CODE APPENDIX, problems in order (Matlab=white, Python=black)

```
%=========================================================================
%== ASSIGNMENT : hw2, Optimization (MAE 5930)
%== AUTHOR     : Jared Hansen
%== DUE        : Thursday, 09/26/2019
%=========================================================================
rng(1776)
clear all; close all; clc;
```

```
%=========================================================================
%=========================================================================
%== PROBLEM 5
%=========================================================================
%=========================================================================

%=========================================================================
%== Create the matrix A and the vector b, filled with ~U[0,1] values
%=========================================================================
A = rand(100,3);
b = rand(100,1);

%=========================================================================
%== 5(B): solve using pinv command
%=========================================================================
%== The analytical solution is: [(A.T * A)^(-1)] * [(A.T) * b]
pinv_soln = pinv((A.')*(A)) * ((A.')*(b));
pinv_soln

%=========================================================================
%== 5(C): solve using quadprog command
%=========================================================================
%== Define matrix H and row vector f
H = ((A.')*(A));
f = (-1.0*(b.')*(A));
qprog_soln = quadprog(H, f);
qprog_soln

%=========================================================================
%== 5(D): solve using fminunc command
%=========================================================================
%== Define the function and an initial guess
fun = @(x) (1/2.0)*(x)*(A.')*(A)*(x.') - (b.')*(A)*(x.');
x0 = [1.0, 1.0, 1.0];
fminunc_soln = fminunc(fun, x0);
fminunc_soln
```

```
clear all; close all; clc;
%=========================================================================
%=========================================================================
%== PROBLEM 6
%=========================================================================
%=========================================================================

%== Define the Rosenbrock function
%rosen = @(x) (1-x(1))^2 + 100*(x(2) - x(1)^2)^2;
%== Define the [x,y] vector = [x(1), x(2)] = [x(1), (x(1))^2]
x(1) = 3.0;
x(2) = (x(1))^2;
%== Define the Hessian of the Rosenbrock function
ros_hess(1,:) = [2-400*x(2)+1200*(x(1)^2)  -400*x(1)];
ros_hess(2,:) = [-400*x(1)  200];
%== Output eigenvalues of the Hessian of the Rosenbrock function
eig(ros_hess)
```

```
clear all; close all; clc;
%=========================================================================
%=========================================================================
%== PROBLEM 7
%=========================================================================
%== For the optimization problem described in Problem 4:
%=========================================================================

%=========================================================================
%== 7(A): solve using fminunc command
%=========================================================================

%== Define the Rosenbrock function
rosen = @(x) (1-x(1))^2 + 100*(x(2) - x(1)^2)^2;
x0 = [2.0, 2.0];
fminunc_rosen_soln = fminunc(rosen, x0);
fminunc_rosen_soln
```

```
clear all; close all; clc;
%=========================================================================
%=========================================================================
%== PROBLEM 8
%=========================================================================
%=========================================================================

%=========================================================================
%== 8(D): Convert the problem to standard form and solve using linprog
%=========================================================================

% Coefficients for the objective function f = x(1) - x(2)
f = [1, -1];
% The matrix for inequality constraints
A(1,:) = [1, 1];
A(2,:) = [-1, 2];
A(3,:) = [1, -3];
% The column vector for inequality constraints
b = [1; 2; 3];
% The vector bounding x on the low side
lb = [-1; -inf];
% The vector bounding x on the high side
ub = [inf; inf];

% Solve the problem using linprog
linprog_soln = linprog(f, A, b, [], [], lb, ub);
linprog_soln
```

```
%==========================================================================
%==========================================================================
%== MATLAB AUTOMATIC OUTPUT BELOW THIS POINT
%==========================================================================
%==========================================================================
```

```
pinv_soln =

    0.3673
    0.3322
    0.2414


Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint
 tolerance.


qprog_soln =

    0.3673
    0.3322
    0.2414


Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.


fminunc_soln =

    0.3673    0.3322    0.2414


ans =

   1.0e+03 *

    0.0001
    7.4019


Local minimum found.

Optimization completed because the size of the gradient is less than
```

```
the value of the optimality tolerance.


fminunc_rosen_soln =

    1.0000    1.0000


Optimal solution found.


linprog_soln =

   -1.0000
    0.5000
```

Published with MATLAB® R2019a