Lab 2

Matlab Optimization Toolbox

MSE 426:

Introduction to Engineering Design Optimization

Selikem Kwadzovia | 301303898 | Selikem K

Sepehr Rezvani | 301291960 |

Tuesday, February 2, 2021

Introduction

The purpose of this lab is to experiment with Matlab's optimization toolbox. More specifically, using the minimum unconstrained (fminunc) and minimum constrained (fmincon) functions. Using these tools, we can find the optimum point, the number of iterations, and the number of evaluations of various functions. Multiple runs were done for each function to see if the results changed between trials. Also, various options were tested such as different stopping criteria, approximate and exact derivatives, and different Hessian methods. We developed the code to do this individually, compiling our results afterwards.

Results and Analysis

For this lab, as mentioned in the introduction, two main functions "fminunc" and "fmincon" were used to solve 5 separate problems and 5 different runs were analyzed. The following were considered: optimum positions, optimum values, the number of iterations and the number of function evaluations.

The results using default conditions for each question can be seen in Tables 1 to 5. In these results, the default initial condition is 1 for all x variables unless otherwise stated. Detailed results providing the position of the optimum values for the default case can be found in the appendices.

Table 1: Optimum Values

Run	1	2	3	4	5
Question 1	7.600632	7.600632	7.600632	7.600632	7.600632
Question 2	9.99E-07	9.99E-07	9.99E-07	9.99E-07	9.99E-07
Question 3	-4.57612	-4.57612	-4.57612	-4.57612	-4.57612
Question 4	0.045675	0.045675	0.045675	0.045675	0.045675
Question 5	-148	-148	-148	-148	-148

Table 2: Number of Iterations

Run	1	2	3	4	5
Question 1	4	4	4	4	4
Question 2	25	25	25	25	25
Question 3	10	10	10	10	10
Question 4	21	21	21	21	21
Question 5	14	14	14	14	14

Table 3: Number of Function Evaluations

Run	1	2	3	4	5
Question 1	15	15	15	15	15
Question 2	130	130	130	130	130
Question 3	36	36	36	36	36
Question 4	76	76	76	76	76
Question 5	107	107	107	107	107

The Optimum values for two different initial conditions can be seen in Tables 4 and 5.

Table 4: Optimum Values using Initial Condition x = 0

Run	1	2	3	4	5
Question 1	6.202272	6.202272	6.202272	6.202272	6.202272
Question 2	0	0	0	0	0
Question 3	-4.57612	-4.57612	-4.57612	-4.57612	-4.57612
Question 4	0.045675	0.045675	0.045675	0.045675	0.045675
Question 5	-148	-148	-148	-148	-148

Table 5: Optimum Values using Initial Condition x = 2

Run	1	2	3	4	5
Question 1	17.13318	17.13318	17.13318	17.13318	17.13318
Question 2	1.56E-06	1.56E-06	1.56E-06	1.56E-06	1.56E-06
Question 3	-4.57612	-4.57612	-4.57612	-4.57612	-4.57612
Question 4	0.045675	0.045675	0.045675	0.045675	0.045675
Question 5	-258	-258	-258	-258	-258

The data in Table 6 to 14 were collected using fmincon with various selected options.

Table 6: Optimum Values with Maximum Evaluations = 100

Run	1	2	3	4	5
Question 3	-4.57612	-4.57612	-4.57612	-4.57612	-4.57612
Question 4	0.045675	0.045675	0.045675	0.045675	0.045675
Question 5	-148	-148	-148	-148	-148

Table 7: Number of Iterations with Maximum Evaluations = 100

Run	1	2	3	4	5
Question 3	10	10	10	10	10
Question 4	21	21	21	21	21
Question 5	13	13	13	13	13

Table 8: Number of Function Evaluations with Maximum Evaluations = 100

Run	1	2	3	4	5
Question 3	36	36	36	36	36
Question 4	76	76	76	76	76
Question 5	100	100	100	100	100

To find exact and approximation function values, a gradient method was used. The Hessian method, which organizes second order partial derivative functions, was used in MATLAB to analyze different problems with varying starting points. Table 9 to 11 Hessian method was used and optimum values for five different iterations were evaluated with specific gradients.

Table 9: Optimum Values with Specified Gradient

Run	1	2	3	4	5
Question 3	-4.52085	-4.52085	-4.52085	-4.52085	-4.52085
Question 4	0.045675	0.045675	0.045675	0.045675	0.045675
Question 5	-148	-148	-148	-148	-148

Table 10: Number of Iterations with Specified Gradient

Run	1	2	3	4	5
Question 3	9	9	9	9	9
Question 4	21	21	21	21	21
Question 5	14	14	14	14	14

Table 11: Number of Function Evaluations with Specified Gradient

Run	1	2	3	4	5
Question 3	59	59	59	59	59
Question 4	76	76	76	76	76
Question 5	107	107	107	107	107

Table 12: Optimum Values with Finite Difference Hessian Approximation

Run	1	2	3	4	5
Question 3	1.5	1.5	1.5	1.5	1.5
Question 4	69.6628	69.6628	69.6628	69.6628	69.6628
Question 5	-45.9602	-45.9602	-45.9602	-45.9602	-45.9602

Table 13: Number of Iterations with Finite Difference Hessian Approximation

Run	1	2	3	4	5
Question 3	1	1	1	1	1
Question 4	16	16	16	16	16
Question 5	1	1	1	1	1

Table 14: Number of Function Evaluations with Finite Difference Hessian Approximation

Run	1	2	3	4	5
Question 3	143	143	143	143	143
Question 4	245	245	245	245	245
Question 5	143	143	143	143	143

The first thing to note is that across the five tests for each variation, the final outcome was the same. This shows that fminunc and fmincon are deterministic when used this way. However, the initial conditions affect the optimum value found. There are different possible local minima depending on the initial values selected.

When the maximum amount of evaluations was set to 100, the values for question 3 and 4 remained constant due to them not reaching the maximum evaluations. Question 5 reached the maximum amount of evaluations, but only had one less iteration than the default result of 14. The optimum value found was the same between the default as well. If the maximum was lowered even further, it is possible that the final result would differ from the default. Reducing the amount of evaluations would save computation time on expensive functions.

The optimum values found using a given gradient varied slightly from the default options. This is likely due to the function approximating the gradient in comparison to a known gradient being provided. This means that providing a gradient will likely produce more accurate results.

When the Hessian was used, the optimum value found was significantly different from the default. It is possible that the hessian approximation was inaccurate for the given conditions, but may save time depending on how it is used.

Conclusion

This lab highlighted the uses of the optimization toolbox in Matlab. The multiple runs showed that fmincon and fminunc were deterministic in nature. Despite reducing the amount of maximum evaluations, the final result was mostly the same. Specifying the gradient did not significantly increase the runtime of the code, and was more accurate. If the gradient is known, specifying it before running the code can prove useful. Lastly, the Hessian approximation provided significantly different results than the default options. It is worthwhile to test and compare the results to other methods to determine the best cases to use it.

Bibliography

[1] G. Wang, “Introduction to Optimum Design: MSE 426/726 Lab Manual.”

Canvas.sfu.ca.

Appendices

Detailed Results

The following tables provide the positions for the optimum values found using default conditions.

Table 15: Minimum Position for x(1)

Run	1	2	3	4	5
Question 1	0.942811	0.942811	0.942811	0.942811	0.942811
Question 2	0.026492	0.026492	0.026492	0.026492	0.026492
Question 3	-0.05654	-0.05654	-0.05654	-0.05654	-0.05654
Question 4	0.786415	0.786415	0.786415	0.786415	0.786415
Question 5	3.70E-07	3.70E-07	3.70E-07	3.70E-07	3.70E-07

Table 16: Minimum Position for x(2)

Run	1	2	3	4	5
Question 1	0.238999	0.238999	0.238999	0.238999	0.238999
Question 2	-0.00265	-0.00265	-0.00265	-0.00265	-0.00265
Question 3	2.032369	2.032369	2.032369	2.032369	2.032369

Question 4	**0.617698**	**0.617698**	**0.617698**	**0.617698**	**0.617698**
Question 5	**5.999995**	**5.999995**	**5.999995**	**5.999995**	**5.999995**

Table 17: Minimum Position for x(3)

Run	**1**	**2**	**3**	**4**	**5**
Question 1	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 2	**0.010028**	**0.010028**	**0.010028**	**0.010028**	**0.010028**
Question 3	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 4	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 5	**1.000002**	**1.000002**	**1.000002**	**1.000002**	**1.000002**

Table 18: Minimum Position for x(4)

Run	**1**	**2**	**3**	**4**	**5**
Question 1	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 2	**0.010054**	**0.010054**	**0.010054**	**0.010054**	**0.010054**
Question 3	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 4	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**

Question 5	**1.50E-05**	**1.50E-05**	**1.50E-05**	**1.50E-05**	**1.50E-05**

Table 19: Minimum Position for x(5)

Run	**1**	**2**	**3**	**4**	**5**
Question 1	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 2	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 3	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 4	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 5	**1.000002**	**1.000002**	**1.000002**	**1.000002**	**1.000002**

Table 20: Minimum Position for x(6)

Run	**1**	**2**	**3**	**4**	**5**
Question 1	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 2	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 3	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 4	**N/A**	**N/A**	**N/A**	**N/A**	**N/A**
Question 5	**1.50E-05**	**1.50E-05**	**1.50E-05**	**1.50E-05**	**1.50E-05**

Code

Codes will be attached to the report. There are three files, "main.m", "myFun1.m" and "myNLCon".