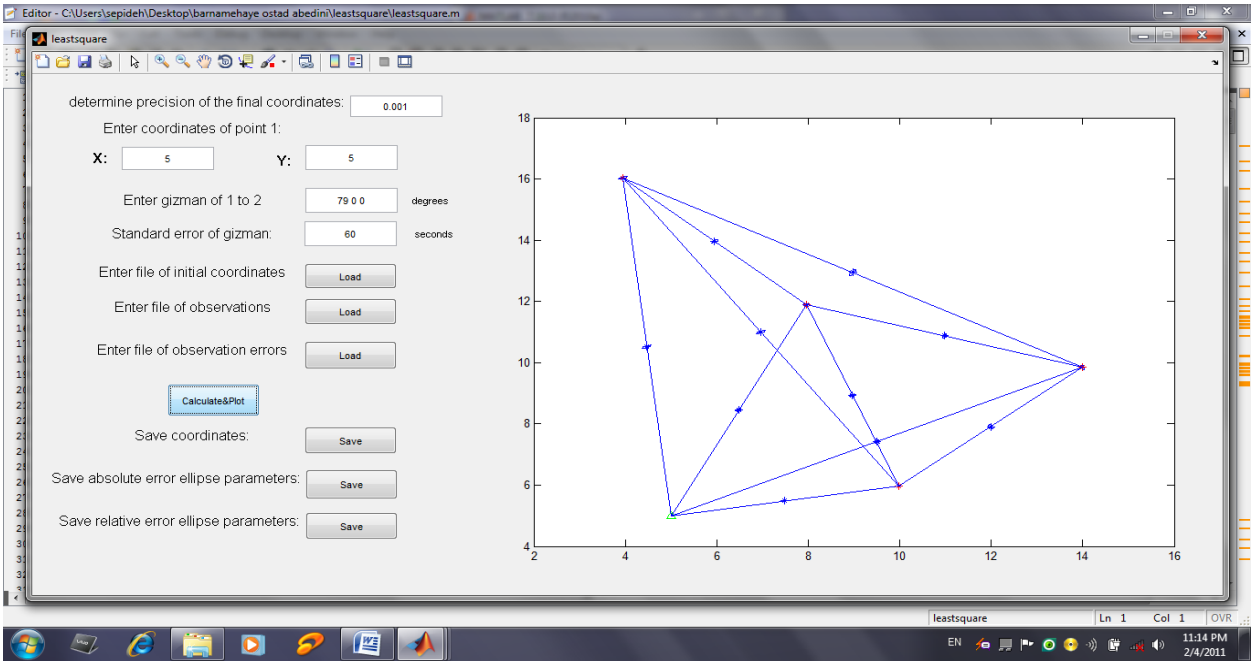
```
    end
    shomarandeh=shomarandeh+1
    x_avvalieh(shomarandeh,1)=sum(x0)/length(x0)
    y_avvalieh(shomarandeh,1)=sum(y0)/length(y0)
    for i=1:(length(handles.coor)-1)
        if handles.obser(i,i)~=0&&handles.obser(i+1,i+1)~=0
        distance=sqrt((handles.coor(i+1,1)-
handles.coor(i,1))^2+(handles.coor(i+1,2)-handles.coor(i,2))^2)
       cosine=(handles.obser(i,i)^2+handles.obser(i+1,i+1)^2-
distance^2)/(2*handles.obser(i,i)*handles.obser(i+1,i+1))
        teta=acos(cosine)
        sinus=handles.obser(i+1,i+1)*sin(teta)/distance
        sigma=asin(sinus)
        if handles.obser(i+1,i+1)^2>handles.obser(i,i)^2+distance^2
            if sigma>0
                sigma=pi-sigma
            end
            if sigma<0
                sigma=-pi-sigma
            end
        end
        if sigma<0
            sigma=2*pi+sigma
        end
        g=gizman(handles.coor(i+1,1)-handles.coor(i,1),handles.coor(i+1,2)-
handles.coor(i,2))
        gizz=g-sigma
        x0(i,1)=handles.coor(i,1)+handles.obser(i,i)*sin(gizz)
        y0(i,1)=handles.coor(i,2)+handles.obser(i,i)*cos(gizz)
        end
    end
    shomarandeh=shomarandeh+1
    x_avvalieh(shomarandeh,1)=sum(x0)/length(x0)
    y_avvalieh(shomarandeh,1)=sum(y0)/length(y0)
    x0=sum(x_avvalieh)/length(x_avvalieh)
    y0=sum(y_avvalieh)/length(y_avvalieh)
    k=1
    for i=1:length(handles.coor)
        for j=1:length(handles.coor)
           if i==j&&handles.obser(i,j)~=0
              l(k,1)=sqrt((x0-handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
              dl(k,1)=handles.obser(i,j)-l(k,1)
              moshtag(k,1)=(x0-handles.coor(i,1))/sqrt((x0-
handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
              moshtag(k,2)=(y0-handles.coor(i,2))/sqrt((x0-
handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
              k=k+1
           end
            if i~=j&&handles.obser(i,j)~=0
                l(k,1)=gizman(handles.coor(i,1)-x0,handles.coor(i,2)-y0)-
gizman(handles.coor(i+1,1)-x0,handles.coor(i+1,2)-y0)
                if l(k,1)<0
                    l(k,1)=2*pi+l(k,1)
                end
                dl(k,1)=handles.obser(i,j)-l(k,1)
```

```
                moshtag(k,1)=(-(handles.coor(i,2)-y0)/((handles.coor(i,1)-
x0)^2+(handles.coor(i,2)-y0)^2))+((handles.coor(i+1,2)-
y0)/((handles.coor(i+1,1)-x0)^2+(handles.coor(i+1,2)-y0)^2))
                moshtag(k,2)=((handles.coor(i,1)-x0)/((handles.coor(i,1)-
x0)^2+(handles.coor(i,2)-y0)^2))-((handles.coor(i+1,1)-
x0)/((handles.coor(i+1,1)-x0)^2+(handles.coor(i+1,2)-y0)^2))
                k=k+1
             end
        end
     end
DX=((moshtag'*moshtag)^(-1))*moshtag'*dl
x0=x0+DX(1,1)
y0=y0+DX(2,1)
e=get(handles.edit1,'string')
e=str2double(e)
E=[e;e]
 while(abs(DX)>=E)
       k=1
     for i=1:length(handles.coor)
         for j=1:length(handles.coor)
            if i==j&&handles.obser(i,j)~=0
               l(k,1)=sqrt((x0-handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
               dl(k,1)=handles.obser(i,j)-l(k,1)
               moshtag(k,1)=(x0-handles.coor(i,1))/sqrt((x0-
handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
               moshtag(k,2)=(y0-handles.coor(i,2))/sqrt((x0-
handles.coor(i,1))^2+(y0-handles.coor(i,2))^2)
               k=k+1
            end
             if i~=j&&handles.obser(i,j)~=0
                l(k,1)=gizman(handles.coor(i,1)-x0,handles.coor(i,2)-y0)-
gizman(handles.coor(i+1,1)-x0,handles.coor(i+1,2)-y0)
                if l(k,1)<0
                    l(k,1)=2*pi+l(k,1)
                end
                dl(k,1)=handles.obser(i,j)-l(k,1)
                moshtag(k,1)=(-(handles.coor(i,2)-y0)/((handles.coor(i,1)-
x0)^2+(handles.coor(i,2)-y0)^2))+((handles.coor(i+1,2)-
y0)/((handles.coor(i+1,1)-x0)^2+(handles.coor(i+1,2)-y0)^2))
                moshtag(k,2)=((handles.coor(i,1)-x0)/((handles.coor(i,1)-
x0)^2+(handles.coor(i,2)-y0)^2))-((handles.coor(i+1,1)-
x0)/((handles.coor(i+1,1)-x0)^2+(handles.coor(i+1,2)-y0)^2))
                k=k+1
             end
        end
     end
DX=((moshtag'*moshtag)^(-1))*moshtag'*dl
x0=x0+DX(1,1)
y0=y0+DX(2,1)
 end
end
set(handles.text3,'string',x0)
set(handles.text5,'string',y0)
X=handles.coor(:,1)
Y=handles.coor(:,2)
plot(X,Y,'b*')
hold on
```

برنامه tsael erauqs

روش least square برای حل شبکه که در آن مشاهدات طولی انجام شده وهمچنین ژیزمان امتداد اول به دوم قرائت شده است

بدین منظور برنامه ای به شکل زیر طراحی شده:



شکل1: نمای کلی برنامه

ابتدا کاربر باید سه فایل در محیط نرم افزار Excel ایجاد کند با فرمت های زیر:

1-مختصات اولیه ی نقاط

شماره ی نقاط X Y

2-طول های قرائت شده

0	5.1	10.2	7.5	11.1
5.1	0	5.6	6.3	11.7
10.2	5.6	0	6.4	11.8
7.5	6.3	6.4	0	5.8
11.1	11.7	11.8	5.8	0

دیده می شود که مثلا درایه ی 12 طول بین دو نقطه ی 12 است

3-خطای قرائت هر طول

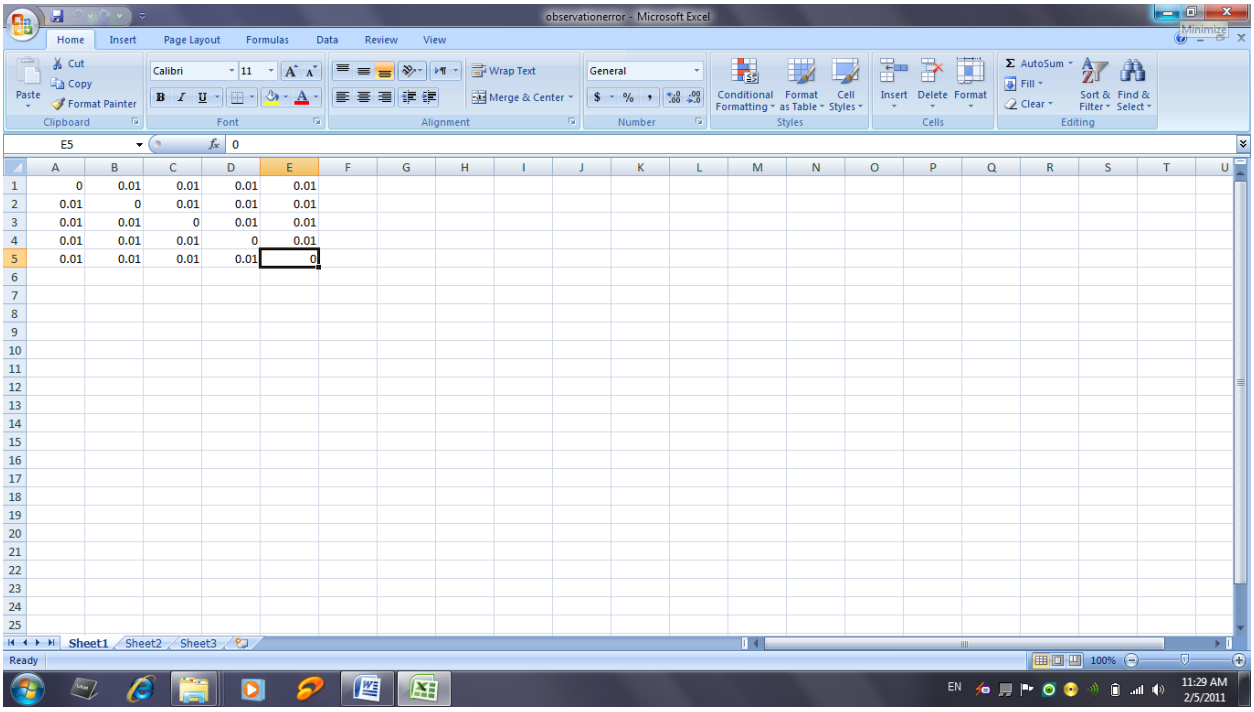
متناظر با هر طول خطای قرائت آن را می نویسیم.

در TEXT BOX اول کاربر دقت نهایی مختصات را برای برنامه مشخص می کند یعنی حلقه ی while تا چه مرحله ای باید ادامه یابد.

شبکه را با این فرض حل می کنیم که مختصات نقطه ی اول به عنوان نقطه ی کنترل بدون هیچ خطایی برای کاربر مشخص باشد پس در TEXT BOX های دوم و سوم کاربر لازم است به ترتیب مختصات X و Y نقطه ی کنترل را وارد کند.

وقتی ژیزمان 2 TO 1 وارد می کنیم باید به صورت (degree minute second) باشد و همچنین دقت ژیزمان نیز باید بر حسب ثانیه وارد شود.

حال توسط pushbutton های load سه فایلی را که قبلا به ترتیب برای مختصات اولیه ی نقاط مجهول، مشاهدات طولی و دقت مشاهدات طولی تهیه کرده بودیم به عنوان ورودی برنامه load می کنیم.



شکل2 :فایل آماده شده برای دقت طول های مشاهده شده

تمام محاسبات مربوط ، رسم شبکه و رسم بیضی های خطای مطلق و نسبی در callback مربوط به pushbutton ، calculate&plot انجام می شود .

برای رسم شبکه پس از بدست آوردن مختصات نهایی و رسم نقاط از تابع line برای وصل دو به دوی نقاط به هم استفاده شده است.

```
for i=1:length(handles.cooravvali)
    for j=i+1:length(handles.cooravvali)
        x=[handles.cooravvali(i,2);handles.cooravvali(j,2)];
        y=[handles.cooravvali(i,3);handles.cooravvali(j,3)];
        line(x,y)
    end
end
```

همچنین برای رسم بیضی های خطا از توابع eig ،axes2ecc و ellipse1 استفاده شده :

یعنی ابتدا ماتریس واریانس-کوواریانس مجهولات را بدست آوردیم که در این مثال چون 4 نقطه ی مجهول داریم یک ماتریس 8*8 خواهد بود .

Sigmaxx=

0.0001- 0.0002 0.0001- 0.0002 0.0005 0.0004 0.0003 0.0014

0.0000- 0.0001- 0.0000 0.0000- 0.0002 0.0000 0.0001 0.0003

0.0009 0.0037 0.0003- 0.0025 0.0022- 0.0025 0.0000 0.0004

0.0014- 0.0051- 0.0003 0.0031- 0.0051 0.0022- 0.0002 0.0005

0.0011 0.0046 0.0001 0.0042 0.0031- 0.0025 0.0000- 0.0002

0.0001- 0.0006- 0.0010 0.0001 0.0003 0.0003- 0.0000 0.0001-

0.0036 0.0101 0.0006- 0.0046 0.0051- 0.0037 0.0001- 0.0002

0.0024 0.0036 0.0001- 0.0011 0.0014- 0.0009 0.0000- 0.0001-

هر بار ماتریس واریانس-کوواریانس یکی از نقاط را استخراج می کنیم

```
for i=2:length(handles.cooravvali)
    ellipse=[sigmaxx(2*(i-1)-1,2*(i-1)-1) sigmaxx(2*(i-1)-1,2*(i-
1));sigmaxx(2*(i-1),2*(i-1)-1) sigmaxx(2*(i-1),2*(i-1))];
    [v,d]=eig(ellipse);
```

حال با بدست آوردن مقادیر ویژه و بردارهای ویژه ی نظیر قطر بزرگ و کوچک بیضی و همچنین جهت گیری قطر بزرگ را نسبت به محور **X** مشخص می کنیم و به صورت زیر به رسم آن و همچنین ذخیره کردن پارامترهای بیضی در یک ماتریس می پردازیم.

```
if d(1,1)>d(2,2)
        semimajordiameter=sqrt(d(1,1));
        semiminordiameter=sqrt(d(2,2));
        majordiameter=2*semimajordiameter;
        minordiameter=2*semiminordiameter;
        angleofmajorwithxaxis=gizman(v(2,1),v(1,1));
        angleofmajorwithxaxistodegree=180*angleofmajorwithxaxis/pi;
        angleofminorwithxaxis=gizman(v(2,2),v(1,2));
        angleofminorwithxaxistodegree=180*angleofminorwithxaxis/pi;
        area=pi*sqrt(semimajordiameter*semiminordiameter);
        ecc=axes2ecc(semimajordiameter,semiminordiameter);

[elat,elon]=ellipse1(handles.cooravvali(i,2),handles.cooravvali(i,3),[semimaj
ordiameter ecc],angleofmajorwithxaxistodegree);
        plot(elat,elon)
        handles.absoluteparameters(i-1,1:7)=[i majordiameter minordiameter
angleofmajorwithxaxistodegree angleofminorwithxaxistodegree ecc area];
        guidata(hObject,handles)
    end
    if d(2,2)>d(1,1)
        semimajordiameter=sqrt(d(2,2));
        semiminordiameter=sqrt(d(1,1));
        majordiameter=2*semimajordiameter;
        minordiameter=2*semiminordiameter;
        angleofmajorwithxaxis=gizman(v(2,2),v(1,2));
        angleofmajorwithxaxistodegree=180*angleofmajorwithxaxis/pi;
        angleofminorwithxaxis=gizman(v(2,1),v(1,1));
        angleofminorwithxaxistodegree=180*angleofminorwithxaxis/pi;
        area=pi*sqrt(semimajordiameter*semiminordiameter);
        ecc=axes2ecc(semimajordiameter,semiminordiameter);

[elat,elon]=ellipse1(handles.cooravvali(i,2),handles.cooravvali(i,3),[semimaj
ordiameter ecc],angleofmajorwithxaxistodegree);
        plot(elat,elon)
```

```
        handles.absoluteparameters(i-1,1:7)=[i majordiameter minordiameter
angleofmajorwithxaxistodegree angleofminorwithxaxistodegree ecc area];
        guidata(hObject,handles)
    end
end
```

در مورد بیضی خطای نسبی نیز وضع به همین منوال است با این تفاوت که هر بار باید یک ماتریس 4*4 از sigmaxx استخراج شود مثلا برای بیضی مربوط بین نقاط 2و5 ماتریس زیر استخراج شود.

Beyzi=

0.0001- 0.0002 0.0003 0.0014

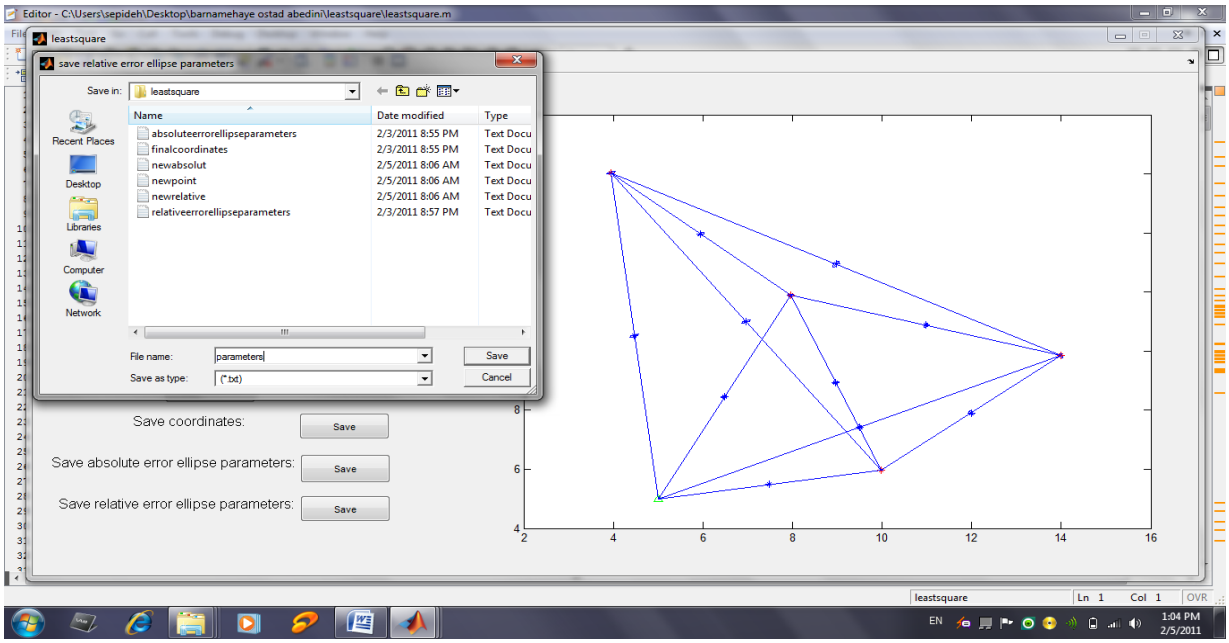
0 0.0001- 0.0001 0.0003

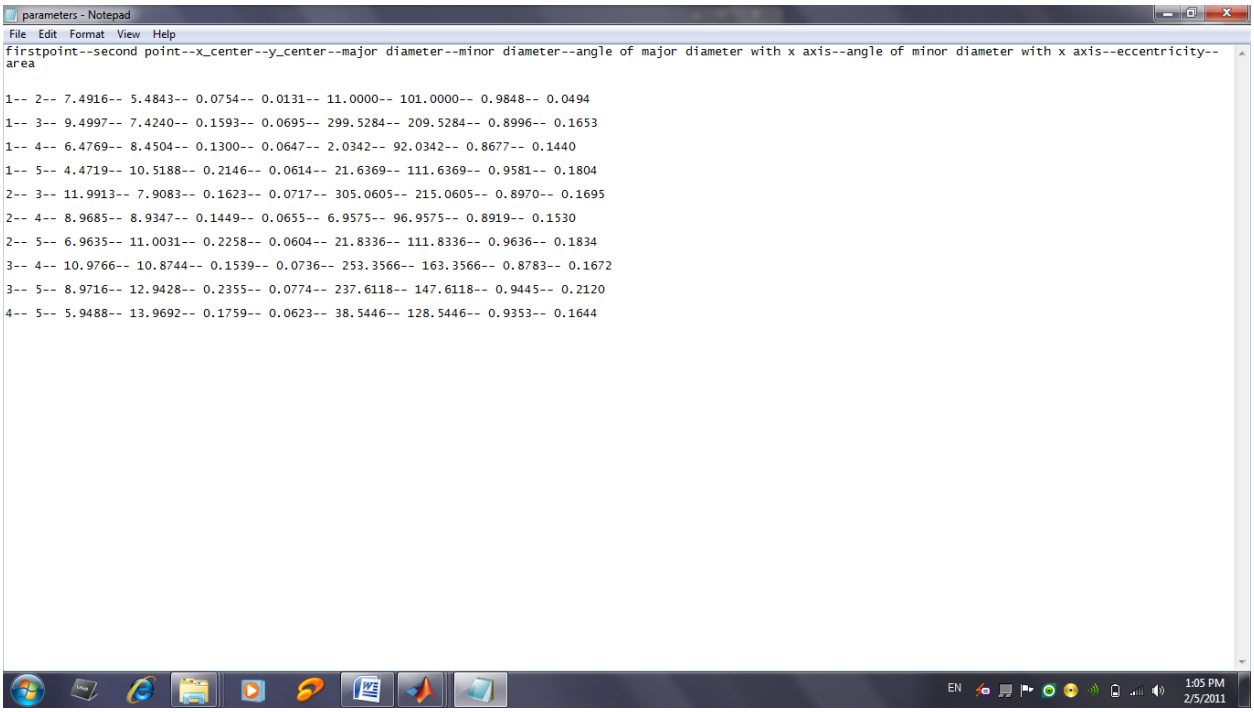
0.0036 0.0101 0.0001- 0.0002

0.0024 0.0036 0 0.0001-

با در نظر گرفتن z=[-1 0 1 0;0 -1 0 1] و ضرب c=z*Beyzi*z' این ماتریس به ماتریس 2*2 ماتریس واریانس_کوواریانس اختلاف مختصات بین 2 و 5 تبدیل شده و از همان روش قبل برای رسم بیضی خطای نسبی استفاده می شود مرکز بیضی نقطه ی وسط بین دو نقطه ی 2 و5 خواهد بود

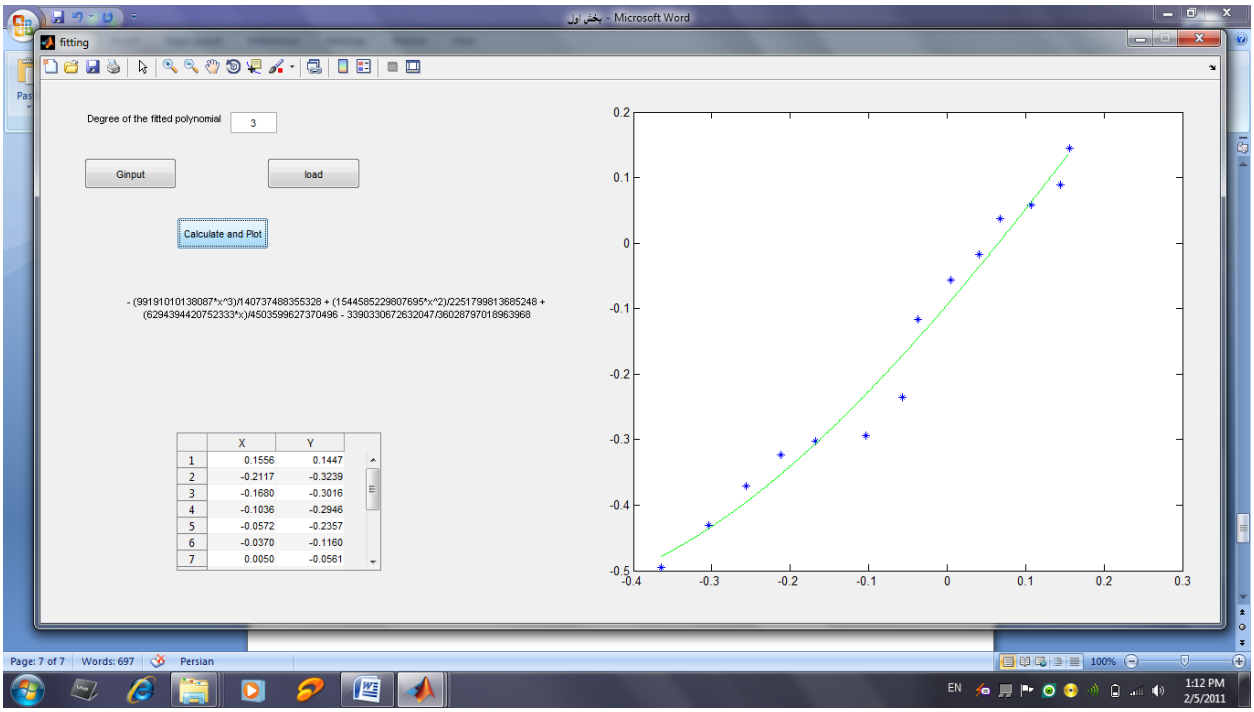
توسط pushbutton های save به ترتیب مختصات نهایی نقاط، پارامترهای بیضی های خطای مطلق و بیضی های خطای نسبی در فایل text برای مثال در اینجا پارامترهای بیضی خطای نسبی را save کرده و فایل text را نشان می دهیم.


Editor - C:\Users\sepideh\Desktop\barnamehaye ostad abedini\leastsquare\leastsquare.m
leastsquare
save relative error ellipse parameters
Save in: leastsquare
Name
Date modified
Type
absoluteerrorellipseparameters 2/3/2011 8:55 PM Text Docu
finalcoordinates 2/3/2011 8:55 PM Text Docu
newabsolut 2/5/2011 8:06 AM Text Docu
newpoint 2/5/2011 8:06 AM Text Docu
newrelative 2/5/2011 8:06 AM Text Docu
relativeerrorellipseparameters 2/3/2011 8:57 PM Text Docu
Recent Places
Desktop
Libraries
Computer
Network
File name: parameters
Save as type: (*.txt)
Save
Cancel
Save coordinates:
Save absolute error ellipse parameters:
Save relative error ellipse parameters:
Save
leastsquare
Ln 1 Col 1
OVR
1:04 PM
2/5/2011


parameters - Notepad
File Edit Format View Help
firstpoint--second point--x_center--y_center--major diameter--minor diameter--angle of major diameter with x axis--angle of minor diameter with x axis--eccentricity--area
1-- 2-- 7.4916-- 5.4843-- 0.0754-- 0.0131-- 11.0000-- 101.0000-- 0.9848-- 0.0494
1-- 3-- 9.4997-- 7.4240-- 0.1593-- 0.0695-- 299.5284-- 209.5284-- 0.8996-- 0.1653
1-- 4-- 6.4769-- 8.4504-- 0.1300-- 0.0647-- 2.0342-- 92.0342-- 0.8677-- 0.1440
1-- 5-- 4.4719-- 10.5188-- 0.2146-- 0.0614-- 21.6369-- 111.6369-- 0.9581-- 0.1804
2-- 3-- 11.9913-- 7.9083-- 0.1623-- 0.0717-- 305.0605-- 215.0605-- 0.8970-- 0.1695
2-- 4-- 8.9685-- 8.9347-- 0.1449-- 0.0655-- 6.9575-- 96.9575-- 0.8919-- 0.1530
2-- 5-- 6.9635-- 11.0031-- 0.2258-- 0.0604-- 21.8336-- 111.8336-- 0.9636-- 0.1834
3-- 4-- 10.9766-- 10.8744-- 0.1539-- 0.0736-- 253.3566-- 163.3566-- 0.8783-- 0.1672
3-- 5-- 8.9716-- 12.9428-- 0.2355-- 0.0774-- 237.6118-- 147.6118-- 0.9445-- 0.2120
4-- 5-- 5.9488-- 13.9692-- 0.1759-- 0.0623-- 38.5446-- 128.5446-- 0.9353-- 0.1644
1:05 PM
2/5/2011

برنامهfitting

برنامه ای به شکل زیر طراحی شده است:



ابتدا کابر در textbox اول درجه ی چند جمله ای را که قرار است به نقاط برازیده شود مشخص می کند به محض ورود عدد غیر صفر در این textbox دوpushbutton، ginput وload فعال می شوند کابر میتواند به دو صورت نقاط را وارد کند یا از طریق ginput و یا load کردن فایل نقاط (مختصات نقاط قبلا در یک فایل text به صورت زیر ذخیره شده است.)

0.1556,0.1447

0.3239-,0.2117-

0.3016-,0.1680-

0.2946-,0.1036-

0.2357-,0.0572-

0.1160-,0.0370-

0.0561-,0.0050

0.0172-,0.0409

0.0678,0.0367

0.1075,0.0577

0.1450,0.0896

0.3705-,0.2564-

0.4313-,0.3035-

0.4952-,0.3649-

اگر تعداد نقاط بیشتر از 2+degree باشد pushbutton ، calculate&plot فعال شده و برنامه به روش least square ضرایب چندجمله ای fitting را محاسبه و آن را رسم می کند و در جدول مختصات نقاط را نمایش می دهد.