Spring 24
207984790

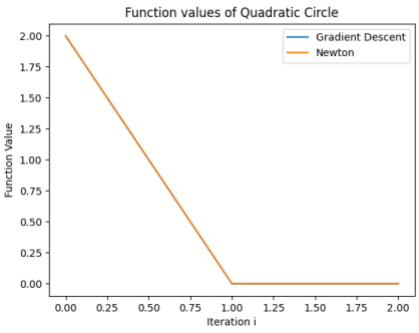
Numerical Optimization

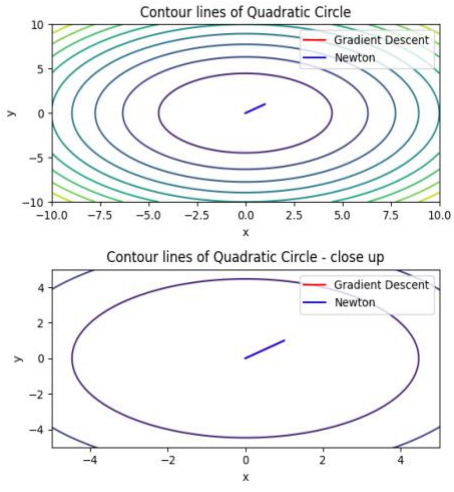
Programming Assignment 1 – Report

The assignment involved the implementation and testing of line search minimization methods for various objective functions. The project was organized into two directories: 'src' containing modules for algorithms and utility functions, and 'tests' containing modules for testing and examples. Within the 'src' directory, the 'unconstrained_min.py' module implemented algorithms supporting gradient descent and Newton's method with the Wolfe condition and backtracking for step length search. The 'utils.py' module provided utilities for plotting contour lines of objective functions and function values at each iteration. The 'tests' directory contained 'test_unconstrained_min.py' for testing the implemented methods on various examples. Each example function was minimized using both methods, and two plots were generated for each example: contour lines with iteration paths (containing 2 subplots with different limits) and function values versus iteration number for comparison. The report presents the final iteration details and success/failure flags for each test, along with plots illustrating the performance of the implemented methods.

Objective Functions:

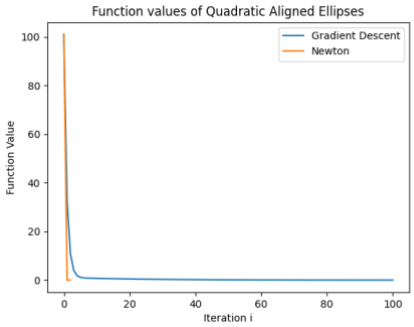
1. *Quadratic Example* $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ *- contour lines are circles:*

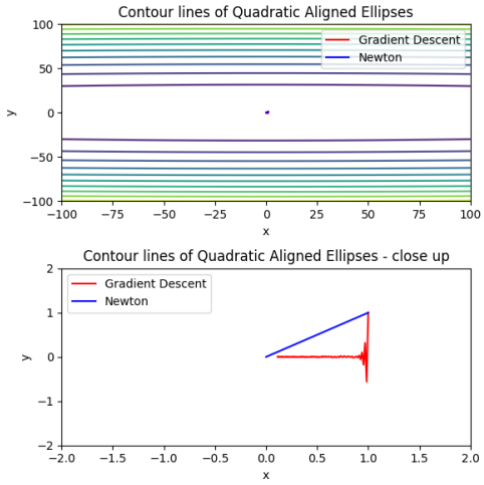




Method	Last Iteration Report	Success
Gradient descent	Iteration: i=1 Current location: x_i=[0. 0.] Objective value: 0.0	True
Newton	Iteration: i=1 Current location: x_i=[0. 0.] Objective value: 0.0	True

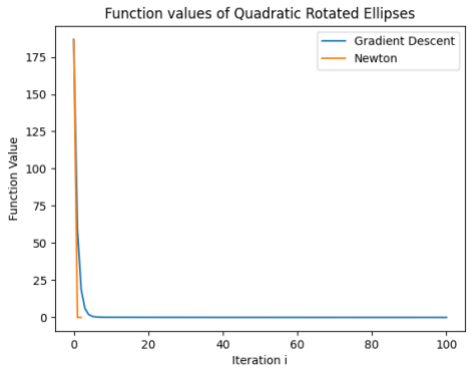
2. *Quadratic Example* $Q = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$ *– contour lines are axis aligned ellipses*

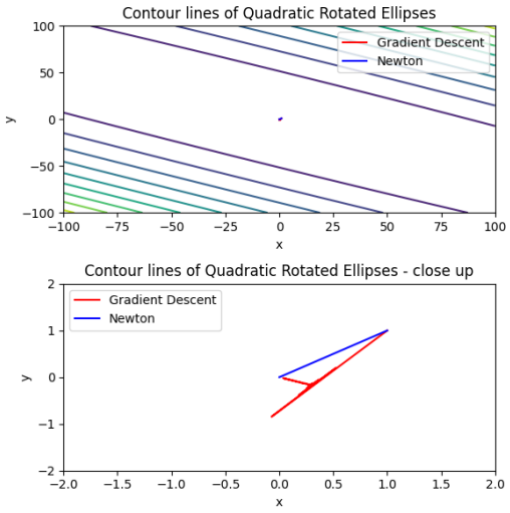




Method	Last Iteration Report	Success
Gradient descent	Iteration: i=99 Current location: x_i=[0.11271997 0.0008856] Objective value: 0.012784220095399295	False
Newton	Iteration: i=1 Current location: x_i=[0. 0.] Objective value: 0.0	True

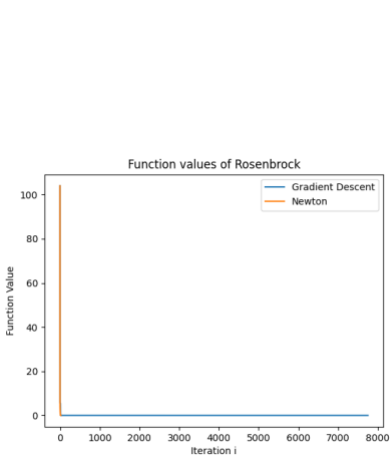
3. *Quadratic Example* $Q = \begin{bmatrix} \frac{\sqrt{3}}{2} & -0.5 \\ 0.5 & \frac{\sqrt{3}}{2} \end{bmatrix}^T \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} & -0.5 \\ 0.5 & \frac{\sqrt{3}}{2} \end{bmatrix}$ *– contour lines are rotated ellipses*

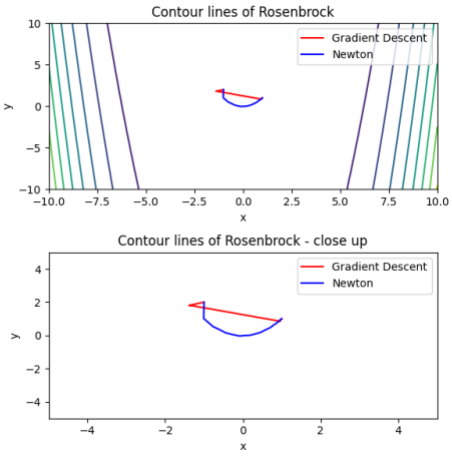




Method	Last Iteration Report	Success
Gradient descent	Iteration: i=99 Current location: x_i=[0.03646722 -0.02068459] Objective value: 0.0017678621294560908	False
Newton	Iteration: i=1 Current location: x_i=[1.41994963e-29 -8.28303950e-30] Objective value: 2.707703457167278e-58	True

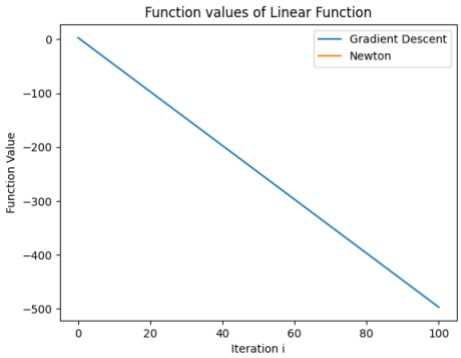
4. *Rosenbrock Function* $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

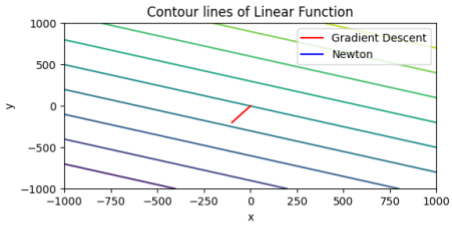


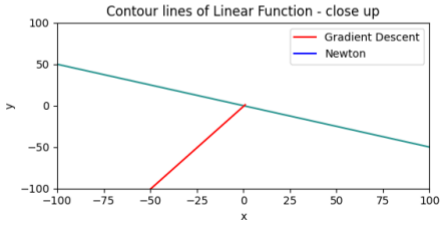


Method	Last Iteration Report	Success
Gradient descent	Iteration: i=7750 Current location: x_i=[0.99989437 0.99978771] Objective value: 1.1267173494484024e-08	True
Newton	Iteration: i=21 Current location: x_i=[1. 1.] Objective value: 4.930380657631324e-30	True

5. *Linear Function*

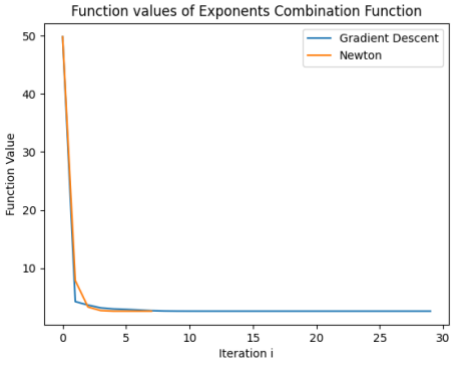


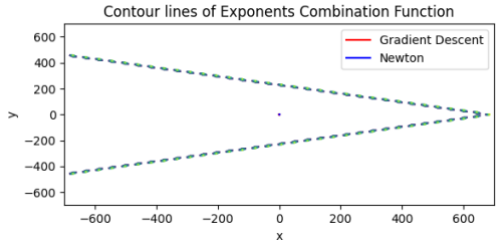


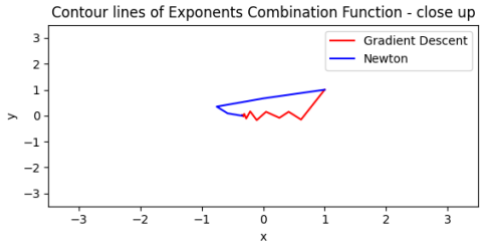


Method	Last Iteration Report	Success
Gradient descent	Iteration: i=99 Current location: x_i=[-99. -199.] Objective value: -497.0	False
Newton	An error occurred Could not solve Newtons method due to np.linalg.LinAlgError (The hessian is not defined!)	False

6. $f(x_1, x_2) = e^{x_1 + 3x_2 - 0.1} + e^{x_1 - 3x_2 - 0.1} + e^{-x_1 - 0.1}$







Method	Last Iteration Report	Success
Gradient descent	Iteration: i=28 Current location: x_i=[-3.46572886e-01 -9.76799665e-07] Objective value: 2.559266696664345	True
Newton	Iteration: i=6 Current location: x_i=[-3.46573590e-01 -3.33659182e-18] Objective value: 2.5592666966582156	True