

گزارش کد متلب:

کد ترازیابی:

هدف این برنامه انجام دادن کل محاسبات عملیات ترازیابی و سرشکن کردن ارتفاع ها می باشد.

متغیر های برنامه:

num_points: تعداد نقاط در عملیات

total_length: طول کل

benchmark: ارتفاع ثابت نهایی

FS_height: ارتفاع قرائت های جلو

BS_height: ارتفاع قرائت های عقب

delta_h: اختلاف ارتفاع ها

main_matrix: ماتریس کلی برای نمایش اطلاعات

e: خطای به دست آمده

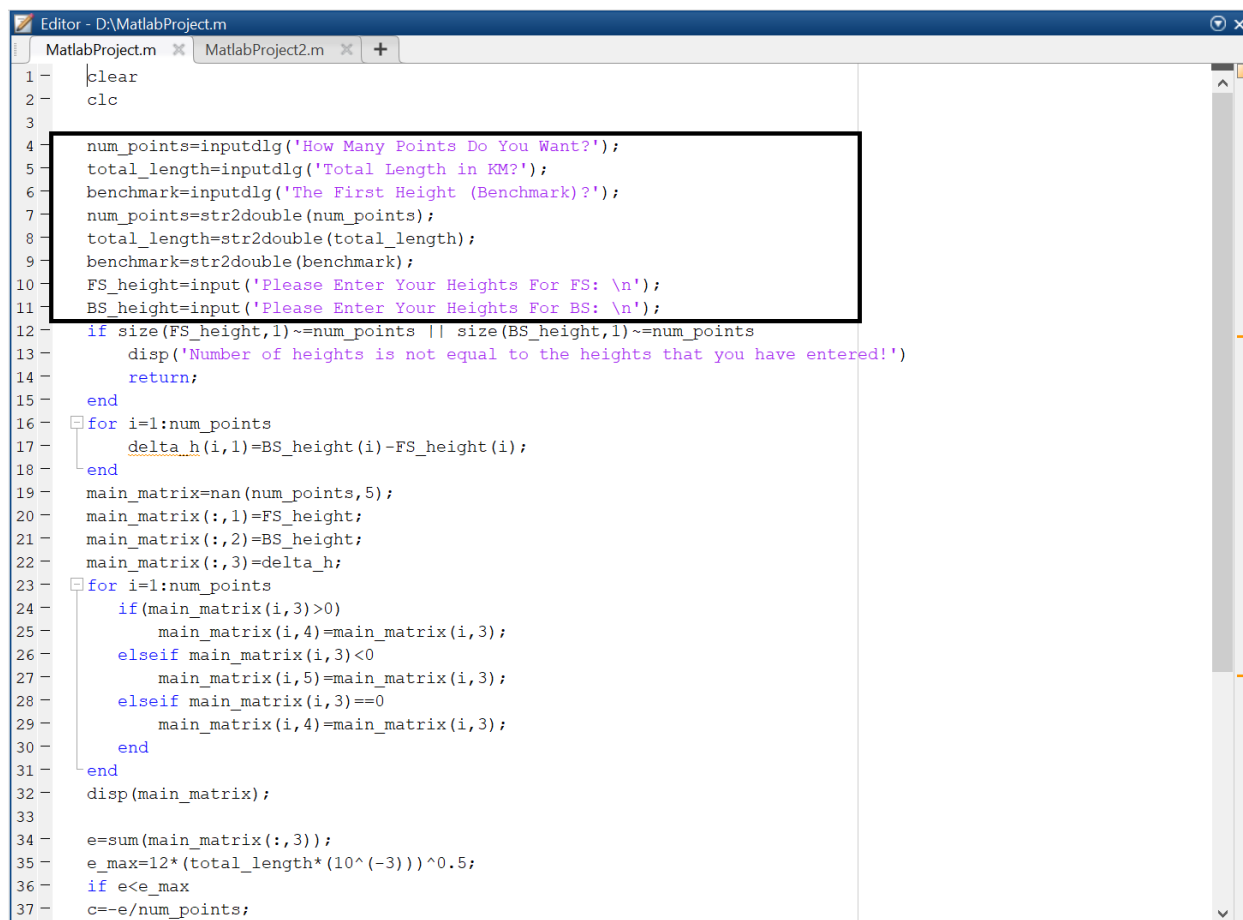
e_max: بیشترین خطای ممکنه

c: میزان سرشکنی برای ارتفاع

new_h: ارتفاع جدید به دست آمده قبل سرشکنی نهایی

final_heights: ارتفاع نهایی سرشکن شده

در ابتدای کد از کاربر تعداد نقاط مدنظر برای ترازیبی را دریافت می کنیم و سپس کل طولی که که بین نقاط است را نیز از کاربر می گیریم که این عدد باید بر حسب کیلومتر باشد، فقط در ابتدای برنامه یک ارتفاع ثابت از کاربر به عنوان benchmark می گیریم که در محاسبات انتهایی برنامه به کار می آید، حال ارتفاع های قراعت شده که fore sight و back sight می باشد را نیز از کاربر می گیریم که توسط عملیات به دست آمده است.



```
1 clear
2 clc
3
4 num_points=inputdlg('How Many Points Do You Want?');
5 total_length=inputdlg('Total Length in KM?');
6 benchmark=inputdlg('The First Height (Benchmark)?');
7 num_points=str2double(num_points);
8 total_length=str2double(total_length);
9 benchmark=str2double(benchmark);
10 FS_height=input('Please Enter Your Heights For FS: \n');
11 BS_height=input('Please Enter Your Heights For BS: \n');
12 if size(FS_height,1)~=num_points || size(BS_height,1)~=num_points
13     disp('Number of heights is not equal to the heights that you have entered!')
14     return;
15 end
16 for i=1:num_points
17     delta_h(i,1)=BS_height(i)-FS_height(i);
18 end
19 main_matrix=nan(num_points,5);
20 main_matrix(:,1)=FS_height;
21 main_matrix(:,2)=BS_height;
22 main_matrix(:,3)=delta_h;
23 for i=1:num_points
24     if main_matrix(i,3)>0
25         main_matrix(i,4)=main_matrix(i,3);
26     elseif main_matrix(i,3)<0
27         main_matrix(i,5)=main_matrix(i,3);
28     elseif main_matrix(i,3)==0
29         main_matrix(i,4)=main_matrix(i,3);
30     end
31 end
32 disp(main_matrix);
33
34 e=sum(main_matrix(:,3));
35 e_max=12*(total_length*(10^(-3)))^0.5;
36 if e<e_max
37     c=-e/num_points;
```

شکل ۱ : ورودی های برنامه

در ابتدا شرطی برای بررسی صحت ورودی های کاربر قرار می دهیم که در صورت غلط بودن آن ها به کاربر هشدار داده و برنامه را اجرا نمی کند.

```
Editor - D:\MatlabProject.m
MatlabProject.m  MatlabProject2.m  +
1 - clear
2 - clc
3
4 - num_points=inputdlg('How Many Points Do You Want?');
5 - total_length=inputdlg('Total Length in KM?');
6 - benchmark=inputdlg('The First Height (Benchmark)?');
7 - num_points=str2double(num_points);
8 - total_length=str2double(total_length);
9 - benchmark=str2double(benchmark);
10 - FS_height=input('Please Enter Your Heights For FS: \n');
11 - BS_height=input('Please Enter Your Heights For BS: \n');
12 - if size(FS_height,1)~=num_points || size(BS_height,1)~=num_points
13 -     disp('Number of heights is not equal to the heights that you have entered!')
14 -     return;
15 - end
16 - for i=1:num_points
17 -     delta_h(i,1)=BS_height(i)-FS_height(i);
18 - end
19 - main_matrix=nan(num_points,5);
20 - main_matrix(:,1)=FS_height;
21 - main_matrix(:,2)=BS_height;
22 - main_matrix(:,3)=delta_h;
23 - for i=1:num_points
24 -     if (main_matrix(i,3)>0)
25 -         main_matrix(i,4)=main_matrix(i,3);
26 -     elseif main_matrix(i,3)<0
27 -         main_matrix(i,5)=main_matrix(i,3);
28 -     elseif main_matrix(i,3)==0
29 -         main_matrix(i,4)=main_matrix(i,3);
30 -     end
31 - end
32 - disp(main_matrix);
33
34 - e=sum(main_matrix(:,3));
35 - e_max=12*(total_length*(10^(-3)))^0.5;
36 - if e<e_max
37 -     c=-e/num_points;
```

شکل ۲: شرط ورودی های کاربر

با داشتن FS و BS از ورودی های کاربر می توان با کم کردن BS از FS میزان تغییر ارتفاع را به دست آورد با انجام این کار کل اختلاف ارتفاع عا را به دست می آوریم و در متغیر delta_h که ماتریسی ستونی است قرار می دهیم سپس ماتریس کلی به نام main_matrix که تمام اطلاعات ما از جمله صعود و سقوط را نمایش می دهیم می سازیم و اطلاعات را وارد آن می کنیم به طوری که ستون اول FS ، ستون دوم BS ، ستون سوم اختلاف ارتفاع ، ستون چهارم میزان افزایش ارتفاع (rise) و در ستون پنجم میزان کاهش ارتفاع (fall) را نمایش می دهد در هر جایی از این ماتریس ۵ در ۵ که اطلاعاتی نداریم NaN قرار می دهیم.

```
Editor - D:\MatlabProject.m
MatlabProject.m  MatlabProject2.m  +
1 - clear
2 - clc
3
4 - num_points=inputdlg('How Many Points Do You Want?');
5 - total_length=inputdlg('Total Length in KM?');
6 - benchmark=inputdlg('The First Height (Benchmark)?');
7 - num_points=str2double(num_points);
8 - total_length=str2double(total_length);
9 - benchmark=str2double(benchmark);
10 - FS_height=input('Please Enter Your Heights For FS: \n');
11 - BS_height=input('Please Enter Your Heights For BS: \n');
12 - if size(FS_height,1)~=num_points || size(BS_height,1)~=num_points
13 -     disp('Number of heights is not equal to the heights that you have entered!')
14 -     return;
15 - end
16 - for i=1:num_points
17 -     delta_h(i,1)=BS_height(i)-FS_height(i);
18 - end
19 - main_matrix=nan(num_points,5);
20 - main_matrix(:,1)=FS_height;
21 - main_matrix(:,2)=BS_height;
22 - main_matrix(:,3)=delta_h;
23 - for i=1:num_points
24 -     if (main_matrix(i,3)>0)
25 -         main_matrix(i,4)=main_matrix(i,3);
26 -     elseif main_matrix(i,3)<0
27 -         main_matrix(i,5)=main_matrix(i,3);
28 -     elseif main_matrix(i,3)==0
29 -         main_matrix(i,4)=main_matrix(i,3);
30 -     end
31 - end
32 - disp(main_matrix);
33
34 - e=sum(main_matrix(:,3));
35 - e_max=12*(total_length*(10^(-3)))^0.5;
36 - if e<e_max
37 -     c=-e/num_points;
```

شکل ۳: مراحل ساخت ماتریس کلی

بعد از نشان دادن main_matrix در command window حال زمان سرشکن کردن ارتفاع ها است که برای این کار باید خطا کم تر از بیشترین خطای ممکنه باشد میزان خطا با جمع کردن اختلاف ارتفاع ها به دست می آید و آن را با e نمایش می دهیم بیشترین خطای ممکنه از رابطه معرفی شده به دست می آید که در آن طول کل باید بر حسب کیلومتر باشد.

با به دست آوردن خطا میزانی که باید به هر ارتفاع اضافه یا کم شود تا ارتفاع ها صحیح شوند محاسبه می شود که با تقسیم کردن خطا بر تعداد نقاط به دست می آید این میزان را با c در کد نمایش می دهیم.

```

Editor - D:\MatlabProject.m
MatlabProject.m  x  MatlabProject2.m  x  +
1 - clear
2 - clc
3
4 - num_points=inputdlg('How Many Points Do You Want?');
5 - total_length=inputdlg('Total Length in KM?');
6 - benchmark=inputdlg('The First Height (Benchmark)?');
7 - num_points=str2double(num_points);
8 - total_length=str2double(total_length);
9 - benchmark=str2double(benchmark);
10 - FS_height=input('Please Enter Your Heights For FS: \n');
11 - BS_height=input('Please Enter Your Heights For BS: \n');
12 - if size(FS_height,1)~=num_points || size(BS_height,1)~=num_points
13 -     disp('Number of heights is not equal to the heights that you have entered!')
14 -     return;
15 - end
16 - for i=1:num_points
17 -     delta_h(i,1)=BS_height(i)-FS_height(i);
18 - end
19 - main_matrix=nan(num_points,5);
20 - main_matrix(:,1)=FS_height;
21 - main_matrix(:,2)=BS_height;
22 - main_matrix(:,3)=delta_h;
23 - for i=1:num_points
24 -     if (main_matrix(i,3)>0)
25 -         main_matrix(i,4)=main_matrix(i,3);
26 -     elseif main_matrix(i,3)<0
27 -         main_matrix(i,5)=main_matrix(i,3);
28 -     elseif main_matrix(i,3)==0
29 -         main_matrix(i,4)=main_matrix(i,3);
30 -     end
31 - end
32 - disp(main_matrix);
33
34 - e=sum(main_matrix(:,3));
35 - e_max=12*(total_length*(10^(-3)))^0.5;
36 - if e<e_max
37 -     c=-e/num_points;

```

شکل ۴ : به دست آوردن خطا و بیشترین خطا

بعد از به دست آوردن C با فرمول های موجود مقدار سرشکنی را با ارتفاع ها طبق کد جمع می کنیم و در نهایت ارتفاع های سرشکن شده را به دست می آوریم.

```

42 - new_H(1,1)=benchmark+main_matrix(1,3);
43 - for i=1:num_points-1
44 -     new_H(i+1,1)=new_H(i,1)+main_matrix(i+1,3);
45 - end
46 - new_H=[benchmark;new_H];
47 - new_H(end)=[];
48 - final_heights=new_H+c_matrix;
49 - disp(final_heights);
50 - elseif e>e_max
51 -     disp('Error is too high! The Process can NOT be done!');
52 - end

```

شکل ۵ : سرشکن کردن

فقط در صورت بزرگتر بودن خطای به دست آمده از بیشترین خطای ممکنه برنامه به کاربر هشدار می دهد که خطا مجاز نیست و عملیات سرشکنی انجام نمی شود و ارتفاع های جدیدی به دست نمی آید.

```
42 - new_H(1,1)=benchmark+main_matrix(1,3);
43 - for i=1:num_points-1
44 -     new_H(i+1,1)=new_H(i,1)+main_matrix(i+1,3);
45 - end
46 - new_H=[benchmark;new_H];
47 - new_H(end)=[];
48 - final_heights=new_H+c_matrix;
49 - disp(final_heights);
50 - elseif e>e_max
51 -     disp('Error is too high! The Process can NOT be done!');
52 - end
```

شکل ۶: شرط بزرگتر بودن خطا از بیشترین خطای ممکنه و عدم اجرای سرشکنی

کد سرشکنی زاویه ها :

در این برنامه هدف سرشکن کردن زاویه های به دست آمده از عملیات می باشد.

متغیر های برنامه:

num_points: تعداد نقاط در عملیات

angles: زوایای به دست آمده از عملیات

sum_angles: مجموع کل زوایای به دست آمده

exact_sum: مجموع زوایای چند ضلعی عملیات

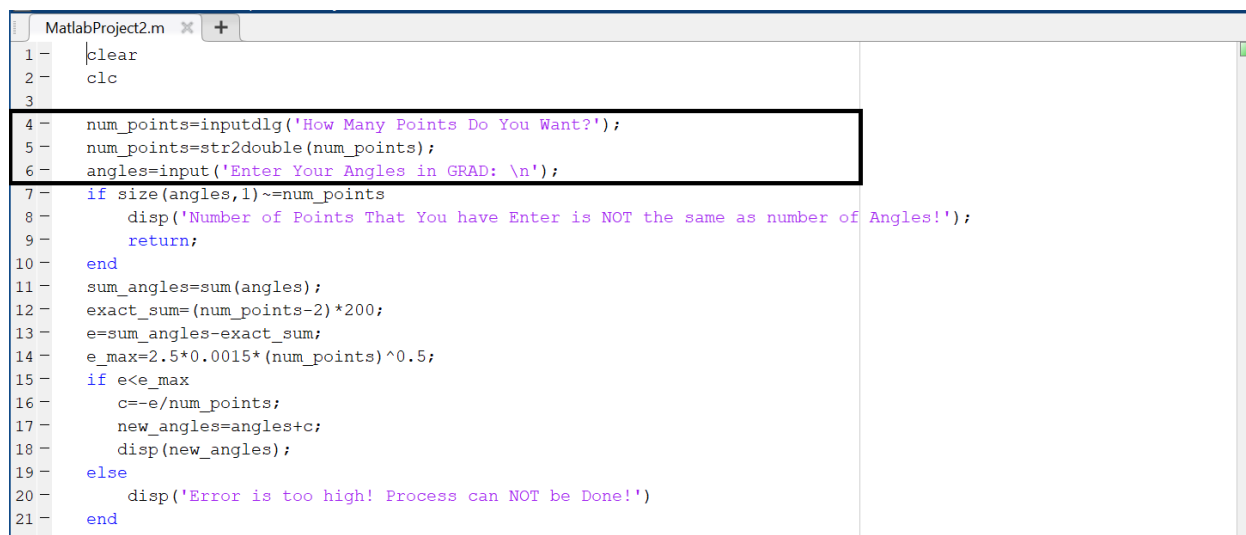
e: خطای به دست آمده

e_max: بیشترین خطای ممکنه

c: مقدار سرشکنی

new_angles: زوایای نهایی سرشکن شده

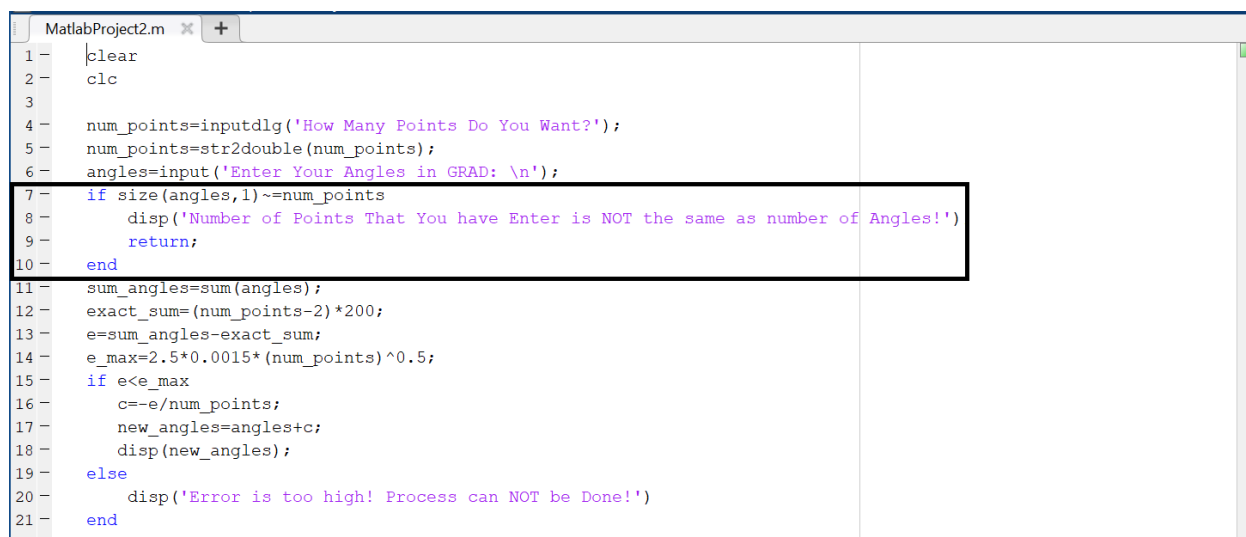
در ابتدای برنامه تعداد نقاط موجود در عملیات از کاربر گرفته می شود و همچنین زوایایی که از انجام عملیات به دست آمده است به طور کلی از کاربر گرفته می شود.

A screenshot of a MATLAB script editor window titled 'MatlabProject2.m'. The code is as follows:

```
1 clear
2 clc
3
4 num_points=inputdlg('How Many Points Do You Want?');
5 num_points=str2double(num_points);
6 angles=input('Enter Your Angles in GRAD: \n');
7 if size(angles,1)~=num_points
8     disp('Number of Points That You have Enter is NOT the same as number of Angles!');
9     return;
10 end
11 sum_angles=sum(angles);
12 exact_sum=(num_points-2)*200;
13 e=sum_angles-exact_sum;
14 e_max=2.5*0.0015*(num_points)^0.5;
15 if e<e_max
16     c=-e/num_points;
17     new_angles=angles+c;
18     disp(new_angles);
19 else
20     disp('Error is too high! Process can NOT be Done!')
21 end
```

شکل ۷: ورودی های برنامه

در همان ابتدا شرطی برای بررسی صحت ورودی های کاربر قرار داده می شود که در صورتی که ورودی های کاربر غلط باشند و یا مشکلی داشته باشند به کاربر هشدار داده می شود و برنامه متوقف می شود.

A screenshot of a MATLAB script editor window titled 'MatlabProject2.m'. The code is identical to the previous one, but a black box highlights the error handling section from line 7 to line 10:

```
7 if size(angles,1)~=num_points
8     disp('Number of Points That You have Enter is NOT the same as number of Angles!')
9     return;
10 end
```

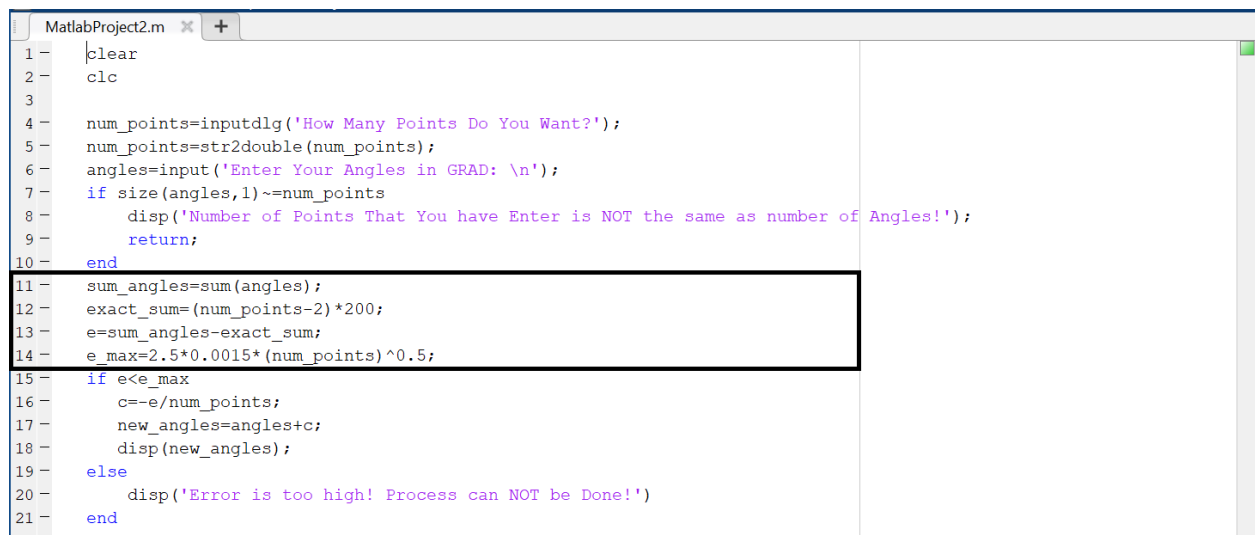
شکل ۸: شرط صحت ورودی های کاربر

مجموع تمام زوایای داده شده به برنامه را به دست می آوریم و آن را با متغیر `sum_angles` نمایش می دهیم.

مجموع دقیق زوایا را با استفاده از رابطه مجموع زوایا برای یک چند ضلعی به دست می آوریم و آن را با متغیر `exact_sum` نمایش می دهیم. فقط باید به این نکته دقت شود که در این برنامه واحد زوایا گرادیان می باشد.

با کم کردن مقدار این دو متغیر خطای ما به دست می آید که آن را با متغیر `e` نمایش می دهیم. برای به دست آوردن بیشترین مقدار خطای ممکنه و مقایسه آن با خطای به دست آمده از رابطه موجود استفاده می کنیم و آن را با متغیر `e_max` نشان می دهیم.

در رابطه موجود برای بیشترین خطای ممکنه نیز باید به این نکته توجه شود که واحد زوایا در این برنامه بر حسب گرادیان می باشد.



```
1 - clear
2 - clc
3
4 - num_points=inputdlg('How Many Points Do You Want?');
5 - num_points=str2double(num_points);
6 - angles=input('Enter Your Angles in GRAD: \n');
7 - if size(angles,1)~=num_points
8 -     disp('Number of Points That You have Enter is NOT the same as number of Angles!');
9 -     return;
10 - end
11 - sum_angles=sum(angles);
12 - exact_sum=(num_points-2)*200;
13 - e=sum_angles-exact_sum;
14 - e_max=2.5*0.0015*(num_points)^0.5;
15 - if e<e_max
16 -     c=-e/num_points;
17 -     new_angles=angles+c;
18 -     disp(new_angles);
19 - else
20 -     disp('Error is too high! Process can NOT be Done!')
21 - end
```

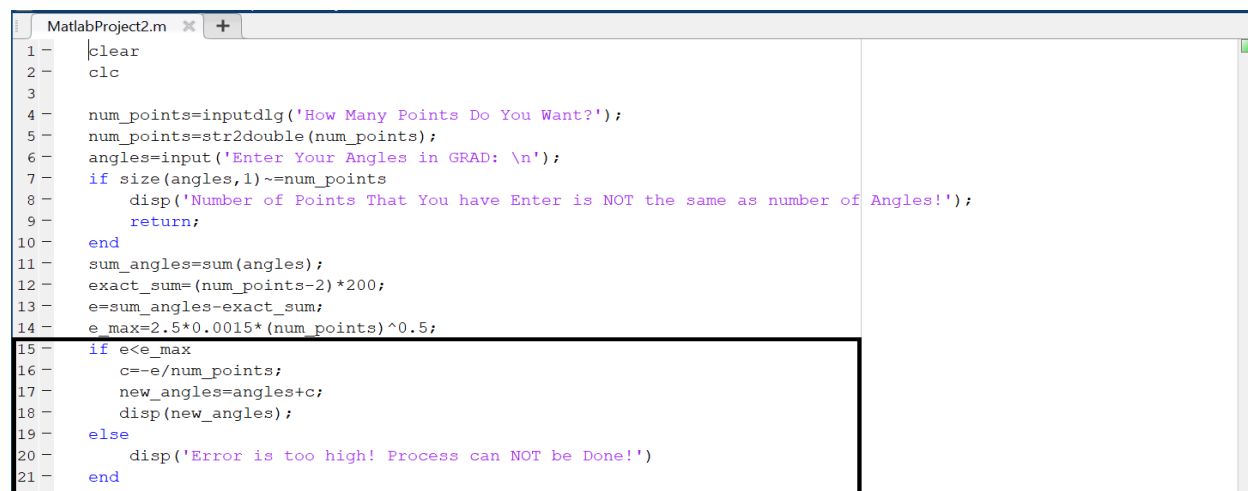
شکل ۹: به دست آوردن مجموع زوایا و خطا

در صورت کم تر بودن خطای به دست آمده از بیشترین خطای ممکنه می توان عملیات سرشکنی را انجام داد.

شرط اول در این قسمت کد همین مطلب را بیان می کند. بعد از بررسی شرط مقدار سرشکنی هر زاویه با تقسیم میزان خطا بر تعداد نقاط به دست می آید که آن را با `C` نشان می دهیم.

با اضافه کردن C به هر زاویه مقدار سرشکن شده آن به دست می آید که زاویه نهایی و صحیح می باشد که زوایای سرشکن شده را در command window نمایش می دهیم.

در صورت بیشتر بودن خطای به دست آمده از بیشترین خطای ممکنه نمی توان عملیات سرشکنی را انجام داد به همین دلیل به کاربر هشدار داده می شود و سرشکنی انجام نمی شود.



```
1 clear
2 clc
3
4 num_points=inputdlg('How Many Points Do You Want?');
5 num_points=str2double(num_points);
6 angles=input('Enter Your Angles in GRAD: \n');
7 if size(angles,1)~=num_points
8     disp('Number of Points That You have Enter is NOT the same as number of Angles!');
9     return;
10 end
11 sum_angles=sum(angles);
12 exact_sum=(num_points-2)*200;
13 e=sum_angles-exact_sum;
14 e_max=2.5*0.0015*(num_points)^0.5;
15 if e<e_max
16     c=-e/num_points;
17     new_angles=angles+c;
18     disp(new_angles);
19 else
20     disp('Error is too high! Process can NOT be Done!')
21 end
```

شکل ۱۰ : سرشکن کردن زوایا