INSPER

Felipe Adeildo da Silva
Maria Eduarda Oliveira Galdino
Robson dos Santos França

Miniprojeto

Otimização pelo Vetor Gradiente

São Paulo
2025

1. Introdução

Este relatório detalha a implementação do método de otimização pelo vetor gradiente para encontrar pontos de máximo e mínimo locais de funções de duas variáveis. O princípio do método é que o vetor gradiente (∇f) aponta para a direção de maior crescimento da função, e seu oposto ($-\nabla f$) para a de maior decrescimento.

Utilizamos este método iterativo para analisar três funções distintas atribuídas ao nosso grupo (Grupo 6), implementando os algoritmos em Python.

Figura 1 - Funções atribuídas ao grupo 6

Grupo 6

Componentes: Maria Eduarda, Robson, Felipe Adeildo

Funções:

$$f(x,y) = x^2 + xy + 4y^2 + 3x + y$$

$$g(x,y) = \sqrt{x^2 + y^2 + 2} + 2x^2e^{-y^2} + (x-2)^2$$

$$h(x,y) = 4e^{-x^2-y^2} + 3e^{-x^2-y^2+4x+6y-13} - \frac{x^2}{4} - \frac{y^2}{6} + 2$$

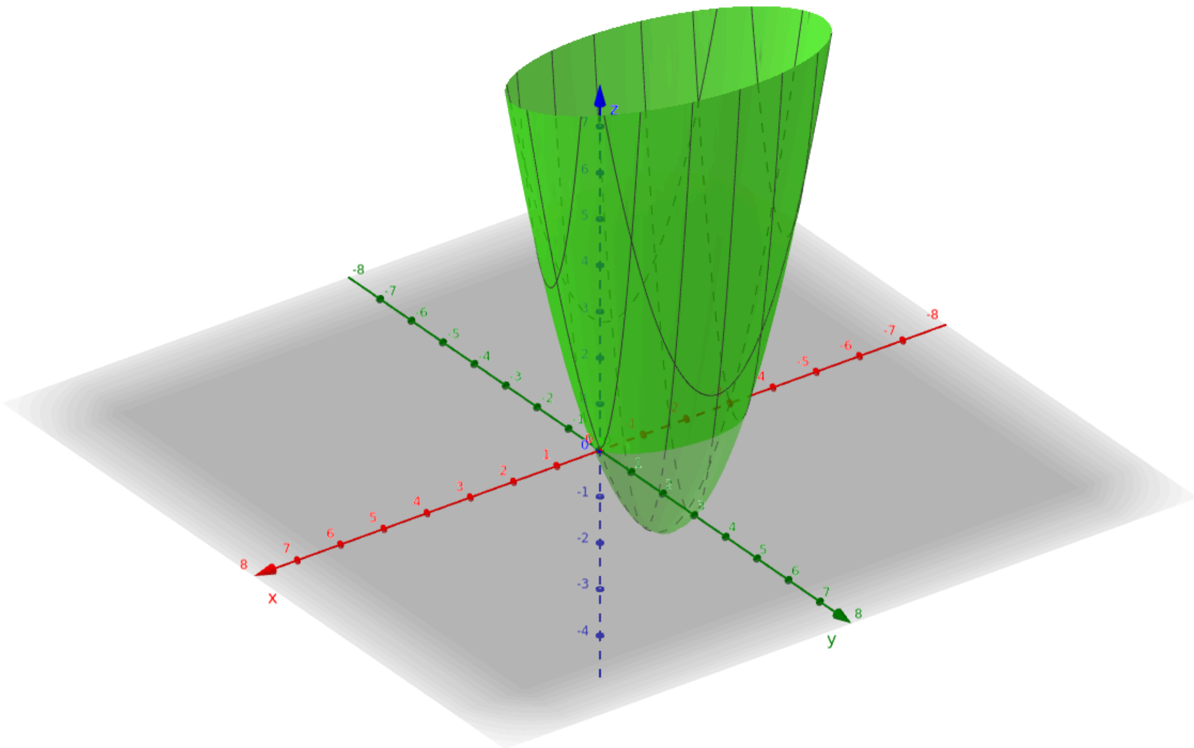
2. Tarefa 1

A primeira tarefa consiste em utilizar do conceito de vetor gradiente para encontrar os pontos críticos da função $f(x,\ y)$, construindo um código (algoritmo), que torne possível essa tarefa.

a)

Ao inserir essa função no GeoGebra Classic, observamos que a figura que surgir desta função [figura 2] possui somente um ponto crítico visível, e este é um ponto de mínimo localizado no negativo do eixo z.

Figura 2 - Função $f(x, y)$ no calculador gráfico GeoGebra



b)

Assim, nosso próximo passo é calcular o vetor gradiente,

- $\frac{\partial f}{\partial x} = 2x + y + 3$
- $\frac{\partial f}{\partial y} = x + 8y + 1$
- $\nabla f(x, y) = (2x + y + 3\,,\ x + 8y + 1)$

c e d)

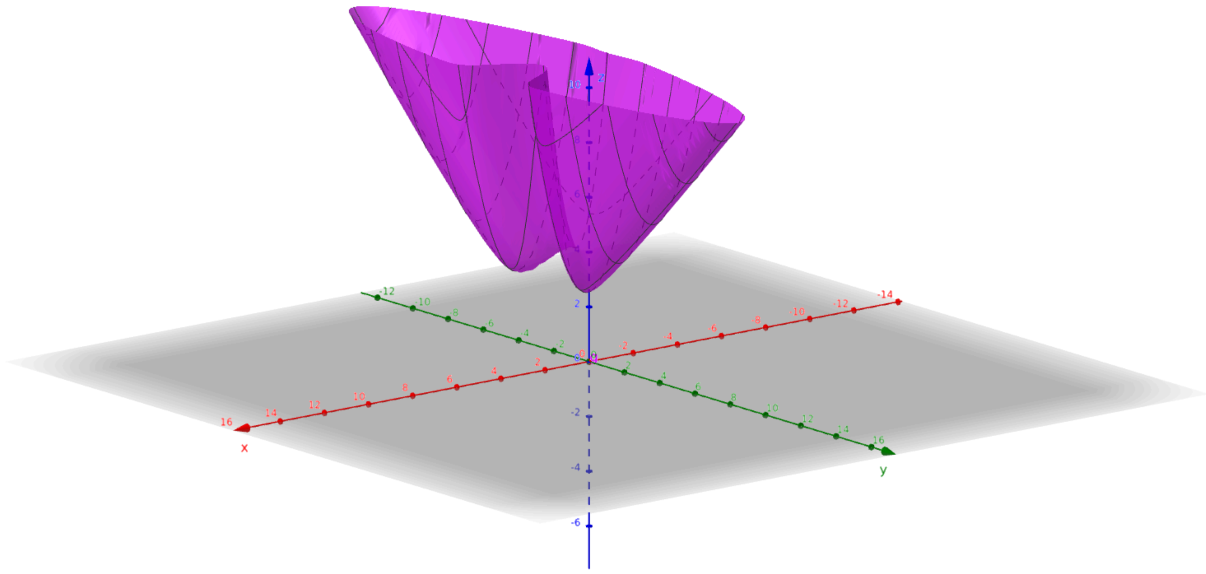
Tabela 1: Resultados da Tarefa 1 com Ponto Inicial (0, 0) e Precisão 0.00005[1]

Passo (α)	**Ponto de Mínimo (x)**	**Ponto de Mínimo (y)**	**Iterações**	**Status**
0.1	-1.532886	0.066594	41	Convergiu
0.15	-1.532982	0.066610	27	Convergiu
0.2	-1.533088	0.066596	20	Convergiu
0.3	nan	nan	atingiu o limite	Divergiu
0.5	nan	nan	atingiu o limite	Divergiu

3. **Tarefa 2**
   A segunda tarefa visa encontrar os dois pontos de mínimo da função $g(x, y)$.

[1] *Nota: "Divergiu" é a interpretação para o resultado* `nan` *com* `10001` *iterações, que é o* `max_iter` *do código.)*

Figura 3 - Função $g(x,\ y)$ no calculador gráfico GeoGebra



3.1 Cálculo do vetor gradiente de $g(x, y)$

- $\frac{\partial g}{\partial x} = \frac{x}{\sqrt{x^2+y^2+2}} + 4xe^{-y^2} + 2(x - 2)$

- $\frac{\partial g}{\partial y} = \frac{y}{\sqrt{x^2+y^2+2}} - 4x^2ye^{-y^2}$

- $\nabla g(x, y) = (\frac{x}{\sqrt{x^2+y^2+2}} + 2x^2e^{-y^2} + 2(x - 2),\ \frac{y}{\sqrt{x^2+y^2+2}} - 4x^2ye^{-y^2})$

3.2 determinação dos pontos de mínimo

O algoritmo de descida do gradiente converge para o mínimo local mais próximo do ponto de partida. Como a função $g(x,\ y)$ possui dois pontos de mínimo, foi necessário alterar o ponto inicial $(x_0,\ y_0)$ para encontrar o segundo mínimo.

Utilizando um passo $\alpha = 0.1$ e precisão de 10^{-5}, executamos o algoritmo duas vezes. O primeiro mínimo foi encontrado partindo da origem (0, 0). O segundo foi encontrado partindo de (2, 0), um ponto escolhido pela proximidade com o termo $(x - 2)^2$ da função. Os resultados estão compilados na Tabela 2.

Tabela 2: Pontos de Mínimo da Função g(x, y)

Ponto Inicial (xo,yo)	**Ponto de Mínimo (x)**	**Ponto de Mínimo (y)**	**Iterações**
(0.0, 0.0)	nan	nan	Não consegui u computar
(0.0, -2.0)	1.85868	-1.93288	37
(0.0, 2.0)	1.86252	1.93288	37

Para encontrar os dois pontos mínimos, foi preciso testar outros pontos. No nosso caso, conseguimos invertendo o sinal de y.

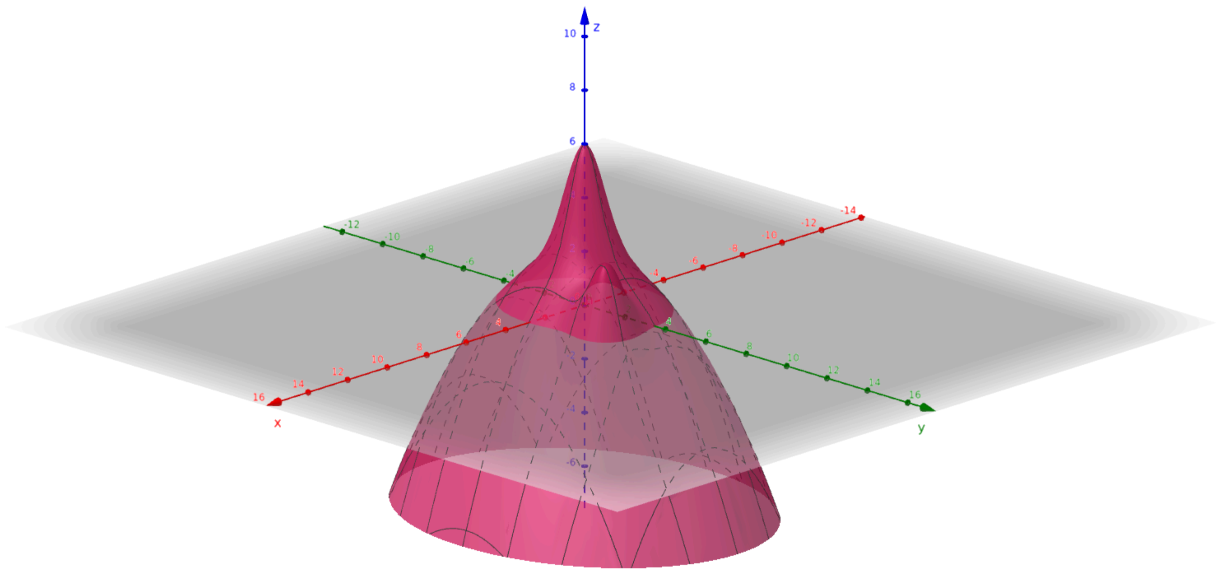
Ponto Inicial (xo,yo)	**Passo (α)**	**Ponto de Mínimo (x)**	**Ponto de Mínimo (y)**	**Iterações**	**Status**
(0.0, 2.0)	0.1	1.86252	1.93288	37	Convergiu
(0.0, 2.0)	0.15	1.85842	1.93279	25	Convergiu

(0.0, 2.0)	0.2	1.85826	1.93264	19	Convergiu
(0.0, 2.0)	0.3	1.85837	1.93240	13	Convergiu
(0.0, 2.0)	0.5	1.85847	1.93251	14	Convergiu
(0.0, -2.0)	0.1	1.85868	-1.93288	37	Convergiu
(0.0, -2.0)	0.15	1.85842	-1.93279	25	Convergiu
(0.0, -2.0)	0.2	1.85826	-1.93264	19	Convergiu
(0.0, -2.0)	0.3	1.85837	-1.93240	13	Convergiu
(0.0, -2.0)	0.5	1.85847	-1.93251	14	Convergiu

Mudando o número dos passos, o número de interações necessárias diminui, porém para passos maiores, como 1 (testado por fora), não é encontrado o ponto mínimo com a precisão desejada, divergindo.

4. Tarefa 3

Figura 4 - Função $h(x, y)$ no calculador gráfico GeoGebra



4.1 Vetor Gradiente de $h(x, y)$

- $\frac{\partial h}{\partial x} = -8xe^{x^2-y^2} + 3(-2x + 4)\,e^{-x^2-y^2+4x+6y-13} - \frac{x}{2}$

- $\frac{\partial h}{\partial y} = -8ye^{-x^2-y^2} + 3(-2y + 6)\,e^{-x^2-y^2+4x+6y-13} - \frac{y}{3}$

- $\nabla h(x, y) = (-8e^{x^2-y^2} + 3(-2x + 4)\,e^{-x^2-y^2+4x+6y-13} - \frac{x}{2},$
  $-8ye^{-x^2-y^2} + 3(-2y + 6)\,e^{-x^2-y^2+4x+6y-13} - \frac{y}{3})$

4.2 Determinação dos Pontos de Máximo

Para a Tarefa 3, o objetivo é encontrar os pontos de máximo. Para isso, o algoritmo de otimização foi modificado da "Descida do Gradiente" para a "Subida do Gradiente". Isso foi feito simplesmente somando o vetor gradiente ao ponto atual em cada iteração, em vez de subtrair:

x_new = x + learning_rate * df_dx

y_new = y + learning_rate * df_dy

Assim como na Tarefa 2, para encontrar os dois pontos de máximo, foi necessário alterar os pontos iniciais. A função h(x,y) é composta por dois "picos", um centrado próximo à origem (0, 0) e outro próximo a (2, 3). Usamos esses valores como nossas estimativas iniciais.

Tabela 3: Pontos de Máximo da Função h(x, y) ($\alpha = 0,1$, precisão de 10^{-5}

Ponto Inicial (xo,yo)	**Ponto de Máximo (x)**	**Ponto de Máximo (y)**	**Iterações**
(0.0, 0.0)	0	0	1
(4.0, 2.0)	1.83822	2.83364	23

5. Tarefa Desafio

A tarefa desafio propõe a implementação de um método de otimização com passo variável para calcular o ponto de mínimo da função f(x, y) da Tarefa 1. Foi implementado o algoritmo *Backtracking Line Search*, que ajusta o tamanho do passo “α” dinamicamente em cada iteração, evitando a divergência e otimizando a velocidade.

Para executar o cálculo, modificamos o main para chamar a nova função gradient_descent_variable, usando a função f e gradient da Tarefa 1:

```
x, y, i = gradient_descent_variable(f, gradient, x0, y0, precision)
```

O resultado obtido para o ponto de mínimo foi idêntico ao da Tarefa 1, porém alcançado em um número muito menor de iterações. A Tabela 4 compara o desempenho do passo variável com os resultados do passo fixo.

Tabela 4: Comparação de Iterações (Passo Fixo vs. Passo Variável)

Método	**Passo (α)**	**Iterações**

Passo Fixo	0.1	41
Passo Fixo	0.15	27
Passo Fixo	0.2	20
Passo Variável	Dinâmico (inicia em 1.0)	19

Análise da Comparação:

O método de passo variável (Backtracking) convergiu em apenas 19 iterações. Isso é significativamente mais rápido que o nosso melhor resultado de passo fixo (20 iterações com $\alpha = 0,2$).

Isso prova que o método de passo variável é superior: ele elimina a necessidade de adivinhar um valor de α e converge mais rapidamente por usar passos grandes quando está longe do mínimo e passos pequenos quando se aproxima dele, garantindo estabilidade e eficiência.