

Universidade Estadual Paulista Julio de Mesquita Filho - UNESP
Instituto de Biociências
Departamento de Biodiversidade e Bioestatística
Programa de Pós-Graduação em Biometria
Otimização Linear & Não Linear

Trabalho Computacional I

Aluno: Richard Castro Júnior

Professora: Daniela Renata Cantane

Botucatu-SP
2024

1 Introdução

Os modelos matemáticos se fundamentam em observações de fenômenos naturais, sociais e outros. Eles têm a finalidade de descrever as leis que governam processos ou sistemas nas áreas física, química, biológica, entre outras, utilizando formulações matemáticas. Esses modelos representam versões simplificadas ou abstratas de situações reais. Normalmente, são desenvolvidos considerando a definição de variáveis e relações matemáticas que delineiam o comportamento relevante ao problema em análise. A validação desses modelos depende da consistência das soluções geradas em relação ao contexto original (Arenales et al., 2011).

Nesse cenário, a Pesquisa Operacional (PO), também conhecida como Otimização, assume um papel importante ao tratar de diversos aspectos e situações relacionadas a problemas complexos. Ela permite a formulação de decisões mais eficazes e a construção de sistemas mais produtivos. Esta disciplina aplica técnicas de modelagem matemática e algoritmos computacionais avançados para estudar, desenvolver e implementar métodos analíticos, fornecendo suporte à tomada de decisões mais informadas em várias áreas de atuação. De modo geral, a análise de problemas de otimização segue a seguinte estrutura:

$$\begin{aligned} &\text{Minimizar (ou maximizar) } f(\mathbf{x}) \\ &\text{Sujeito a } \mathbf{x} \in \Omega \end{aligned}$$

em que $f(\mathbf{x})$ é a função objetivo; \mathbf{x} representa as variáveis de decisão; e Ω é o conjunto viável, que contém as restrições do problema (Arenales et al., 2011).

A minimização de uma função contínua de n variáveis, sem restrições, é um dos problemas clássicos da otimização não linear. Há diversas situações do mundo real que podem ser modeladas dessa forma. Quando a função é diferenciável, a condição necessária de primeira ordem para identificar minimizadores é que o gradiente deve ser igual a zero. À medida que o número de variáveis ou a complexidade da função aumenta, métodos isolados tornam-se insuficientes para encontrar até mesmo pontos estacionários. Assim, é necessário recorrer a métodos numéricos, que geralmente são iterativos. Trata-se de algoritmos que operam com um dado ponto x_k , determinando-se uma direção d_k ao longo da qual é possível, em princípio, reduzir o valor da função objetivo. Em seguida, calcula-se um tamanho de passo que permita uma diminuição adequada (Martínez e Santos, 1995).

Consideramos que $\|\cdot\|$ representa a norma euclidiana, embora muitos resultados sejam inde-

pendentes dessa escolha. A aproximação x_{k+1} satisfaz $f(x_{k+1}) < f(x_k)$ se $\nabla f(x_k) \neq 0$. Para a definição desses algoritmos, as direções executadas reduzem o valor de $f(x)$. Assim, dado $x \in \mathbb{R}^n$ e dizemos que $d \in \mathbb{R}^n$ é uma direção de descida a partir de x se existe $\epsilon > 0$ tal que, para todo $t \in (0, \epsilon]$, $f(x + td) < f(x)$ (Martínez e Santos, 1995; Nocedal e Wright, 1999; Shor, 1979, Ribeiro e Karas, 2013).

A direção $d = -\nabla f(x)$ é chamada de direção de máxima descida a partir de x . Se considerarmos todas as direções com norma euclidiana unitária no espaço, é fácil perceber que a derivada direcional mais negativa ocorre nessa direção. A solução do problema

Minimizar $\bar{f}(x)$ sujeita a $\|x - \bar{x}\| \leq \epsilon$, em que \bar{f} é qualquer função tal que $\nabla \bar{f}(\bar{x}) = \nabla f(\bar{x})$, é um ponto $x(\epsilon)$ tal que

$\frac{x(\epsilon) - \bar{x}}{\|x(\epsilon) - \bar{x}\|}$ tende ao vetor unitário na direção de máxima descida quando ϵ tende a 0 (Martínez e Santos,

Os métodos que seguem o algoritmo básico utilizam as direções de descida dado $x_k \in \mathbb{R}^n$ tal que $\nabla f(x_k) \neq 0$. Escolhe-se uma direção de descida d_k e um $t_k > 0$ tais que $f(x_k + t_k d_k) < f(x_k)$. Em seguida, defina $x_{k+1} = x_k + t_k d_k$.

Naturalmente, espera-se que a aplicação do algoritmo resultasse, após algumas iterações, a um minimizador global de f . No entanto, isso não será possível, pois o algoritmo é incapaz até mesmo de nos conduzir a pontos estacionários no limite. Uma das razões desta falha em encontrar minimizadores ou até mesmo pontos estacionários é que o requisito de que $f(x_k + t_k d_k)$ seja menor que $f(x_k)$ é um objetivo excessivamente modesto. Na realidade, um descenso mais eficaz pode ser alcançado ao longo de direções de descida (Martínez e Santos, 1995).

Para contornar isso, os métodos de busca, como método de Seção Áurea e Condição de Armijo, substituem o descenso simples e são úteis para refutar alguns dos contra-exemplos que podem ser elaborados para contestar a condição de descenso simples. Esses métodos de busca são projetados para otimizar o processo de minimização, permitindo que o algoritmo explore mais efetivamente o espaço de soluções. O método da Seção Áurea, por exemplo, utiliza a proporção áurea para dividir o intervalo de busca, garantindo que cada iteração se concentre nas regiões mais promissoras, enquanto a Condição de Armijo ajusta o tamanho do passo com base na taxa de redução observada na função objetivo, promovendo um avanço mais eficiente em direção ao mínimo.

2 Objetivos

Desenvolver e implementar métodos de otimização irrestrita, incluindo o método do gradiente, método de Newton, método de gradientes conjugados e método quasi-Newton. Comparar o desempenho dos métodos implementados em termos de convergência e eficiência, utilizando funções não lineares e quadráticas. Avaliar as estratégias de busca método da seção áurea e busca de Armijo, para determinar o impacto na convergência dos métodos de otimização.

3 Procedimento

Os métodos de otimização irrestrita (método do gradiente, método de Newton, método de gradientes conjugados e método de quasi-Newton) foram implementados utilizando o MATLAB®. Trata-se de uma linguagem de programação que expressa diretamente matrizes e arranjos matemáticos, combinando um ambiente de desktop ajustado para análise iterativa e processos de design com uma linguagem de programação.

Foram implementados também os métodos de busca de seção áurea e busca de Armijo. Com isso, os códigos foram executados utilizando 4 funções distintas (não lineares/quadráticas) e os métodos foram comparados, incluindo uma análise da convergência, contendo tabelas e gráficos. Para cada um das funções (função de raiz quadrada suavizada, função logarítmica, função seno e função quadrática) foi adicionada a matriz *bfw62b*, extraída do *site* <https://math.nist.gov/MatrixMarket/> e implementando a partir da função *mmread*.

Para comparar a taxa de convergência dos métodos de otimização, foi analisado como cada método se aproxima do ponto ótimo nas quatro funções objetivo escolhidas. O procedimento começa implementando cada método e fixando tolerância $\epsilon = 10^{-6}$ da convergência. A métrica principal é o número de iterações necessárias para atingir a tolerância especificada no gradiente, fornecendo uma medida objetiva da eficiência. Durante a execução, a norma do gradiente será monitorada ao longo das iterações como uma indicação da qualidade da convergência. Além disso, gráficos foram gerados para mostrar a redução do valor da função objetivo em função das iterações, ilustrando visualmente a velocidade de convergência.

O desempenho dos métodos de busca foi avaliado em combinação com os algoritmos de otimização. Para cada algoritmo, ambas as estratégias de busca foram aplicadas e suas eficiências foram comparadas com base em métricas chave: tempo total de execução (em segundos), valor final al-

cançado da função objetivo e norma do gradiente na solução. Além disso, o número de iterações foi contabilizado para determinar como a escolha do método de busca influencia a convergência.

Por fim, foi avaliada a sensibilidade de cada método às características específicas das funções objetivo escolhidas. Comparações são apresentadas para identificar quais métodos são mais robustos para cada tipo de função, destacando suas vantagens e limitações.

3.1 Métodos de otimização irrestrita

3.1.1 Método do gradiente

O Método do Gradiente é uma técnica de otimização iterativa amplamente utilizada para encontrar o mínimo (ou máximo) de uma função diferenciável em problemas de otimização irrestrita. Ele se baseia no fato de que a direção negativa do gradiente de uma função ($-\nabla f(x)$) indica o sentido de maior declínio no valor da função em um ponto específico.

O processo de busca começa em um ponto arbitrário x_0 e caminha sobre a linha determinada pelo gradiente, até estar próxima suficientemente da solução. A cada passo do processo iterativo, o valor de α pode ser alterado para permitir uma convergência mais rápida, certificando-se que a função converge α cada iteração; caso contrário, o valor de α deve ser reduzido para que o processo se estabilize. Segue-se abaixo o algoritmo deste método:

1. Entrada:

- Função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$
- Ponto inicial $x_0 \in \mathbb{R}^n$
- Taxa de aprendizagem $\alpha > 0$
- Número máximo de iterações N

2. Inicialização: Defina $k = 0$

3. Enquanto $k < N$ fazer:

- (a) Calcule o gradiente $\nabla f(x_k)$
- (b) Atualize o ponto: $x_{k+1} = x_k - \alpha \nabla f(x_k)$
- (c) Se $\|\nabla f(x_{k+1})\| < \epsilon$ (critério de parada), **pare**

4. Defina $k = k + 1$

Saída: Aproximação do minimizador x_k

3.1.2 Método de Newton

O método de Newton é um procedimento iterativo utilizado para encontrar as raízes de uma função diferenciável f . Essas raízes são as soluções da equação $f(x) = 0$. O algoritmo baseia-se na ideia de linearizar a função em torno de um ponto inicial e, em seguida, usar essa linearização para encontrar uma aproximação melhorada da raiz.

Além de encontrar raízes de funções, o método de Newton pode ser aplicado à derivada f' de uma função f que é duas vezes diferenciável. Dessa forma, é possível encontrar as raízes da derivada, que correspondem às soluções da equação $f'(x) = 0$. O método de Newton, ao utilizar informações da primeira e da segunda derivada, oferece uma convergência quadrática, o que significa que a precisão da solução melhora significativamente a cada iteração, desde que o ponto inicial esteja suficientemente próximo da raiz verdadeira.

Dessa forma, o método de Newton não apenas fornece uma maneira eficaz de localizar raízes, mas também permite localizar extremos de funções em diversas aplicações práticas. A seguir, é apresentado o algoritmo deste método:

1. **Entrada:**

- Função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$
- Ponto inicial $x_0 \in \mathbb{R}^n$
- Número máximo de iterações N
- Tolerância $\epsilon > 0$

2. **Inicialização:** Defina $k = 0$

3. **Enquanto** $k < N$ **fazer:**

- Calcule o gradiente $\nabla f(x_k)$
- Calcule a Hessiana $H(x_k)$
- Verifique se $\|\nabla f(x_k)\| < \epsilon$ (critério de parada)
 - Se verdadeiro, **pare**
- Resolva o sistema linear: $H(x_k)d_k = -\nabla f(x_k)$

(e) Atualize o ponto:

$$x_{k+1} = x_k + d_k$$

4. Defina $k = k + 1$

Saída: Aproximação do minimizador x_k

3.1.3 Método de gradientes conjugados

Este é um método iterativo que se baseia na mesma ideia do Método dos Gradientes, que é minimizar uma função quadrática. No entanto, notaremos que existe um limite para o número de iterações necessárias para que o Método dos Gradientes Conjugados converja à solução do sistema linear. Assim como no Método dos Gradientes, trocaremos o problema de encontrar uma solução para o sistema $Ax = b$ pelo problema equivalente de encontrar um minimizador de

$$f(x) = \frac{1}{2}x^T Ax - b^T x,$$

em que A é simétrica e definida positiva.

Dada a função $f(x) = \frac{1}{2}x^T Ax - b^T x$, temos:

$$\nabla f(x) = Ax - b,$$

$$\nabla^2 f(x) = A.$$

Encontrar a solução do sistema linear $Ax = b$ é equivalente a encontrar o ponto x que satisfaz $\nabla f(x) = Ax - b = 0$, ou seja, o minimizador da função f . Assim, o algoritmo deste método pode ser descrito da seguinte forma:

1. Entrada:

- Função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$
- Ponto inicial $x_0 \in \mathbb{R}^n$
- Número máximo de iterações N
- Tolerância $\epsilon > 0$

2. Inicialização:

- Calcule o gradiente inicial $r_0 = \nabla f(x_0)$
- Defina $p_0 = -r_0$
- Defina $k = 0$

3. Enquanto $k < N$ fazer:

- Calcule a solução do problema unidimensional: $\alpha_k = \arg \min_{\alpha} f(x_k + \alpha p_k)$
- Atualize o ponto: $x_{k+1} = x_k + \alpha_k p_k$
- Calcule o novo gradiente: $r_{k+1} = \nabla f(x_{k+1})$
- Se $\|r_{k+1}\| < \epsilon$ (critério de parada), **pare**
- Atualize a direção: $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
- Defina: $p_{k+1} = -r_{k+1} + \beta_k p_k$

4. Defina $k = k + 1$

Saída: Aproximação do minimizador x_k

3.1.4 Método de quasi-Newton

Os métodos Quasi-Newton constituem uma classe de técnicas de otimização que são uma extensão dos métodos Newton-Raphson, mas se mostram mais eficientes do ponto de vista computacional, pois não exigem o cálculo direto da matriz Hessiana da função objetivo. Em vez disso, eles aproximam a matriz Hessiana utilizando informações do gradiente da função.

Esses métodos são especialmente úteis quando o cálculo da matriz Hessiana é computacionalmente caro ou impraticável. Eles funcionam atualizando iterativamente uma aproximação da matriz Hessiana com base nas mudanças no gradiente ao longo das iterações.

Os algoritmos de métodos Quasi-Newton são baseados na seguinte ideia: a iteração da matriz W_n segue a regra

$$W_{n+1} = W_n + E_n,$$

em que E_n é uma matriz definida negativa (ou positiva). Observe que, se W_0 é negativa (ou positiva) definida (geralmente, usa-se $W_0 = I$), então W_n será positiva (ou negativa) definida para qualquer $n \geq 0$. A seguir, é apresentada a implementação do método BFGS (Broyden-Fletcher-Goldfarb-Shanno):

1. Entrada:

- Função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$
- Ponto inicial $x_0 \in \mathbb{R}^n$
- Número máximo de iterações N
- Tolerância $\epsilon > 0$

2. Inicialização:

- Calcule o gradiente inicial $g_0 = \nabla f(x_0)$
- Defina $H_0 = I$ (matriz identidade)
- Defina $k = 0$

3. Enquanto $k < N$ fazer:

- (a) Se $\|g_k\| < \epsilon$ (critério de parada), **pare**
- (b) Calcule a direção de descida: $p_k = -H_k g_k$
- (c) Determine o tamanho do passo: $\alpha_k = \arg \min_{\alpha} f(x_k + \alpha p_k)$
- (d) Atualize o ponto: $x_{k+1} = x_k + \alpha_k p_k$
- (e) Calcule o novo gradiente: $g_{k+1} = \nabla f(x_{k+1})$
- (f) Calcule a diferença: $s_k = x_{k+1} - x_k$
- (g) Calcule a diferença do gradiente: $y_k = g_{k+1} - g_k$
- (h) Atualize a matriz Hessiana aproximada: $H_{k+1} = H_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k}$

4. Defina $k = k + 1$

Saída: Aproximação do minimizador x_k

3.2 Métodos de busca

3.2.1 Seção áurea

Pelo método de seção áurea, busca-se a solução numérica para o problema de encontrar máximos ou mínimos de uma função escalar de uma única variável, sem restrições. Inicialmente, considera-se

que essa função é não linear, uma vez que, no caso linear em uma dimensão, não existem máximos ou mínimos finitos.

O objetivo é determinar o mínimo de uma função unidimensional através de um processo em etapas. Inicialmente, um intervalo $[x_L, x_U]$ é definido dentro do qual o mínimo será buscado. Para detectar a ocorrência de um mínimo, selecionamos dois pontos dentro desse intervalo, escolhidos com base na razão áurea: $x_a = x_L + d$, $x_b = x_U - d$, $d = (\phi - 1)(x_U - x_L)$.

Em seguida, calcula-se o valor da função nesses dois pontos. Dois resultados podem ocorrer:

1. Se $f(x_a) < f(x_b)$, então $f(x_a)$ é o mínimo do intervalo, e o domínio à esquerda de x_a (de x_L a x_b) pode ser eliminado da busca, pois não contém o mínimo. Nesse caso, x_b se torna o novo x_L para a próxima iteração.
2. Se $f(x_b) < f(x_a)$, então $f(x_b)$ é o mínimo do intervalo, e o domínio à direita de x_a (de x_a a x_U) pode ser eliminado da busca. Nesta situação, x_a se torna o novo x_U para a próxima iteração.

Como os valores de x_a e x_b foram escolhidos usando a razão áurea, não é necessário calcular todos os valores da função na próxima iteração.

Para completar o algoritmo, determina-se o novo x_a utilizando a equação mencionada, com base nos novos valores de x_L e x_U . No segundo caso, o procedimento é análogo. Demonstra-se que, a cada iteração, o intervalo é reduzido em aproximadamente 61,8%. Se o objetivo for encontrar o máximo de uma função $f(x)$, basta encontrar o mínimo da função com sinal trocado: $F(x) = -f(x)$.

Segue-se abaixo o algoritmo deste método:

1. Entrada:

- Função $f : [a, b] \rightarrow \mathbb{R}$ a ser minimizada
- Tolerância $\epsilon > 0$
- Número máximo de iterações N

2. Inicialização:

- Calcule $\phi = \frac{1 + \sqrt{5}}{2}$ (razão áurea)
- Defina $b' = b - \frac{b - a}{\phi}$

- Defina $a' = a + \frac{b-a}{\phi}$

3. **Enquanto** $|b-a| > \epsilon$ **e** $k < N$ **fazer:**

(a) Calcule $f(b')$ e $f(a')$

(b) **Se** $f(b') < f(a')$:

- Defina $a = a'$
- Atualize $b' = b - \frac{b-a}{\phi}$
- Atualize $a' = a + \frac{b-a}{\phi}$

(c) **Caso contrário:**

- Defina $b = b'$
- Atualize $a' = a + \frac{b-a}{\phi}$
- Atualize $b' = b - \frac{b-a}{\phi}$

4. **Saída:** Aproximação do ponto mínimo $x^* = \frac{a+b}{2}$

3.2.2 Condição de Armijo

A Condição ou Busca de Armijo é uma técnica de busca linear inexata amplamente utilizada para a determinação do comprimento de passo. Esta regra garante que a função objetivo diminua de maneira suficiente em relação ao comprimento de passo escolhido, promovendo assim uma convergência mais eficaz do algoritmo.

Consiste em encontrar um $\alpha_k \in [0, \delta)$, com $\delta > 0$, tal que a seguinte condição seja satisfeita: $f(x_k + \alpha_k d_k) \leq f(x_k) + \eta_1 \alpha_k \nabla f(x_k)^T d_k$, em que $x_k \in \mathbb{R}^n$ é o ponto atual, $d_k \in \mathbb{R}^n \setminus \{0\}$ é uma direção de descida, e $\eta_1 \in (0, 1)$ é um parâmetro que controla a taxa de redução. Essa busca é projetada para evitar comprimentos de passo grandes que não resultem em um decrescimento significativo da função objetivo.

A seguir, o algoritmo implementado é apresentado:

1. Entrada:

- Função objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$
- Ponto atual $x_k \in \mathbb{R}^n$
- Direção de descida d_k
- Parâmetros $\alpha_0 > 0$, $\beta \in (0, 1)$, $\sigma \in (0, 1)$

2. Inicialização:

- Defina $\alpha = \alpha_0$

3. Enquanto $f(x_k + \alpha d_k) > f(x_k) + \sigma \alpha \nabla f(x_k)^T d_k$ fazer:

- (a) Atualize α : $\alpha = \beta \alpha$

4. Saída: Tamanho do passo α que satisfaz a condição de Armijo.

4 Resultados e discussão

A análise dos resultados dos métodos de otimização irrestrita revela diferenças significativas entre as abordagens utilizadas, especialmente em relação à eficiência e ao tempo de convergência. A Tabela 1 apresenta os resultados obtidos por cada métodos implementada, executando-os para uma função quadrática.

Tabela 1: Resultados dos Métodos de Otimização para a Função Quadrática

Método	Busca	Iterações	Função Objetivo	Tempo (s)	Norma do Gradiente
Gradiente	Seção Áurea	2	$8,999271 \times 10^{-17}$	0,01	$1,897253 \times 10^{-8}$
Gradiente	Armijo	3	$8,637315 \times 10^{-24}$	0,02	$5,877607 \times 10^{-12}$
Newton	Seção Áurea	402315	$2,500046 \times 10^{-13}$	354,14	$9,999875 \times 10^{-7}$
Newton	Armijo	2	$2,625377 \times 10^{-30}$	0,02	$3,240572 \times 10^{-15}$
Gradientes Conjugados	Seção Áurea	12	Derivada muito próxima de zero!	-	-
Gradientes Conjugados	Armijo	20	$4,230032 \times 10^{-15}$	829,97	$1,300723 \times 10^{-7}$
Quasi-Newton	Seção Áurea	11	$1,163421 \times 10^{-15}$	0,24	$6,821659 \times 10^{-8}$
Quasi-Newton	Armijo	12	$7,013593 \times 10^{-14}$	0,07	$5,296494 \times 10^{-7}$

Os resultados sugerem que, para problemas de otimização quadrática, a escolha da estratégia de busca pode influenciar significativamente o número de iterações necessárias para a convergên-

cia. Embora alguns métodos sejam teoricamente mais robustos, a aplicação prática pode revelar limitações significativas, principalmente em relação à computação e ao tempo de convergência.

O método do gradiente, por exemplo, apresentou um bom desempenho quando combinado com os dois métodos de busca. Com poucas iterações, conseguiu atingir uma função objetivo muito próxima de zero, demonstrando uma rápida convergência. Entretanto, a busca de Armijo destacou-se pela eficiência ao encontrar uma função objetivo de menor valor e com uma convergência mais precisa.

O método de Newton, por sua vez, mostrou-se mais complexo em termos de desempenho. Quando utilizado com a seção áurea, o número de iterações foi exorbitante, totalizando 402315, o que é indicativo de uma convergência lenta neste caso específico. No entanto, a busca de Armijo se revelou mais eficiente, alcançando uma solução com apenas duas iterações e um valor de função objetivo ainda menor.

Os métodos de gradientes conjugados e quasi-Newton apresentaram um comportamento variado. O gradiente conjugado, ao utilizar a busca de Armijo, levou 20 iterações e um tempo computacional elevado, indicando que não ser a melhor escolha para problemas quadráticos simples. Além disso, ao ser executado o método em combinação a seção áurea, não foi possível obter um valor convergente, pois a derivada a partir da 12^a iteração aproximava a um valor extremamente próximo de zero. Por outro lado, o método quasi-Newton, apesar de ter um desempenho razoável com os métodos de busca, também não conseguiu se igualar à eficiência do método do gradiente com as mesmas estratégia e de Newton, combinado à Busca de Armijo.

A Tabela 2 a seguir apresenta os resultados obtidos a partir desses métodos aplicados à função da raiz quadrada suavizada. Os resultados obtidos com a implementação dos diferentes métodos de otimização também demonstram variações significativas para a função raiz quadrada suavizada.

O método do gradiente com a estratégia de seção áurea superou a busca de Armijo em termos de número de iterações e tempo computacional necessários para convergir. Enquanto a seção áurea levou 568.297 iterações em 197,39 segundos, a busca de Armijo requereu 1.136.595 iterações em 118,90 segundos. Essa diferença indica que a escolha da estratégia de busca pode impactar consideravelmente o desempenho do método do gradiente.

Por outro lado, o método de Newton exibiu um desempenho mais instável. Ao utilizar a seção áurea, o método de Newton não conseguiu convergir dentro do limite de 1.416.319 iterações, mesmo com a tolerância relaxada para 10^{-5} . Somente com uma tolerância de 10^{-4} é que o método

Tabela 2: Resultados dos Métodos de Otimização para a Função Raiz Quadrada Suavizada

Método	Busca	Iterações	Função Objetivo	Tempo (s)	Norma do Gradiente
Gradiente	Seção Áurea	568297	1,000000	197,39	$9,999991 \times 10^{-7}$
Gradiente	Armijo	1136595	1,000000	118,90	$9,999990 \times 10^{-7}$
Newton	Seção Áurea	1	1,000015	0,23	$1,830520 \times 10^{-5}$
Newton	Armijo	1	1,000015	0,07	$1,830370 \times 10^{-5}$
Gradientes Conjugados	Seção Áurea	4359934	1,000000	567,06	$9,999998 \times 10^{-7}$
Gradientes Conjugados	Armijo	2260	1,000000	0,28	$9,994375 \times 10^{-7}$
Quasi-Newton	Seção Áurea	6	1,000000	0,17	$6,472623 \times 10^{-7}$
Quasi-Newton	Armijo	6	1,000000	0,04	$8,410578 \times 10^{-7}$

de Newton com seção áurea convergiu em uma única iteração, em 0,23 segundos, como para a estratégia de busca de Armijo, com convergência em apenas 1 iteração e tolerância de 10^{-4} , com tempo de 0,07.

O método de gradientes conjugados com a estratégia de seção áurea apresentou desempenho inferior ao método do gradiente, pois precisou de 4.359.934 iterações e 567,06 segundos para convergir. No entanto, a utilização da busca de Armijo melhorou significativamente o desempenho do método de gradientes conjugados, que convergiu em 2.260 iterações e 0,28 segundos.

O método quasi-Newton, por sua vez, demonstrou um desempenho consistente, com resultados semelhantes para ambas as estratégias de busca e destacando-se pela convergência mais rápida e eficiente entre os métodos testados. Com a seção áurea, o método quasi-Newton convergiu em 6 iterações em 0,17 segundos, enquanto com a busca de Armijo, a convergência também ocorreu em 6 iterações, porém em um tempo um pouco menor, de 0,04 segundos.

A seguir, são apresentados os resultados obtidos a partir desses métodos aplicados à função logarítmica na Tabela 3. Os resultados obtidos com a aplicação de diferentes métodos de otimização para a função logarítmica, com uma tolerância de 1×10^{-6} , revelam um desempenho variado entre as abordagens testadas.

Ao utilizar o método do gradiente com a seção áurea, observou-se que o algoritmo convergiu em 248.329 iterações, levando 78,35 segundos e resultando em uma função objetivo de aproximadamente $-0,693147$. A norma do gradiente foi de $9,999911 \times 10^{-7}$, indicando que a solução estava próxima do ótimo. Quando foi aplicada a busca de Armijo, o método do gradiente levou mais tempo, com 496.659 iterações em 50,63 segundos, mas alcançou o mesmo valor da função objetivo.

Tabela 3: Resultados dos Métodos de Otimização para a Função Logarítmica

Método	Busca	Iterações	Função Objetivo	Tempo (s)	Norma do Gradiente
Gradiente	Seção Áurea	248329	$-6,931470 \times 10^{-1}$	78,35	$9,999911 \times 10^{-7}$
Gradiente	Armijo	496659	$-6,931470 \times 10^{-1}$	50,63	$9,999923 \times 10^{-7}$
Newton	Seção Áurea	146879	$-6,931472 \times 10^{-1}$	143,05	$9,999857 \times 10^{-7}$
Newton	Armijo	3	$-6,931472 \times 10^{-1}$	0,03	$2,270994 \times 10^{-14}$
Gradientes Conjugados	Seção Áurea	2069152	0	646,47	$2,617699 \times 10^{-27}$
Gradientes Conjugados	Armijo	2246	$-6,931471 \times 10^{-1}$	0,44	$9,994292 \times 10^{-7}$
Quasi-Newton	Seção Áurea	119	$-6,931471 \times 10^{-1}$	1,00	$9,235893 \times 10^{-7}$
Quasi-Newton	Armijo	126	$-6,931471 \times 10^{-1}$	0,33	$8,773117 \times 10^{-7}$

Essa diferença sugere que a seção áurea pode ser mais eficiente neste contexto.

Ao testar o método de Newton com a seção áurea, o algoritmo convergiu em 146.879 iterações e 143,05 segundos, obtendo uma função objetivo ligeiramente melhor, $-0,693472$, com uma norma do gradiente de $9,999857 \times 10^{-7}$. No entanto, quando a busca de Armijo foi utilizada, o método de Newton apresentou desempenho excepcional, convergindo em apenas 3 iterações e 0,03 segundos. Essa eficiência notável indica que, para esta função, o método de Newton com busca de Armijo é altamente eficaz.

O método de gradientes conjugados teve um desempenho misto. Com a seção áurea, foram necessárias 2.069.152 iterações e um tempo computacional de 646,47 segundos, mas o resultado foi uma função objetivo de 0, o que indica que o método encontrou um ponto crítico. Em contrapartida, com a busca de Armijo, o método de gradientes conjugados convergiu em 2.246 iterações em apenas 0,44 segundos, resultando em uma função objetivo de aproximadamente $-0,693471$. Essa diferença substancial no número de iterações e tempo sugere que a busca de Armijo é mais adequada para este método.

Por fim, os resultados do método quasi-Newton mostraram uma performance competitiva. Com a seção áurea, o método convergiu em 119 iterações em 1,00 segundo, enquanto com a busca de Armijo, o método levou 126 iterações em 0,328 segundos, ambos resultando em uma função objetivo próxima de $-0,693471$. Ambos os métodos mostraram eficiência, mas a busca de Armijo novamente se destacou em termos de tempo computacional.

Os resultados obtidos através da aplicação de diferentes métodos de otimização à função seno, apresentados na Tabela ??, com uma tolerância de $1,000000 \times 10^{-6}$, mostram um desempenho

variado entre os métodos e as estratégias de busca.

Tabela 4: Resultados dos Métodos de Otimização para a Função Seno

Método	Busca	Iterações	Função Objetivo	Tempo (s)	Norma do Gradiente
Gradiente	Seção Áurea	17487	$-1,000000$	2,74	$9,990435 \times 10^{-7}$
Gradiente	Armijo	34977	$-1,000000$	1,43	$9,992073 \times 10^{-7}$
Newton	Seção Áurea	87734	$-1,000000$	40,59	$9,998681 \times 10^{-7}$
Newton	Armijo	100000	$-0,005386$	19,36	$8,495605 \times 10^{-4}$
Gradientes Conjugados	Seção Áurea	51126	$1,000000$	6,68	$1,068732 \times 10^{-8}$
Gradientes Conjugados	Armijo	419	$-1,000000$	0,00	$9,804306 \times 10^{-7}$
Quasi-Newton	Seção Áurea	164	$-7,957116 \times 10^{-9}$	0,27	$4,837315 \times 10^{-7}$
Quasi-Newton	Armijo	1	$-0,005386$	0,01	$8,495605 \times 10^{-4}$

Iniciando com o método do gradiente, a utilização da seção áurea resultou em 17.487 iterações, levando 2,74 segundos para convergir para uma função objetivo de $-1,000000$. A norma do gradiente foi de $9,990435 \times 10^{-7}$, indicando que a solução estava próxima do ótimo. Ao empregar a busca de Armijo, o número de iterações aumentou para 34.977, mas o tempo computacional foi reduzido para 1,43 segundos. Isso sugere que, apesar do aumento no número de iterações, a eficiência em termos de tempo foi melhorada.

Ao aplicar o método de Newton, com a seção áurea, o algoritmo convergiu em 87.734 iterações, levando 40,59 segundos, e alcançou o mesmo valor de função objetivo, $-1,000000$. No entanto, quando a busca de Armijo foi utilizada, o número de iterações aumentou para 100.000, resultando em uma função objetivo de $-0,005386200$ e um tempo de 19,36 segundos. Essa situação indica que o método de Newton, ao usar a busca de Armijo, não conseguiu encontrar o mínimo da função de forma eficaz.

O desempenho do método de gradientes conjugados foi mais positivo. Com a seção áurea, o método convergiu em 51.126 iterações, obtendo uma função objetivo de $1,000000$ em 6,68 segundos, o que não corresponde ao mínimo global. Em contraste, ao utilizar a busca de Armijo, o método convergiu rapidamente em apenas 419 iterações, alcançando o valor ótimo de $-1,000000$ em apenas 0,09 segundos. Essa diferença substancial na eficiência sugere que a busca de Armijo é a melhor escolha para este método.

Por fim, os resultados do método quasi-Newton mostraram um desempenho competitivo. Com a seção áurea, o método convergiu em 164 iterações em 0,27 segundos, mas não atingiu o mínimo

global, resultando em uma função objetivo de $-7,957116 \times 10^{-9}$. Quando a busca de Armijo foi utilizada, o método se mostrou excepcionalmente eficiente, convergindo em apenas uma iteração e em 0,0085 segundos, mas resultando em uma função objetivo de $-0,005386200$, que não corresponde ao mínimo desejado.

5 Conclusão

A análise dos resultados dos métodos de otimização irrestrita revela que a escolha da estratégia de busca impacta significativamente a eficiência e o tempo de convergência. O método do gradiente se destacou, especialmente com a busca de Armijo, que proporcionou uma rápida convergência e uma função objetivo de menor valor. O método de Newton, embora eficaz com a busca de Armijo, apresentou limitações quando combinado à seção áurea, resultando em um número excessivo de iterações. Os métodos de gradientes conjugados e quasi-Newton mostraram desempenhos variados, com o quasi-Newton demonstrando uma convergência rápida e consistente. Em geral, os resultados indicam que a busca de Armijo tende a ser a abordagem mais eficiente para a maioria dos métodos testados, destacando-se em termos de tempo computacional e número de iterações.

Referências

- Arenales, Marcos *et al.* Pesquisa Operacional: para cursos de engenharia. Rio de Janeiro: Editora Campus, 2011.
- Martínez, J. M.; Santos, S. A. Métodos computacionais de otimização. Campinas: IMECC-UNICAMP, 1995. Versão revisada - setembro 2020.
- Nocedal, J.; Wright, S. J. *Numerical optimization*. Nova Iorque: Springer, 1999.
- Ribeiro, A. A.; Karas, E. W. Otimização contínua: aspectos teóricos e computacionais. Brasil: Cengage Learning, 2013.
- Shor, N. Z. *Minimization methods for non-differentiable functions*. Tradução de K. C. Kiwiel e A. Ruszczyński. Berlim; Nova Iorque; Tóquio: Springer-Verlag, 1979.

APÊNDICE

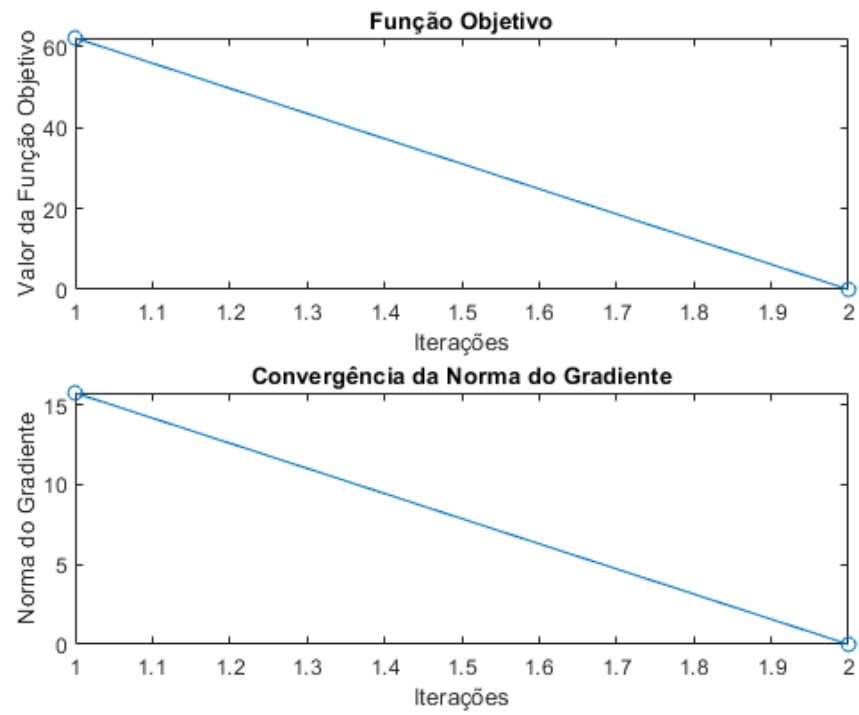


Figura 1: Teste de convergência do método gradiente com seção áurea para a função quadrática.

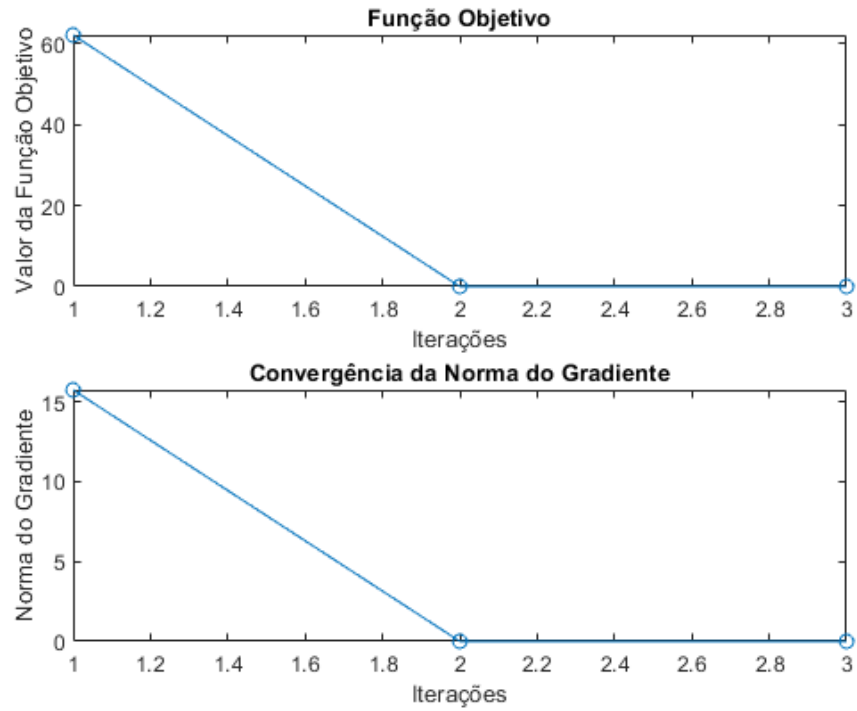


Figura 2: Teste de convergência do método gradiente com busca de Armijo para a função quadrática.

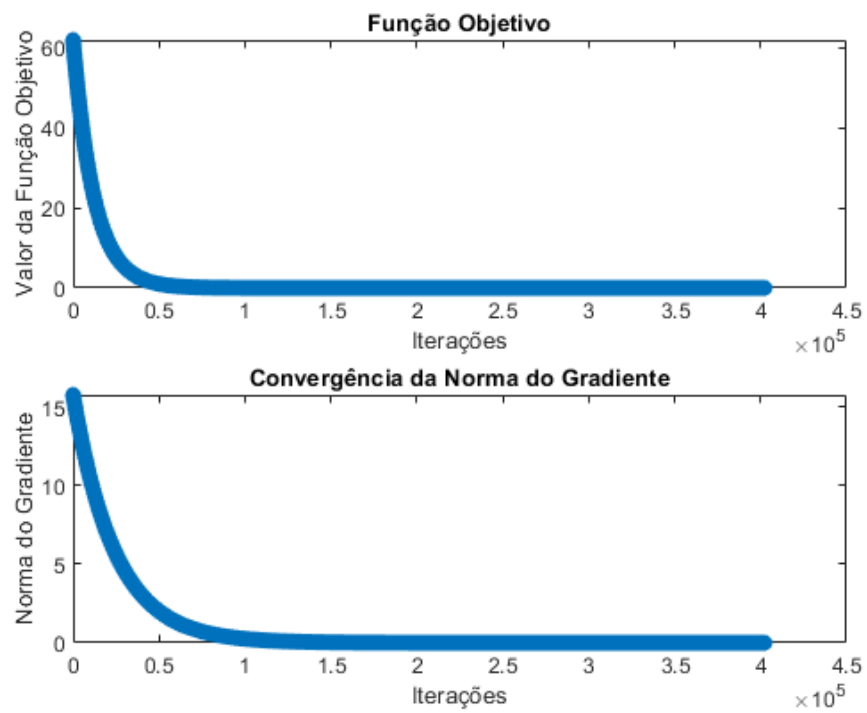


Figura 3: Teste de convergência do método de Newton com seção áurea para a função quadrática.

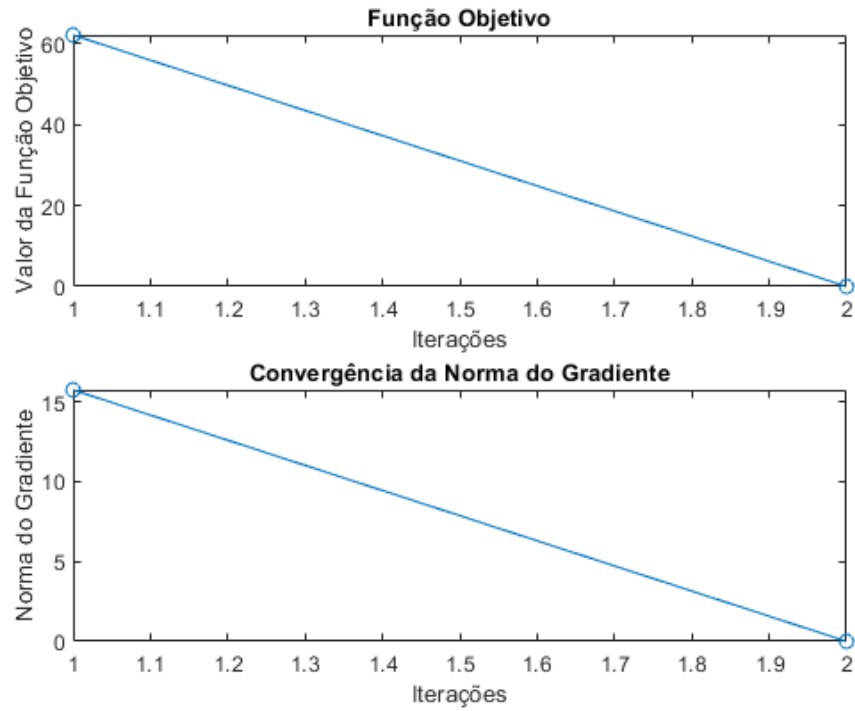


Figura 4: Teste de convergência do método de Newton com busca de Armijo para a função quadrática.

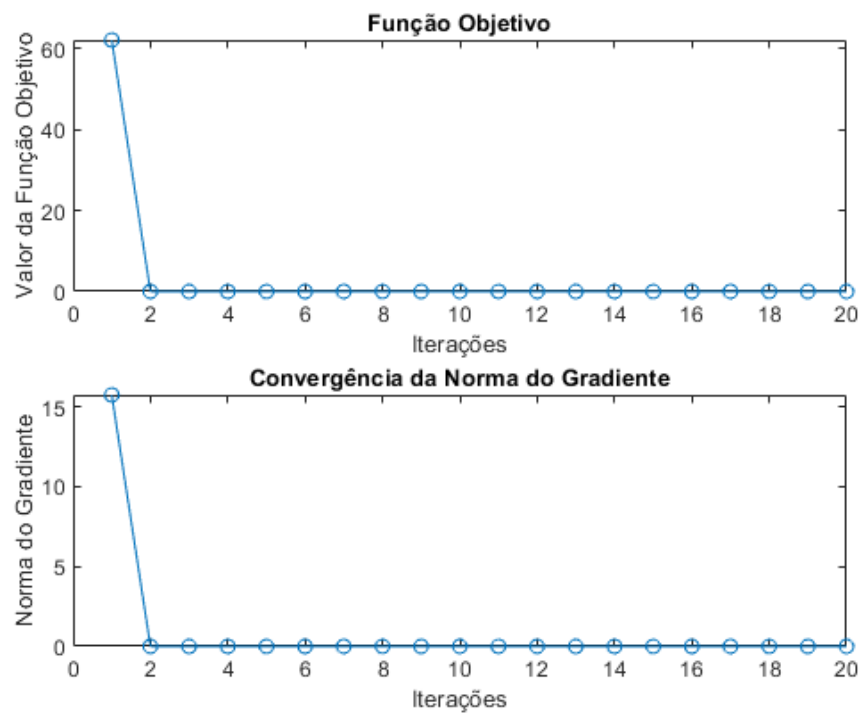


Figura 5: Teste de convergência do método de gradientes conjugados com busca de Armijo para a função quadrática.

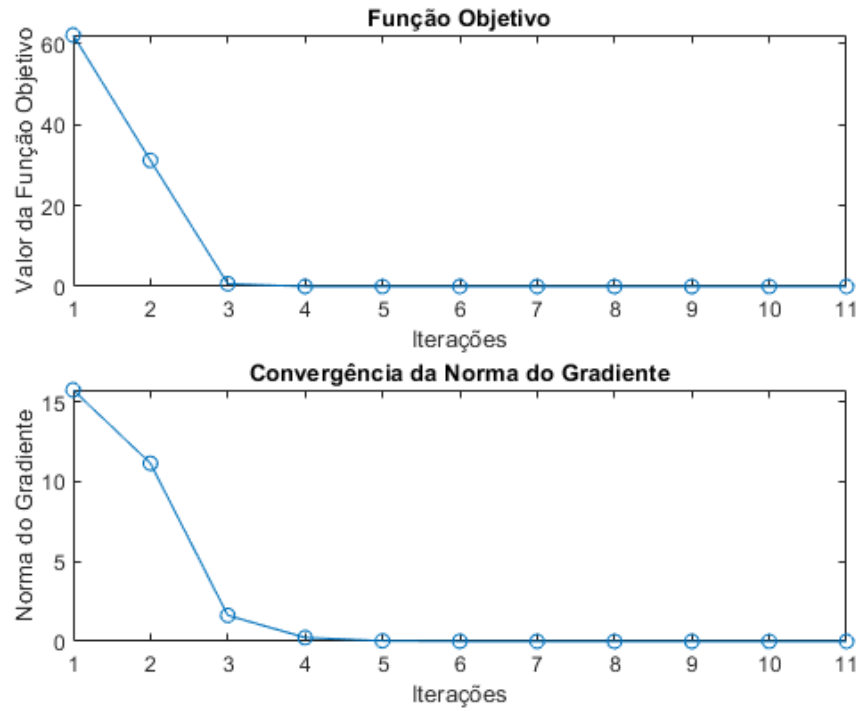


Figura 6: Teste de convergência do método de quasi-Newton com seção áurea para a função quadrática.

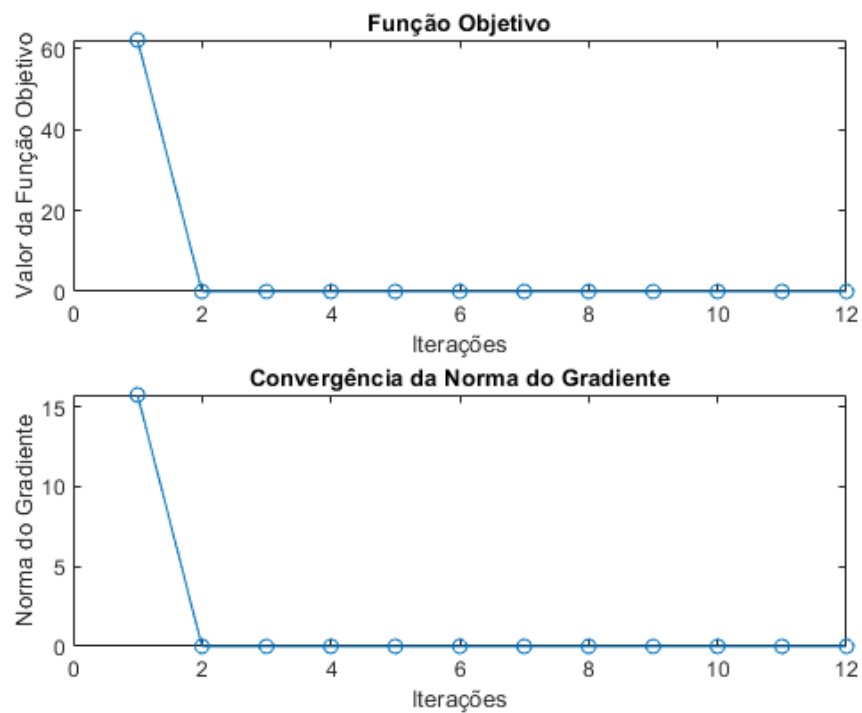


Figura 7: Teste de convergência do método de quasi-Newton com busca de Armijo para a função quadrática.

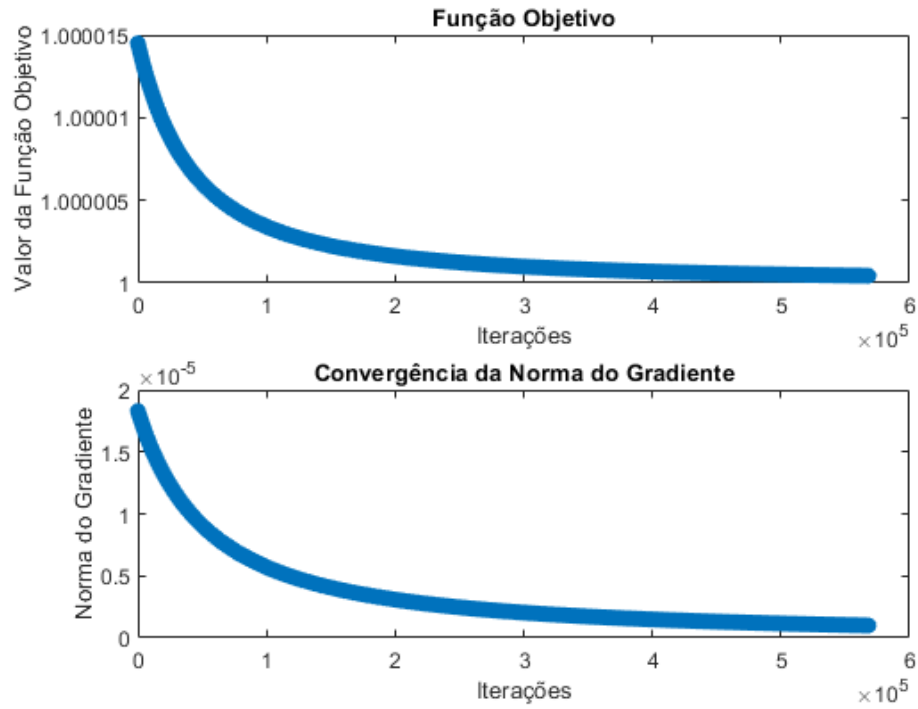


Figura 8: Teste de convergência do método gradiente com seção áurea para a função de raiz quadrada suavizada.

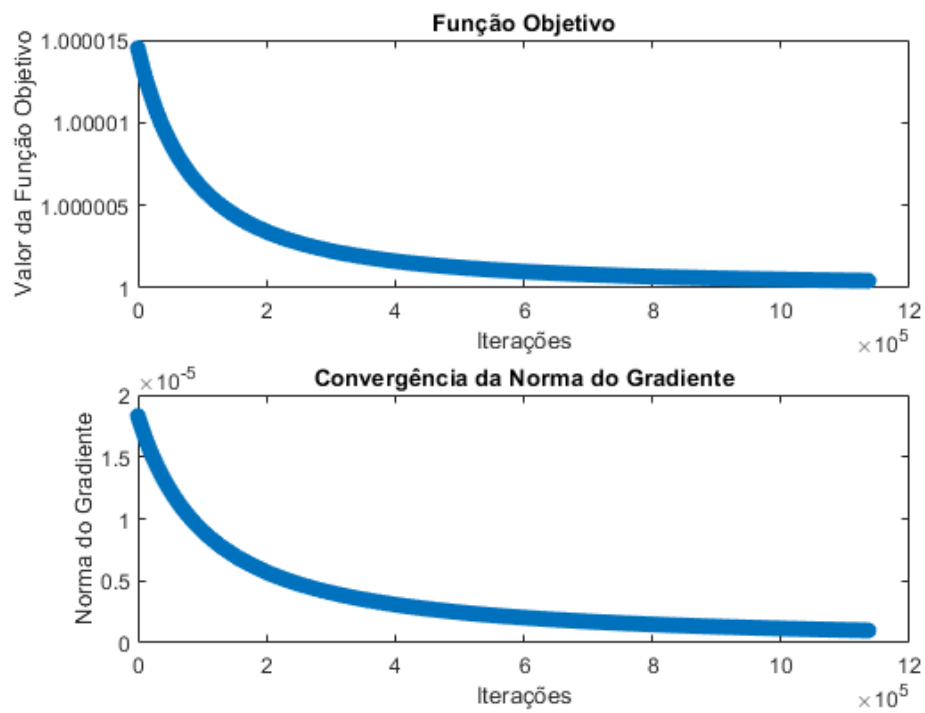


Figura 9: Teste de convergência do método gradiente com busca de Armijo para a função de raiz quadrada suavizada.

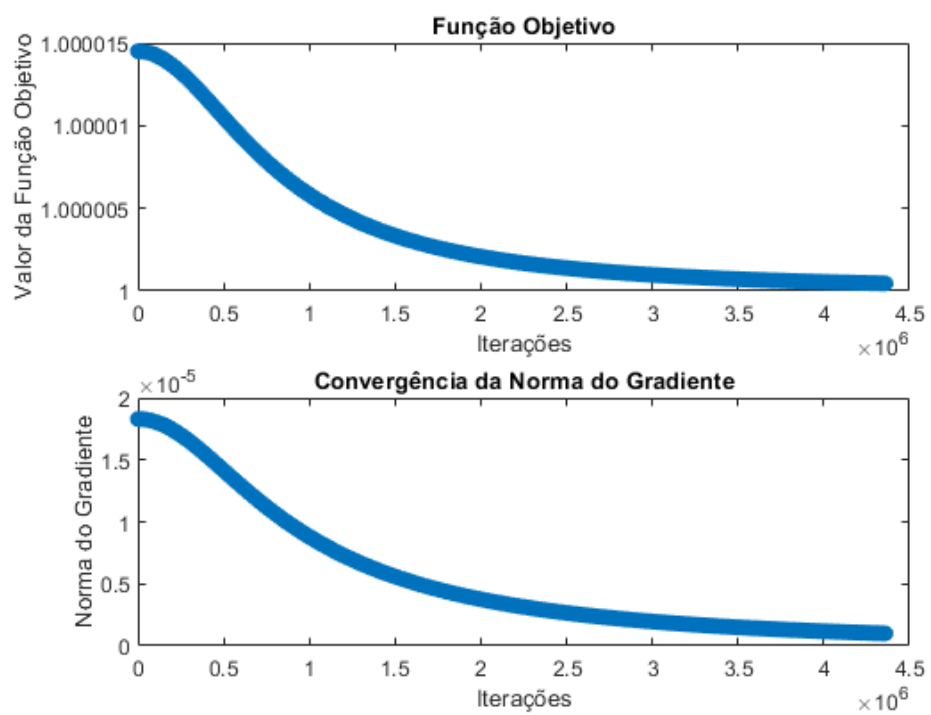


Figura 10: Teste de convergência do método de gradientes conjugados com seção áurea para a função de raiz quadrada suavizada.

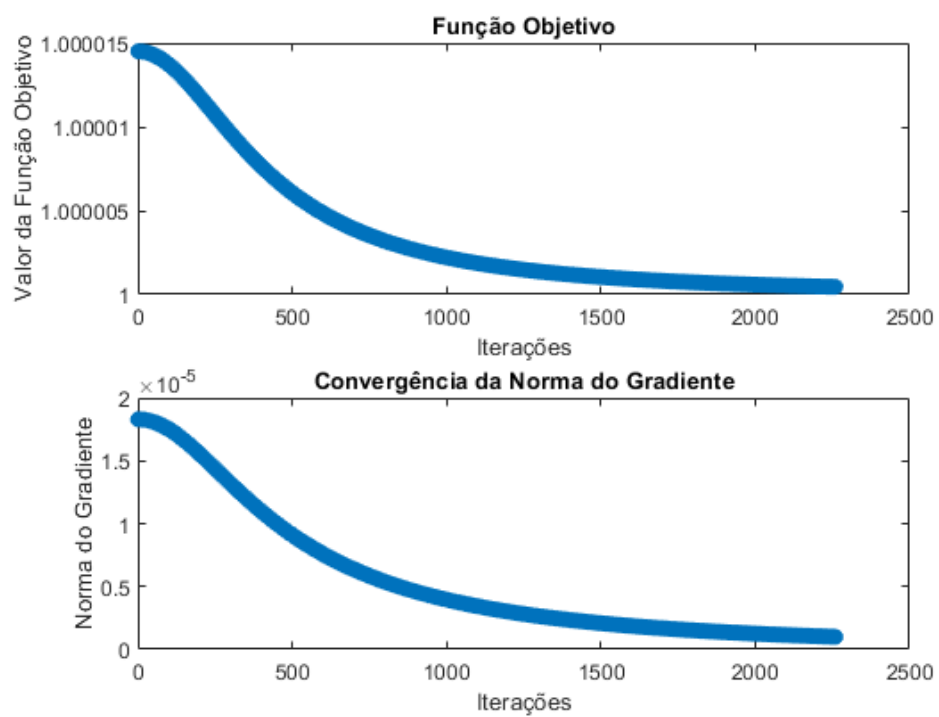


Figura 11: Teste de convergência do método de gradientes conjugados com busca de Armijo para a função de raiz quadrada suavizada.

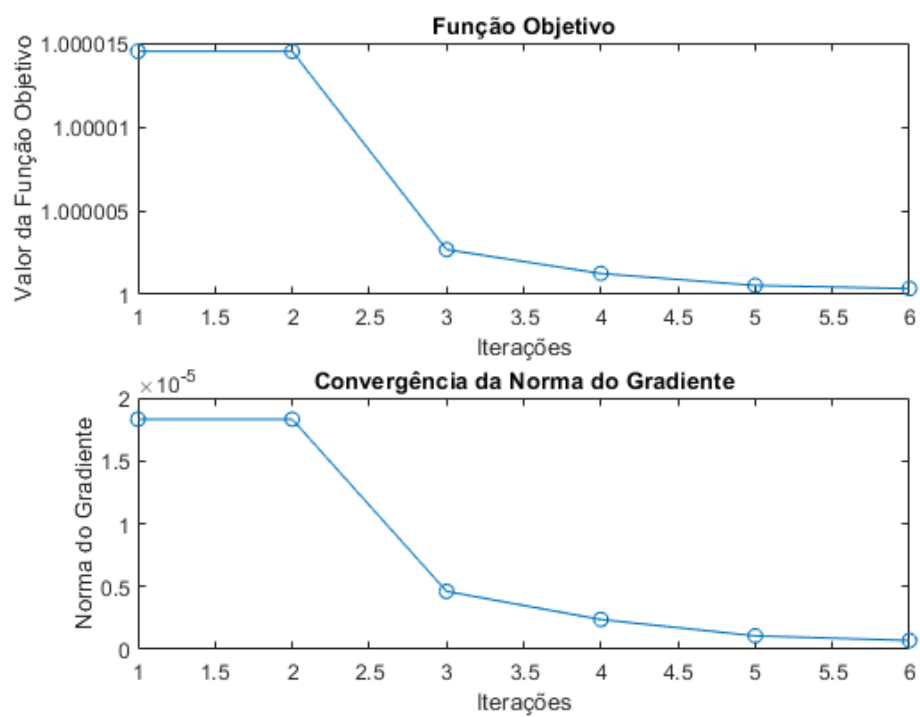


Figura 12: Teste de convergência do método de quasi-Newton com seção áurea para a função de raiz quadrada suavizada.

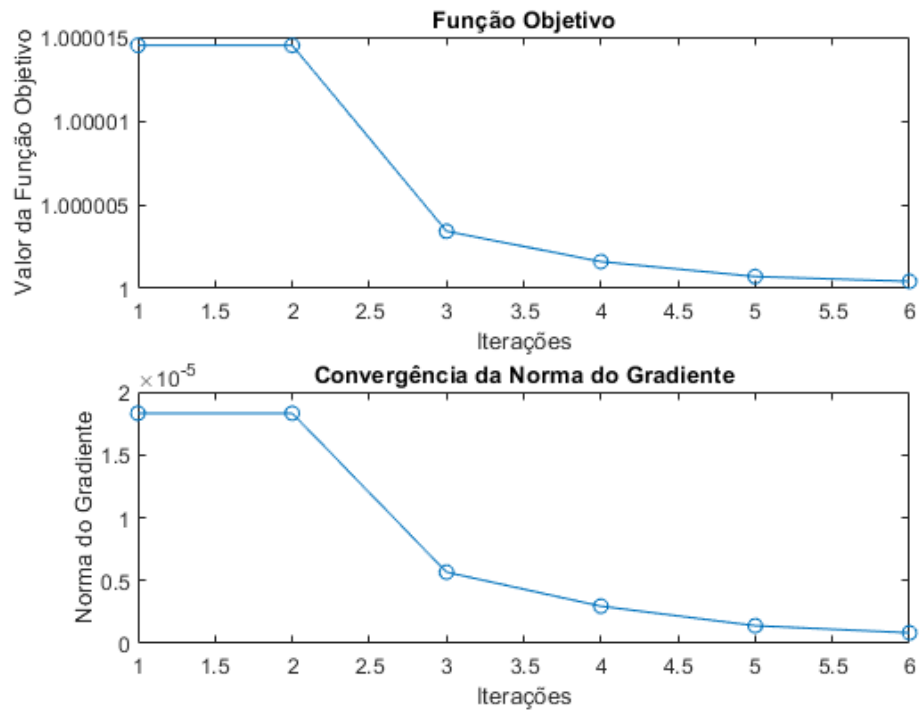


Figura 13: Teste de convergência do método de quasi-Newton com busca de Armijo para a função de raiz quadrada suavizada.

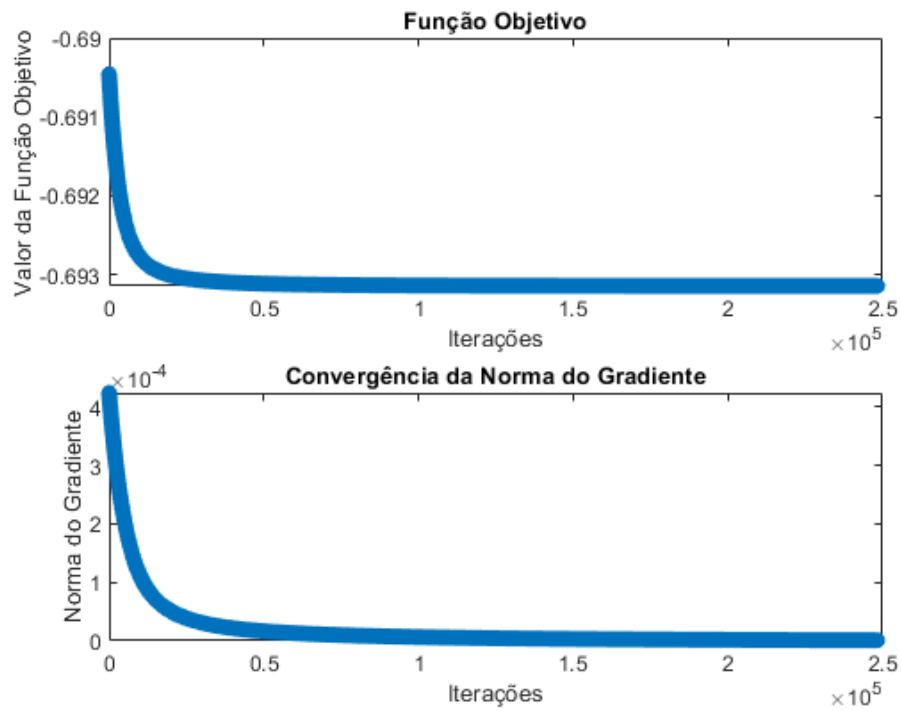


Figura 14: Teste de convergência do método gradiente com seção áurea para a função logarítmica.

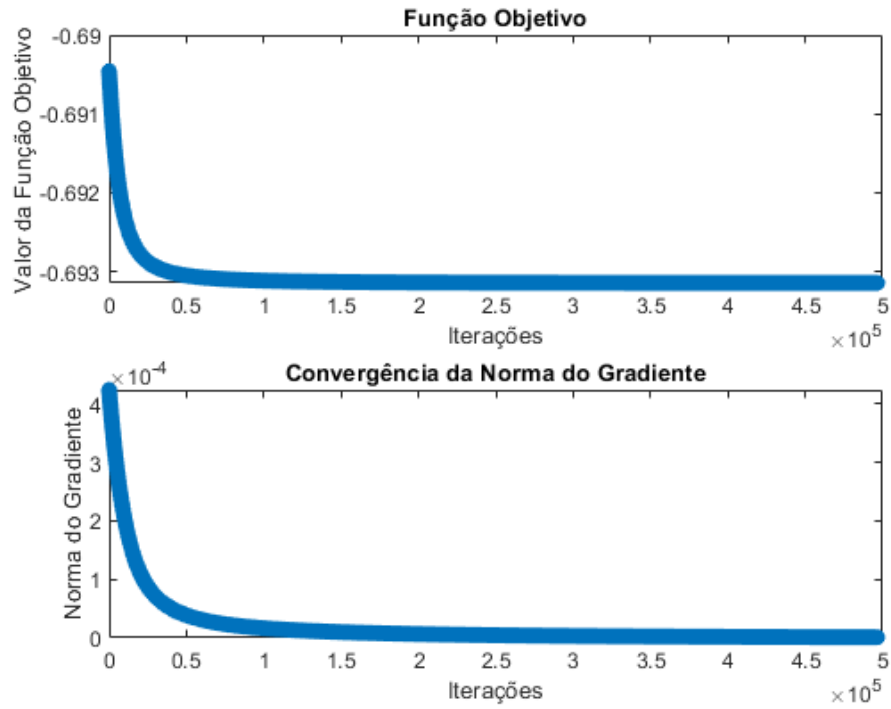


Figura 15: Teste de convergência do método gradiente com busca de Armijo para a função logarítmica.

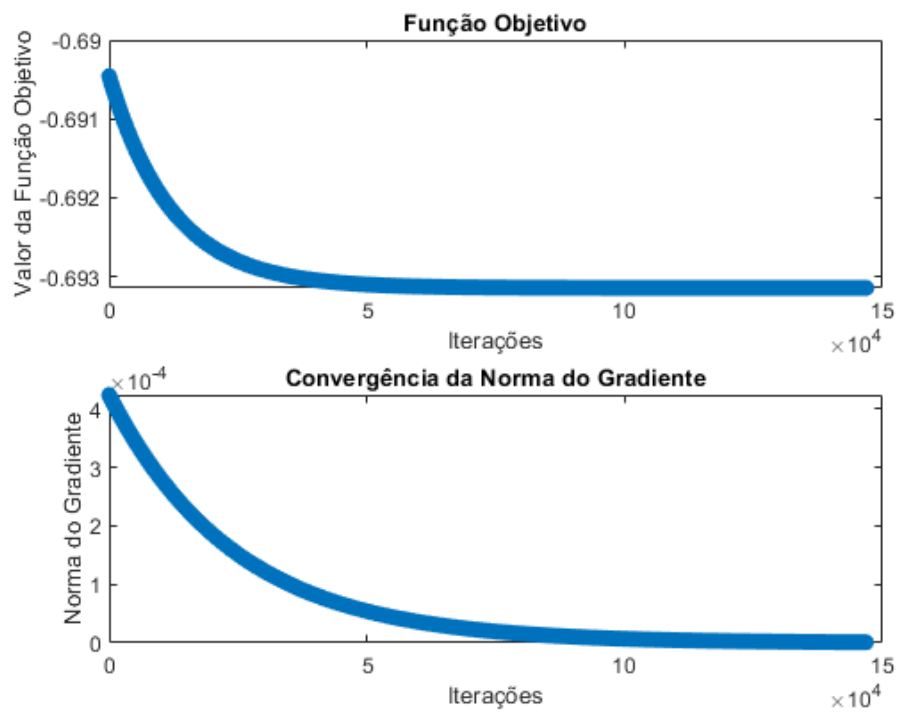


Figura 16: Teste de convergência do método de Newton com seção áurea para a função logarítmica.

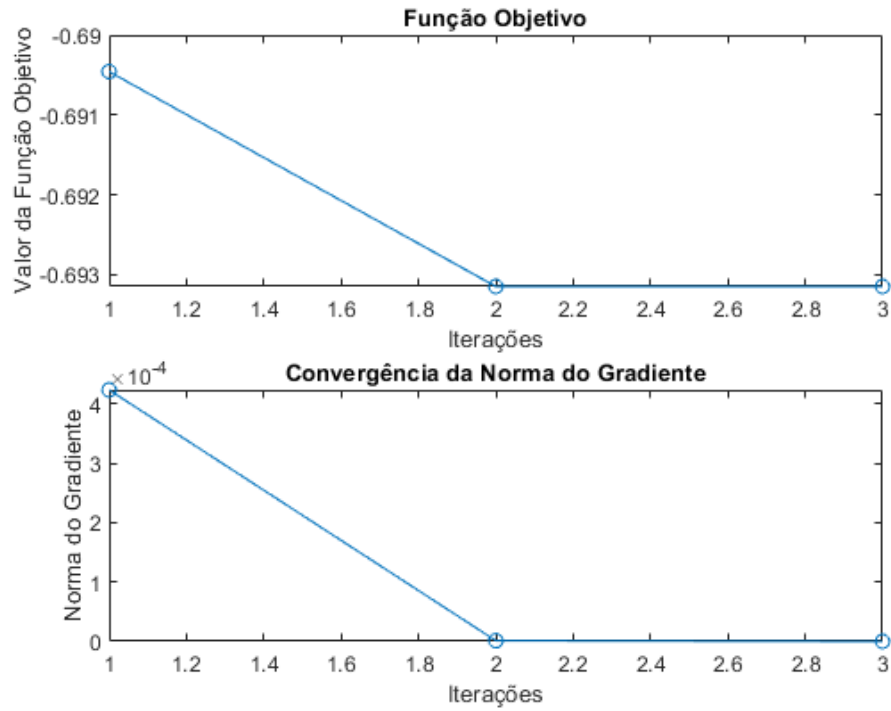


Figura 17: Teste de convergência do método de Newton com busca de Armijo para a função logarítmica.

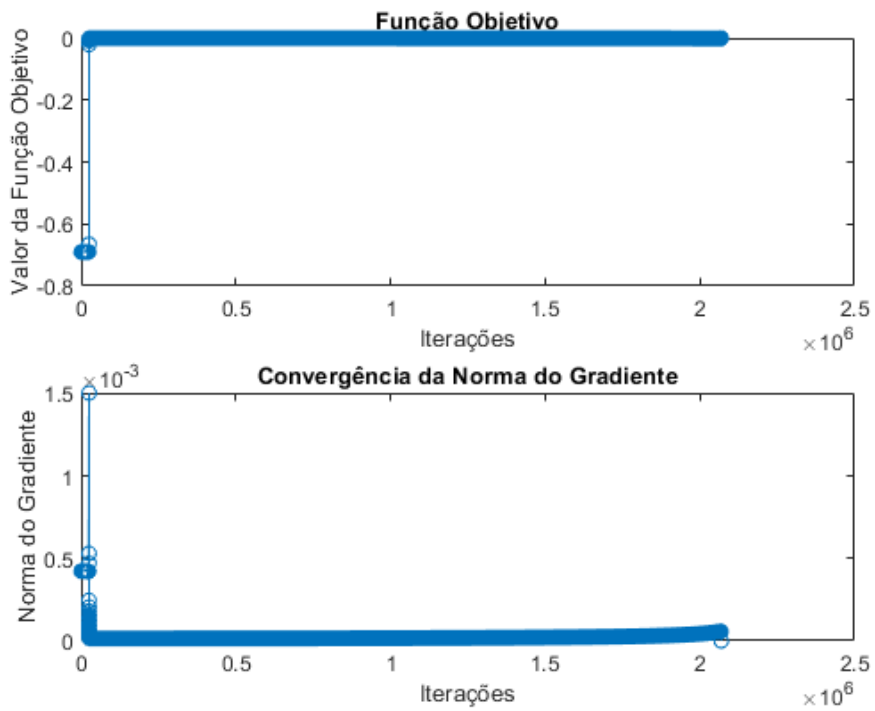


Figura 18: Teste de convergência do método de gradientes conjugados com seção áurea para a função logarítmica.

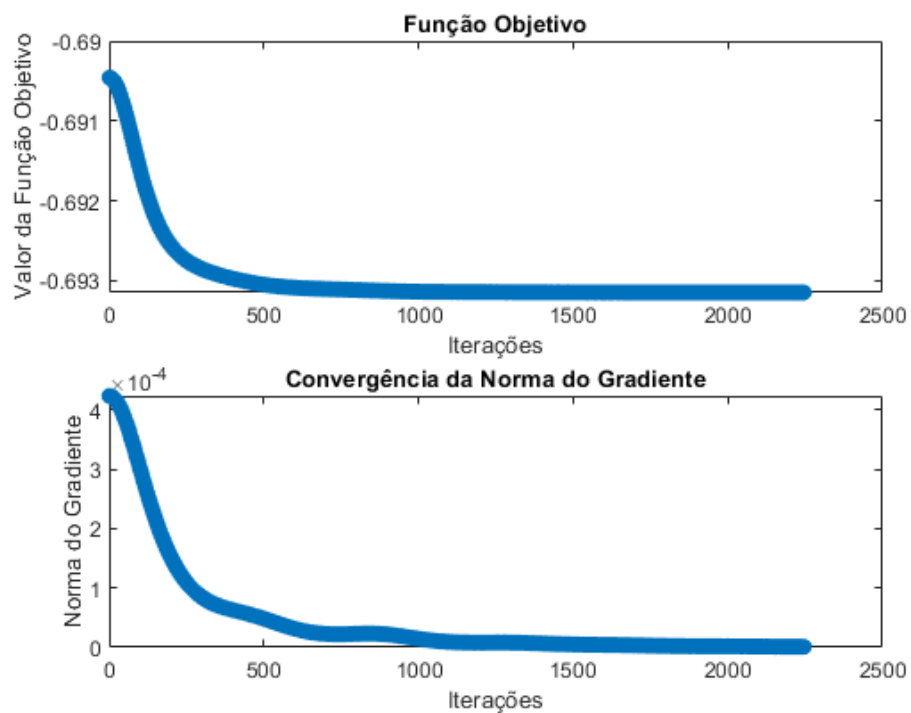


Figura 19: Teste de convergência do método de gradientes conjugados com busca de Armijo para a função logarítmica.

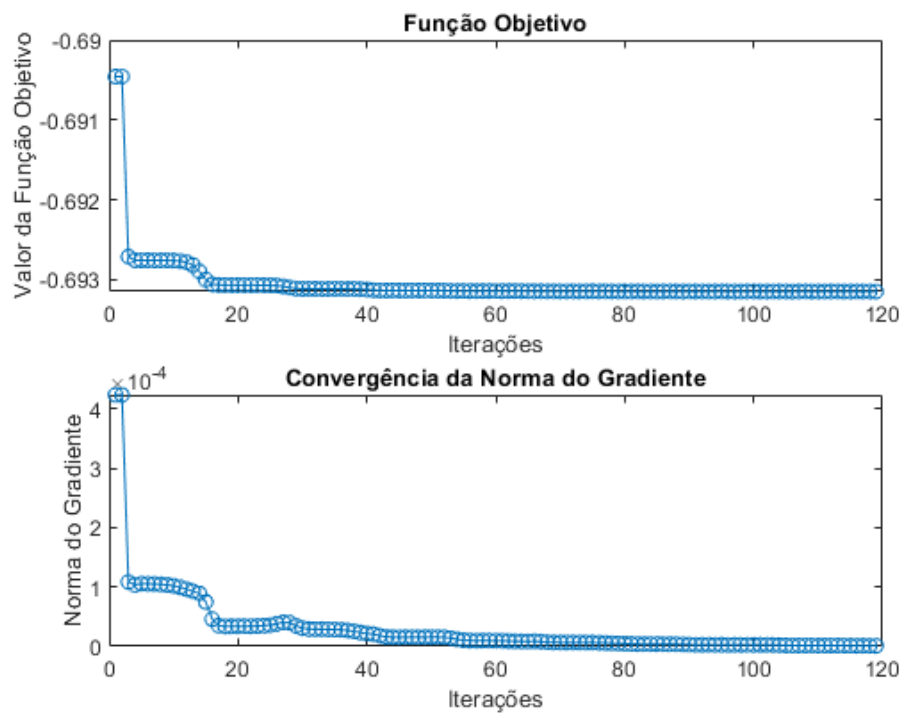


Figura 20: Teste de convergência do método de quasi-Newton com seção áurea para a função logarítmica.

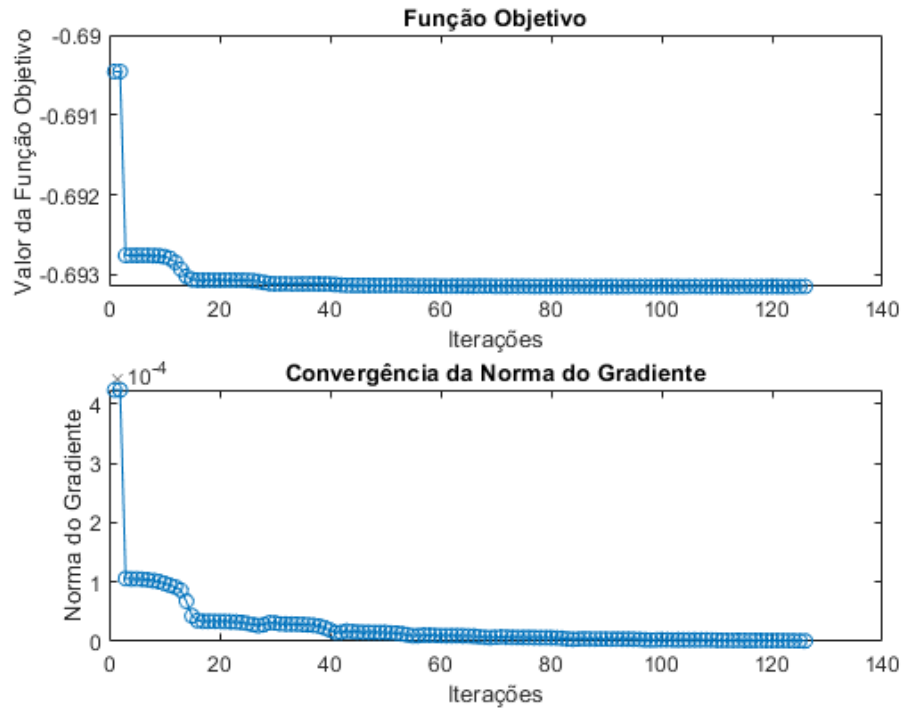


Figura 21: Teste de convergência do método de quasi-Newton com busca de Armijo para a função logarítmica.

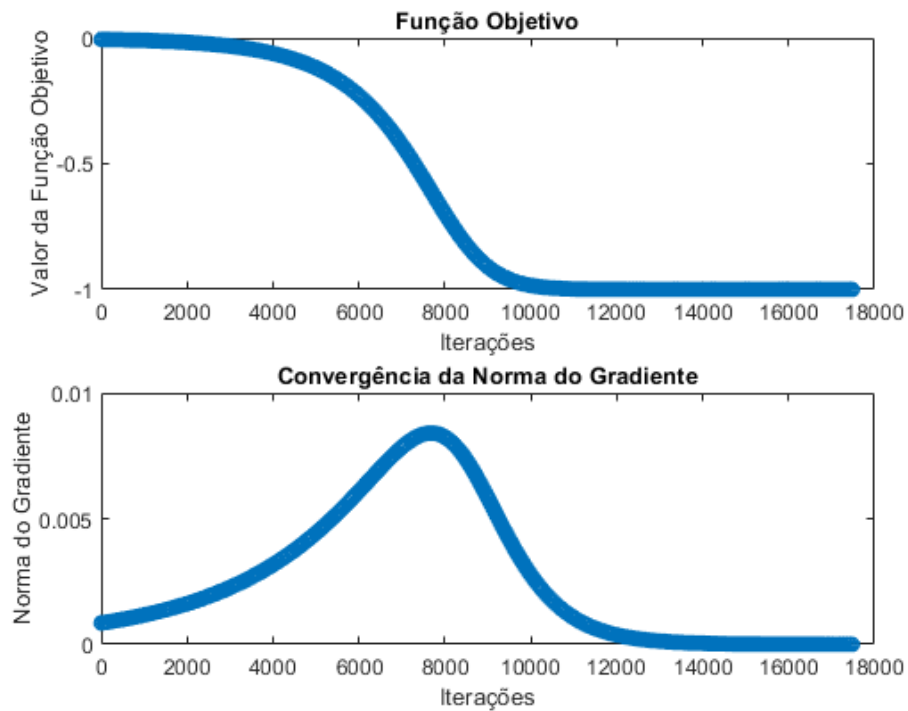


Figura 22: Teste de convergência do método gradiente com seção áurea para a função seno.

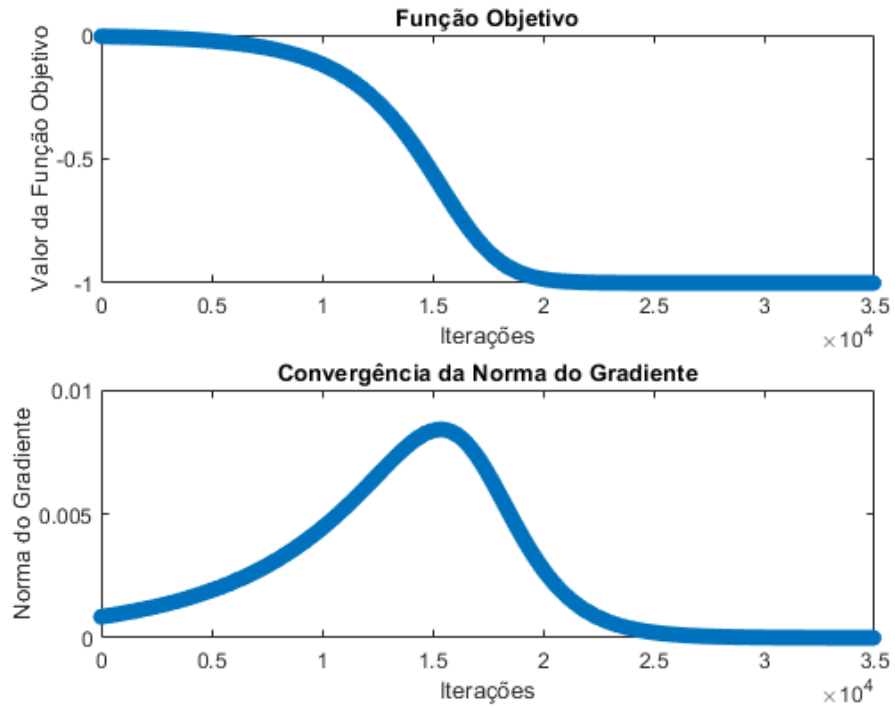


Figura 23: Teste de convergência do método gradiente com busca de Armijo para a função seno.

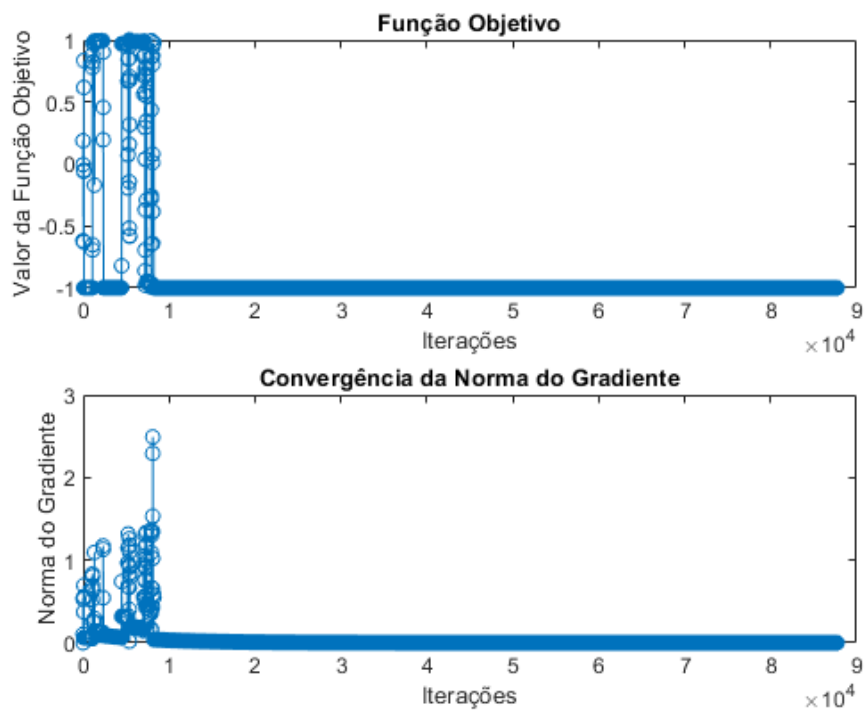


Figura 24: Teste de convergência do método de Newton com seção áurea para a função seno.

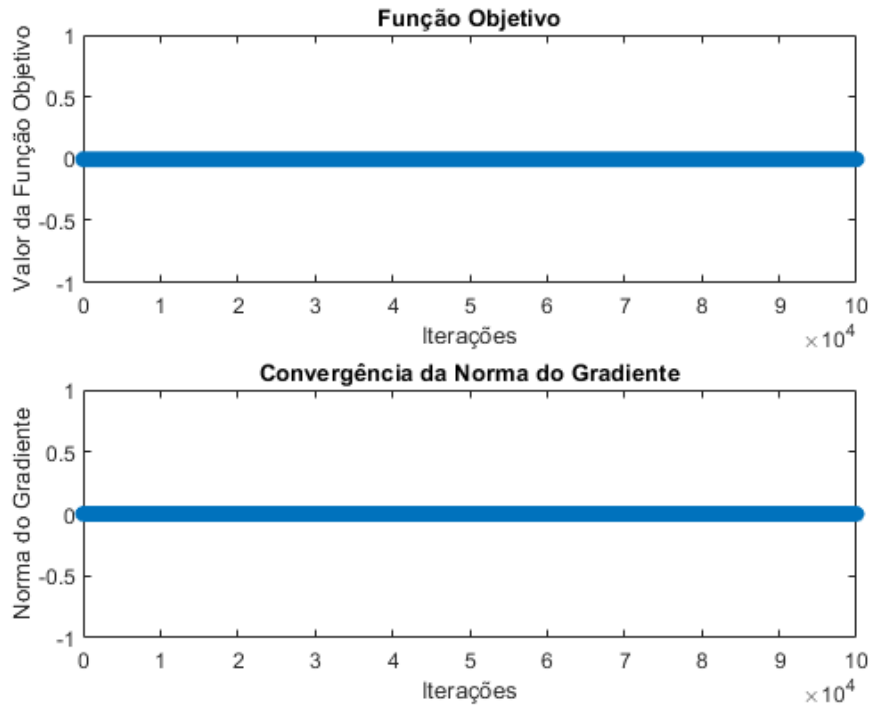


Figura 25: Teste de convergência do método de Newton com busca de Armijo para a função seno.

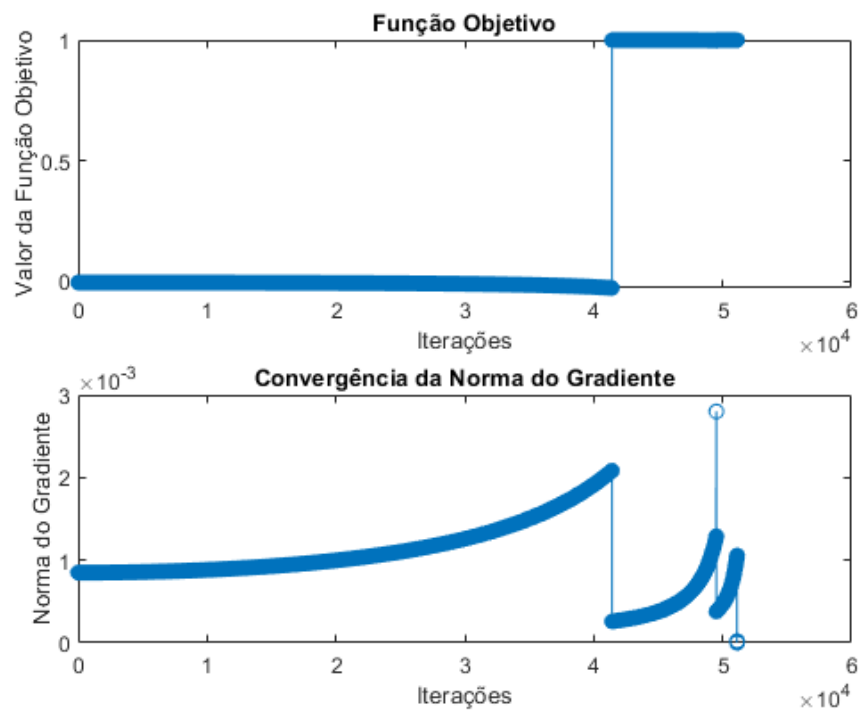


Figura 26: Teste de convergência do método de gradientes conjugados com seção áurea para a função seno.

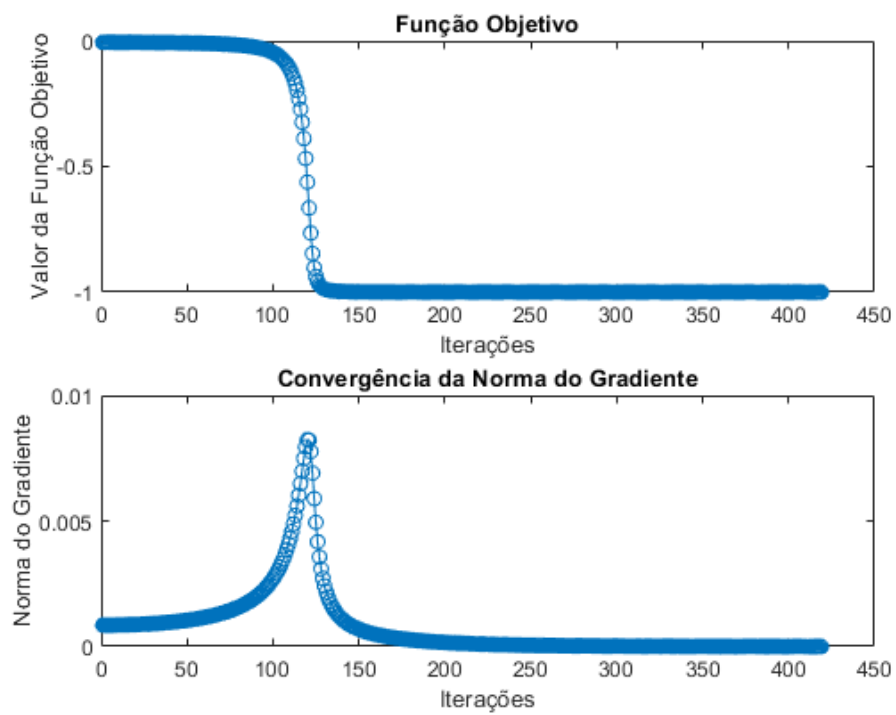


Figura 27: Teste de convergência do método de gradientes conjugados com busca de Armijo para a função seno.