

CPE-737 Otimização Aspectos Teóricos e Métodos Numéricos

Projeto #2

Métodos numéricos para minimização vetorial

trabalho realizado por

Arthur DOS SANTOS XAUD
Carolina CALVO POSE SANTOS NEVES
Gabriel FELIPPE DA CRUZ PACHECO

Prof. Afonso DEL NERO GOMES

10 de dezembro de 2015

Sumário

1	Considerações Iniciais	1
1.1	Modalidade	1
1.2	Busca Espacial	1
1.3	Busca Linear	2
1.4	Critérios de Parada	2
1.5	Interface	2
1.6	Resultados	3
2	Método do Gradiente	4
2.1	Descrição	4
2.2	Resultado	5
3	Método do Gradiente Conjugado	5
3.1	Descrição	5
3.2	Resultado	7
4	Método de Newton	8
4.1	Descrição	8
4.2	Resultado	9
5	Método de Newton Modificado	10
5.1	Descrição	10
5.2	Resultado	10
6	Método de Quasi-Newton	11
6.1	Descrição	11
6.2	Resultado	12
7	Resultados Complementares	13
7.1	Outras funções de duas variáveis	13
7.2	Funções com mais de duas variáveis	14
8	Fluxogramas	16

1 Considerações Iniciais

Nos algoritmos de otimização apresentados nas seções a seguir (Método do Gradiente, Gradiente Conjugado, Newton, Newton Modificado e Quase-Newton), considera-se que as funções de entrada de cada um desses algoritmos são suaves. Tais funções são também vetoriais tais que $f: \mathbb{R}^n \rightarrow \mathbb{R}$.

1.1 Modalidade

O problema de busca de um mínimo para uma dada função vetorial em uma determinada região envolve novamente o conceito de modalidade de funções porém, desta vez, a aboradagem é multivariável.

Uma função unimodal em uma dada região possui apenas um mínimo naquela região. Dessa maneira, se uma função $f(x)$ é unimodal em uma região S , então esta região enquadra um mínimo e é de interesse nos métodos numéricos.

Entretanto, funções podem ter mais de um mínimo, e até mesmo máximos ou pontos de sela, em uma determinada região sendo denominadas então como multimodais. Essas funções também são de interesse nos métodos numéricos e serão abordadas nesse trabalho.

1.2 Busca Espacial

As funções vetoriais $f: \mathbb{R}^n \rightarrow \mathbb{R}$ são representadas graficamente de acordo com a quantidade de variáveis n independentes do problema. O número de variáveis do problema determina o espaço de busca do mínimo da função objetivo.

Funções de apenas uma variável representam curvas bidimensionais em \mathbb{R}^2 , as de duas variáveis representam superfícies tridimensionais em \mathbb{R}^3 , as de três variáveis hiperplanos em \mathbb{R}^4 , e assim por diante para \mathbb{R}^n . Por esse motivo, a representação gráfica de funções com mais de duas variáveis é inviável.

Neste trabalho, para funções com duas variáveis serão gerados gráficos 3D e curvas de nível com os pontos percorridos pelo algoritmo. Para funções com mais de duas variáveis, o resultado mostrado será o um gráfico com o valor da função e do módulo do gradiente.

1.3 Busca Linear

Embora a busca por mínimos em funções vetoriais seja feita no espaço \mathbb{R}^n , uma vez determinado um vetor de direção de descida $f_d : \mathbb{R} \rightarrow \mathbb{R} \mid d_k(\alpha(x)) = f_d(\alpha(x))$, é possível utilizar as técnicas de busca linear apresentadas anteriormente para encontrar o passo de máxima descida nessa direção.

Neste trabalho, foram utilizados na busca linear o método de enquadramento por três pontos para determinar o enquadramento inicial e o método de Brent para a busca do mínimo. Ambos os métodos estão detalhadamente descritos no projeto #1.

1.4 Critérios de Parada

Nos métodos a seguir foram utilizados dois critérios de parada: tolerâncias absolutas e relativas ao erro, e a verificação do módulo do gradiente. Os critérios são baseados na escolha de três parâmetros: tolerância absoluta ϵ_a , tolerância relativa ϵ_r e tolerância do gradiente ϵ_g .

O critério de tolerâncias ao erro é calculado como uma ponderação entre a tolerância absoluta ϵ_a e a relativo ϵ_r . Caso a condição (1) seja satisfeita, o algoritmo é parado.

$$|f(x^{k+1}) - f(x^k)| < \epsilon_a + \epsilon_r |f(x^k)| \quad (1)$$

Da mesma forma, nos métodos a seguir, valores para o módulo do gradiente muito próximos de zero indicam proximidade de um ponto de extremo. Portanto, caso o gradiente seja suficientemente pequeno, dentro de uma tolerância ϵ_g , o critério (2) também é utilizado como condição de parada.

$$\|\nabla f(x)\| < \epsilon_g \quad (2)$$

1.5 Interface

Para testar todos os métodos foi criada uma interface que permite ao usuário escolher a função objetivo vetorial, o método de otimização e os parâmetros necessários. A interface ainda permite a visualização do resultado da otimização de forma agradável e detalhada, utilizando tabelas de dados e gráficos 2D e 3D, como pode ser visto na figura 1.

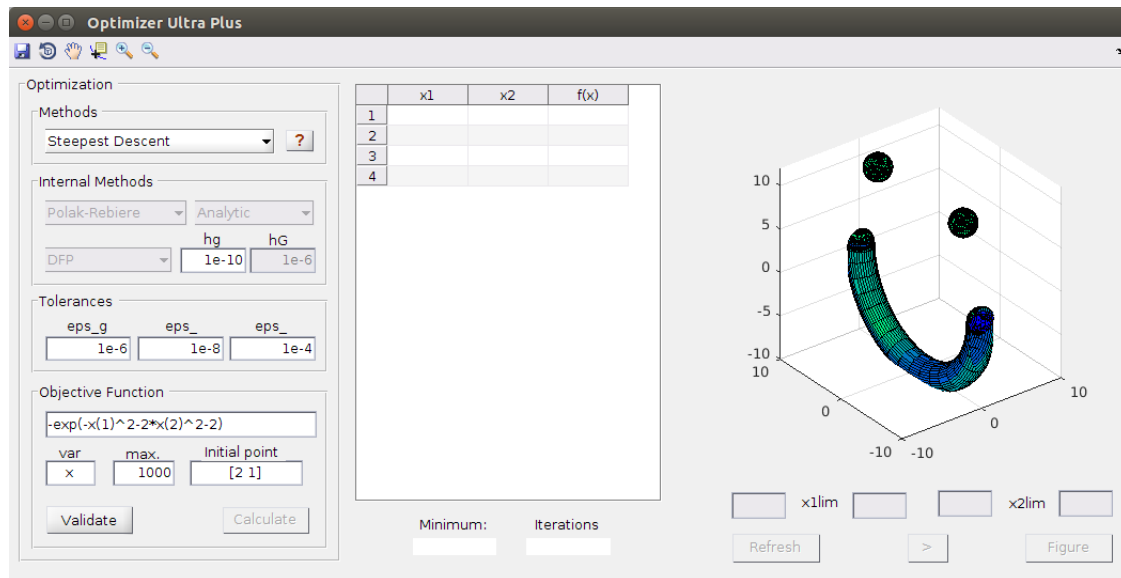


Figura 1: Interface Gráfica

1.6 Resultados

Neste relatório, apresenta-se o resultado da convergência em cada um dos cinco métodos para a mesma função, presente na equação 3. As tolerâncias nos critérios de parada utilizados para gerar os resultados estão descritos na tabela 1. Em todos os casos, o ponto inicial utilizado foi $x_0 = [-2 \ 1]$.

$$f = (1 - x_1)^2 + 2(x_1 - x_2^2)^2 \quad (3)$$

Função de Rosenbrock

Tolerância	Valor
Relativa (ϵ_r)	1×10^{-4}
Absoluta (ϵ_a)	1×10^{-8}
Gradiente (ϵ_g)	1×10^{-6}

Tabela 1: Tolerâncias para os critérios de parada

Outras funções podem ser testadas utilizando a interface gráfica anexa a este relatório, conforme demonstrado na apresentação do trabalho em sala de aula.

2 Método do Gradiente

2.1 Descrição

O método do Gradiente, também chamado de método de descida máxima (MDM), utiliza o conceito de gradiente de uma função vetorial para determinar a direção de avanço. Como o vetor gradiente aponta para a direção de máximo crescimento da função vetorial f , o vetor contrário ao gradiente aponta para a direção de máximo decréscimo dessa função.

Para encontrar o vetor gradiente de uma função vetorial f deve-se realizar o cálculo analítico ou numérico das derivadas parciais da função para cada variável independente em determinado. Ou seja,

Dado $f: \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciável no ponto $x^0 \in \mathbb{R}^n$:

$$\begin{bmatrix} f'_{x_1}(x^0) \\ f'_{x_2}(x^0) \\ \vdots \\ f'_{x_n}(x^0) \end{bmatrix} = g(x^0) = \nabla f(x^0) \in \mathbb{R}^n \quad (4)$$

A direção de descida para o algoritmo é dada de forma iterativa utilizando o vetor unitário da direção contrária ao gradiente.

$$d^k = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|} \quad (5)$$

Uma vez definida a direção, utiliza-se o método de Brent de busca linear nesta direção para encontrar o passo que realiza a maior descida. Ou seja, o novo ponto x^{k+1} irá depender de d^k , x^k e um α . Como a direção de descida d^k já foi calculada, e o ponto anterior x^k é conhecido, o novo ponto x^{k+1} depende apenas de α .

$$x^{k+1} = x^k + \alpha d^k = x^{k+1}(\alpha) \quad (6)$$

Dessa forma, podemos escrever o problema da busca linear da seguinte forma:

$$\alpha^k = \min\{\tilde{f}(\alpha) = f(x^k + \alpha d^k)\} \quad (7)$$

Com os três parâmetros conhecidos, é possível calcular novos pontos através de 6 até o algoritmo atingir um determinado critério de parada.

O fluxograma do algoritmo completo pode ser encontrado em 10.

Observação: É importante notar que utilizando o método do gradiente, duas direções de descida consecutivas $[d^k \ d^{k+1}]$ são sempre perpendiculares.

2.2 Resultado

O método conseguiu encontrar um mínimo em 11 iterações como pode ser visto na tabela 2. Conforme esperado, as direções de descida estão sempre perpendiculares conforme pode ser observado na figura 2.

Iter	$x(1)$	$x(2)$	$f(x)$
0	-2.0000	1.0000	27.0000
1	-0.7602	-0.6531	5.9148
2	0.5570	0.3348	0.5922
3	0.4075	0.5341	0.3809
4	1.0287	1.0000	0.0025
5	1.0201	1.0114	0.0004
6	1.0050	1.0000	0.0001
7	1.0035	1.0020	0.0000
8	1.0009	1.0000	0.0000
9	1.0006	1.0003	0.0000
10	1.0001	1.0000	0.0000
11	1.0001	1.0001	0.0000

Tabela 2: Tabela de Resultados: Método do Gradiente

3 Método do Gradiente Conjugado

3.1 Descrição

O método do gradiente conjugado (MGC) é muito semelhante ao método do gradiente porém tem o objetivo de corrigir problemas de convergência lenta do MDM para problemas mal condicionados. No novo algoritmo, a nova direção de descida é calculada através de uma ponderação entre a direção contrária ao gradiente e a direção de descida calculada na iteração anterior.

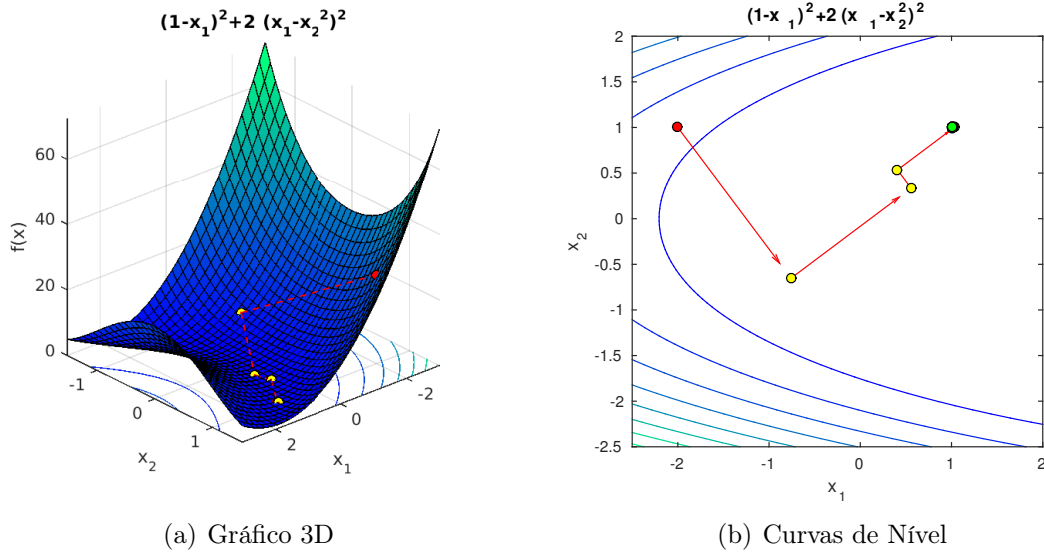


Figura 2: Método do Gradiente

$$d^{k+1} = -\nabla f(x) + \beta_k d^k \quad (8)$$

Essa mudança impede que a direção de descida se mova sempre ortogonal, tornando a evolução do algoritmo mais suave.

O cálculo de β pode ser feito utilizando dois métodos: *Polak-Rebière* ou *Fletcher-Reeves* conforme as equações (9) e (10).

$$\beta^k = \frac{(g^{k+1})^T (g^{k+1} - g^k)}{g^{kT} g^k} = \frac{\|g^{k+1}\| - [g^{k+1}]^T g^k}{\|g^k\|} \quad (9)$$

Polak-Rebière

$$\beta^k = \frac{(g^{k+1})^T g^k}{g^{kT} g^k} = \frac{\|g^{k+1}\|}{\|g^k\|} \quad (10)$$

Fletcher-Reeves

O restante do algoritmo segue os mesmos passos do MDM conforme mostrado no fluxograma 11.

3.2 Resultado

Durante a execução foi utilizado o método de *Polak-Rebière*. O Gradiente Conjugado conseguiu encontrar um mínimo em 7 iterações como pode ser visto na tabela 3. Observa-se que o método encontrou um mínimo diferente do MDM pois possui uma etapa que suaviza a direção de descida impedindo possibilitando direções não perpendiculares como pode ser observado na figura 3.

Iter	$x(1)$	$x(2)$	$f(x)$
0	-2.0000	1.0000	27.0000
1	-0.7602	-0.6531	5.9148
2	0.3566	-0.2932	0.5604
3	0.7883	-0.9225	0.0527
4	1.0059	-1.0115	0.0006
5	1.0094	-1.0052	0.0001
6	1.0015	-0.9998	0.0000
7	0.9999	-0.9999	0.0000

Tabela 3: Tabela de Resultados: Método do Gradiente Conjugado

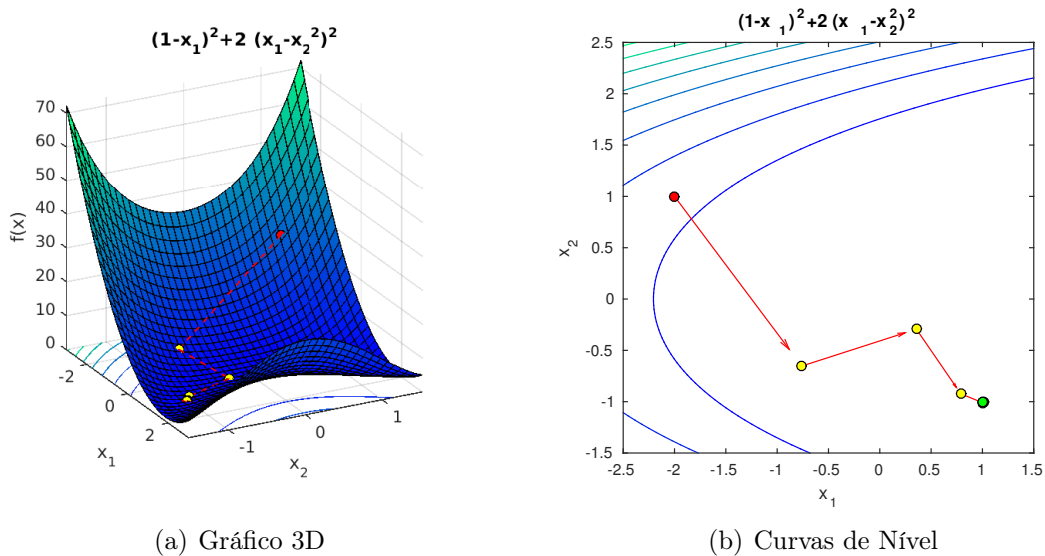


Figura 3: Método do Gradiente Conjugado

4 Método de Newton

4.1 Descrição

Os métodos de Newton são baseados na minimização de funções quadráticas, sendo a função $f(x)$ aproximada por uma quadrática a cada iteração. Como o mínimo de funções quadráticas é bem conhecido, segue que este mínimo é utilizado como passo para a nova iteração.

Para executar o método de Newton é necessário conhecimento do vetor gradiente $g(x)$ e da matriz Hessiana $G(x)$. Ambos podem ser obtidos analiticamente ou numericamente.

Para encontrar a matriz Hessiana de uma função vetorial f deve-se realizar o cálculo analítico ou numérico das derivadas parciais do gradiente para cada variável independente em determinado ponto. Ou seja,

Dado $f: \mathbb{R}^n \rightarrow \mathbb{R} \in \mathbb{C}^2$ e $x^0 \in \mathbb{R}^n$:

$$\begin{bmatrix} f'_{x_1 x_1}(x^0) & f'_{x_1 x_2}(x^0) & \dots & f'_{x_1 x_n}(x^0) \\ f'_{x_2 x_1}(x^0) & f'_{x_2 x_2}(x^0) & \dots & f'_{x_2 x_n}(x^0) \\ \vdots & & \ddots & \\ f'_{x_n x_1}(x^0) & f'_{x_n x_2}(x^0) & \dots & f'_{x_n x_n}(x^0) \end{bmatrix} = \nabla^2 f(x^0) = H(x^0) = G(x^0) \quad (11)$$

Utilizando os conhecimentos prévios de funções quadráticas é fácil notar que o mínimo para funções do tipo $f(x) = \frac{1}{2}x^T A x + x^T b$ com $A > 0$ é $x^* = -A^{-1}b$. A série de Taylor também fornece uma aproximação de segunda ordem para funções vetoriais conforme a equação (13).

$$f_{quad}(x) = f(x^k) + (x - x^k)^T \nabla f(x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k) \quad (12)$$

$$= f(x^k) + (x - x^k)^T g(x^k) + \frac{1}{2}(x - x^k)^T G(x^k)(x - x^k) \quad (13)$$

Aproximação de segunda ordem por Taylor

A aproximação quadrática de $f(x)$ para $G(x) > 0$ possui mínimo em $(x - x^k)^* = -[G^k]^{-1}g^k$. Dessa forma, o ponto para nova é atualizado através de (15).

$$x^{k+1} = x^k + d^k \quad (14)$$

$$d^k = -[G^k]^{-1}g^k \quad (15)$$

Uma vez definida a regra para atualização de x^k basta executar o algoritmo de forma iterativa conforme o fluxograma 12.

4.2 Resultado

Durante a execução do algoritmo foi utilizado o cálculo analítico (simbólico) do Gradiente e da Hessiana da função vetorial. O método encontrou o mínimo em apenas uma iteração como pode ser visto na tabela 4. Isso acontece pois a direção de descida é dado por uma aproximação de segunda ordem da função que nesse caso se encaixa corretamente com a função. Esse resultado pode ser observado graficamente na figura 4.

Iter	$x(1)$	$x(2)$	$f(x)$
0	-2.0000	1.0000	27.0000
1	1.0000	1.0000	0.0000

Tabela 4: Tabela de Resultados: Método de Newton

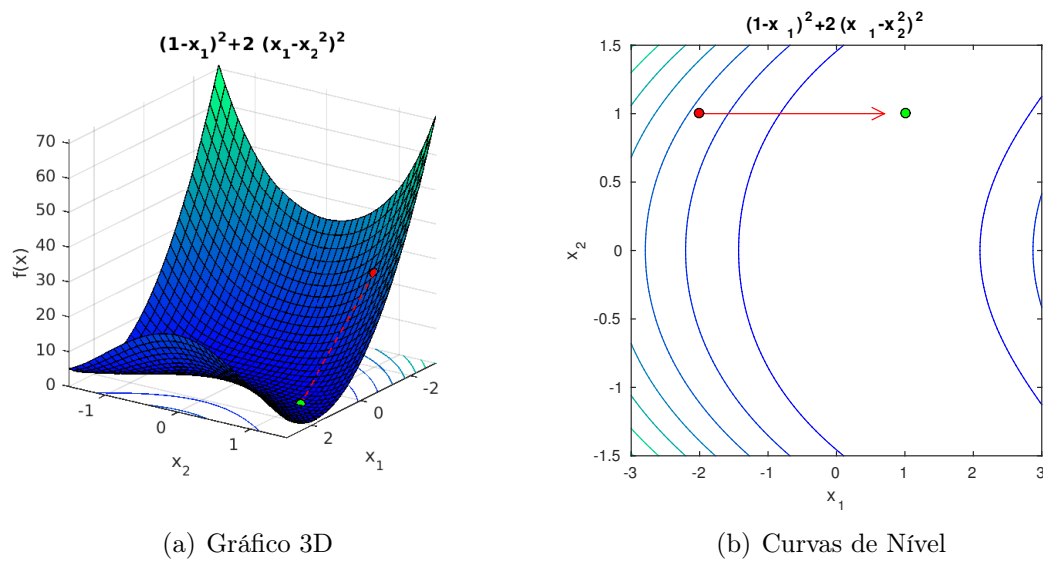


Figura 4: Método de Newton

5 Método de Newton Modificado

5.1 Descrição

O método de Newton modificado foi criado para contornar problemas de convergência quando a Hessiana da função objetivo não é definida positiva. O algoritmo verifica se a Hessiana é positiva definida ou não em determinado ponto. Se ela for definida positiva, o algoritmo segue de forma equivalente ao método de Newton. Caso contrário, é feita uma correção nos autovalores da Hessiana somando uma matriz identida vezes um γ conforme a equação (16).

$$F^k = G^k + \gamma \mathbb{I}_n \quad (16)$$

Correção da Hesssiana

Se a Hessiana for positiva definida $\gamma = 0$, porém se ela não for positiva definida, γ deverá ser grande o suficiente tal que todos os autovalores de F sejam maiores que 0, tornando a Hessiana positiva definida. Ou seja:

$$\gamma \in \mathbb{R} | \lambda_i(G^k + \gamma \mathbb{I}_n) > 0 \quad (17)$$

Com isso, a direção de descida passa a ser $d^k = -[F]^{-1}g(k)$

O restante do algoritmo funciona da mesma forma conforme pode ser observado no fluxograma 13.

5.2 Resultado

Da mesma forma que o Método de Newton, o Método de Newton Modificado também encontra o mínimo com apenas uma iteração como pode ser visto na tabela 5. Esse resultado era esperado pois a modificação feita no método tem o objetivo de contornar problemas de convergência apenas. Em situações onde o Método de Newton converge, o modificado irá convergir da mesma forma.

Iter	$x(1)$	$x(2)$	$f(x)$
0	-2.0000	1.0000	27.0000
1	1.0000	1.0000	0.0000

Tabela 5: Tabela de Resultados: Método de Newton Modificado

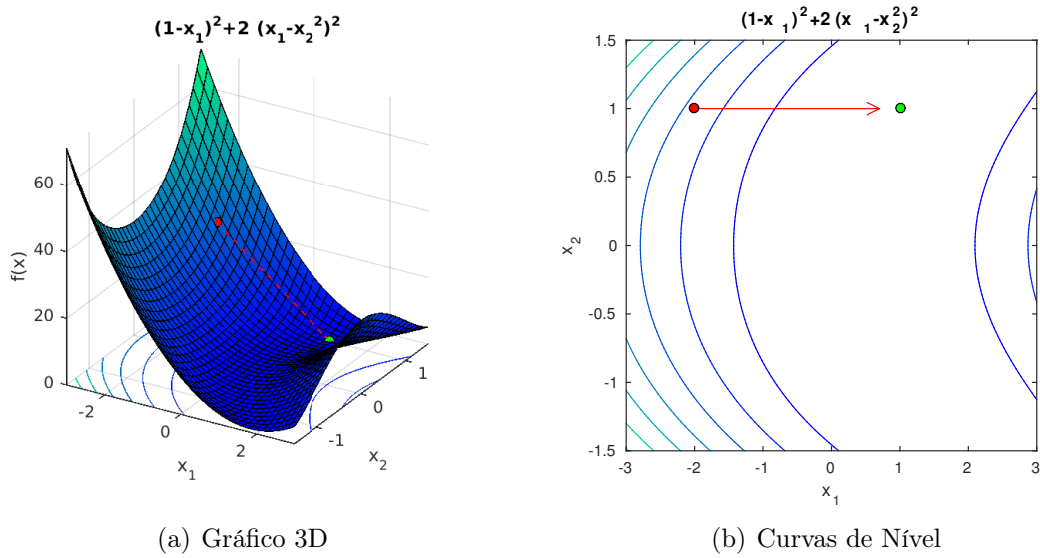


Figura 5: Método de Newton Modificado

6 Método de Quasi-Newton

6.1 Descrição

O objetivo do método de Quasi-Newton é facilitar o cálculo da inversa da Hessiana, uma vez que derivar duas vezes a função f e ainda fazer a inversão da matriz Hessiana são tarefas extremamente custosas.

$$H = [G^k]^{-1} \quad (18)$$

Portanto, nesse método, são utilizadas aproximação para a inversa da Hessiana. Nesse contexto, duas aproximações são propostas: A de *Davidon-Fletcher-Powell* (DFP) e a de *Broyden-Fletcher-Goldfarb-Shanno* (BFGS).

$$H^{k+1} = H^k - \frac{H\gamma\gamma^T H}{\gamma^T \gamma} + \frac{\delta\delta^T}{\delta^T \gamma} \quad (19)$$

Davidon-Fletcher-Powell

$$H^{k+1} = H^k - \frac{\delta\gamma^T H + H\gamma\delta^T}{\delta^T \gamma} + \left(1 + \frac{\gamma^T H \gamma}{\delta^T \gamma}\right) \frac{\delta\delta^T}{\delta^T \gamma} \quad (20)$$

Broyden-Fletcher-Goldfarb-Shanno

Onde $\delta^k = x^{k+1} - x^k$ e $\gamma^k = g^{k+1} - g^k$

O restante do algoritmo funciona de forma equivalente ao método de Newton conforme explicitado no fluxograma 14.

6.2 Resultado

Durante a execução do algoritmo foi utilizado método de DFP para o cálculo aproximado da Hessiana. O método encontrou o mínimo em 6 iterações como pode ser visto na tabela 6. Isso acontece devido ao cálculo aproximado da Hessiana que dificulta a entrada no critério de parada como pode ser visto na figura 6.

Iter	$x(1)$	$x(2)$	$f(x)$
0	-2.0000	1.0000	27.0000
1	-0.7602	-0.6531	5.9148
2	0.3566	-0.2932	0.5604
3	0.9398	-0.9812	0.0047
4	0.9757	-0.9815	0.0009
5	0.9998	-1.0000	0.0000
6	1.0000	-1.0000	0.0000
7	1.0000	-1.0000	0.0000

Tabela 6: Tabela de Resultados: Método de Quasi-Newton

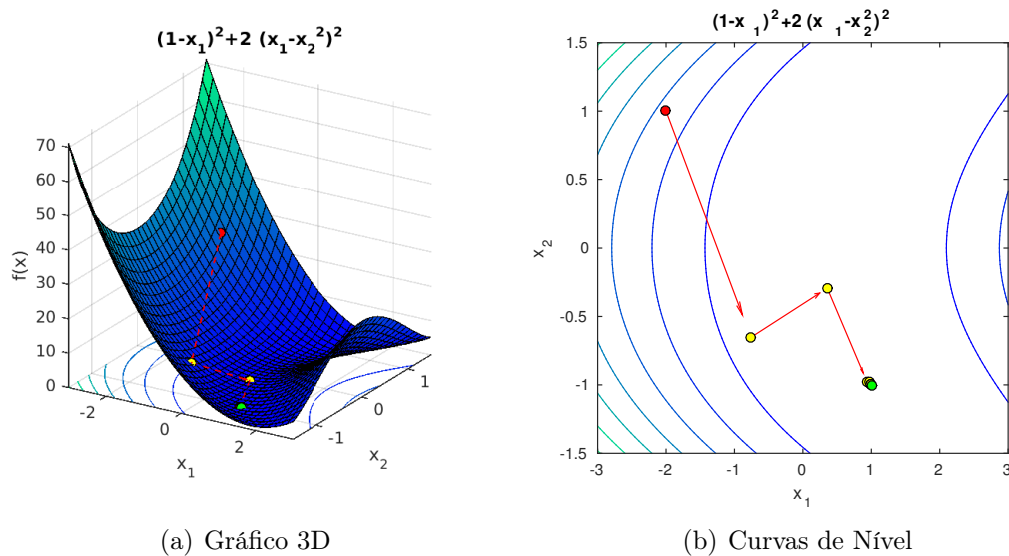


Figura 6: Método de Quasi-Newton

7 Resultados Complementares

7.1 Outras funções de duas variáveis

Além dos resultados apresentados para comparação entre os métodos, foram feitos também testes em outras funções. Esses testes apresentam resultados gráficos interessantes permitindo melhor visualização do funcionamento dos métodos.

Um dos exemplos executados durante a apresentação do trabalho foi a utilização da Gradiente Conjugado para a função (21).

$$f = -e^{(-x_1^2 - 2x_2^2 - 2)} \quad (21)$$

Função Exponencial

O resultado mostrado na figura 7 descreve claramente como o gradiente conjugado suaviza a direção de descida do gradiente seguindo uma trajetória curva.

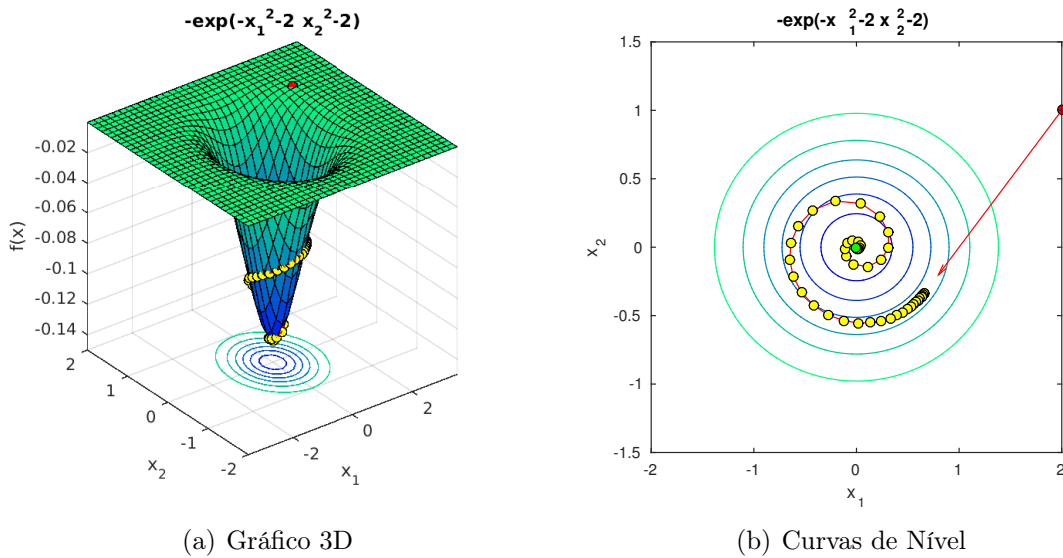


Figura 7: Gradiente Conjugado aplicado a função exponencial

Outro resultado interessante obtido com o Gradiente Conjugado aparece ao aplicar o método à função (22).

$$f = \sin(x_1)^2 + \sin(x_2)^2 \quad (22)$$

Função Seno

O resultado apresentado na figura 8 mostra que a função possui infinitos mínimos locais. O algoritmo naturalmente encontra o mínimo mais próximo do ponto inicial.

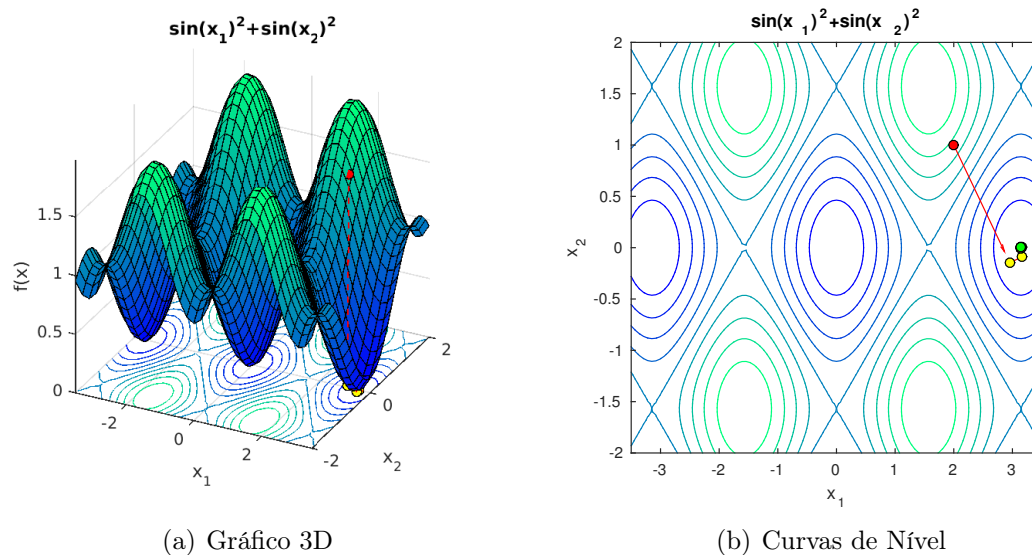


Figura 8: Gradiente Conjugado aplicado a função seno

Outro resultado interessante para a função seno é que, por não ser uma função positiva definida, a Hessiana por vezes não necessariamente garante a convergência para um mínimo. Pelo método de Newton, que utiliza a Hessiana pura, podemos convergir para um ponto de máximo ou de sela. No entanto, usando o método de Newton-Modificado ou o Quase-Newton, conseguimos contornar esse problema garantindo uma "Hessiana" sempre positiva definida.

7.2 Funções com mais de duas variáveis

Durante todo o relatório foram apresentados apenas resultados para funções bivariáveis. O motivo é que para essas funções a exibição dos resultados é muito mais agradável.

Entretanto, a interface criada permite também a utilização de funções como mais de duas variáveis, apresentando novamente a tabela com os resultados da otimização. Já os gráficos não podem ser gerados como anteriormente portanto foi utilizada a abordagem de mostrar o decaimento da função objetivo e do módulo do gradiente no tempo (iterações).

O módulo do gradiente indo para zero indica que um ponto de extremo foi atingido. Dessa forma, avaliando os dois gráficos é possível identificar se a função foi para algum ponto de mínimo.

A função utilizada para teste foi a função exponencial (23) com três variáveis. Foi utilizado o Método do Gradiente com ponto inicial $x_0 = [2 \ 1 \ 1]$.

$$f = -e^{(-x_1^2 - 2x_2^2 - x_3^2 - 2)} \quad (23)$$

Função Exponencial

O método encontrou o mínimo em 4 iteração como pode ser visto na tabela 7. O resultado gráfico para o algoritmo, conforme descrito anteriormente, mostra o valor da função objetivo e o módulo do gradiente. Analisando a figura 9 é possível ver claramente a função decaindo com o número de iterações até o módulo do gradiente chegar a zero.

Iter	$x(1)$	$x(2)$	$x(3)$	$f(x)$
0	2.0000	1.0000	1.0000	-0.0001
1	0.6154	-0.3846	0.3077	-0.0627
2	0.2198	0.1099	0.1099	-0.1244
3	0.0676	-0.0423	0.0338	-0.1341
4	0.0242	0.0121	0.0121	-0.1352
5	0.0074	-0.0046	0.0037	-0.1353

Tabela 7: Tabela de Resultados: Método do Gradiente para três variáveis

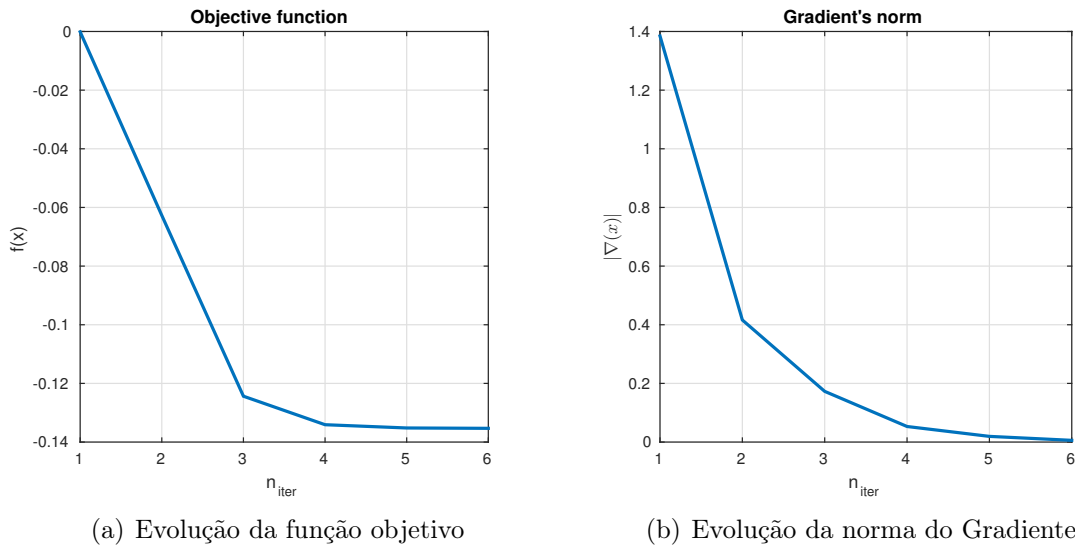


Figura 9: Método do Gradiente para três variáveis

8 Fluxogramas

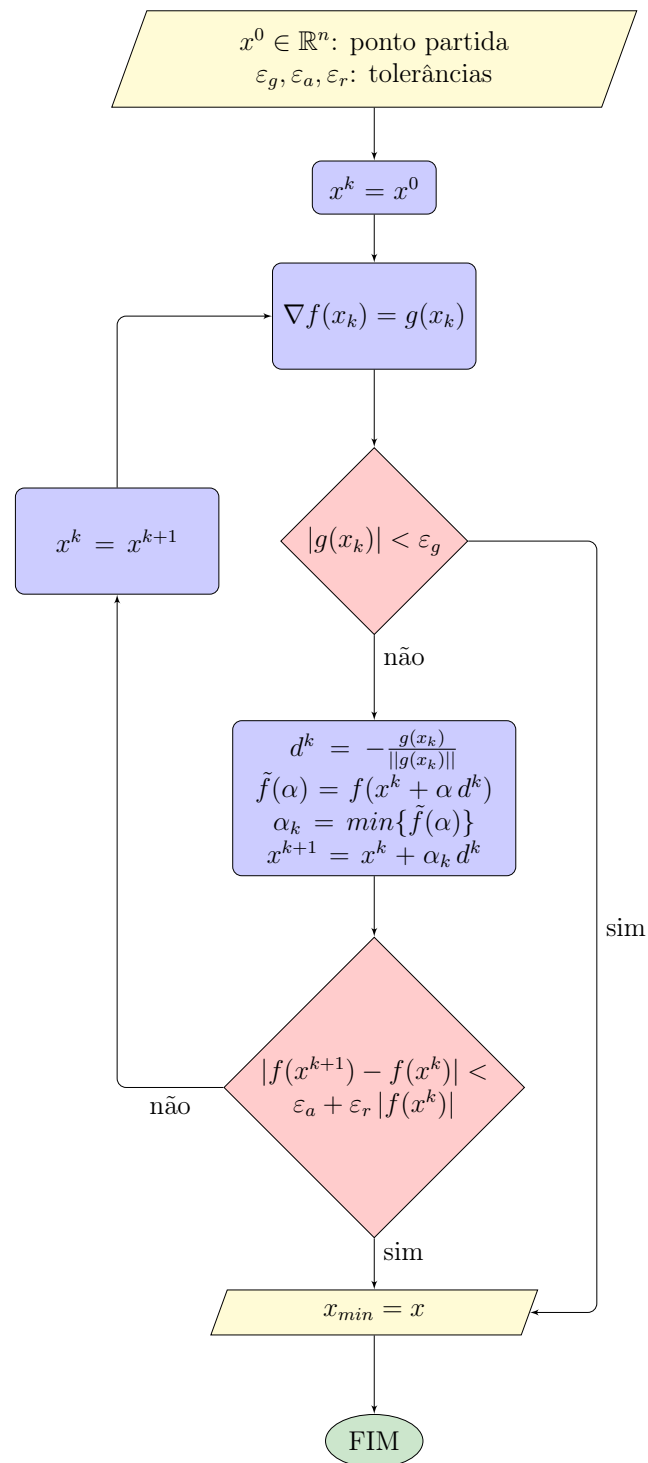


Figura 10: Método de Descida Máxima (Gradiente)

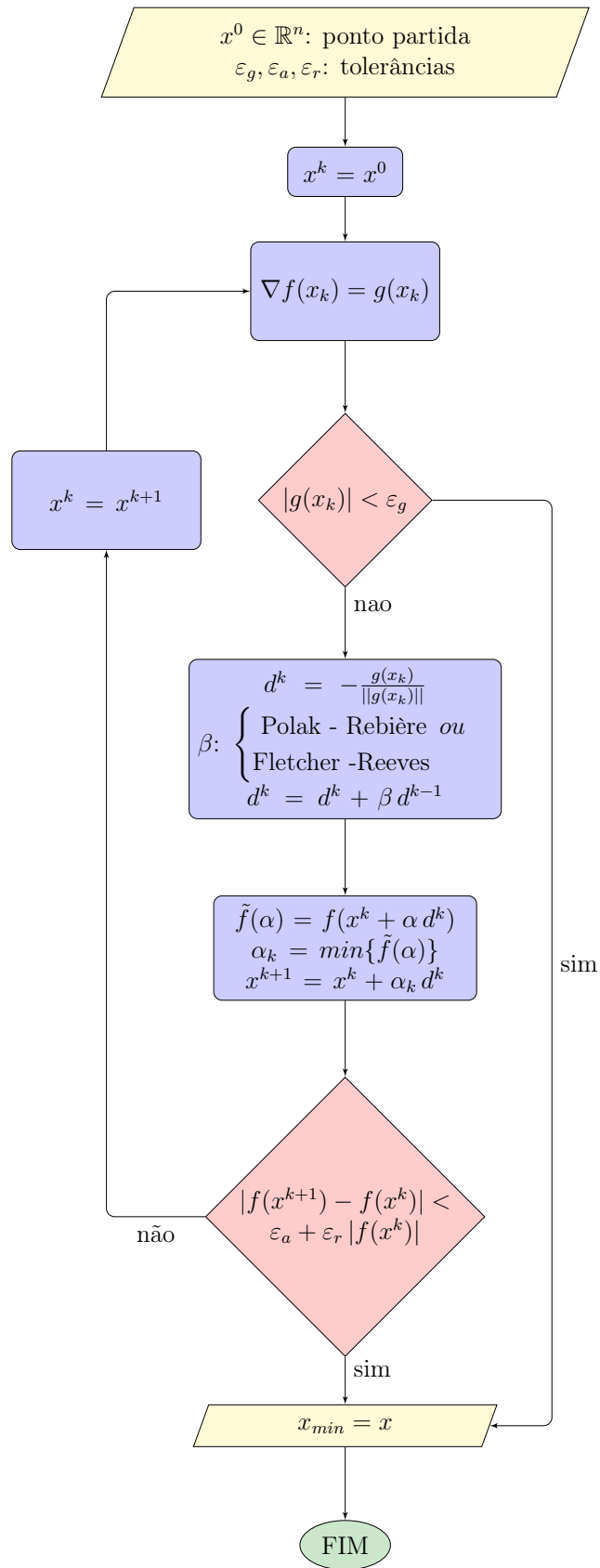


Figura 11: Método do Gradiente Conjugado

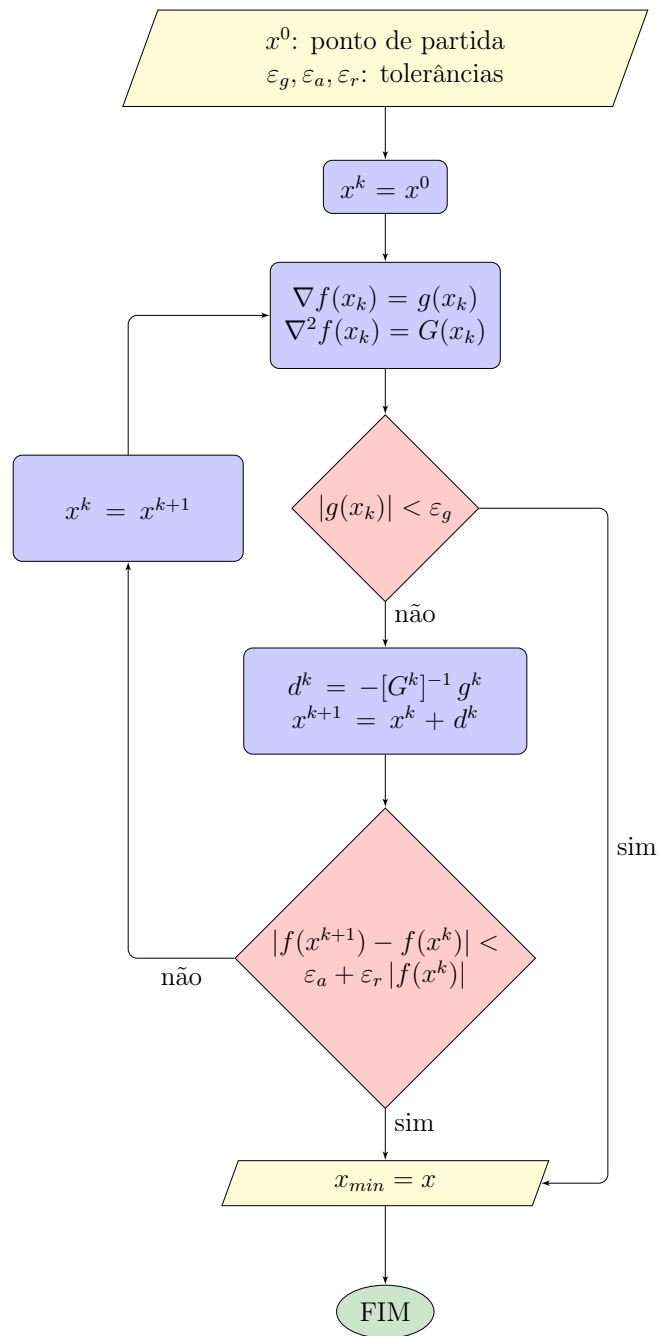


Figura 12: Método de Newton

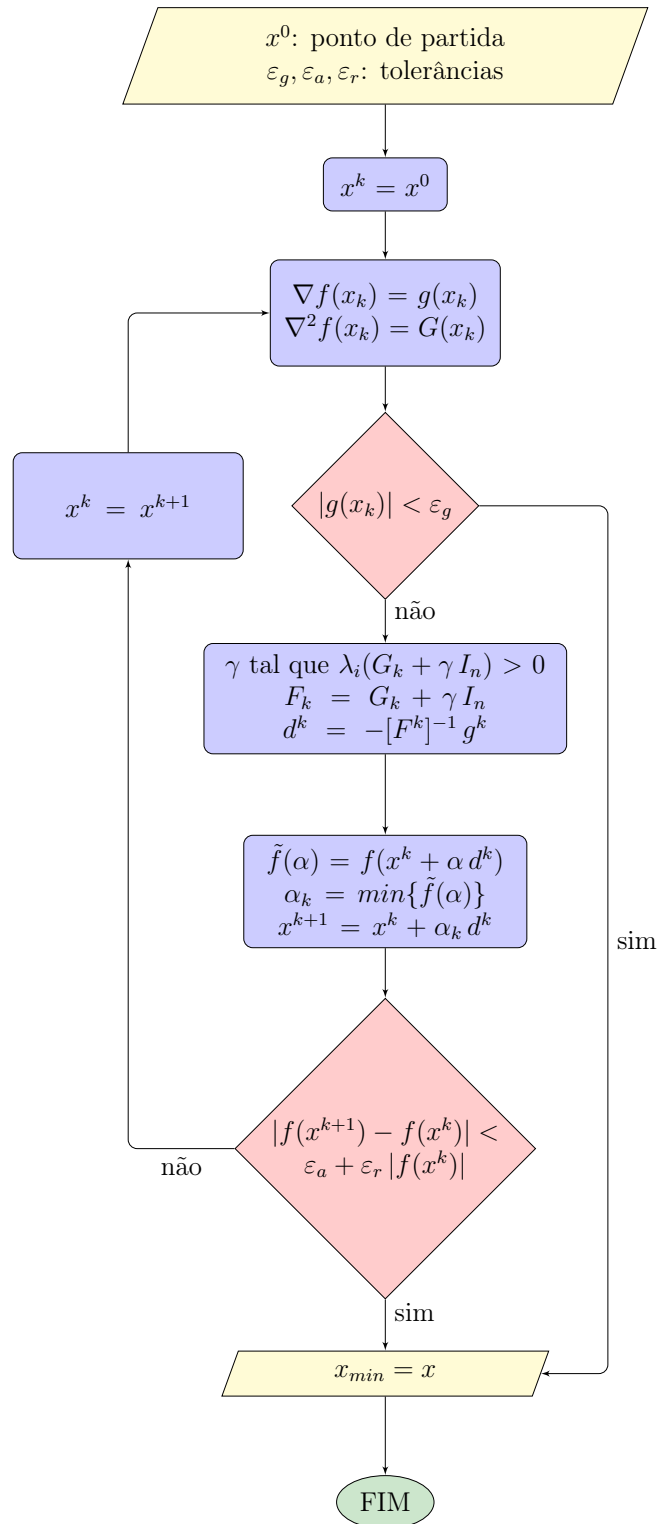


Figura 13: Método de Newton Modificado

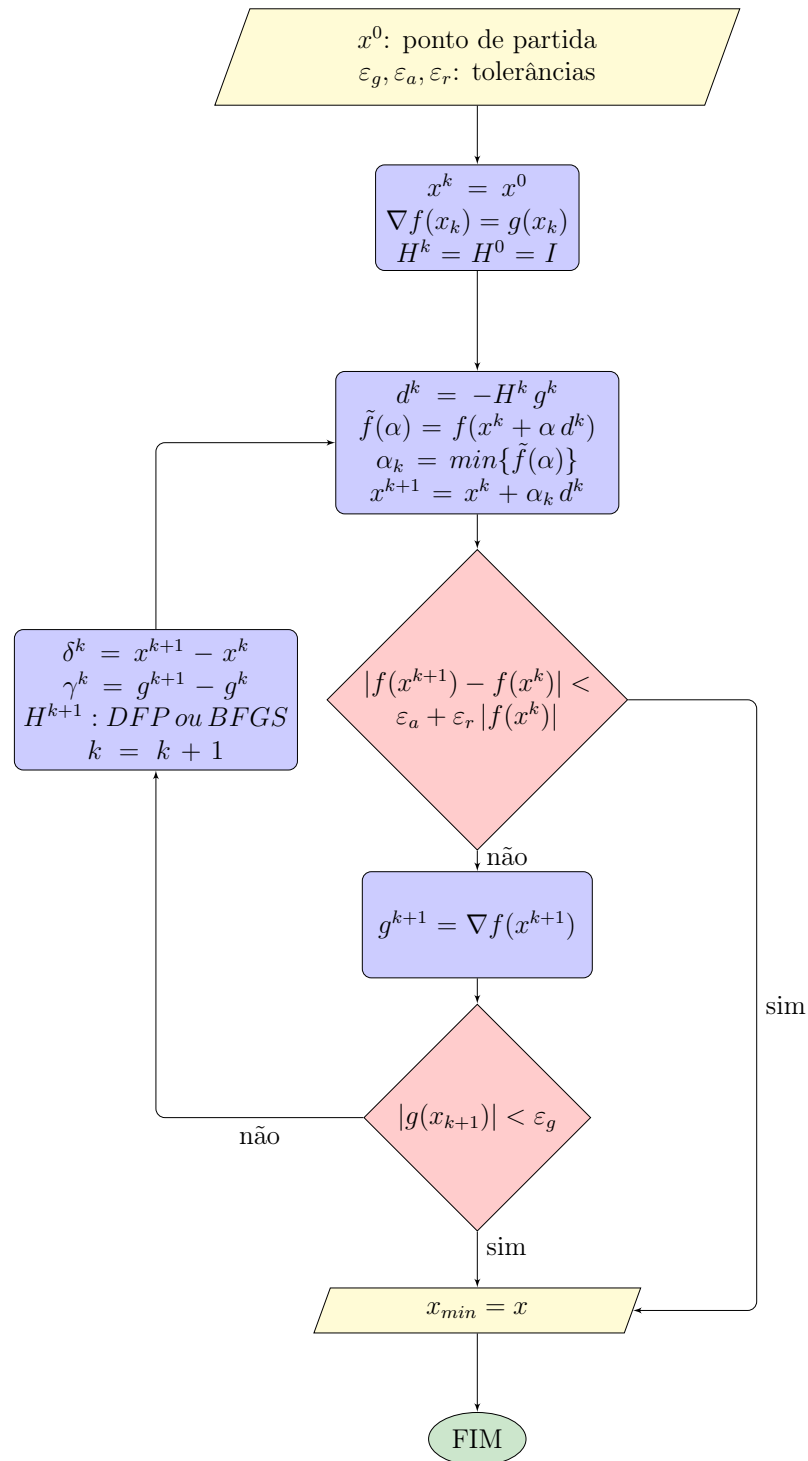


Figura 14: Método de Quase-Newton