Faculty of Arts and Sciences
Department of Computer Science
CMPS 351 – Numerical Optimization
Assignment 1 – Due Jan 28, 2019

Reading Material:

- Introduction to Python, Numpy, and Matplotlib. Tutorials may be found on the web and there are some links on Moodle. Please make sure you are comfortable working with vectors and matrices, plotting, and writing functions.
- Appendices A.1 and A.2 of reference 1 review linear algebra and calculus.

1. Function Plot. The Rosenbrock function is commonly used to test the behavior of optimization algorithms. The function is defined as:

$$f(x) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2$$

The minimizer of this function is the point $x = [1, 1]^t$. Draw contour lines of the function in the region near the minimum. You will need to use the function `meshgrid` first to generate a grid, compute the function values at the grid points, and then use the `contour` function and tune its parameters to obtain the desired contour plot.

2. Gradient Computation. Compute the partial derivatives of the Rosenbrock function analytically, and use them to write a function that returns the gradient of the Rosenbrock function.

```
def rosen_grad(x)
```

3. Backtracking Line Search. Write a routine for a backtracking line search method using the steepest descent direction. Test the performance of the algorithms on Rosenbrock's function starting at the point $x^0 = [-1.2, 1]^t$ by finding the number of iterations till convergence to a gradient norm of 10^{-5}. The header of this routine should look something like this:

```
def steepest_descent_bt(fun, grad, x0)
```

where the arguments `fun` and `grad` are Python functions that return the objective function value and its gradient respectively.

4. Performance Assessment. Assess the performance of the steepest descent method on the problem above by generating a semilog plot of the error $|f^k - f^*|$ vs. iteration count. Comment on the convergence rate.