Exploration of Function Coerciveness and Optimization Techniques

Kerem Zengin

July, 2023

Motivation

We explore the properties of the function $f(x_1, x_2) = x_1^4 - 4x_1x_2 + x_2^4$ especially its coerciveness. Coerciveness is an important function property such that guarantees the existence of a global minimizer. We graphically visualized the function, analyzed its global minimum, compared optimization algorithms like gradient descent and gradient descent with momentum.

Contents

1 Coercive Analysis and Visualization of Function

Definition. [1] A continuous function $f(x)$ that is defined on all of $\mathbb{R}^n$ is coercive if $f(x) \to +\infty$ as $||x|| \to \infty$ That is, for any constant $M > 0$ there exists a constant $R_M > 0$ such that $f(x) > M$ whenever $\|x\| > R_M$.

Now consider $f(x_1, x_2) = x_1^4 - 4x_1x_2 + x_2^4$.

$$x_1^4 - 4x_1x_2 + x_2^4 = x_1^4 + x_2^4(1 - \frac{4xy}{x^4 + y^4}).$$

As $(x, y) \to \infty$, $\frac{4xy}{x^4+y^4} \to 0$ and $x_1^4 + y^4 \to \infty$. So $f(x_1, x_2) = x_1^4 - 4x_1x_2 + x_2^4 \to \infty$. That shows that $f(x_1, x_2) = x_1^4 - 4x_1x_2 + x_2^4$ is coercive.

Theorem. Let $f(x)$ be a continuous function defined on all of $\mathbb{R}^n$. If $f(x)$ is coercive, then $f(x)$ has a global minimizer. Furthermore, if the first partial derivatives of $f(x)$ exist on all of $\mathbb{R}^n$, then any global minimizers of $f(x)$ can be found among the critical points of $f(x)$.

The gradient of the function is given by $\nabla f(x_1, x_2) = (4x_1^3 - 4x_2, -4x_1 + 4x_2^3)$. The critical points can be found by setting the gradient to zero. This yields the equations $x_2 = x_1^3$ and $x_1 = x_2^3$. Consequently, we have $x_1 = x_1^9$, which gives us the critical points $(0, 0)$, $(1, 1)$, and $(-1, -1)$. To determine the global minimizers, we evaluate the function at these critical points:

$$\begin{aligned} f(0, 0) &= 0, \\ f(1, 1) &= -2, \\ f(-1, -1) &= -2. \end{aligned}$$

Thus, the points $(1, 1)$ and $(-1, -1)$ are global minimizers of the function.

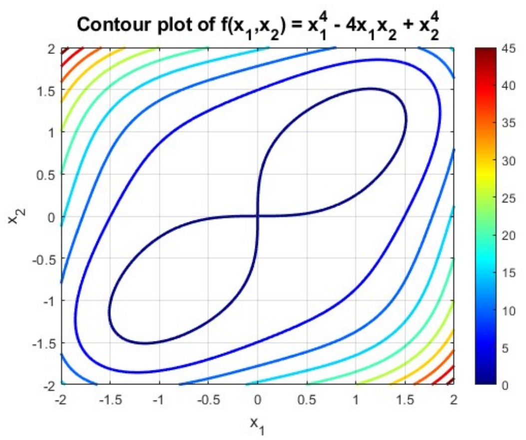

Figure 1: Contour Plot

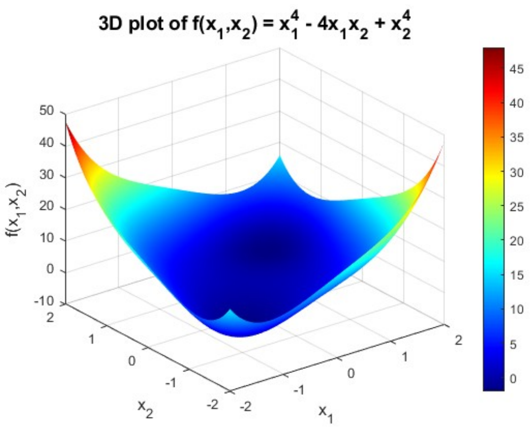

Figure 2: 3-D Plot

[1] Jim Lambers MAT419/519 SummerSession 2011-12

2 Gradient Descent *vs* Gradient Descent with Momentum

2.1 Gradient Descent for $f(x_1, x_2) = x_1^4 - 4x_1x_2 + x_2^4$

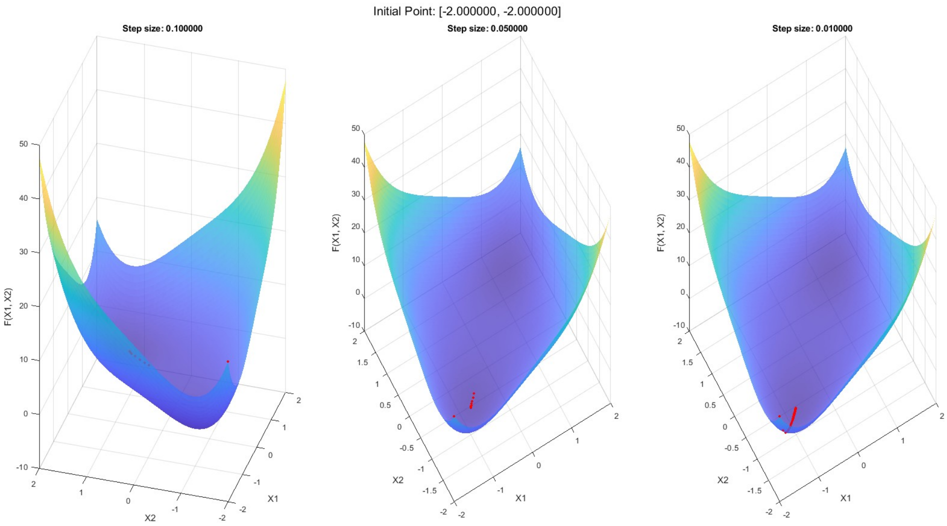

Figure 3: GD on inital point (-2,-2) with different stepsizes

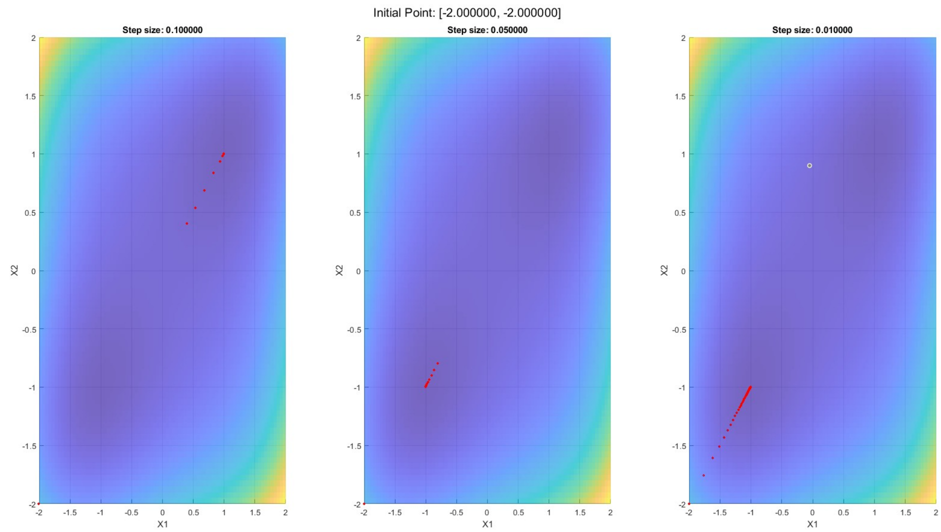

Figure 4: Iteration on xy plane.

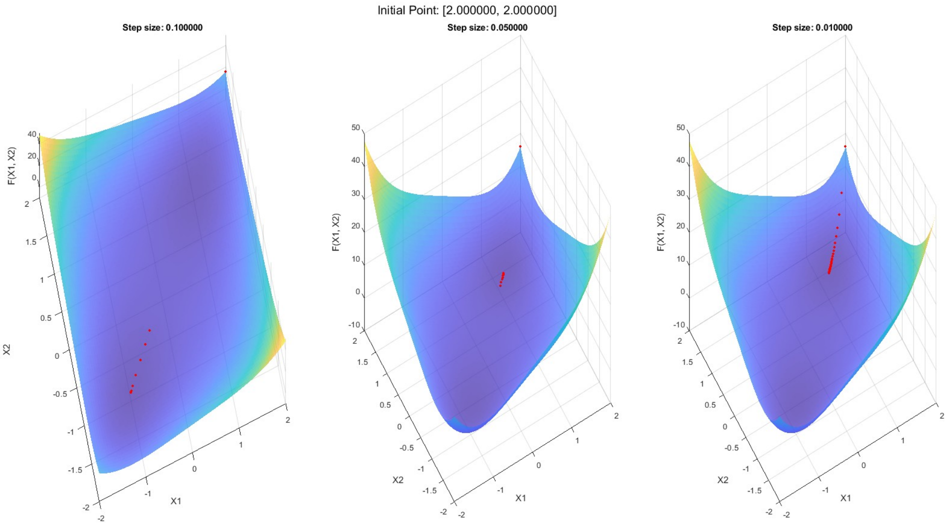

Figure 5: GD on inital point (2,2) with different stepsizes

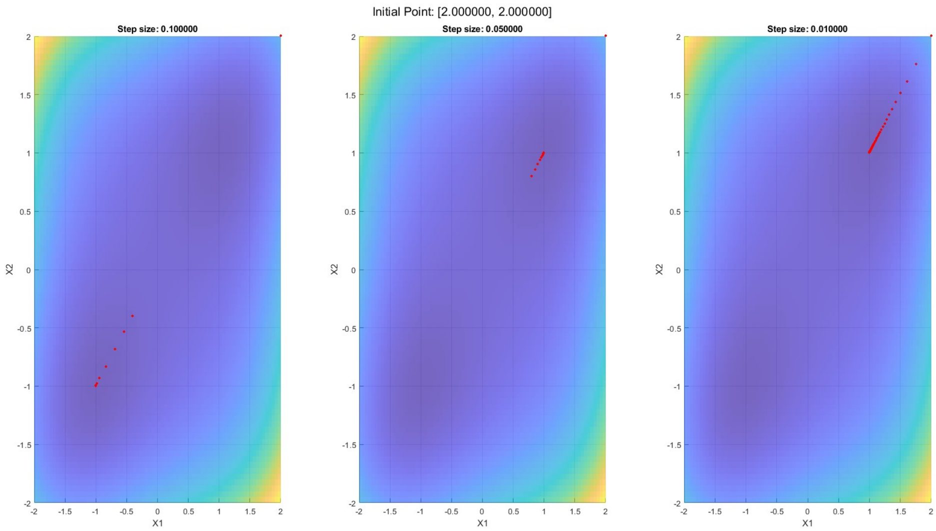

Figure 6: Iteration on xy plane.

- **Initial Points:** The algorithm was run starting from two different initial points, (2,2) and (-2,-2).
- **Effect of Step Size:** The step size seems to have a impact on the number of iterations needed for the algorithm to converge. As the step size decreases, the number of iterations increases. This is as expected because smaller steps mean the algorithm takes longer to reach the minimum.
- **Final Points:** For the same initial points, the algorithm seems to converge to the same final points regardless of the step size, but the final points are different depending on the initial points.

2.2 Gradient Descent with Momentum for $f(x_1, x_2) = x_1^4 - 4x_1x_2 + x_2^4$

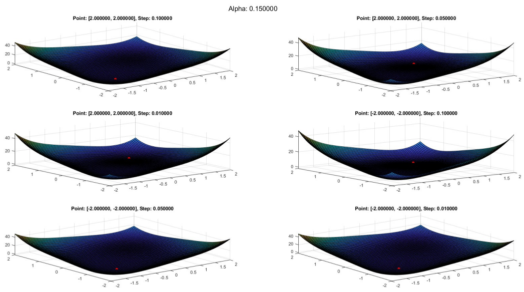

Figure 7: Momentum coefficient = 0.15

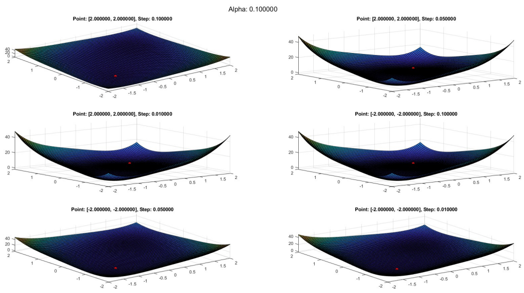

Figure 8: Momentum coefficient = 0.1

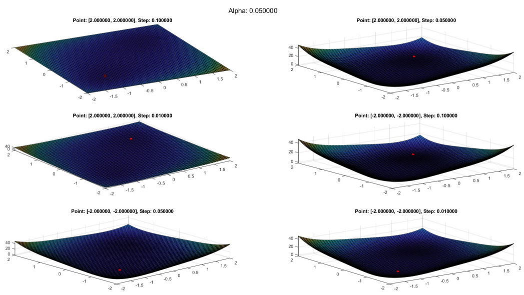

Figure 9: Momentum coefficient = 0.05

- **Initial Points:** The algorithm was tested from two different initial points, (2,2) and (-2,-2).
- **Effect of Step Size**: The step size still has a significant effect on the number of iterations required for convergence. As the step size decreases, the number of iterations increases.
- **Effect of Momentum Coefficient:** The alpha value, seems to influence the iteration count. As alpha increases, the number of iterations decreases. This suggests that adding momentum can help the algorithm converge faster, likely by helping it overcome local minima.
- **Final Points:** For the same initial points and the same alpha, the algorithm converges to the same final points regardless of the step size. However, the final points differ based on the initial points.

Initial_X1	Initial_X2	Step_Size	Alpha	Final_X1	Final_X2	Iterations
2	2	0.1	0.05	-1	-1	14
2	2	0.05	0.05	1	1	30
2	2	0.01	0.05	1	1	173
-2	-2	0.1	0.05	1	1	14
-2	-2	0.05	0.05	-1	-1	30
-2	-2	0.01	0.05	-1	-1	173
2	2	0.1	0.1	-1	-1	16
2	2	0.05	0.1	1	1	27
2	2	0.01	0.1	1	1	162
-2	-2	0.1	0.1	1	1	16
-2	-2	0.05	0.1	-1	-1	27
-2	-2	0.01	0.1	-1	-1	162
2	2	0.1	0.15	-1	-1	18
2	2	0.05	0.15	1	1	23
2	2	0.01	0.15	1	1	151
-2	-2	0.1	0.15	1	1	18
-2	-2	0.05	0.15	-1	-1	23
-2	-2	0.01	0.15	-1	-1	151

Table 1: GD_momentum results

Initial_X1	Initial_X2	Step_Size	Final_X1	Final_X2	Iterations
2	2	0.1	-1	-1	15
2	2	0.05	1	1	32
2	2	0.01	1	1	183
-2	-2	0.1	1	1	15
-2	-2	0.05	-1	-1	32
-2	-2	0.01	-1	-1	183

Table 2: GD results

- From the results in the tables, it can be seen that the number of iterations required to reach the minimum point is generally lower with gradient descent with momentum than gradient descent. This suggests that the momentum term helps to speed up the convergence of the algorithm, especially when the step size is small.
- For both methods, the starting point effects the final point to which the algorithm converges. It looks that the algorithm converges to either (-1,-1) or (1,1) depending on the starting point. This is okey with the fact that the function has multiple local minima.
- In both methods, a larger step size leads to faster convergence.
- In the GD with momentum method, a higher momentum term results in faster convergence. This is because the momentum term helps the algorithm to remember the previous direction of the gradient.