Homework 2
ME/SYS 564-A
Fall 2018

1. **Surrogate modeling** (follow-up on Week 4 in-class exercise): [25pt]
   Use the Rosenbrock function (Eqn. (16) from Yang 2010), and use my sample codes as much as you want. Submit your commented code and results for the following:
   a. Sample the design space with 100 points in (1) a full factorial and (2) an optimal Latin hypercube. Present your 2 designs of experiments (DOEs) in side-by-side scatter plots of the design space (x vs y).
   b. Using the full factorial sample, fit 2 linear regression models: 1 using only linear terms, and the other using linear, interaction, and quadratic terms, following the form:
      $$f(x,y) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 xy + \beta_4 x^2 + \beta_5 y^2$$
      Measure and report the R^2 value and MSE value of each of the two models.
   c. Use MATLAB's nftool function to fit a neural network to the same set of inputs and outputs. Try this with 10 neurons and again with 30 neurons, using all three of the training algorithm options. Record the R^2 and MSE values under each of these 6 scenarios, and report these 12 values in a table.
   d. Briefly (in 3-5 sentences) compare linear regression and neural networks for this application. Which seems to have the best fit? Which do you prefer and why?

2. **Derivative-free algorithm** (follow up to Week 3 in-class programming exercise): [25pt]
   Write a MATLAB code to do a coordinate search (i.e., Hooke-Jeeves) to minimize the 2-dimensional Rosenbrock function (same as Problem 1). Search in the four cardinal directions {[1,0], [0,1], [-1,0], [0,-1]}, and when the step size decreases, reduce it by a factor of 2. Perform 100 iterations before stopping. Execute the code under nine scenarios. Use all combinations of the three starting points {(0,0), (5,5), (-5,-5)} and three different initial step sizes (0.5, 1, and 2). Submit your commented code and the following:

   a. For each of the nine scenarios, present a set of plots that show:
      i. Progression of the objective function over time (f vs. iteration), and
      ii. Movement in the design space (x_1 vs. x_2).
   b. Briefly (in 3-5 sentences) discuss the results – do you think this is a good algorithm? What do you like/dislike about it?