CHE 362

OPTIMIZATION WITH MATLAB

INTRODUCTION:

There are three main toolboxes which are generally used for optimization purposes in MATLAB;

1. **Curve fitting toolbox**

   Graphical tools and command-line functions for fitting curves and surfaces from data are available in Curve Fitting Toolbox. We can perform exploratory data analysis using this toolbox. Pre-processing and post-processing of data, comparing candidate models, and removing outliers are possible using this toolbox. The inbuilt library of linear and nonlinear models can be used to conduct regression analysis. We can also specify our customized model equations. The library provides us the starting conditions and optimized solver parameters to enhance the quality of our results. The nonparametric modeling techniques like splines, interpolation, and smoothing are also possible with this toolbox.

2. **Global optimization toolbox**

   Global Optimization Toolbox provides techniques, which search for global solutions to problems that have multiple stationary points. We can employ these solvers to optimize problems where the objective or constraint functions are continuous, discontinuous, stochastic, or even does not possess derivatives.

3. **Optimization toolbox**

   Optimization Toolbox comprises algorithms for standard and large-scale optimization. These algorithms solve unconstrained and constrained continuous and discrete problems. This toolbox comprises functions for linear programming, quadratic programming, nonlinear optimization, nonlinear least squares, solving systems of nonlinear equations, binary integer programming, and multi-objective optimization.

Table 1: Description of MATLAB function used for optimization

Function	Description
Curve fitting toolbox	
cftool	Open Curve Fitting Tool
fit	Fit model to data
sftool	Open Surface Fitting Tool
Optimization toolbox	
bintprog	Solve binary integer programming problems

fminbnd	Find minimum of single-variable function on fixed interval
fgoalattain	Solve multiobjective goal attainment problems
Fmincon	Find minimum of constrained nonlinear multivariable function
Fminimax	Solve minimax constraint problem
fminsearch	Find minimum of unconstrained multivariable function using derivative-free
fminunc	Find minimum of unconstrained multivariable function
linprog	Solve linear programming problems
Global Optimization problem	
ga	Find minimum of function using genetic algorithm
Gamultiobj	Find minima of multiple functions using genetic algorithm
patternsearch	Find minimum of function using pattern search
simulannealbnd	Find unconstrained or bound-constrained minimum of function of several

Example 1: Write the MATLAB code for solving the problem given below;

Minimize $31-11x1 - 5x2 + 3x1 + x2$

Solution

The function "fminunc" is used for solving unconstrained multivariable optimization problem. Follow the steps given below;

Step 1: Create an M file

```
function f = myfun(x)
f = 31-11*x(1)-5*x(2) + 3*x(1)^2 + x(2)^2;
```

Step 2: Save this file with file name "myfun"

Step 3: Write the code to optimized the problem using a guess starting point/initial value of $x_1=1$ and $x_2=1$. Note that we have only two variables x_1 and x_2

```
x0 = [1,1];
[x,fval] = fminunc(@myfun,x0);
x,fval
```

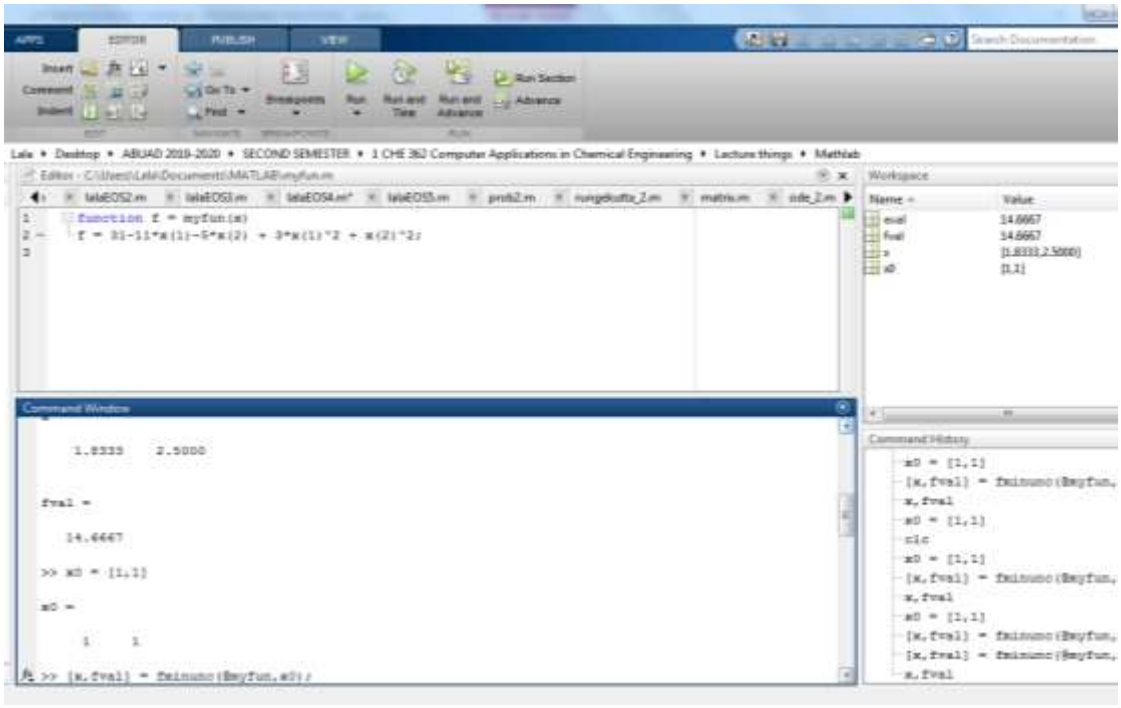
Note: The above command will be written on the command window

Step 4: Run the programme

The result is

```
x =
1.8333          2.5000
```

fval =
14.6667



Example 2: Solve the optimization problem with constraints;

Minimize $31-11x_1 - 5x_2 + 3x_1 + x_2$

subject to

$x_1 + x_2 \leq 3$

$x_1^2 + x_2^2 = 8$

Solution:

Step 1: Create a M file for the objective function (objfun.m):

```
function f = objfun(x)
f = 31-11*x(1)-5*x(2) + 3*x(1)^2 + x(2)^2;
```

Step 2: Create another M file for the constraint functions (confun.m):

```
function [c, ceq] = confun(x)
% Nonlinear inequality constraints
c = x(1)+x(2)-3;
% Nonlinear equality constraints
ceq = x(1)^2+x(2)^2-8;
```

Step 3: Invoke constrained optimization routine:

```
x0 = [1,1];            % Make a starting guess at the solution
options = optimset('Algorithm','active-set');
fmincon(@objfun, x0, [ ], [ ], [ ], [ ], [ ], [ ], @confun, options);
[x,fval]
```

which gives the result:

Ans =

1.8333 2.5000 14.6667

i.e

x1, x2 and Minimum function