MSE 426 - Design Optimzation
Lab 2 Report

Written For: Dr. Gary Wang

Justin Ortiz Luis - 301279195

February 6, 2020

1 Abstract

This report focuses on the methodologies and MATLAB programming to determine the optimal solution to different numerical models of systems. To conserve time and limit human especially for multivariable and non-linear systems, the user must design programs to assist in the calculations.

The methodologies implemented in this lab are the two MATLAB functions that come with the optimization toolbox: fmincon and fminunc. Fminunc is an unconstrained method of finding local minimums of a function where fmincon finds the local minimums within a constrained range. The fmincon uses multiple methods of input constraints: lienar inequalities, equalities, bounding, and non linear conditions. Moreover, the user is able to have full customization of these options to suit their needs. These options include adjusting the calculation tolerances, number of function evaluations, specifiying the objective gradient, and hessian approximation. Through full implementation of these functions and options, the user will be able to optimize many systems.

Contents

List of Figures

2 Introduction

The significance of optimization becomes increasingly important as technology is progressing ever so quickly. The difficulty in modelling systems is becoming more complex as nonlinearity is very common in systems, requiring more advanced computations. To optimize said systems, we must find the minimum or "minimize" the said function that represents the system. Through the use of the Optimization Toolbox in MATLAB, the knowledge and skills acquired will be broken down in core components such as:

- Using the functions in the Optimization Toolbox
- Set specific constraints to the problem and variables of interest
- Modify options to the function used for the computations

During this lab session, we focused on using "fmincon" and "fminunc", in which both functions are used to find the local minimums of the system function representation. Fminunc finds the local minimum with no listed constraints, where Fmincon finds the local minimum with a listed set of constraints.

3 Results and Discussion

3.1 Question 1

For question 1, we were required to go through the examples in the lab manual to implement the fmincon function.

3.2 Question 2

For question 2, we were required to solve for the roots of the five different systems. The function constraints were all written as separate files where the calculations were done in a separate main script.

Part 1

The results to the first part is shown below:

Question 2-1 Results						
fminunc (no constraints)						
Starting Point	Optimum x1	Optimum x2	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
(0,0)	0.0105	0.2909	6.2023	3	15	1
(1,1)	0.9428	0.239	7.6006	4	15	1
(2,2)	1.7955	0.1596	17.1332	5	18	1
(3,3)	2.5755	0.06	40.3616	6	21	2
(4,4)	0.52	0.2671	6.1608	49	201	0
(5,5)	3.2403	-0.0455	80.1595	14	66	1

Figure 1: Results for Question 2-1

Part 2

The results to the second part is shown below:

Question 2-2 Results								
fminunc (no constraints)								
Starting Point	Optimum x1	Optimum x2	Optimum x3	Optimum x4	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
(0,0,0,0)	0	0	0	0	0	1	45	5
(1,1,1,1)	0.0265	-0.0026	0.01	0.0101	9.99E-07	25	130	1
(2,2,2,2)	0.0063	-0.0006	-0.0124	-0.0124	1.56E-06	26	135	1
(3,3,3,3)	-0.0245	0.0024	-0.0037	-0.0037	1.87E-06	32	165	1
(4,4,4,4)	-0.0302	0.003	-0.0154	-0.0155	1.80E-06	32	170	1
(5,5,5,5)	0.0033	-0.0003	-0.0163	-0.0163	2.57E-06	37	195	1

Figure 2: Results for Question 2-2

Part 3

The results to the third part is shown below:

Question 2-3 Results						
fmincon (constrained)						
Starting Point	Optimum x1	Optimum x2	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
(0,0)	-0.0565	2.0324	-4.5761	12	41	1
(1,1)	-0.0565	2.0324	-4.5761	10	36	1
(2,2)	-0.0565	2.0324	-4.5761	10	33	1
(3,3)	-0.0565	2.0324	-4.5761	10	33	1
(4,4)	-0.0565	2.0324	-4.5761	11	36	1
(5,5)	-0.0565	2.0324	-4.5761	9	30	1

Figure 3: Results for Question 2-3

Part 4

The results to the fourth part is shown below:

Question 2-4 Results						
fmincon (constrained)						
Starting Point	Optimum x1	Optimum x2	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
(0,0)	0.7864	0.6177	0.0457	24	84	1
(1,1)	0.7864	0.6177	0.0457	21	76	1
(2,2)	0.7864	0.6177	0.0457	26	85	1
(3,3)	0.7864	0.6177	0.0457	26	87	1
(4,4)	0.7864	0.6177	0.0457	25	83	1
(5,5)	0.7864	0.6177	0.0457	50	191	1

Figure 4: Results for Question 2-4

Part 5

The results to the fifth part is shown below:

Question 2-5 Results										
fmincon (constrained)										
Starting Point	Optimum x1	Optimum x2	Optimum x3	Optimum x4	Optimum x5	Optimum x6	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
(0,0,0,0,0,0)	0	6	1	0	1	0	-147.9997	107	107	1
(1,1,1,1,1,1)	0	6	1	0	1	0	-147.9997	107	107	1
(2,2,2,2,2,2)	5	1	1	0	1	0	-2.58E+02	12	92	1
(3,3,3,3,3,3)	5	1	5	0	5	0	-2.90E+02	77	77	1
(4,4,4,4,4,4)	5	1	5	0	5	0	-2.90E+02	98	98	1
(5,5,5,5,5,5)	5	1	5	6	5	10	-2.98E+02	86	86	1

Figure 5: Results for Question 2-5

3.3 Question 3

Part 1

For part 1, we varied the OptimalityTolerance paramaeter for first-order optimality. Applying more than one variation to this option, we have set our fixed initial root estimate at [0,0]. The results to the first part is shown below:

Question 3-1 Results						
fmincon (constrained): Optimality Tolerance						
OptimalityTolerance	Optimum x1	Optimum x2	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
10	1	1	1.5	0	3	1
1	-0.0911	1.9746	-4.5325	3	12	1
1e-6 (Default)	-0.0565	2.0324	-4.5761	10	33	1
1.00E-03	-0.0565	2.0316	-4.5753	7	24	2
1.00E-09	-0.0565	2.0324	-4.5761	12	48	2

Figure 6: Results for Question 3-1

Part 2

For part 2, the MaxFunctionEvaluations parameter was modified, changing the maximum number of functions evaluatiosn that the system is allowed to computate. Similarly to part 1, an initial estimation point of [0,0] is used. The results to the second part is shown below:

Question 3-2 Results						
fmincon (constrained): MaxFunctionEvaluations						
MaxFunctionEvaluations	Optimum x1	Optimum x2	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
3000 (Default)	-0.0565	2.0324	-4.5761	10	36	1
100	-0.0565	2.0324	-4.5761	10	33	1
50	1.7955	0.1596	17.1332	5	18	1
30	2.5755	0.06	40.3616	6	21	2
20	0.52	0.2671	6.1608	49	201	0

Figure 7: Results for Question 3-2

Part 3

For part 3, we varied the SpecifyObjectiveGradient which allows us to add the gradient of the function in the calculation for faster and more reliable computations. The default option is set to false, therefore we have set the new option to be true. The results to the third part is shown below:

Question 3-3 Results						
fmincon (constrained): Specify Objective Constrant: Set true (initially false as default)						
Starting Point	Optimum x1	Optimum x2	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
(0,0)	-0.0565	2.0324	-4.5761	12	41	1
(1,1)	-0.0565	2.0324	-4.5761	10	36	1
(2,2)	-0.0565	2.0324	-4.5761	10	33	1
(3,3)	-0.0565	2.0324	-4.5761	10	33	1
(4,4)	-0.0565	2.0324	-4.5761	11	36	1
(5,5)	-0.0565	2.0324	-4.5761	9	30	1

Figure 8: Results for Question 3-3

Part 4

For part 4, we varied the HessianApproximation parameter, there are different algorithms for the Hessian calculation, the default is 'bfgs', but for this scenario, we chose 'lbfgs'. The results to the fourth part is shown below:

Question 3-4 Results						
fmincon (constrained): HessianApproximation (Use lbfgs instead of bfgs)						
Starting Point	Optimum x1	Optimum x2	Optimum f	# of Iterations	# of Function Evaluations	Exitflag
(0,0)	-0.0565	2.0324	-4.5761	13	44	1
(1,1)	-0.0565	2.0324	-4.5761	13	46	1
(2,2)	-0.0565	2.0324	-4.5761	11	36	1
(3,3)	-0.0565	2.0324	-4.5761	12	39	1
(4,4)	-0.0565	2.0324	-4.5761	13	42	1
(5,5)	-0.0565	2.0324	-4.5761	12	39	1

Figure 9: Results for Question 3-4

3.4 Discussion

After extensive use of the two functions: fminunc and fmincon, we learned about the key features of each function.

Fminunc takes up to 3 inputs: the system representation f(x), the initial choice of the root, and the options (any modifications to the default parameter settings). In MATLAB, fminunc is implemented in the following syntax. x = fminunc(fun,x0,options)

Furthermore, the fmincon function can take up to 10 inputs, and the syntax for implementation is shown below. x = fmincon(fun,x0,A,B,Aeq,beq,lb,ub,nonlcon,options)

By modifying the input "options" in question 3, we can make an analysis on how it affects the results.

3.5 Optimality Tolerance

This parameter has significant impact to the computational results. As we increase the tolerance, it increases the error of computation, overall decreasing accuracy. It can be concluded that this has a negative relationship with output accuracy.

3.6 MaxFunctionEvaluations

This parameter also affects computational results. As we decrease the number of evaluations, the accuracy decreases. If we do not allow a sufficient amount of computations, then the function will not converge to the proper local minimum. Thus, this parameter has a negative relationship to the computation accuracy.

3.7 SpecifyObjectiveGradient

This parameter did not show significant impact when using the generic test points as previously used. However, by changing the initial points to random values such as [200,400], it is shown that the first-order optimality showed higher accuracy.

Insert result differences

Through this comparison, we can conclude that with x % increased accuracy, it could potentially be very helpful for applications requiring zero first-order optimality error.

3.8 HessianApproximation

When the optimization tool is computing to find roots for the given function, hessian approximation is used. By default, the BFGS algorithm is used (Broyden-Fletcher-Goldfarb-Shanno). The input used for modification is the LBFGS, the limited memory version of the BFGS algorithm. The LBFGS method computes the results faster but there is decreased accuracy. This is because LBFGS approximates for every next answer to increase computation speed, whereas the default algorithm calaculates an exact answer for every iteration. For this lab, there is no noticeable difference, which is probably due to a simple system function and lack of power from the computer.

4 Conclusions and Recommendations

For this lab we learned and put into practice how the use the MATLAB optimizatoin toolbox, specifically fmincon and fminunc to calculate the local minima of said system functions. We also adjusted different parameters and to determine desired values. When using fmincon, are able to input constraints of linear inequalities, equalities, bounding, and nonlinear conditions to restrict the domain and range of each variable. Lastly, we are able to adjust the options of the minimization function to alter tolerance, number of evaluations, specify the objective gradient, and hessian approximation.

A recommendation for future labs, it might have been more efficient and beneficial to be taught the specific uses or theories behind some of the options. It was hard to realize and distinguish the difference between the suboptions modified in question 3. However, if given more direction on the actual differences, it would have been easier to compare experimental and theoretical results to determine why there were noticeable differences, otherwise we could be possibly making incorrect assumptions.

5 References

[1] https://www.motorcyclecruiser.com/stopping-power