Computer Exercise 1

Edoardo Mangia

In this assignment, different methods are used to minimize some given functions.
Along with the implementation of the methods, some considerations will be highlighted.

Contents

1 Description of the Methods Used

- Steepest Descent Method:

  The steepest descent method is an iterative optimization algorithm used to minimize a function. It updates the current point by moving in the direction of the negative gradient of the function at the current point. The update rule is given by:

  $$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

  Where:

  - x_k is the current point at iteration k,
  - $\nabla f(x_k)$ is the gradient of the function f at x_k,
  - α_k is the step size at iteration k.

  The method proceeds as follows:

  1. Initialize with an initial guess x_0.
  2. Compute the gradient at the current point x_k, i.e., $\nabla f(x_k)$.
  3. Update the current point using the update rule:

     $$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

     where α_k is typically determined using a line search.
  4. Repeat steps 2 and 3 until convergence, i.e., until $\|\nabla f(x_k)\|$ is sufficiently small.

  This method is particularly effective for smooth, convex functions but may converge slowly for poorly conditioned problems.

- Newton's method:

  Newton's method is an iterative optimization algorithm used to find the stationary points of a function, which are candidates for local minima or maxima. The update rule is based on the Taylor series expansion and is given by:

  $$x_{k+1} = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

  Where:

  - x_k is the current point at iteration k,
  - $\nabla f(x_k)$ is the gradient of the function f at x_k,
  - $\nabla^2 f(x_k)$ is the Hessian matrix of second derivatives of f at x_k.

  The method proceeds as follows:

  1. Initialize with an initial guess x_0.
  2. Compute the gradient $\nabla f(x_k)$ and the Hessian $\nabla^2 f(x_k)$ at the current point x_k.
  3. Update the current point using the update rule:

     $$x_{k+1} = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

  4. Repeat steps 2 and 3 until convergence, i.e., until $\|\nabla f(x_k)\|$ is sufficiently small or until x_k does not change significantly.

  Newton's method typically converges faster than the steepest descent method, especially near the optimum, but it requires the computation of the Hessian matrix, which can be computationally expensive.

- Newton's method (Levenberg-Marquardt)

  The Levenberg-Marquardt algorithm is an optimization method that blends the Newton method and gradient descent, specifically designed for minimizing nonlinear least squares problems. The update rule combines the Hessian matrix with a damping factor to ensure stability during optimization:

  $$x_{k+1} = x_k - \left(\nabla^2 f(x_k) + \lambda_k I\right)^{-1} \nabla f(x_k)$$

  Where:

  – x_k is the current point at iteration k,
  – $\nabla f(x_k)$ is the gradient of the function f at x_k,
  – $\nabla^2 f(x_k)$ is the Hessian matrix of second derivatives of f at x_k,
  – λ_k is the damping parameter,
  – I is the identity matrix.

  The method proceeds as follows:

  1. Initialize with an initial guess x_0 and choose an initial damping parameter λ_0.
  2. Compute the gradient $\nabla f(x_k)$ and the Hessian $\nabla^2 f(x_k)$ at the current point x_k.
  3. Update the current point using the update rule:

     $$x_{k+1} = x_k - \left(\nabla^2 f(x_k) + \lambda_k I\right)^{-1} \nabla f(x_k)$$

  4. Adjust the damping parameter:
     – If the update reduces the objective function, decrease λ_k (e.g., $\lambda_{k+1} = \frac{\lambda_k}{10}$).
     – If the update does not reduce the objective function, increase λ_k (e.g., $\lambda_{k+1} = 10\lambda_k$).
  5. Repeat steps 2-4 until convergence, i.e., until $\|\nabla f(x_k)\|$ is sufficiently small or until x_k does not change significantly.

  The Levenberg-Marquardt method is particularly effective for nonlinear least squares problems, providing a robust approach that combines the rapid convergence of Newton's method with the stability of gradient descent.

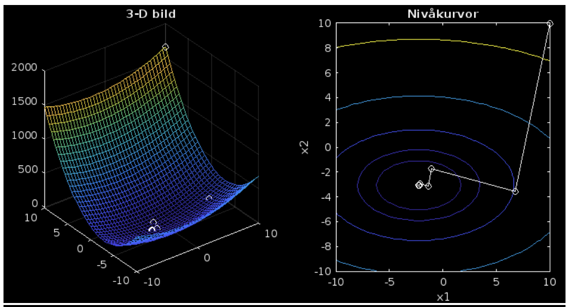
2 Function 1

2.1

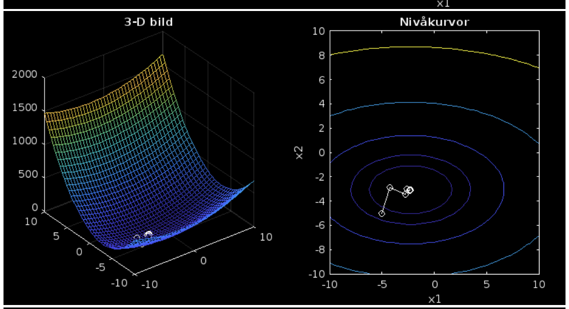
The function considered is

$$f(x_1, x_2) = 2(x_1 + 1)^2 + 8(x_2 + 3)^2 + 5x_1 + x_2$$

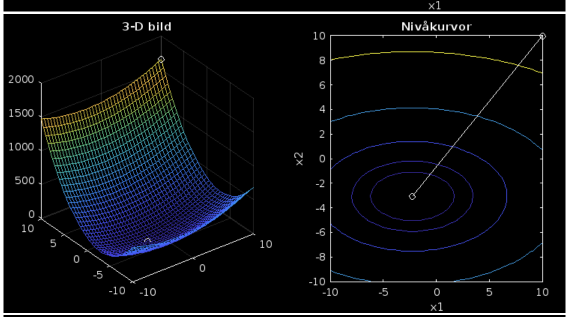
These are the results after studying the function:



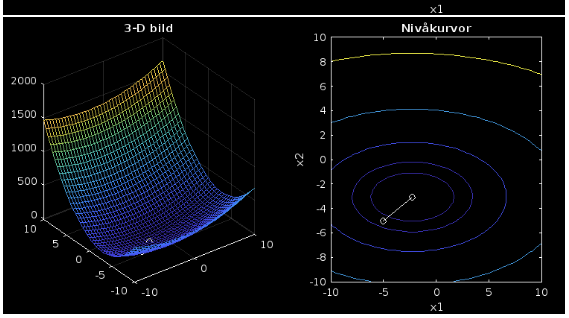
Steepest Descend method
Starting point: (10, 10)
Function value: -11.1562
Solution point: (-2.2491, -3.06255)
No. of iterations: 9



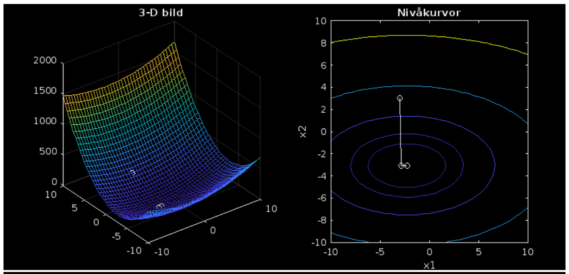
Steepest Descend method
Starting point: (-5, -5)
Function value: -11.1562
Solution point: (-2.25225, -3.06234)
No. of iterations: 9



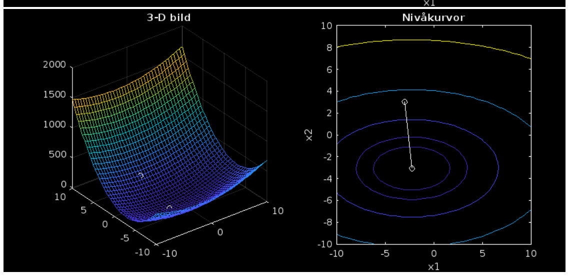
Newton's (unit step) method
Starting point: (10, 10)
Function value: -11.1562
Solution point: (-2.25, -3.0625)
No. of iterations: 1



Newton's (unit step) method
Starting point: (-5, -5)
Function value: -11.1562
Solution point: (-2.25, -3.0625)
No. of iterations: 1



Steepest Descend method
Starting point: (-3, 3)
Function value: -11.1562
Solution point: (-2.2512, -3.06255)
No. of iterations: 3



Newton's (unit method) method
Starting point: (-3, 3)
Function value: -11.1562
Solution point: (-2.25, -3.06255)
No. of iterations: 1

2.2

The point obtained is locally optimal. If we can prove that the function is also convex, we can say that this point will also be globally optimal.

To prove that $f(x_1, x_2)$ is convex, we use the Hessian matrix test.

The gradient of the function $f(x_1, x_2)$ is given by:

$$\nabla f(x_1, x_2) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right)$$

First, we calculate the partial derivatives:

$$\frac{\partial f}{\partial x_1} = 2 \cdot 2(x_1 + 1) + 5 = 4(x_1 + 1) + 5$$

$$\frac{\partial f}{\partial x_2} = 2 \cdot 8(x_2 + 3) + 1 = 16(x_2 + 3) + 1$$

Thus, the gradient is:

$$\nabla f(x_1, x_2) = (4(x_1 + 1) + 5, 16(x_2 + 3) + 1)$$

The Hessian matrix $\nabla^2 f$ contains the second-order partial derivatives:

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

We compute the second-order derivatives:

$$\frac{\partial^2 f}{\partial x_1^2} = \frac{d}{dx_1}(4(x_1 + 1) + 5) = 4$$

$$\frac{\partial^2 f}{\partial x_2^2} = \frac{d}{dx_2}(16(x_2 + 3) + 1) = 16$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = 0$$

Thus, the Hessian matrix is:

$$\nabla^2 f = \begin{bmatrix} 4 & 0 \\ 0 & 16 \end{bmatrix}$$

The Hessian matrix is diagonal with entries 4 and 16, both of which are positive. Since the Hessian matrix is positive definite, the function $f(x_1, x_2) = 2(x_1 + 1)^2 + 8(x_2 + 3)^2 + 5x_1 + x_2$ is convex.

Fundamental Theorem of Global Optimality:
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function defined on a closed and bounded set $S \subseteq \mathbb{R}^n$. The point $\mathbf{x}^* \in S$ is said to be a global minimizer of f if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in S.$$

If f is a convex function, then any local minimizer is also a global minimizer. Specifically, if S is convex and f is convex on S, then:

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) \quad \forall \mathbf{x} \in S.$$

Furthermore, if f is continuous on S, then f attains its minimum and maximum values on S..

Since the function is convex, for the Fundamental Theorem of Global Optimality, the obtained point is also globally optimal.

2.3

The Steepest Descent method uses the gradient of the function to determine the search direction and a fixed coefficient α, chosen arbitrarily, to determine the step length.
The update rule for a new point x_k, starting from a point x_k, is defined as

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

The Newton's method instead, uses the Hessian of the function to determine the step length.
The update rule for a new point x_k, starting from a point x_k, is defined as

$$x_{k+1} = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

The curvature information from the Hessian, allows the Newton's method to take bigger steps when the function is steeper. On the contrary, it will take smaller steps when the function is less steep.

This method converges to the optimal point quadratically with each iteration.
In other words, the error, or difference between the current point x_n and the optimal point $x*$, is reduced quadratically with each iteration.

Because of this, when the function considered is quadratic, the method reaches a optimal point in just 1 iteration.

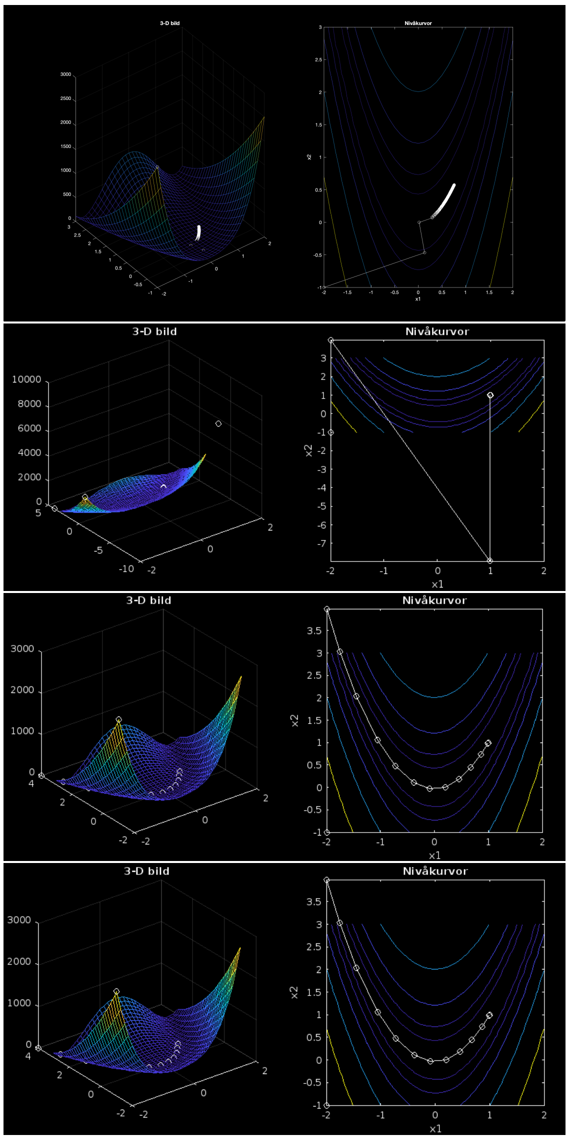
3 Function 2 (Rosenbrock's function)

3.1

The function considered is

$$f(x_1, x_2) := 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

These are the results after studying the function:



Steepest Descend method
Starting point: (-2, -1)
Function value: 0.0586
Solution point: (0.7596, 0.5741)
No. of iterations: 201

Newton's (unit step) method
Starting point: (-2, -1)
Function value: 1.9932e-27
Solution point: (1, 1)
No. of iterations: 5

Newton's (Marquardt's) method
Starting point: (-2, -1)
Function value: 2.6568e-11
Solution point: (0.9999, 0.9999)
No. of iterations: 14

Newton's (modified) method
Starting point: (-2, -1)
Function value: 2.6568e-11
Solution point: (0.9999, 0.9999)
No. of iterations: 14

3.2

The Rosenbrock function is a non-convex function commonly used as a performance test problem for optimization algorithms.

With the same method as previously stated we can find out that the hessian of function 2.
Thus, the Hessian matrix is:

$$\nabla^2 f(x_1, x_2) = \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}$$

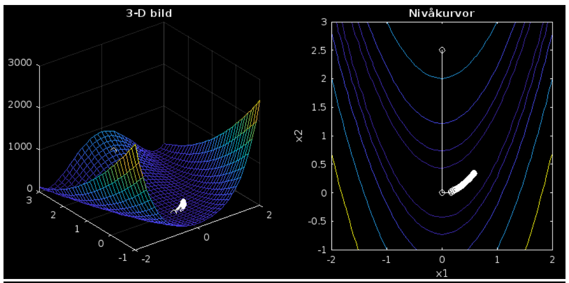
We can se that the matrix is not positive definite since for eg x1 = 1 and x2 = 4.

$$\nabla^2 f(x_1, x_2) = \begin{bmatrix} -398 & -400 \\ -400 & 200 \end{bmatrix}$$

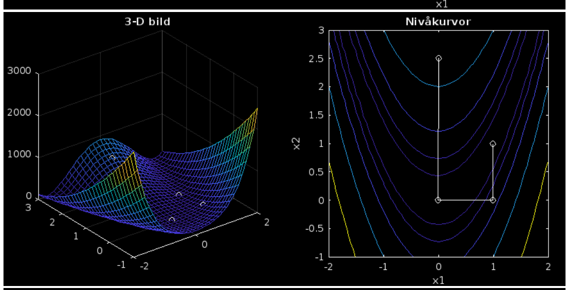
The global minimum is found in $(x_1, x_2) = (1, 1)$, where $f(1, 1) = 0$.

By using the Newton's method, in it's different iterations, it was possible to converge to the minimum in significantly less iterations, compared to using the steepest descent method.

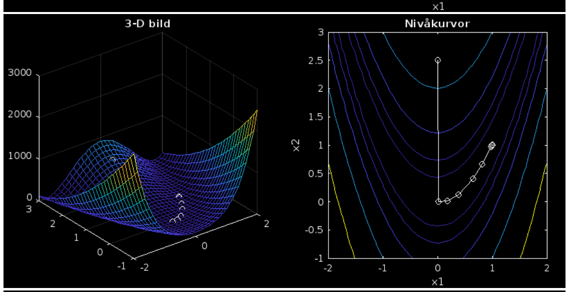
3.3



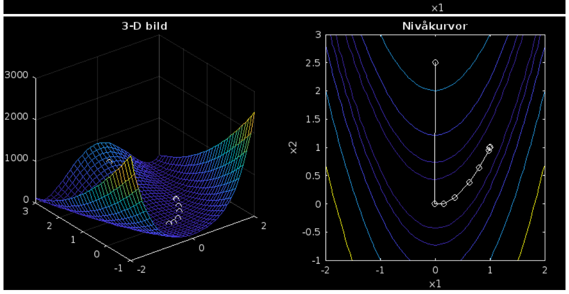
Steepest Descent method
Starting point: (0, 2.5)
Function value: 0.17365
Solution point: (0.5846, 0.3385)
No. of iterations: 100



Newton's (unit step) method
Starting point: (0, 2.5)
Function value: 6.4029e-07
Solution point: (0.9992, 0.9984)
No. of iterations: 3



Newton's (Marquardt's) method
Starting point: (0, 2.5)
Function value: 9.5434e-11
Solution point: (1.000, 1.000)
No. of iterations: 8



Newton's (modified) method
Starting point: (0, 2.5)
Function value: 2.3718e-09
Solution point: (1.000, 1.000)
No. of iterations: 8

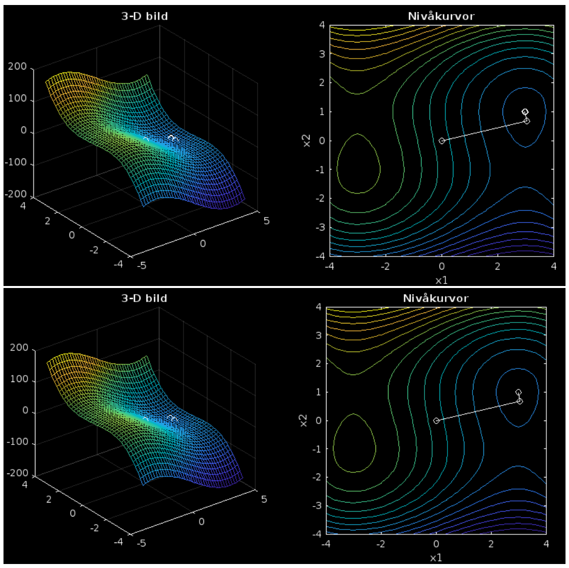
4 Function 4

4.1

The function considered is

$$f(x_1, x_2) := x_1^3 + 2x_2^3 - 27x_1 - 6x_2$$

These are the results after studying the function:



Steepest Descent method
Starting point: (0, 0)
Function value: -58
Solution point: (2.999, 0.999)
No. of iterations: 4

Newton's (Marquardt's) method
Starting point: (0, 0)
Function value: -58
Solution point: (3, 0.999)
No. of iterations: 3

By choosing $x_1, x_2 = (0, 0)$ as the starting point, we find that the Newton's method (unit step) doesn't behave well and can't find an optimal point in the first iteration.

The reason for this, goes back to how the method rely on the information of the Hessian of the function to search for new points.

In $x_1, x_2 = (0, 0)$, the Hessian of the function is 0 and therefore the computation results in an error.

For the same starting point, the other methods, do not rise errors during the computation.

This is why:

- The Steepest Descent method, does not rely on the Hessian of the function to calculate the next optimal point in each iteration.
- The Levenberg-Marquardt modification of the Newton's method, doesn't use the Hessian of the function, but rather a first-order derivative approximation of the latter, the Jacobian matrix.

4.2

Studying the function, we can find some stationary points.

In particular:

- Local minimum in $(-3, -1)$ (Function Value: 58).
- Saddle point in $(-3, 1)$ (Function Value: 50).
- Saddle point in $(3, -1)$ (Function Value: -50).
- Local minimum in $(3, 1)$ (Function Value: -58).