Università della Svizzera italiana | Institute of Computing CI

Optimization Methods **2024**

Student: Jeferson Morales Mariciano

Discussed with: Leonardo Birindelli

Assignment 2

Due date: Friday, 12 April 2024, 12:00 AM

Exercise 1

Considering the highly non-linear Rosenbrock's function:

$$f(x, y) := (1 - x)^2 + 100(y - x^2)^2 \tag{1}$$

1, 2, 3

Implement in MATLAB two functions: Newton's method (Newton.m), Steepest descent (Gradient) method (GD.m). Both methods can be run with backtracking algorithm (backtracking.m) with step size $\beta = 1$. Use the following values for the backtracking parameters: $\tilde{\alpha} = 1, \rho = 0.9$. You can choose the parameter $c_1 \in [0.5, 10^{-4}]$.

Minimize the Rosenbrock's function 1 by using the Steepest Descent (Gradient) method with backtracking and fixed step size $\beta = 1$. Use starting value $x_0 = (0, 0)$, maximum number of iterations $N = 50000$ and tolerance $TOL = 10^{-6}$.

Minimize the Rosenbrock's function 1 by using Newton method with backtracking and fixed step size $\beta = 1$. Use same parameters as for SD.

Matlab scripts are provided in */code* folder. The 2 main files to run are: *GD.m, Newton.m.* They handle both computations and visualization of the Rosenbrock's function with the corresponding methods.

4, 5, 6

Plot the obtained iterates on the energy landscape in 2D. Analyze convergence behaviour of the methods by plotting the gradient norm and the function value at each iteration. Compare and comment on the the performances of the different methods.

Figure 1: Convergence comparison between methods with Backtracking

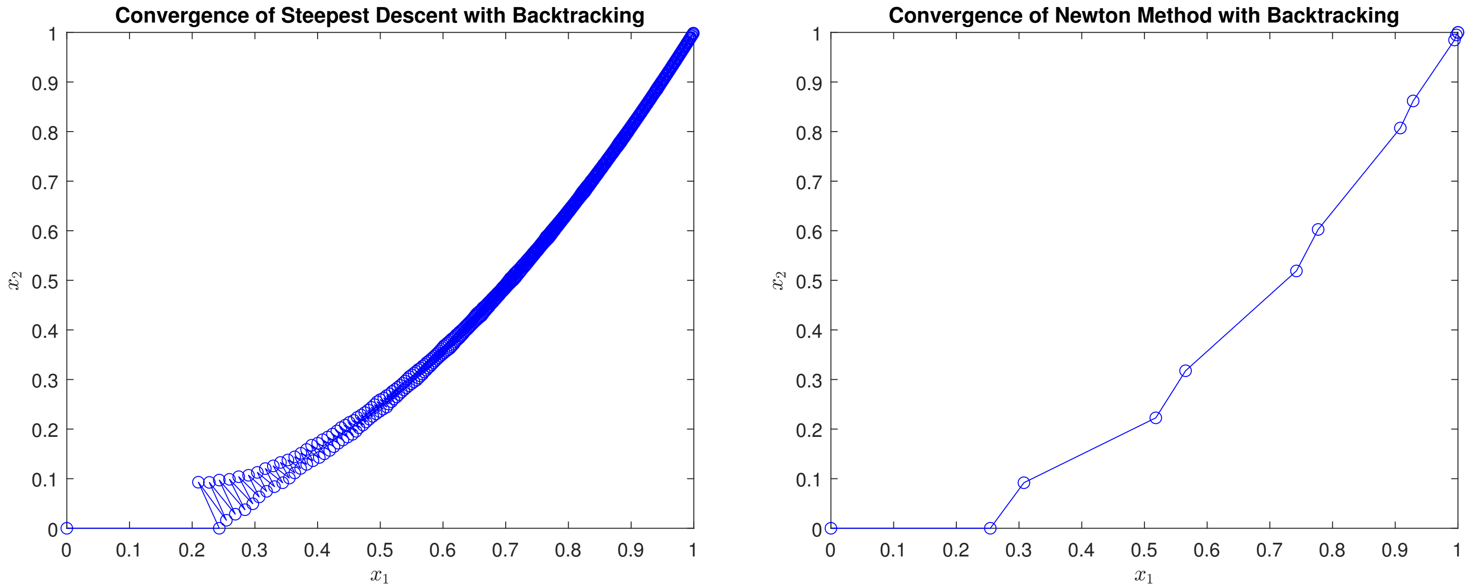


Figure 2: Visualization of Steepest Descent with Backtracking

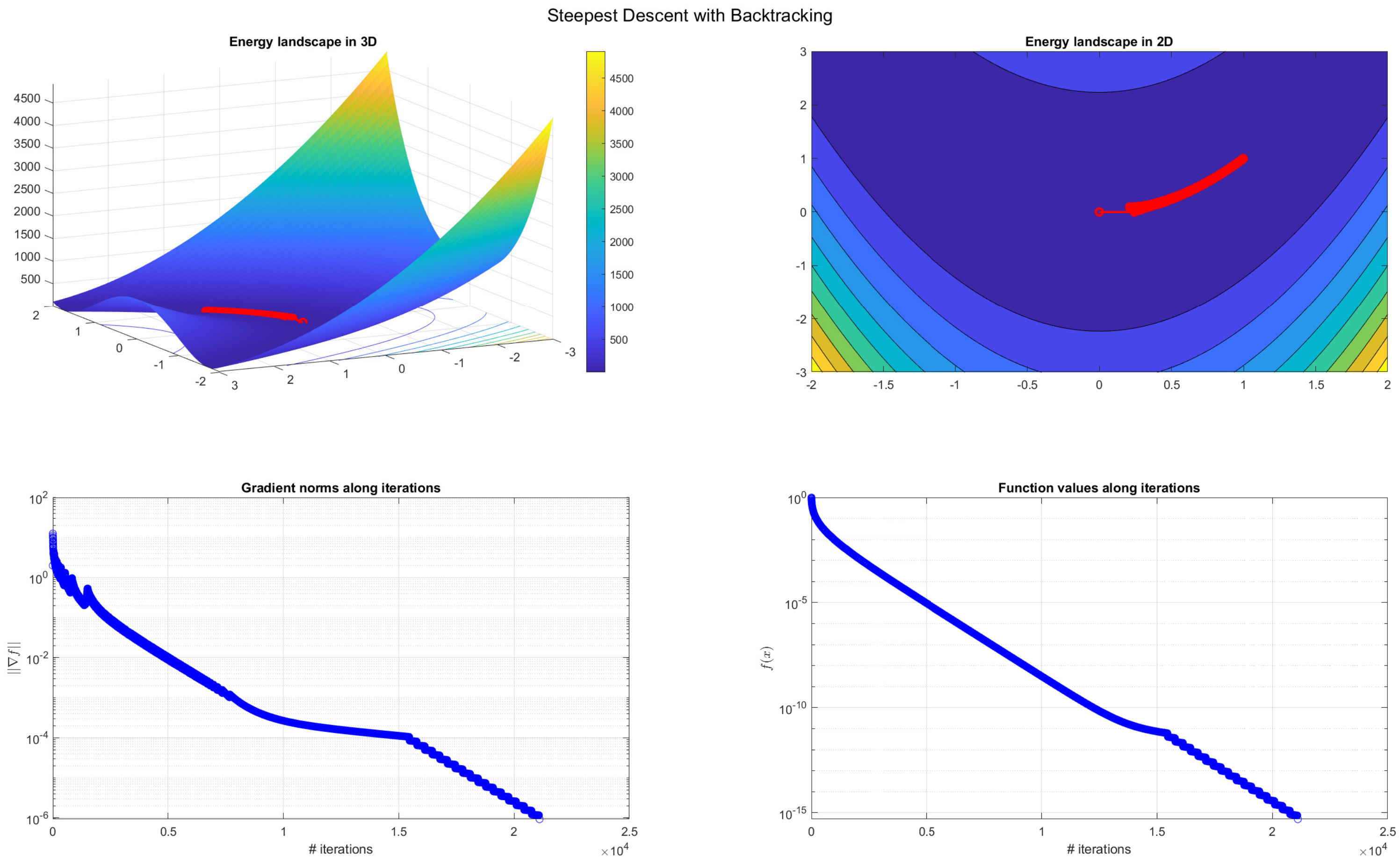


Figure 3: Visualization of Newton Method with Backtracking

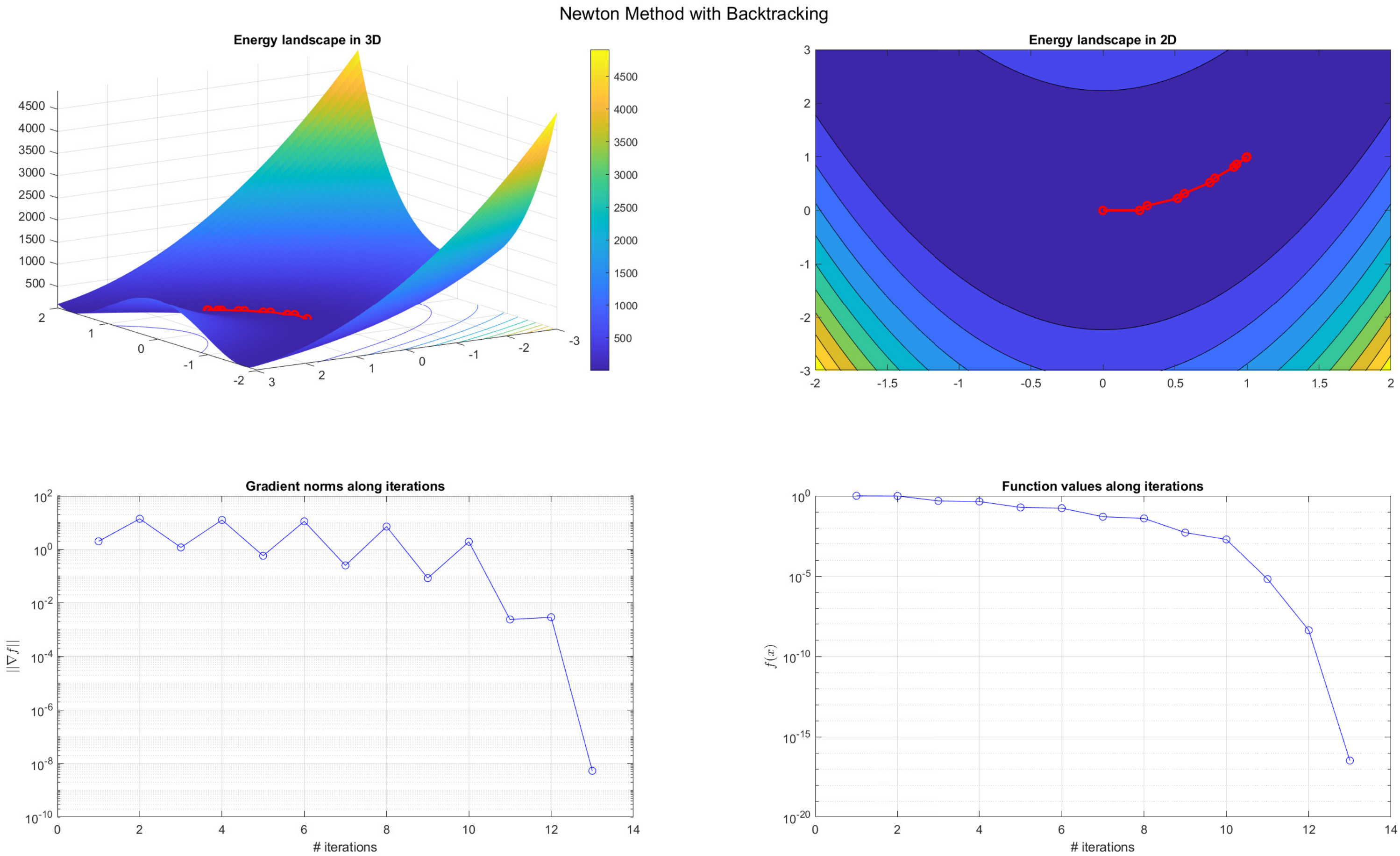
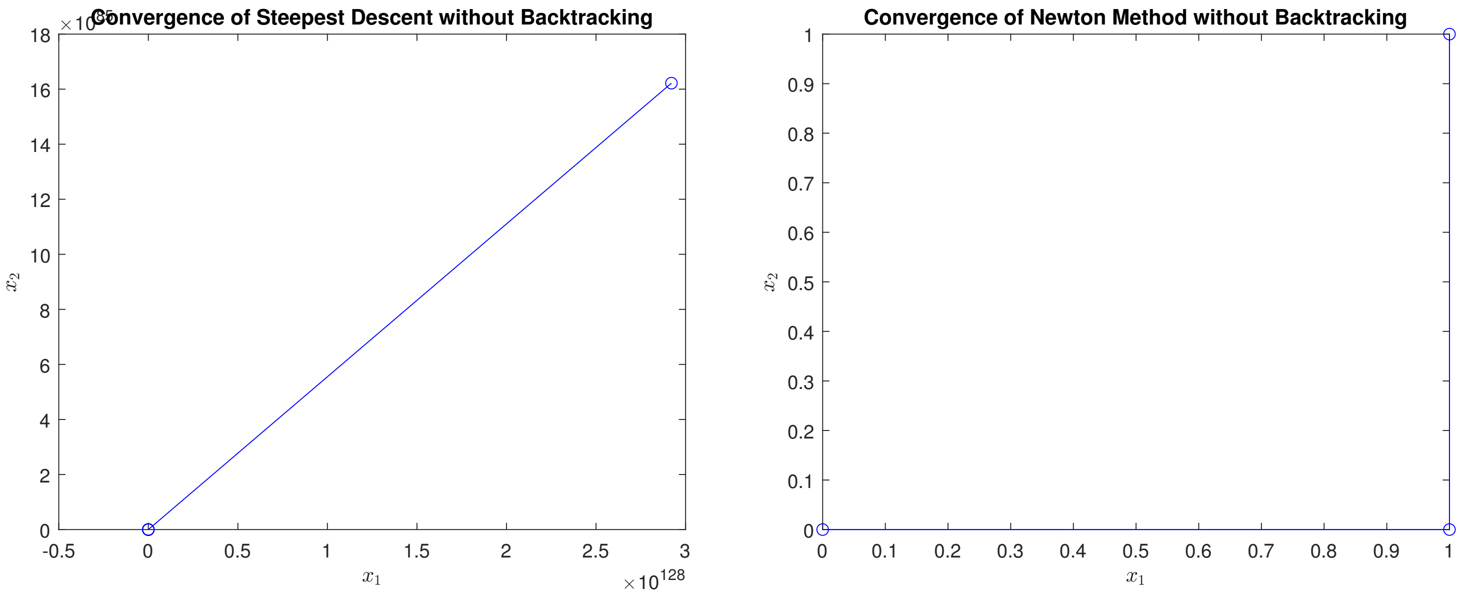
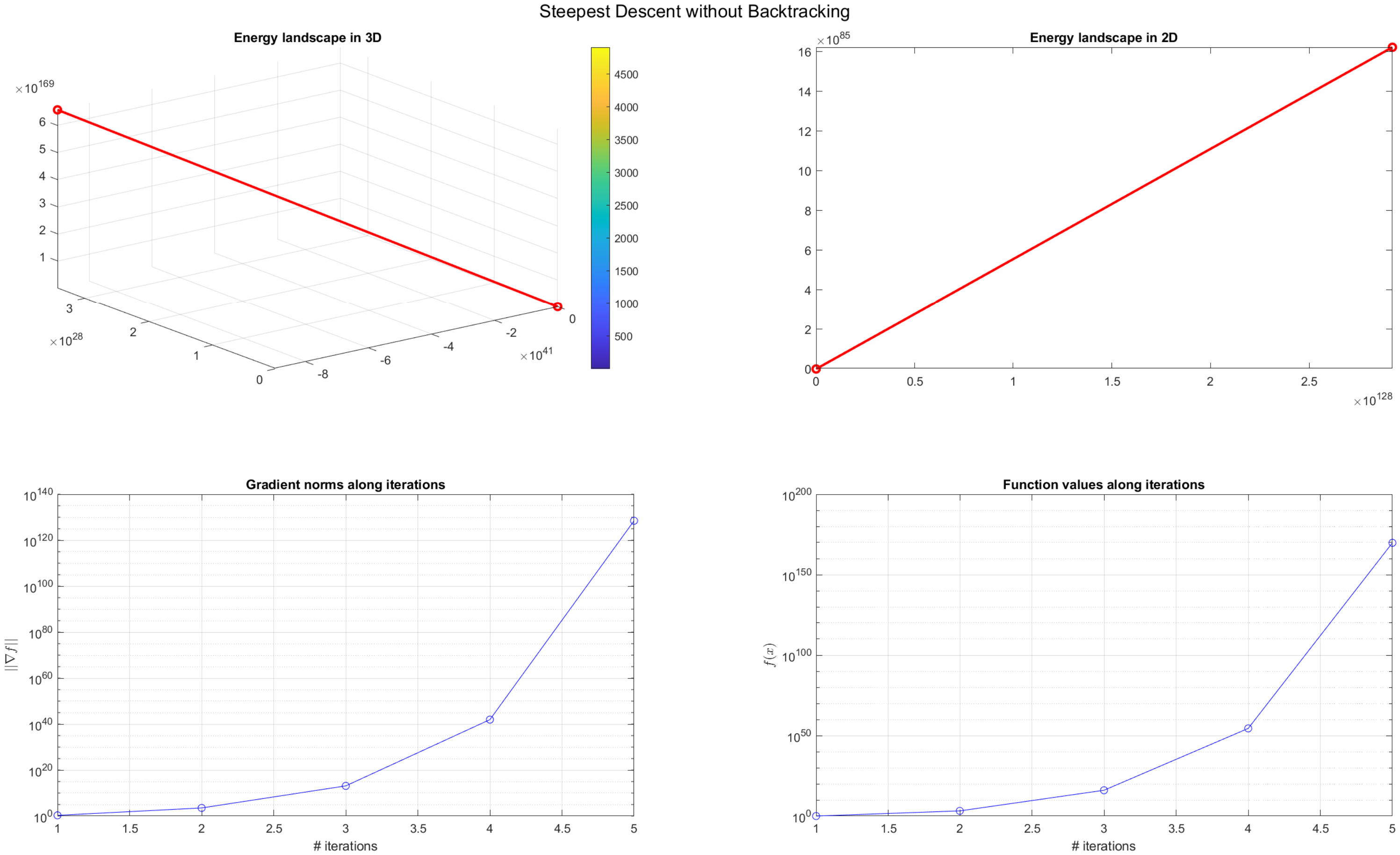


Figure 4: Convergence comparison between methods without Backtracking



Chosen value for parameter $c_1 = 1e-4$. Both methods are run with backtracking algorithm with step size $\beta = 1$. For the Newton's method without backtracking, when it reaches the local minimum

Figure 5: Visualization of Steepest Descent without Backtracking

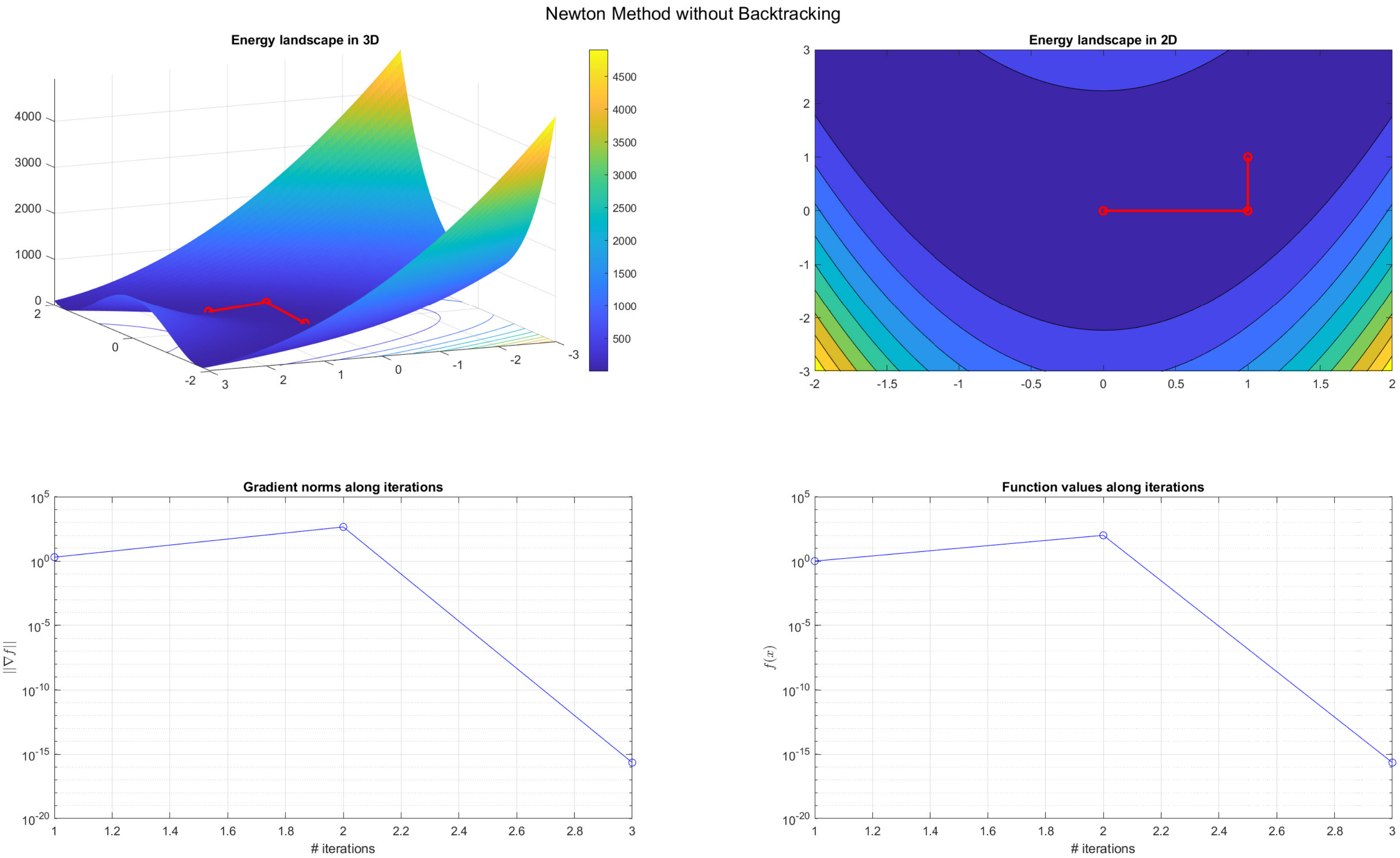


at $x^* = (1, 1)$, in order to plot it logarithmically the value $0 \mapsto EPS$ with $EPS = 2.2204e - 16$ in order to visualize symbolically the convergence to 0.

Convergence behavior: Newton method converges faster than Steepest Descent, regardless of the usage of the backtracking algorithm to compute the step length α. The Steepest Descent method converges after 21103 iterations with backtracking as in Figure 2 and it does not converge at all without backtracking as Figure 5 show: notice that the energy landscape is not shown due to the linear search going out of bounds of the rendered part of the Rosenbrock function. Meanwhile, the Newton method converges after 13 iterations with backtracking as in Figure 3 and in 3 iterations, so even less, without backtracking as in Figure 6. The gradient norm and the function value at each iteration are plotted for both methods with and without the Backtracking algorithm in the before-mentioned figures. Such behavior is explained because the Newton method uses the Hessian matrix to compute the step direction and will converge quadratically for all x in the neighborhood of the solution point $\mathcal{N}(x^*)$ such that $\nabla f(x^*)$ is positive definite, thus the Hessian matrix $H_f(x) = \nabla^2 f(x)$ is also positive definite.

Performances of methods: in addition, the convergence comparison between methods with and without backtracking plotting the input values x_1, x_2 of the function along iterations converging to the local minima is shown in Figure 1 for backtracking comparison, and Figure 4 for $\beta = 1$ fixed step length size. The steepest descent is a first-order method and its convergence is linear from Theorem 3.3. Its characterized by zigzagging behavior when searching for the local minima. On the

Figure 6: Visualization of Newton Method without Backtracking



other hand, Theorem 3.5 states that the Newton method is a second-order method, its convergence is quadratic and its sequence of gradient norms converges quadratically to 0. For higher dimension and orders of convergences, the Newton method is faster and suggested. Remember that the Newton method is not always the best choice, since it requires the computation of the Hessian matrix which is computationally expensive $O(n^3)$ because needs to compute the second derivative and solve a linear system. Hence, beware of Newton's method, specially for large scale problems and keep in mind that the Hessian matrix is not always positive definite, meaning it could not even yield a descent direction.

Exercise 2

1, 2

Implement the BFGS method (BFGS.m) with backtraking for the step size β. Test your implementation by minimizing the Rosenbrock's function. Use starting values $x_0 = (0, 0)$, $H_0 = I$, maximum number of iterations $N = 500$ and tolerance $TOL = 10^{-6}$.

Matlab scripts are provided in */code* folder. The main file to run is *BFGS.m*. It handles both computations and visualization of the Rosenbrock's function with the corresponding method.

3, 4

Plot the obtained iterates on the energy landscape in 2D. Analyze convergence behaviour of the methods by plotting the gradient norm and the function value at each iteration.

Figure 7: Visualization of BFGS

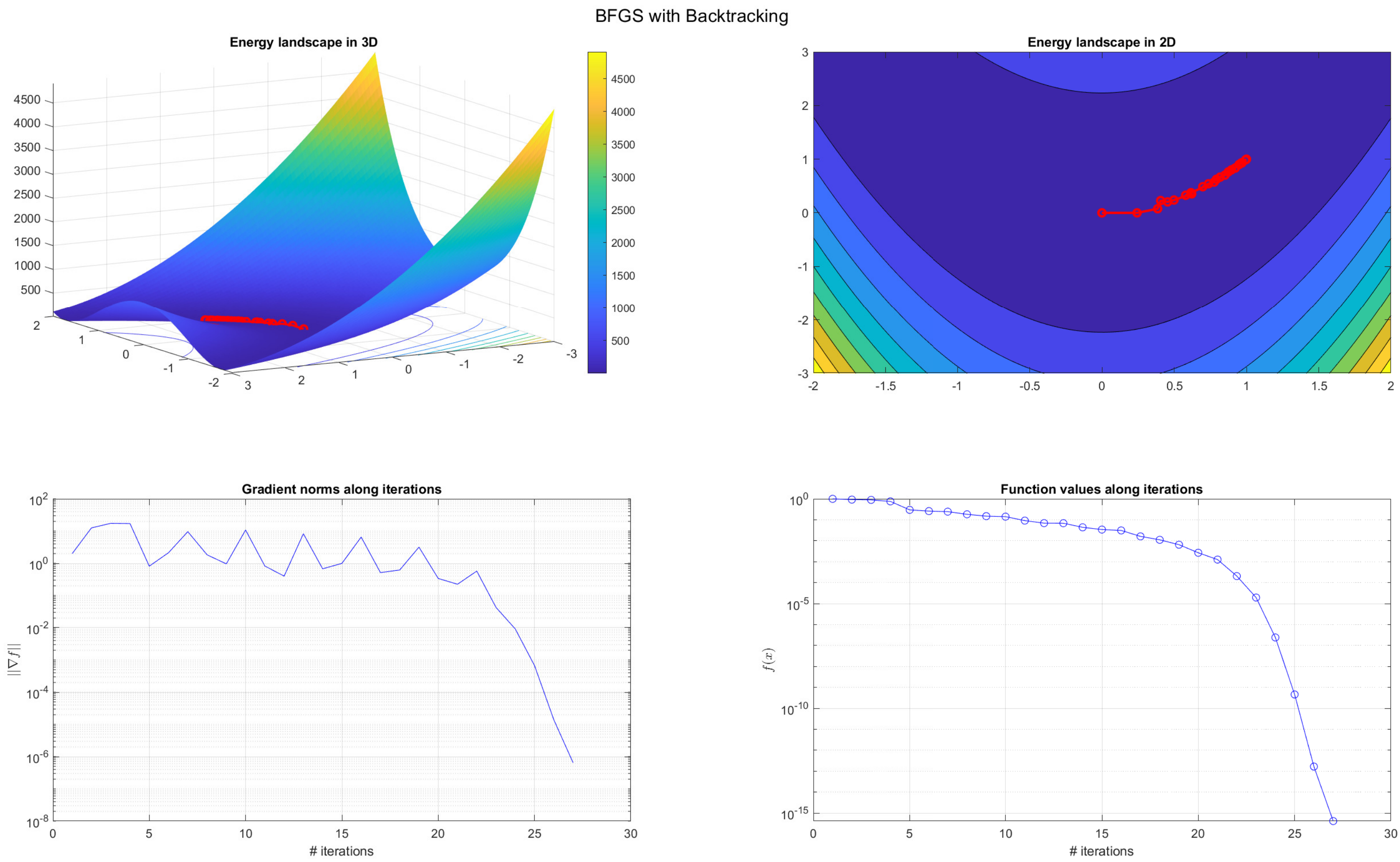
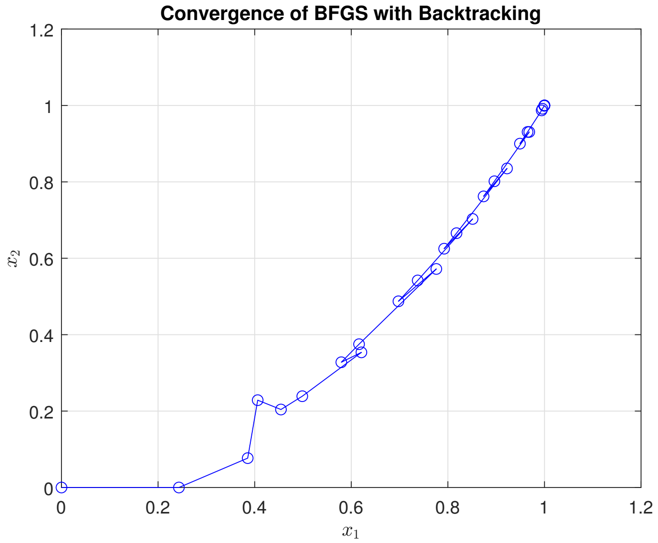


Figure 8: Convergence of BFGS with Backtracking



The plots regarding the energy landscape both 3D and 2D together with the convergence behavior of the BFGS method with respect to gradient norms and function values are visualized in Figure 7. The BFGS method convergence stop because it reaches a local minima approximation below the tolerance $TOL = 10^{-6}$ when computing the norm of the gradient $\nabla f(x^*)$.

5

Produce a table in which you compare the number of iterations required by BFGS, by Newton's method (with backtracking) and by Steepest descent method (with backtracking). You can use the results from the previous exercise. Comment the results by comparing the different methods.

Table 1: Comparison of Iterations for Rosenbrock's Function

Newton	BFGS	Steepest Descent
13	27	21103

In Table 1, it is shown the number of iterations required by the Steepest Descent, Newton and BFGS methods to minimize the Rosenbrock's function in ascending order of convergence. The iteration counter starts at 1, so not entering the loop is considered as 1 iteration. Order of convergence speed: Newton, BFGS, Steepest Descent. The reason relies on the order of the method dictating the convergence rate: steepest descent convergence rate is linear, Newton is quadratic and BFGS which is a quasi-Newton method with a superlinear convergence rate. The BFGS has such convergence rate because it approximates the inverse Hessian matrix H_k which combines the most recently observed information about the objective function with the existing knowledge embedded in the approximation H_{k-1}. It is suggested to use such quasi-newton method for large scale problems instead of the Newton method to avoid the need to compute second derivatives and solve linear systems.

Exercise 3

Let $f : \mathbb{R} \to \mathbb{R}$ be given by $f = \frac{1}{2}x^T Ax - b^T x$ with A symmetric positive definite. How many iterations does the SD method take to minimize the function f if we use the optimal step length? Please, prove your answer.

From Theorem 3.3, when the steepest descent method with exact line searches α_{opt}:

$$x_{k+1} = x_k - \underbrace{\left(\frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}\right)}_{\alpha_{opt}} \nabla f_k \qquad \text{where Q is SPD}$$

is applied to the strongly convex quadratic function of the form:

$$f(x) = \frac{1}{2}x^T Qx - b^T x \qquad \text{where Q is SPD}$$

as the requested function f from the exercise is, the error norm:

$$\frac{1}{2}||x - x^*||_Q^2 = f(x) - f(x^*)$$

satisfies:

$$||x_{k+1} - x^*||_Q^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 ||x_k - x^*||_Q^2$$
$$\text{with } 0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n,\ \lambda \in \Lambda = \mathit{spectrum}(Q)$$

The special case of the last formula see that convergence is achieved in one iteration if all eigenvalues are equal $\lambda_i = \lambda_j$ for $i \neq j$, $\forall \lambda \in \Lambda = \mathit{spectrum}(Q)$.

$$\underbrace{||x_{k+1} - x^*||_Q^2}_{||\cdot||^2 \geq 0} \leq 0$$

In this case, Q is a multiple of the identity matrix I, so the contours are circles and the steepest descent direction always points at the solution.