Università della Svizzera italiana | Institute of Computing CI

Optimization Methods **2024**

Student: Jeferson Morales Mariciano

Discussed with: Leonardo Birindelli

Assignment 1

Due date: Tuesday, 19 March 2024, 12:00 AM

1. Exercise 1

Vector-valued function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x_1, x_2) = 200(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{1}$$

1.

Compute gradient $\nabla f : \mathbb{R}^2 \to \mathbb{R}^2$ and Hessian $H_f : \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$.

$$\begin{aligned}
\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &\mapsto 200(x_2 - x_1^2)(x_2 - x_1^2) + (1 - x_1)(1 - x_1) \\
&= 200(x_2^2 - 2x_2x_1^2 + x_1^4) + (1 - 2x_1 + x_1^2) \\
&= 200x_2^2 - 400x_2x_1^2 + 200x_1^4 + 1 - 2x_1 + x_1^2 \\
\nabla f &= \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -800x_2x_1 + 800x_1^3 + 2x_1 - 2 \\ 400x_2 - 400x_1^2 \end{bmatrix} \\
H_f &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} -800x_2 + 2400x_1^2 + 2 & -800x_1 \\ -800x_1 & 400 \end{bmatrix}
\end{aligned}$$

2.

Write Taylor's expansion of f up to 2nd order around point $(x_1, x_2) = (0, 0)$.

$$\text{Given } x_0 = (0,0),\ f(x_0) = 1,\ \nabla f(x_0) = \begin{bmatrix} -2 \\ 0 \end{bmatrix},\ H_f(x_0) = \begin{bmatrix} 2 & 0 \\ 0 & 400 \end{bmatrix}, \text{ incognites } h = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{aligned}
f(x_0 + h) &= f(x_0) + h^T \nabla f(x_0) + \frac{1}{2} h^T H_f(x_0) h + o(||h||^2) \\
&= 1 + \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} -2 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 400 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + o(||h||^2) \\
&= 1 - 2x_1 + \frac{1}{2} \begin{bmatrix} 2x_1 & 400x_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + o(||h||^2) \\
&= 1 - 2x_1 + x_1^2 + 200x_2^2 + o(||h||^2) \approx f(x_1, x_2)
\end{aligned}$$

2. Exercise 2

Quadratic minimization problem with $A \in \mathbb{R}^{n\times n}$ is SPD $\wedge\, x, b \in \mathbb{R}^n$.

$$\min_{x\in\mathbb{R}^n} J(x) = \frac{1}{2}x^T Ax - b^T x \tag{2}$$

1.

Compute gradient and Hessian of J.

$$\nabla J(x) = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} - b = Ax - b$$

$$H_J(x) = A$$

2.

Write down the 1st Order Necessary Conditions.

If x^* is local minimizer $\wedge\, f$ continuously differentiable in open neighborhood $\mathcal{N}(x^*) \therefore \nabla f(x^*) = 0$. x^* stationary point $\because \nabla f(x^*) = 0$, from Th 2.2 any local minimizer is a stationary point.

$$\nabla f(x^*) = 0 \Rightarrow \nabla J(x^*) = Ax^* - b \Rightarrow Ax^* - b = 0 \Rightarrow Ax^* = b$$

3.

Write down the 2nd Order Necessary and Sufficient Conditions.

2nd Order Necessary Conditions: If x^* local minimizer of $f \wedge \exists \nabla^2 f$ continuous in open neighborhood $\mathcal{N}(x^*) \therefore \nabla f(x^*) = 0 \wedge \nabla^2 f(x^*)$ is positive semidefinite.

$$\nabla f(x^*) = 0 \Rightarrow \nabla J(x^*) = Ax^* - b \Rightarrow Ax^* = b$$

$$\nabla^2 f(x^*) \Rightarrow \nabla^2 J(x^*) = H_J = A \text{ positive semidefinite } \therefore \text{ eigenvals } \lambda \geq 0\ \forall \lambda \in \Lambda \wedge x^T Ax \geq 0$$

2nd Order Sufficient Conditions: Suppose $\nabla^2 f$ continuous in open neighborhood $\mathcal{N}(x^*) \wedge \nabla f(x^*) = 0 \wedge \nabla^2 f(x^*)$ positive definite $\therefore x^*$ strict local minimizer of f.

$$\nabla f(x^*) = 0 \Rightarrow \nabla J(x^*) = Ax^* - b \Rightarrow Ax^* = b$$

$$\nabla^2 f(x^*) \Rightarrow \nabla^2 J(x^*) = H_J = A \text{ positive definite } \therefore \text{ eigenvals } \lambda > 0\ \forall \lambda \in \Lambda \wedge x^T Ax > 0$$

3. Exercise 3

Function

$$f(x, y) = x^2 + \mu y^2 \tag{3}$$

1.

Write down its quadratic form.

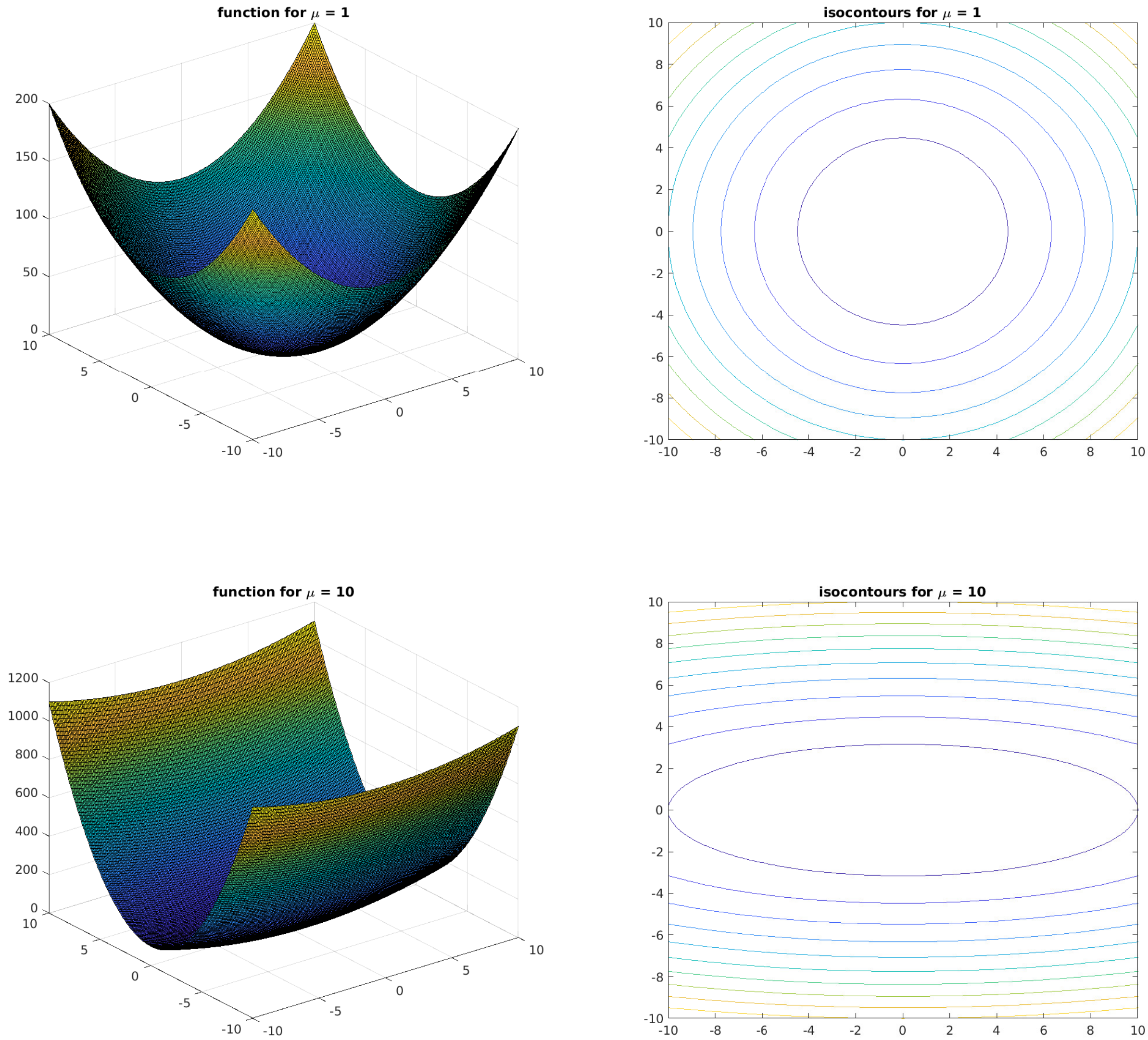
$$
\begin{aligned}
f(x,y) &= x^2 + \mu y^2 \\
&= \begin{bmatrix} x & \mu y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
&= \frac{1}{2} \vec{x}^T A \vec{x} - \vec{b}^T \vec{x}
\end{aligned}
$$

Quadratic Form $\frac{1}{2}\vec{x}^T A\vec{x} - \vec{b}^T\vec{x}$ with $\vec{x} = \begin{bmatrix} x \\ y \end{bmatrix}$, $A \in \mathbb{R}^{2\times 2}$, A is SPD, $\vec{b} \in \mathbb{R}^2$, $\vec{b} = \vec{0}$.

2.

Plot the surface of the functions and the corresponding contour plot for values $\mu = 1$ and $\mu = 10$. In both cases use the square $[-10, 10] \times [-10, 10]$. Comment on the behaviour of the isolines.

Figure 1: Plot of function with $\mu = 1, 10$



The matlab script is contained in *code/ex3_2.m* file.
The behavior of the isolines $\propto \mu$ parameter that affects the steepness of the function in the y-dimension. For $\mu = 1$, the isolines are circular, indicating that the function increases evenly in all directions from the origin. A symmetric paraboloid from the origin is the result of our isotropic quadratic function where its coefficients are equal, resulting in a radially symmetric growth. For $\mu = 10$, the isolines are ellipses because the function grows faster on the y-dimension resulting in a paraboloid steeper along the y-axis and flatter on the x-axis. This behavior is due to the scalar coefficient μ. Generally, the isolines indicates how the function behave in the space: closer contour lines mean that the function changes rapidly.

3.

Considering that A is a symmetric positive-definite matrix, find the exact optimal step-length α. Show your computations.

Recalling the formula to compute the step length in ideal case for quadratic forms:

$$\alpha_k = \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T A \nabla f_k}$$

Note that A is SPD $\therefore A = A^T$. $A = \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix}$. $\nabla f_k = Ax - b = \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$ with $b = \vec{0}$.
By plugging the previously obtained values into the definition:

$$\begin{aligned}
\alpha_k &= \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T A \nabla f_k} \\
&= \frac{\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}}{\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2\mu \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}} \\
&= \frac{4x^2 + 4\mu^2 y^2}{8x^2 + 8\mu^3 y^2} \\
&= \frac{1}{2} \cdot \frac{x^2 + \mu^2 y^2}{x^2 + \mu^3 y^2} \\
&= \alpha_{opt} \text{ for quadratic forms}
\end{aligned}$$

For $\mu = 1$:

$$\alpha_k = \frac{1}{2} \cdot \frac{x^2 + y^2}{x^2 + y^2} = \frac{1}{2}$$

For $\mu = 10$:

$$\alpha_k = \frac{1}{2} \cdot \frac{x^2 + 100y^2}{x^2 + 1000y^2}$$

4.

Write a Matlab code for the gradient method with maximum number of iterations $N = 100$ and a tolerance $tol = 10 - 8$. Minimize f for $\mu = (1, 10)$ and starting points: $(x_0, y_0) = (10, 0), (0, 10), (10, 10)$.

The matlab code is contained and extensively documented in the script *code/ex3_4.m*. All the generated plots for the next exercise 3.5 are done through such script.

5.

For each case plot the iterations on the energy landscape in 2D (the plot of the objective function), the log10 of the norm of the gradient and the value of the energy function (objective function) as functions of the iterations. Comment the results.

Figure 2: Case $\mu = 1, x_0 = (10, 0)$

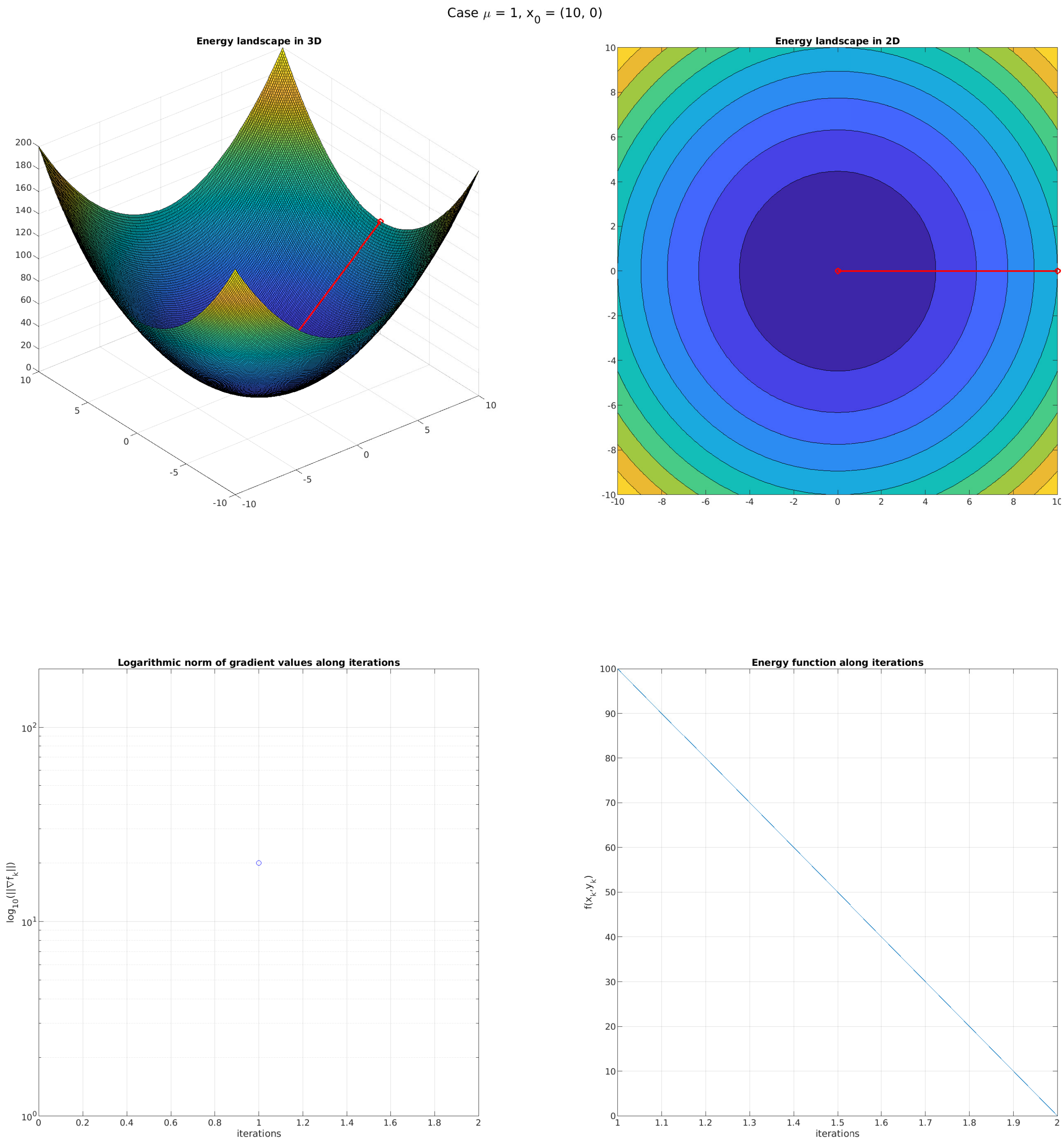


Figure 3: Case $\mu = 1, x_0 = (0, 10)$

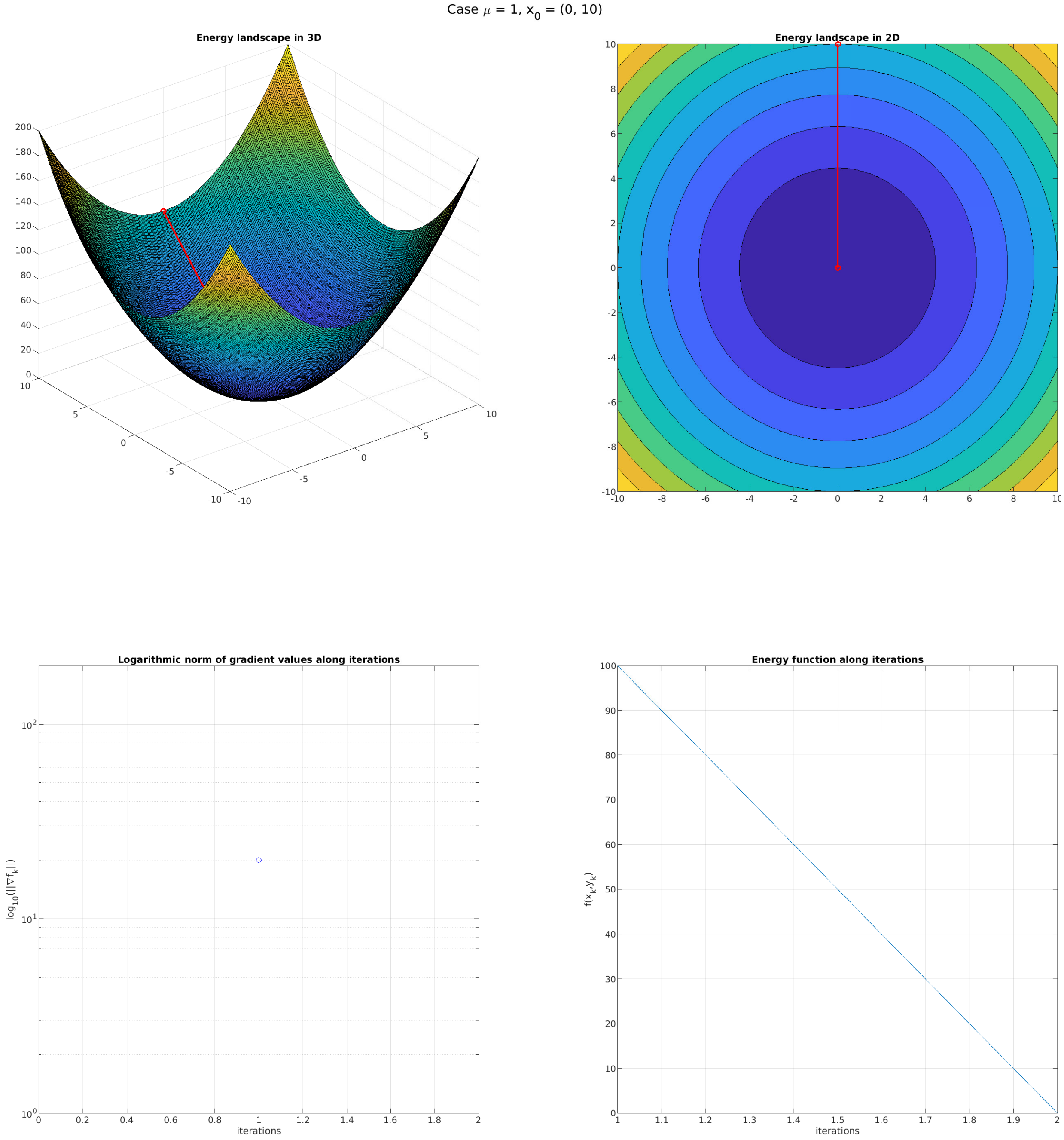


Figure 4: Case $\mu = 1, x_0 = (10, 10)$

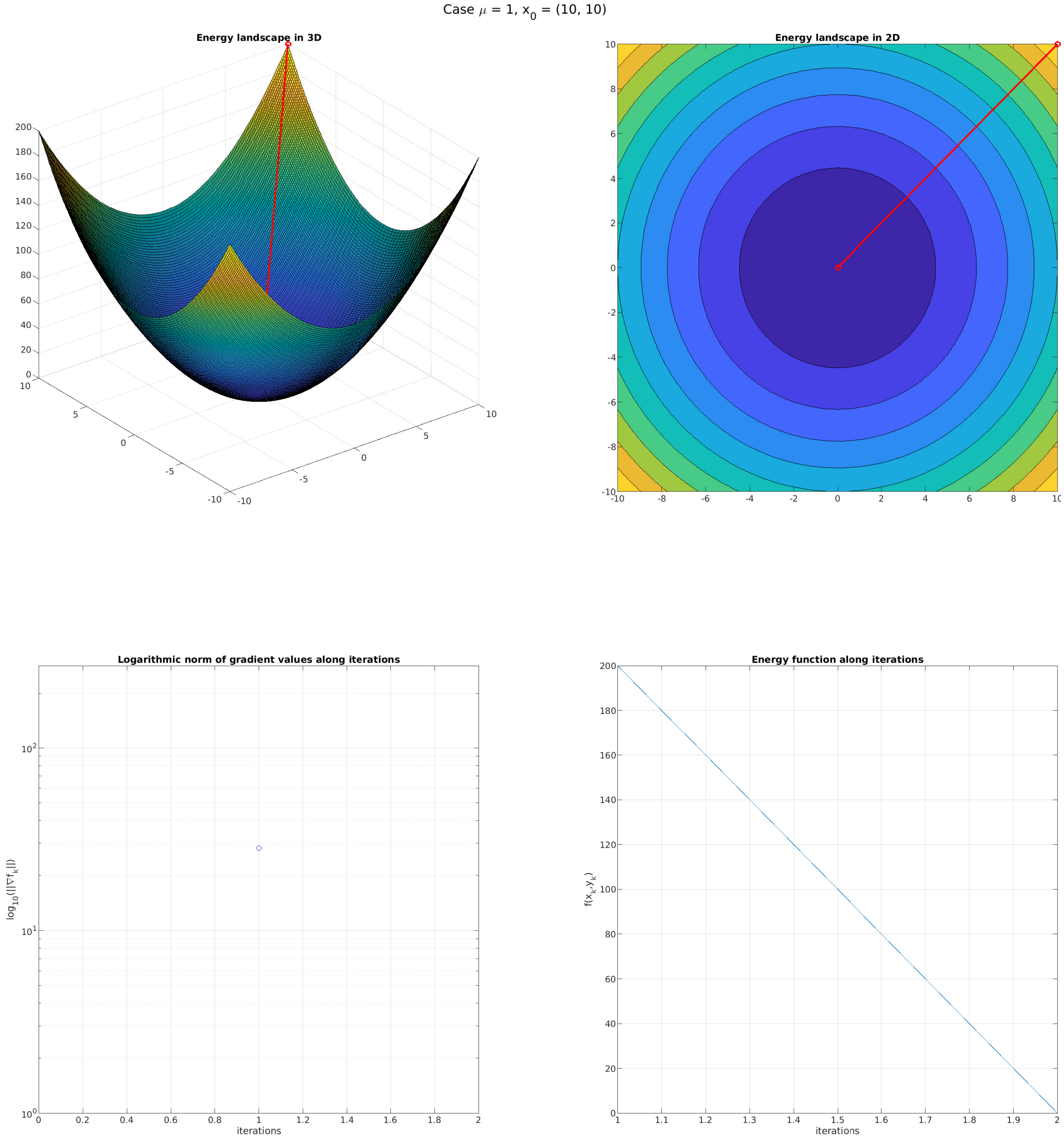


Figure 5: Case $\mu = 10, x_0 = (10, 0)$

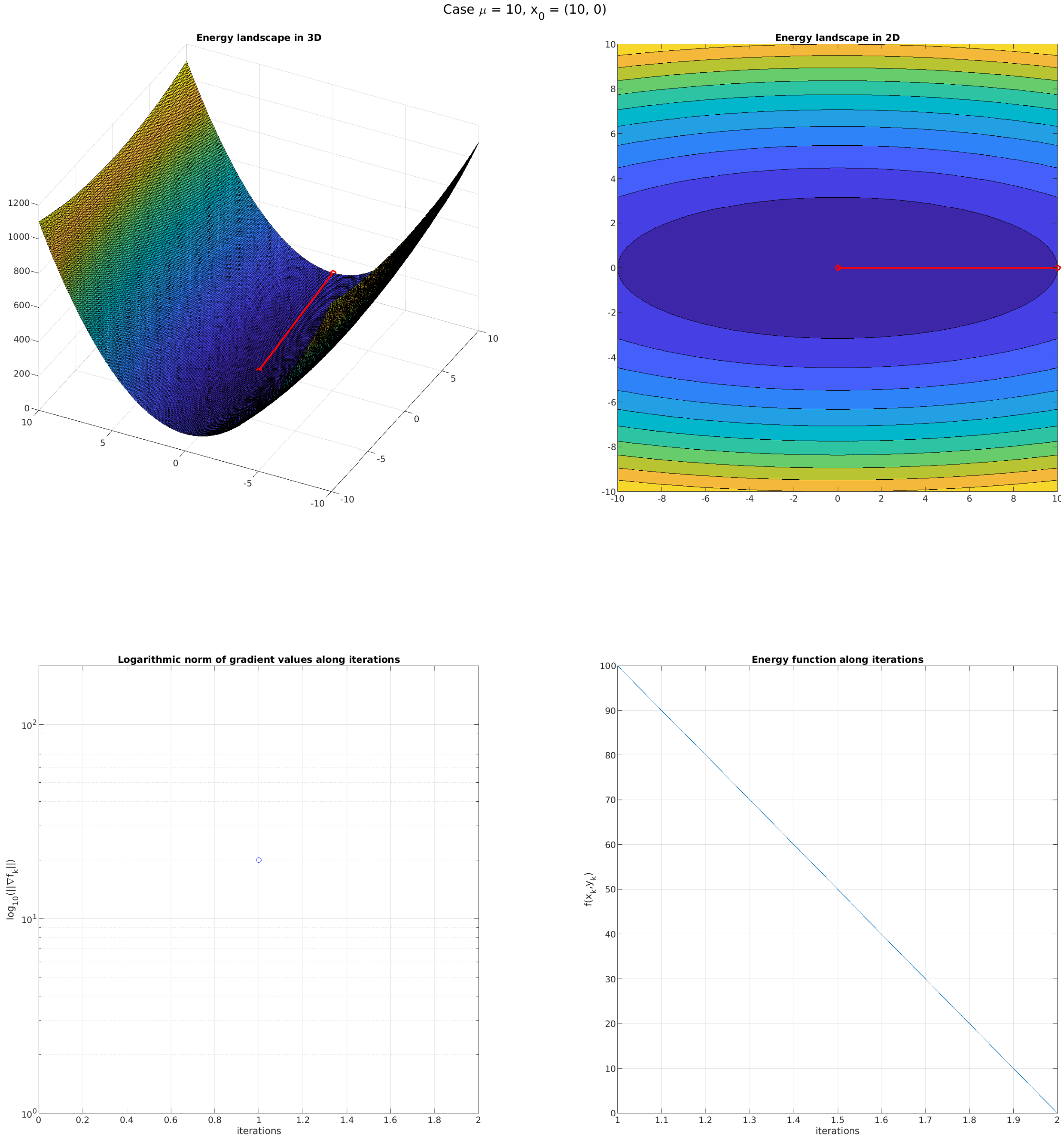


Figure 6: Case $\mu = 10, x_0 = (0, 10)$

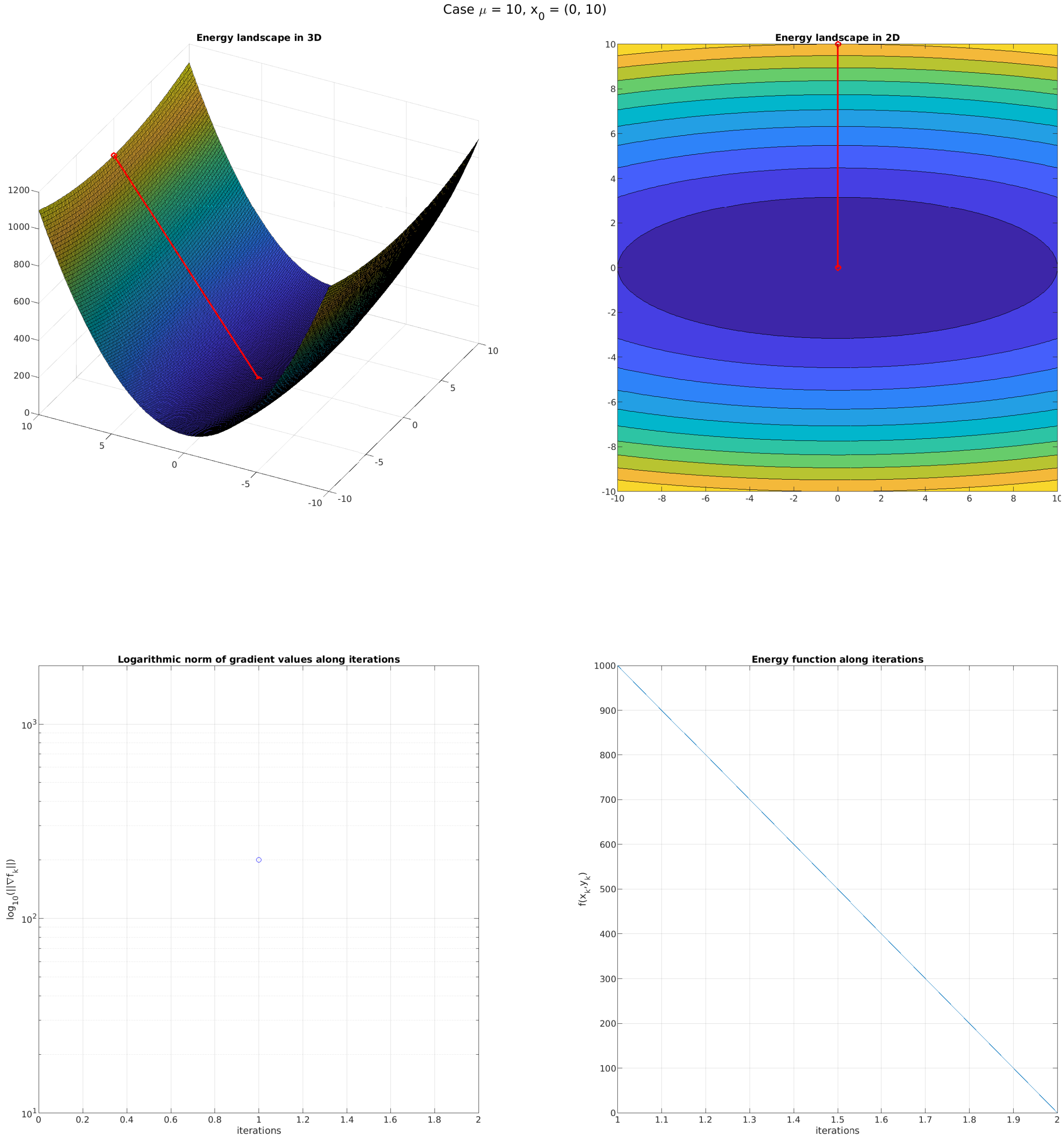
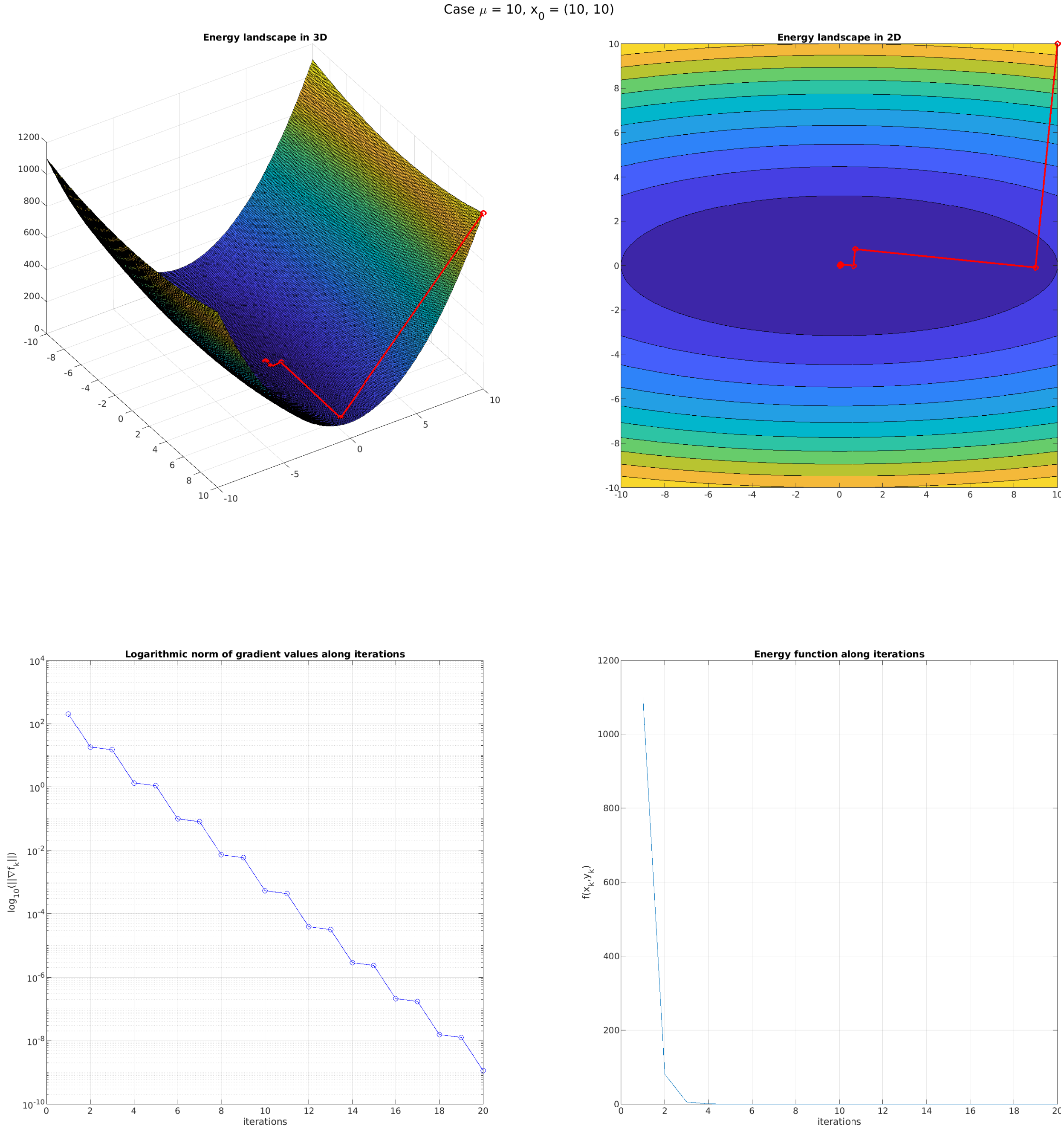


Figure 7: Case $\mu = 10, x_0 = (10, 0)$



μ values

The parameter μ can be either the value 1 or 10. As previously stated in exercise 3.2, the plots are clustered in 2 groups of 3 samples with respect to μ value, and whenever it is 1, it's value allows for a radially symmetric expansion of the function as a symmetric paraboloid from the origin which is the local minimizer x^*.

Convergence & Isocontours

Related to μ values, isocontours and the energy landscape clearly suggest the change in steepness. The convergence of the gradient method is reflected through the isolines and the initial point in every test case since they are important for the method performance. Moreover, from Th 3.3 notice that the eigenvalues $\lambda \in \Lambda = spectrum(A)$ for $\mu \neq 1$ are not equal $\lambda_i \neq \lambda_j$ for $i \neq j$, meaning that it cannot exploit one-iteration convergence.

Number of Iterations

For the last plot $\mu = 10$, $x_0 = (10, 0)$, the eigenvalues $\lambda \in \Lambda$ are not all equal and the convergence happens after zigzagging when searching for the local minima. This emphasize that the starting point x_0 heavily impacts the convergence rate of the steepest descent method, despite the function is writable in Quadratic Form and hence strongly convex.