Assignment requirements/suggestions:

Email me your code that implements or solves the given problems with each problem in a separate file.

Please reuse code between different problems and do not cut and paste. Include functions that are used in multiple problems in their own file.

Make your code clean and readable, including comments where appropriate.

Make any plots and figures professional looking and readable. Label things appropriately (axes, legends, etc.) and present the data is a clear manner.

Ensure your code runs and provide an code example that shows how I would run your code for different inputs. I will return code that throws an error to you to fix.

Try to make your code efficient but do not spend too much time optimizing the code for speed.

Please provide any specific results in a Word or PDF formats.

1. [10 points] Write a code that scans a univariate objective implements the Powell method for minimizing a multivariate objective function described at the end of the assignment.
2. [10 points] Write a code that implements any gradient-based method for minimizing a multivariate objective function.
3. [10 points] Write a function that implements the N-dimensional Rosenbrock objective function:
$$F(\boldsymbol{x}) = \sum_{i=1}^{N-1} \left[b\left(x_{i+1} - x_i^2\right)^2 + (a - x_1)^2 \right]$$
Write a function or functions to test your code from the first two problems using the Rosenbrock function for various values of N and for various initial estimates.
4. [20 points] Write a function that implements the following two-dimensional objective function:
$$F(x,y) = \left(x^2 + y - 11\right)^2 + \left(x + y^2 - 7\right)^2.$$
Determine how many local minimums and maximums and their location for this objective function.
5. [20 points] Using the code from the first two problems or by implementing an alternative method, write a function that fits given data to a line through the origin using total least squares.
The total least squares method fits the data to a function by minimizing the sum of the minimum Eucledian distance from each data point to the curve. For the curve $y = ax + b$, one problem formulation would be
$$\text{Minimize } F(a) = \sum_{i=1}^{N} \left[(x_i - z_i(a,b))^2 + (y_i - [a z_i(a,b) + b])^2 \right],$$
where N is the number of data points (x_i, y_i) and
$$z_i = \frac{x_i + a y_i - ab}{a^2 + 1}.$$
Your function should take an 2xN array of data (the fitting data) and return an optimal a value that minimizes the objective function as well as the value of F at the minimum. Also, the code should create

a plot of the objective function over meaningful range of a and b values as well as a plot of the data and the best fit line.

To test your code, write a function that creates data by adding noise to a linear function with a given slope and intercept. For example, evaluate the formula

$$y_i = mx_i + b + \varepsilon\,(2R_i - 1)$$

for some set of N points x_i for $i = 1, 2, \ldots, N$, where m is the slope, b is the intercept, R_i is a random variable selected from uniform random with range $[0, 1]$, and ε is the magnitude of the noise. For small or zero ε, you code should return a result close to the chosen slope and intercept.

6. [20 points] Suppose a robot starts at the origin and must reach a target point P in the first quadrant of the Cartesian plane. The robot initially moves in a straight line to an intermediate point (x, y) and then turns and travels along a straight line towards the target P. The robot can move anywhere in the first quadrant and along the horizontal and vertical axes except for a circle centered at the point C with radius r.
   (a) Determine an objective whose minimum would describe an optimal route from the origin to the target.
   (b) Using your code from the first two problems or by implementing an alternative method, write a function that minimizes the objective function. This function should input a target point, P, and the center and radius of the excluded circle and return the optimal intermediate point. Additionally, your code should plot the objective function over a meaningful range of x and y values as well as a plot of the optimal path from the origin to the target point and the excluded circle.
   (c) Solve the minimization problem for the following parameters:
      1. $P = (5, 3)$, $C = (2, 2)$, $r = 1$.
      2. $P = (2, 4)$, $C = (2, 2)$, $r = 1$.
      3. $P = (3, 3)$, $C = (2, 2)$, $r = 1$.

Powell's Method Powell's method is a gradient-free search method similar to Univariate Search and Hooke and Jeeves. The basic algorithm works by searching in N search directions (a univariate search). The resulting direction that connects the current guess and the next guess is then added to the list of N search directions. The direction that contributed the most to new search direction is then removed.

The algorithm is

Given: Objective function $F(\boldsymbol{x})$, an initial estimate $\boldsymbol{x}_0 \in \mathbb{R}^N$, and a increment tolerance, $\varepsilon > 0$

1. Initialize set of N search direction $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$ with search directions aligned with the coordinate axes: $\boldsymbol{d}_k = \hat{\boldsymbol{e}}_k$.
2. Repeat for $k = 1, 2, \ldots$
   (a) Set $\boldsymbol{z} = \boldsymbol{x}_k$
   (b) For $i = 1$ to N
      i. Determine the value s_i^* that minimizes the function $f(s) = F(\boldsymbol{z} + s\boldsymbol{d}_i)$
      ii. Set $\boldsymbol{z} = \boldsymbol{z} + s_i^* \boldsymbol{d}_i$
   (c) Find the most successful search direction as

$$i^* = \underset{i=1,2,\ldots,N}{\arg\max} \, ||s_i^* \boldsymbol{d}_i||$$

   (d) Replace the most successful search direction, $\boldsymbol{d}_{i^*}$, with the new search direction

$$\boldsymbol{d}_{i^*} = \boldsymbol{z} - \boldsymbol{x}_k$$

   (e) Set $\boldsymbol{x}_{k+1} = \boldsymbol{z}$
3. Until $||\boldsymbol{x}_{k+1} - \boldsymbol{x}_k|| < \varepsilon$