



Trabalho 2

INTRODUÇÃO À OTIMIZAÇÃO

Alunos:

Cayo Valsamis

Gabriel Pelielo

Rafael Accácio

Rodrigo Moysés

Professor:

Afonso Celso del Nero

13 de Fevereiro de 2016

Conteúdo

1	Introdução	2
2	Método da Descida Máxima	2
3	Método do Gradiente Conjugado	8
4	Método de Newton	8
5	Método de Newton Modificado	9
6	Método de Quase Newton	11
7	Conclusão	13

1 Introdução

O trabalho detalhado neste relatório se baseia em analisar métodos numéricos para realizar busca de mínimos de funções vetoriais. Foram implementados cinco métodos diferentes de localização de mínimos:

- Método da Descida Máxima (ou gradiente)
- Método do Gradiente Conjugado
- Método de Newton
- Método de Newton Modificado
- Método de Quase Newton

Todos os métodos foram implementados na plataforma *MATLAB*, através de um programa de interface gráfica usado para escolher o método desejado e inserir alguns valores necessários para executar a busca. O objetivo do trabalho foi comparar esses métodos de acordo com os quesitos de tempo de execução, número necessário de iterações e por fim qualificá-los de acordo com cada função inserida.

2 Método da Descida Máxima

O método de Descida máxima, ou método do gradiente, se baseia em um método muito simples para encontrar o mínimo de uma função. A partir de um ponto inicial calcula-se a direção de maior de crescimento, de descida máxima ou sentido inverso do gradiente, que dão nome ao método, e faz-se um avanço nessa mesma direção.

Como sabemos, o gradiente de uma função representa sua direção de maior crescimento. Sabendo disso utiliza-se a direção oposta, mas a questão que fica é qual o avanço é necessário. Para resolver isso, a direção do gradiente foi normalizada, d^k , e criou-se uma variável, α , que indica o avanço a ser feito.

Com o gradiente calculado naquele ponto, usa-se um método de minimização unidimensional qualquer para minimizar a função $x^k - \alpha d^k$. Foi escolhido o método da seção áurea, devido a sua grande flexibilidade e por encontrar o mínimo com poucas iterações e baixo esforço computacional. Após a minimização utilizamos o valor de α que a minimiza como avanço.

Dessa forma faz-se a iteração para encontrar o próximo ponto, que a princípio está mais perto do mínimo. E assim faz-se até convergir ao ponto de mínimo da função (considerando que a mesma o possui).

Como métodos de parada foram escolhidos o tamanho da norma do vetor diferença de gradiente em cada iteração, a norma entre a diferença entre os pontos em cada iteração e finalmente o número de iterações.

Para verificar a convergência do método vemos como são as direções de avanço do algoritmo e também sua norma. Sabe-se a partir de [?] que cada iteração tem direção perpendicular a anterior e que a convergência dá-se quando cada avanço é menor que o anterior, assim como vemos na figura 1

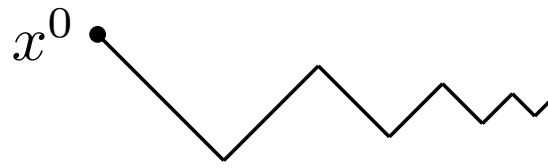


Figura 1: Representação do avanço convergente em zig-zag com ângulos retos.

Lei de Iteração:

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

Após construir o algoritmo, foram feitos testes usando algumas funções:

- $f_1(x, y) = x^2 + y^2$
- $f_2(x, y) = -e^{-x^2 - y^2}$
- $f_3(x, y) = \cos\left(\frac{xy}{5}\right) + \sin\left(\frac{xy}{5}\right)$
- $f_4(x, y) = |x + y|$

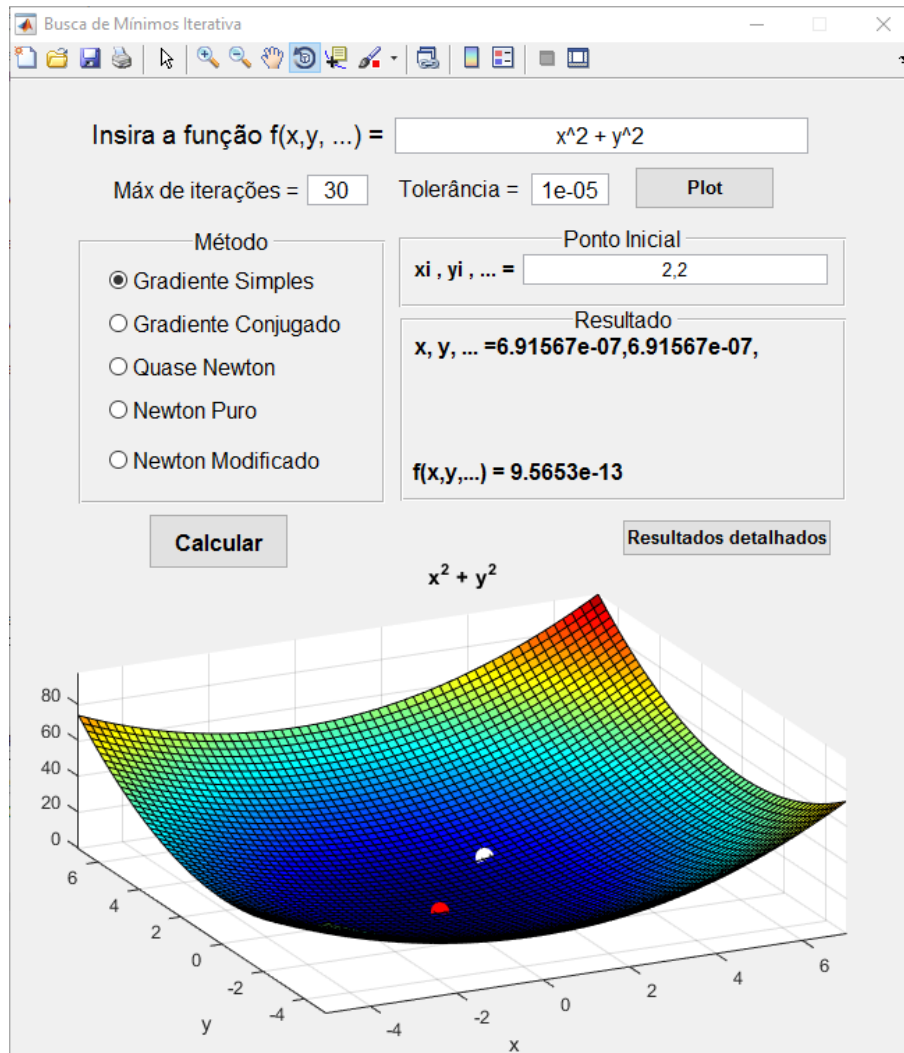


Figura 2: Janela de inicialização de $f_1(x, y)$

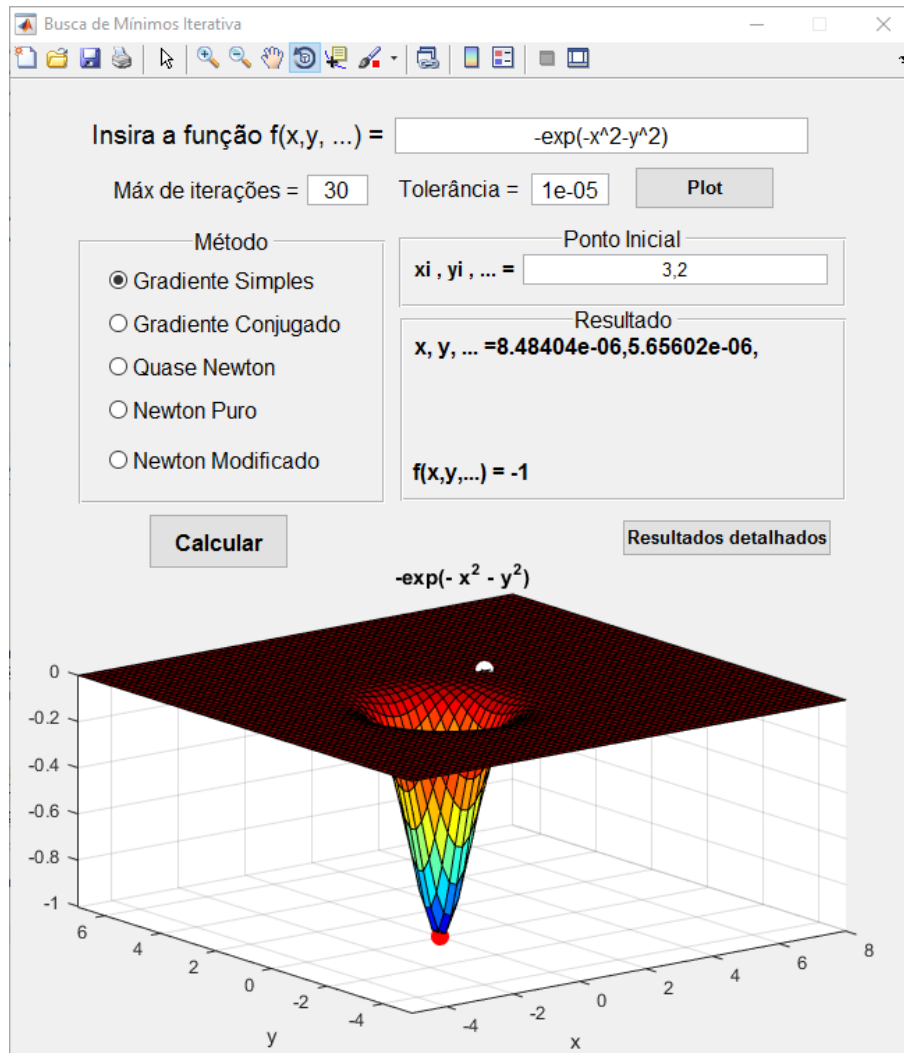


Figura 3: Janela de inicialização de $f_2(x,y)$

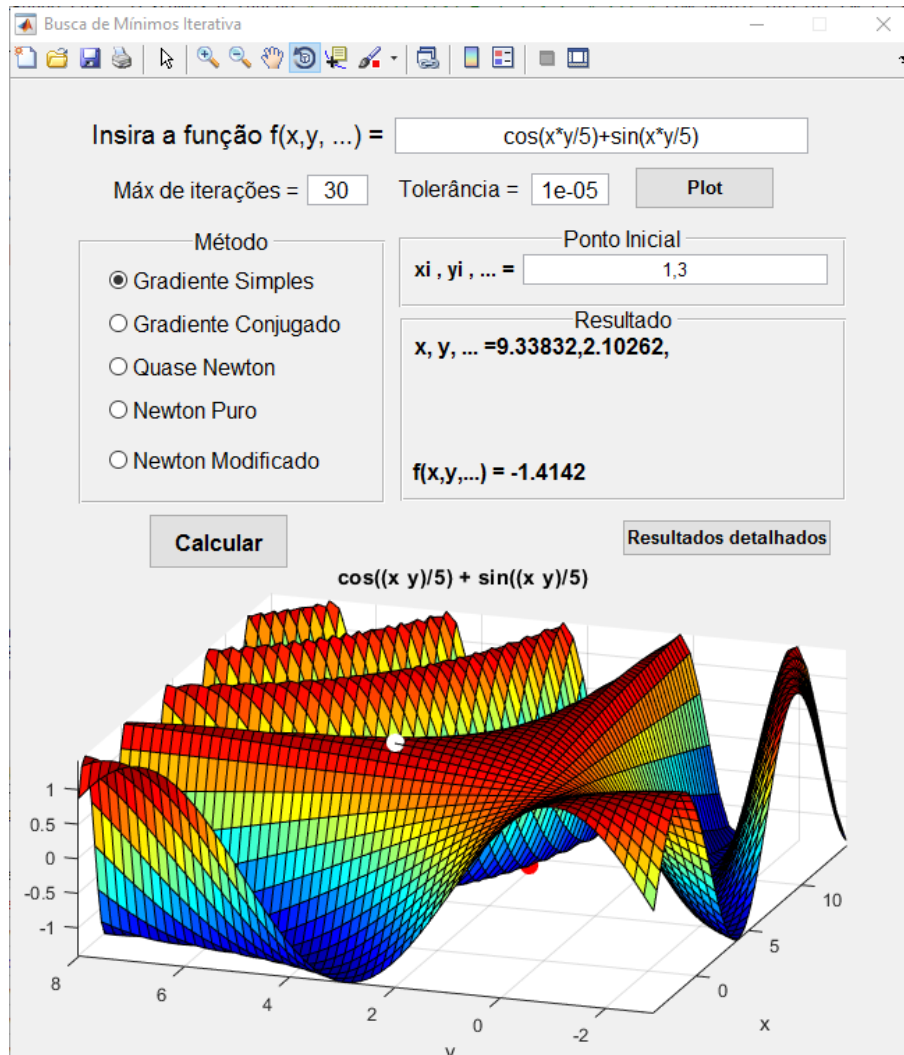


Figura 4: Janela de inicialização de $f_3(x,y)$

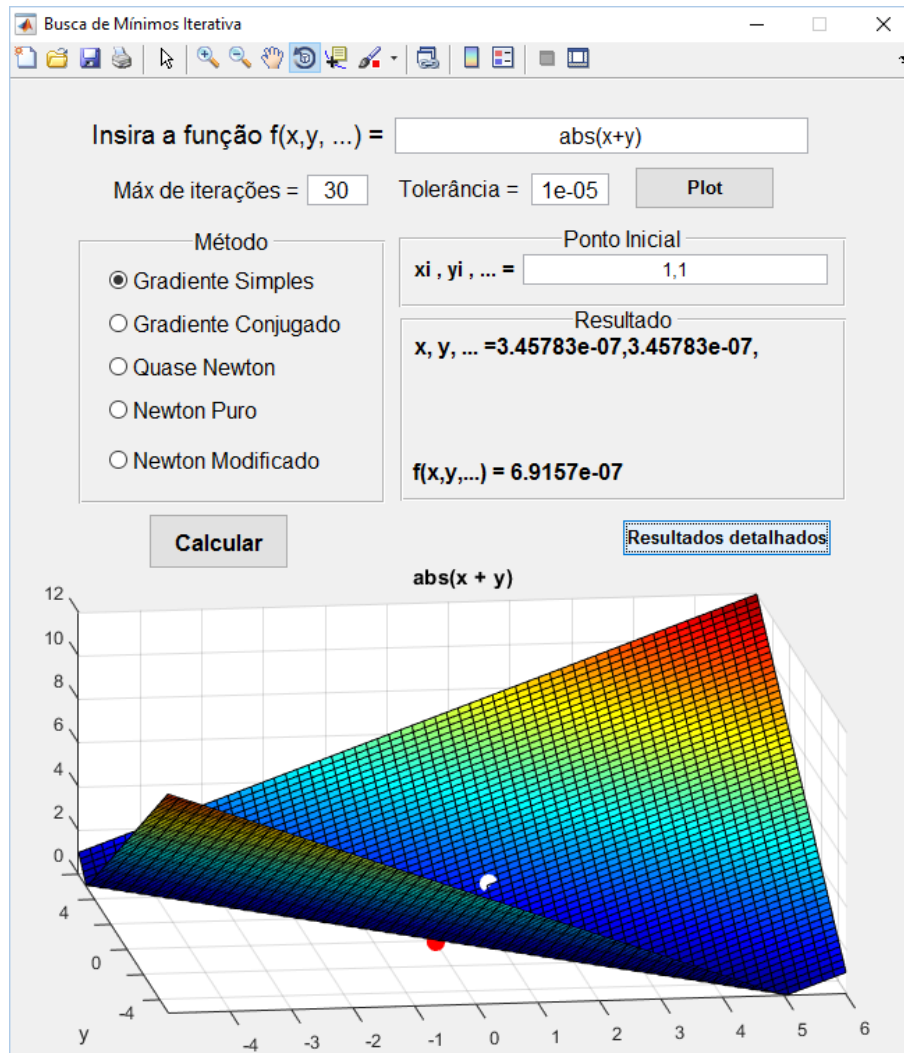


Figura 5: Janela de inicialização de $f_4(x,y)$

3 Método do Gradiente Conjugado

O método do Gradiente Conjugado nada mais é do que um ajuste sobre o método do Gradiente descrito na seção 2. As mudanças de direção abruptas, conforme ilustrado na figura 1, são suavizadas com a adição do coeficiente de inércia β , que conserva uma fração da direção anterior. Assim, a direção de avanço toma a seguinte forma:

$$d^{k+1} = -g(x^{k+1}) + \beta_k d^k \quad (2)$$

A Lei de Iteração é a mesma do método do Gradiente:

$$x^{k+1} = x^k + \alpha_k d^k \quad (3)$$

$$\tilde{f}(\alpha) = f(x^k + \alpha d^k) \quad (4)$$

A função $\tilde{f}(\alpha)$ é minimizada utilizando a razão áurea.

Já o coeficiente β é uma relação entre o gradiente atual e o anterior.

Segundo *Polak-Rebière*:

$$\beta_k = \frac{\|g^{k+1}\| - [g^{k+1}]^T g^k}{\|g^k\|} \quad (5)$$

Com essa suavização das direções de avanço, é esperado que esse método convirga com menos iterações do que o método do Gradiente.

4 Método de Newton

Os métodos de Newton se baseiam em encontrar uma aproximação quadrática $q(x)$ a partir do teorema de Taylor para a função objetivo $f(x)$, e assim, encontrar o seu mínimo.

Lei de Iteração:

$$x^{k+1} = x^k - [G^k]^{-1} g^k \quad (6)$$

Teoricamente, a vantagem do método de Newton em relação aos outros é que, para funções quadráticas, ele converge em apenas uma iteração. Porém, um dos problemas do método é ele ser de segunda ordem, o que significa que ele depende tanto do valor da hessiana quanto do gradiente da função objetivo, e alguns casos, a convergência pode ser comprometida devido ao fato de que a hessiana em um dado ponto pode não ser positiva definida.

Depois de construído o algoritmo, foram testadas quatro funções para efeitos de comparação:

- $f_1(x, y) = x^2 + y^2$
- $f_2(x, y) = -e^{-x^2 - y^2}$
- $f_3(x, y) = \cos\left(\frac{xy}{5}\right) + \sin\left(\frac{xy}{5}\right)$
- $f_4(x, y) = |x + y|$

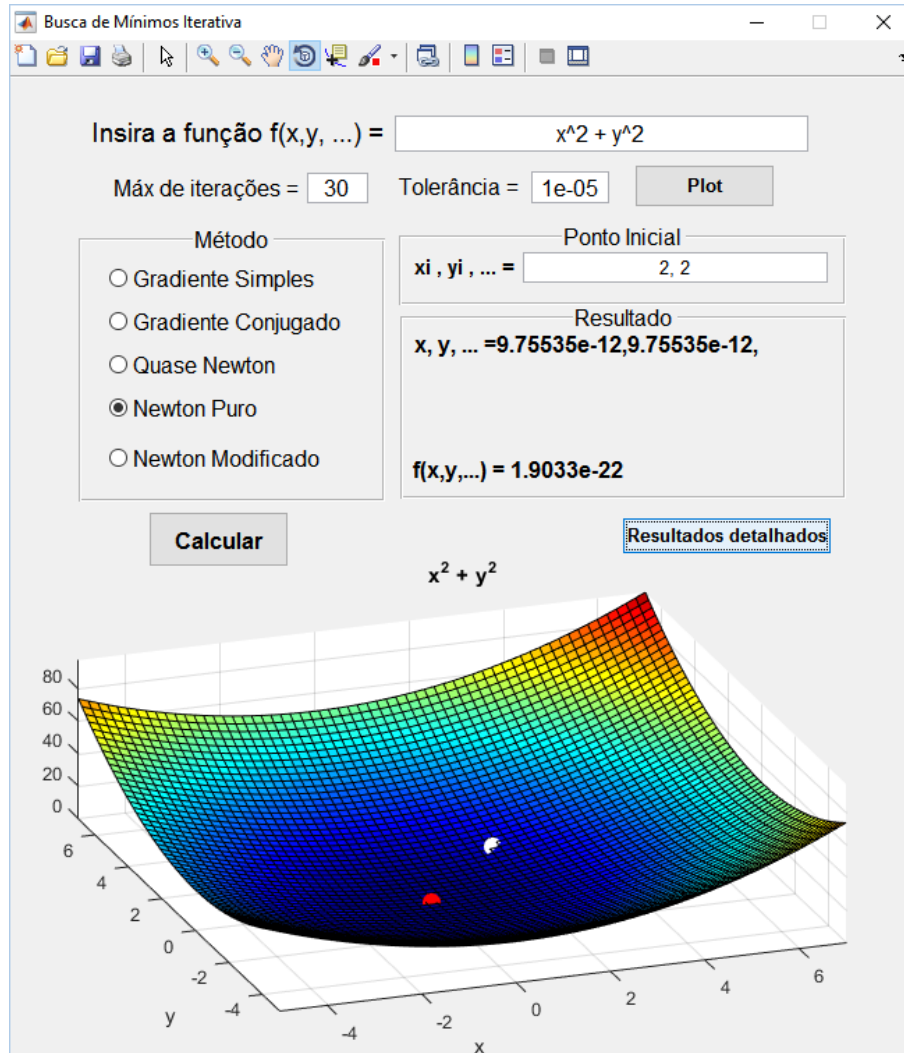


Figura 6: Janela de inicialização de $f_1(x, y)$

5 Método de Newton Modificado

Para melhorar os problemas de convergência do método de Newton, foram feitas algumas alterações, como diminuir o avanço em uma dada direção d^k , da forma:

$$x^{k+1} = x^k + \alpha_k d^k \quad (7)$$

onde α_k minimiza $\tilde{f}(\alpha) = f(x^k + \alpha d^k)$.

O problema resultante do sinal negativo da hessiana pode ser desviado usando um truque matricial segundo a equação 8.

$$F^k = G^k + \gamma I_n \quad (8)$$

Sendo I_n a matriz identidade de ordem n , é sempre possível encontrar γ tal que F_k tenha seus autovalores positivos e possa, assim, ser usada para gerar uma direção

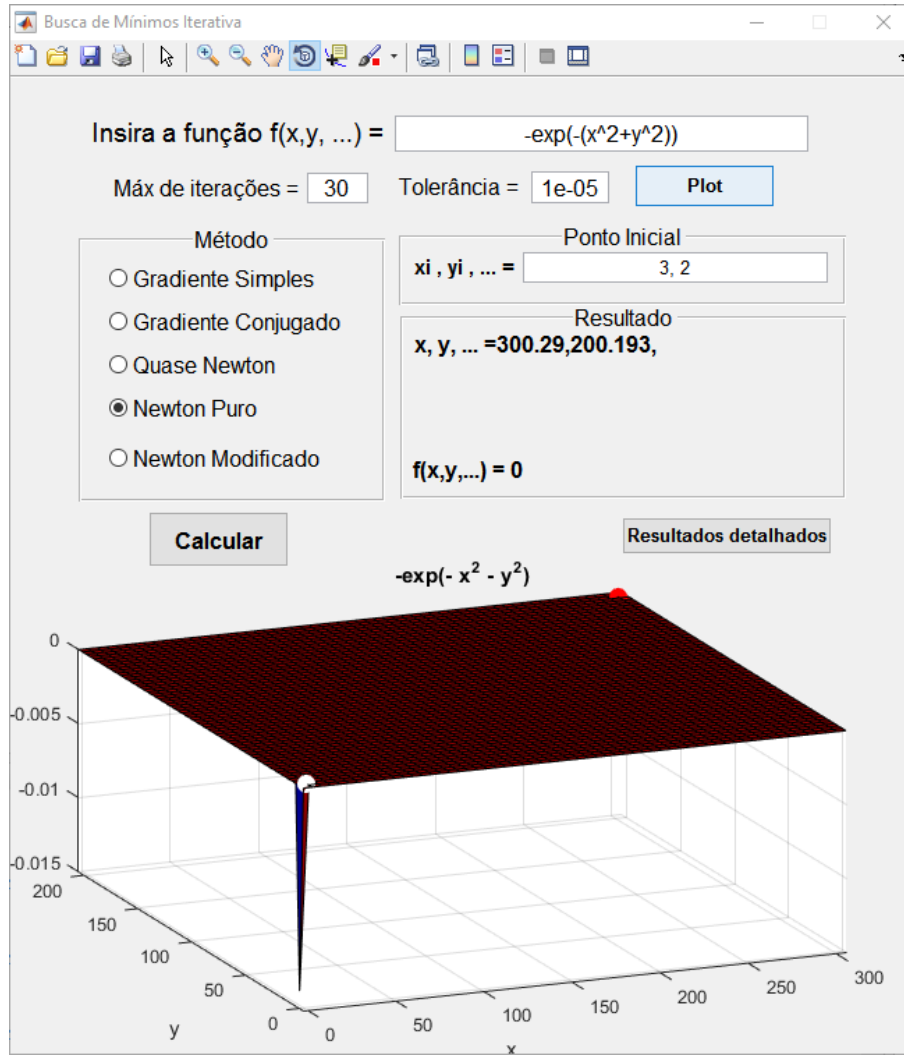


Figura 7: Janela de inicialização de $f_2(x, y)$

d^k que seja realmente de descida. Em que F^k tem autovalores positivos para poder gerar uma direção d^k de descida segundo a equação 9.

$$d^k = -[F^k]^{-1}g^k = -[\nabla^2 f(x^k) + \gamma I_n]^{-1}\nabla f(x^k) \quad (9)$$

Depois de construído o algoritmo, foram testadas quatro funções para efeitos de comparação:

- $f_1(x, y) = x^2 + y^2$
- $f_2(x, y) = -e^{-x^2-y^2}$
- $f_3(x, y) = \cos(\frac{xy}{5}) + \sin(\frac{xy}{5})$
- $f_4(x, y) = |x + y|$

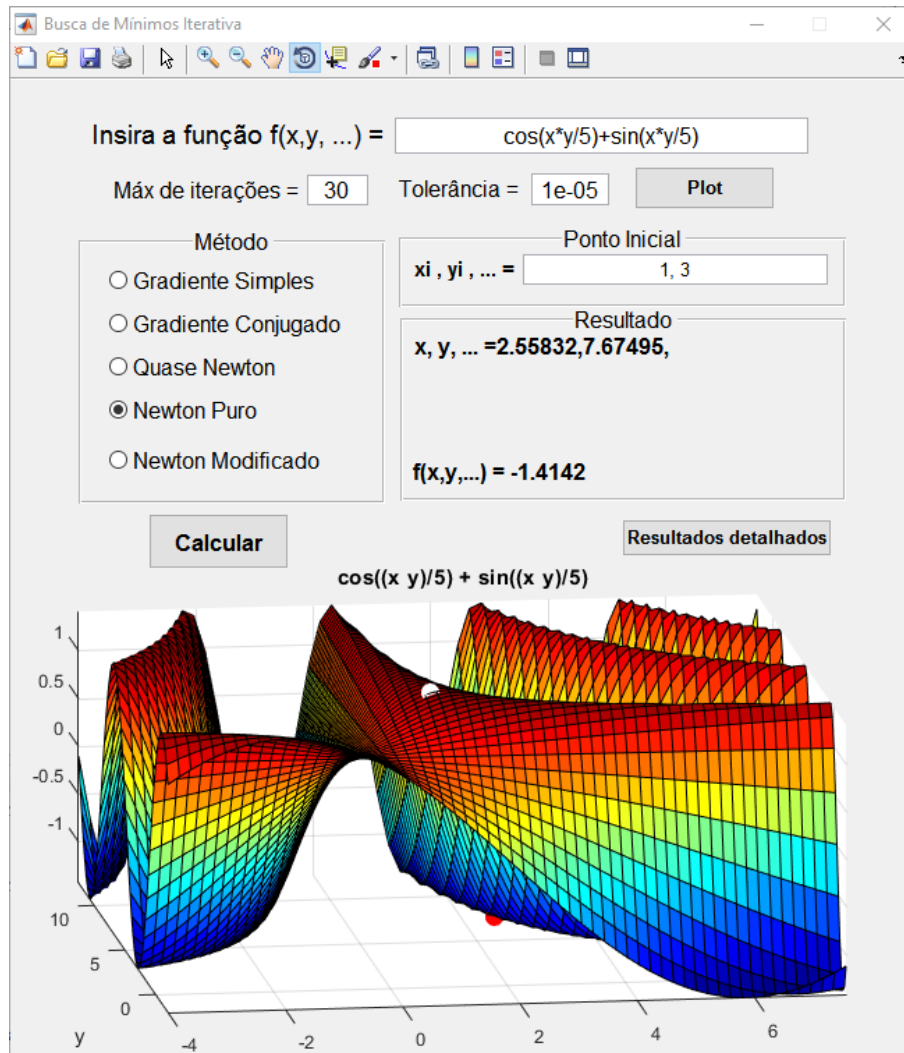


Figura 8: Janela de inicialização de $f_3(x, y)$

6 Método de Quase Newton

Assim como os outros métodos de Newton, esse método tem como intenção facilitar o cálculo da Hessiana. Deseja-se chegar a ela sem derivar duas vezes a função objetivo. As iterações são regidas por:

$$x^{k+1} = x^k + \alpha_k d^k = x^k - \alpha [F^k]^{-1} g^k \quad (10)$$

onde F^k contém informações sobre a Hessiana G^k e deve ser obtido de forma simples. A tarefa de obter F^k é substituída pela de obter sua inversa $H^k = [F^k]^{-1}$. Em cada passo esta matriz deve exibir, de algum modo, características da Hessiana, ou melhor, de sua inversa. Vejamos como conseguir isso. Utilizando a expansão em série de Taylor do gradiente de f , podemos notar que H , em cada iteração, pode ser

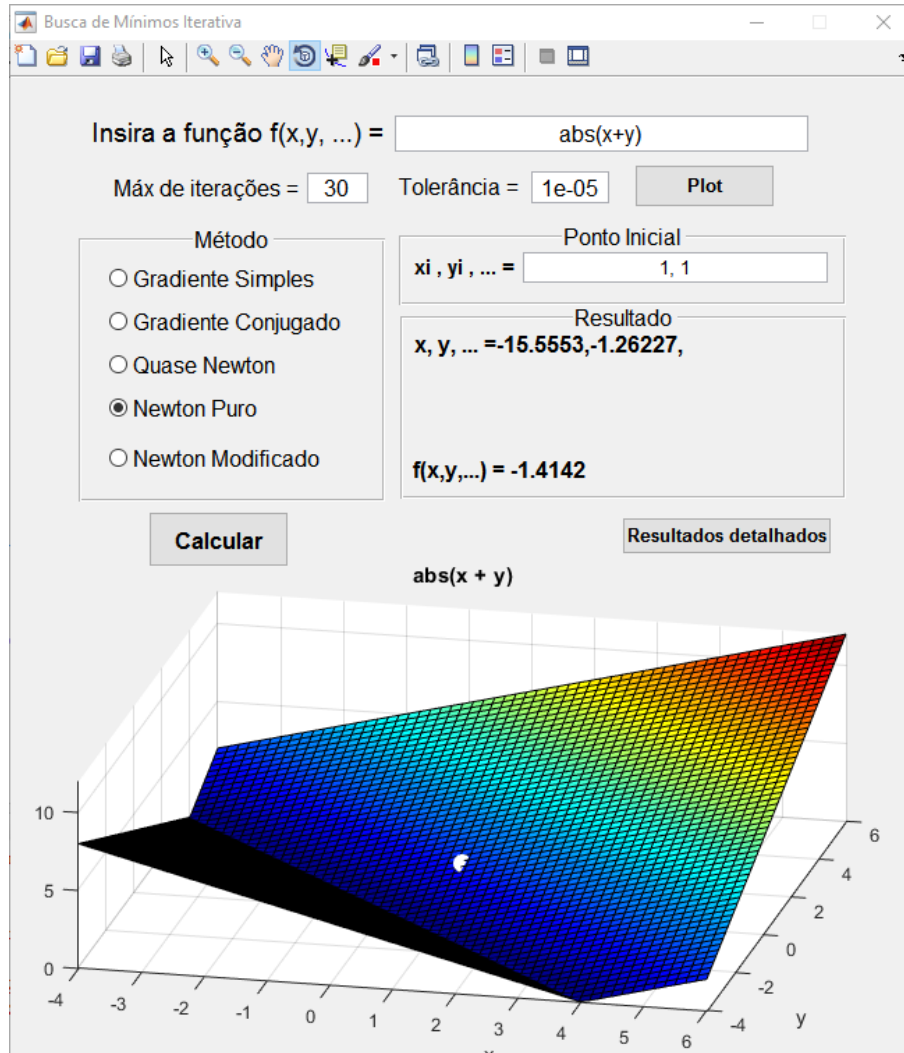


Figura 9: Janela de inicialização de $f_4(x, y)$

delimitado pela relação

$$H^{k+1}\gamma^k = \delta^k = \begin{cases} \gamma^k = g^{k+1} - g^k \\ \delta^k = x^{k+1} - x^k \end{cases} \quad (11)$$

chamada de condição de quase-Newton. Para gerar matrizes H satisfazendo esta restrição utilizamos, em nosso programa, a fórmula de Broyden-Fletcher-Goldfarb-Shanno (BFGS):

$$H^{k+1} = H - \frac{\delta\gamma^T H + H\gamma\delta^T}{\delta^T\gamma} + \left(1 + \frac{\gamma^T H\gamma}{\delta^T\gamma}\right) \frac{\delta\delta^T}{\delta^T\gamma} \quad (12)$$

Para comprovar a eficiência do método, o testamos utilizando as seguintes funções:

- $f_1(x, y) = x^2 + y^2$
- $f_2(x, y) = -e^{-x^2-y^2}$

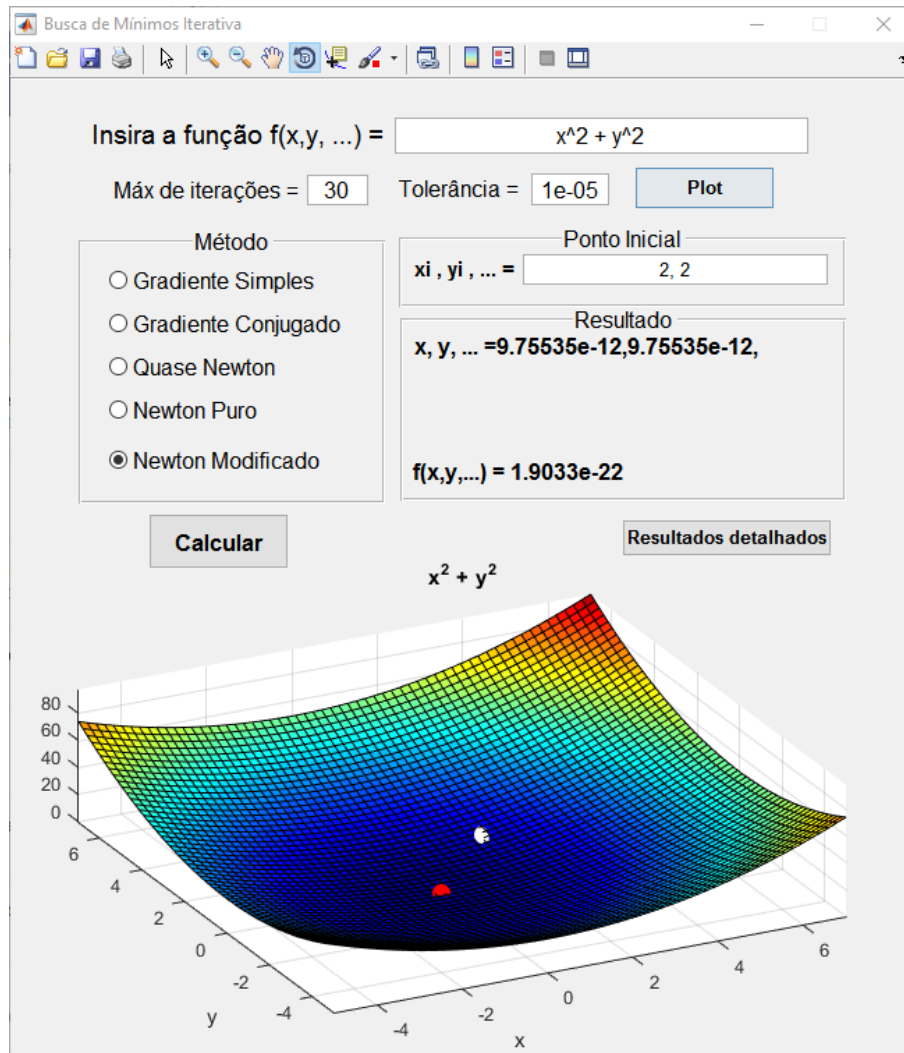
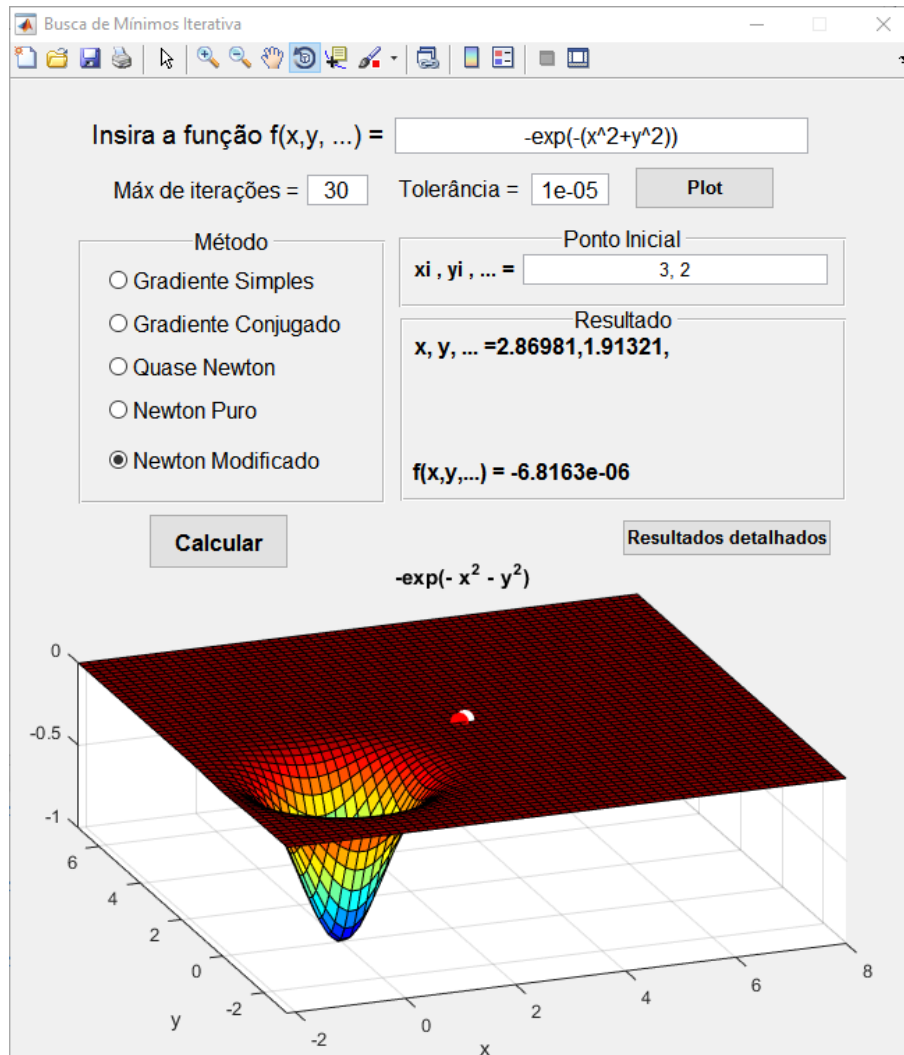
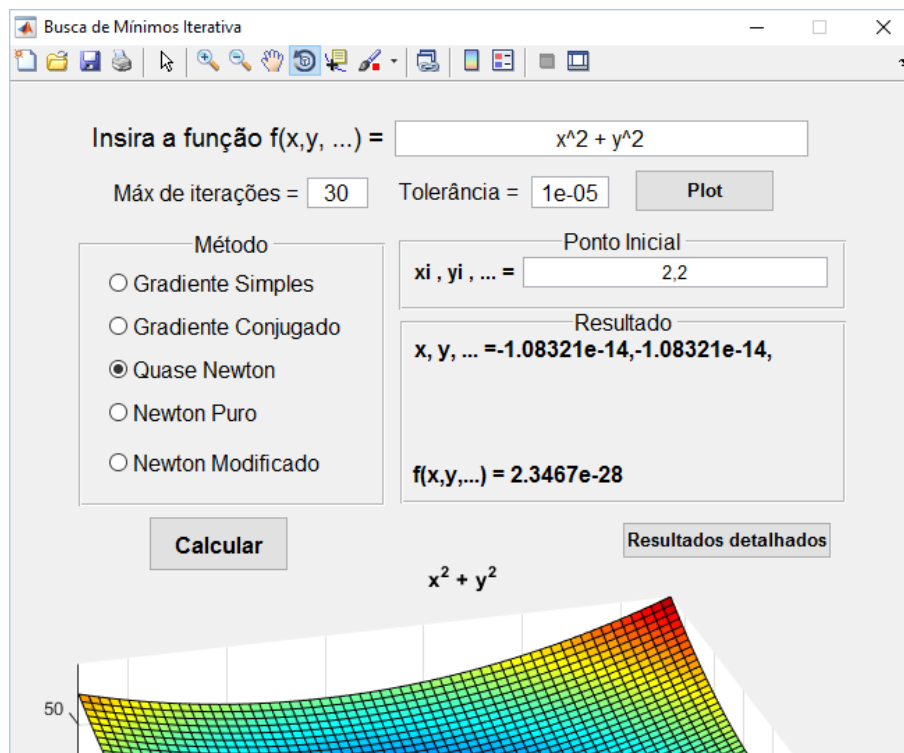
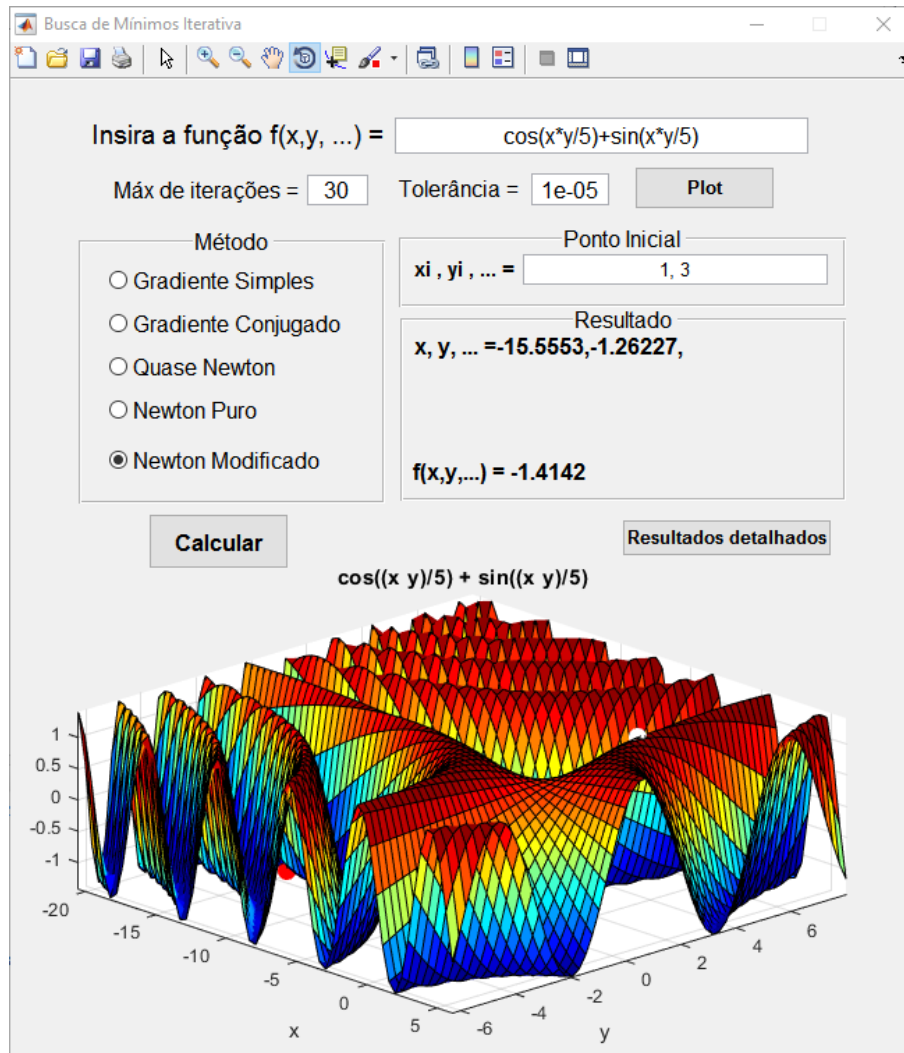
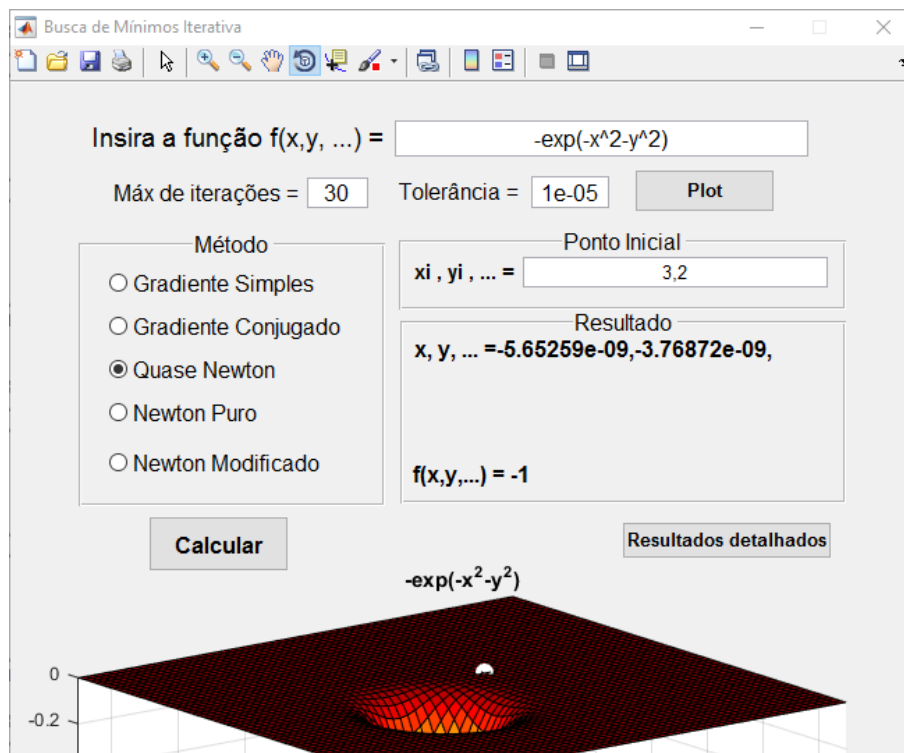


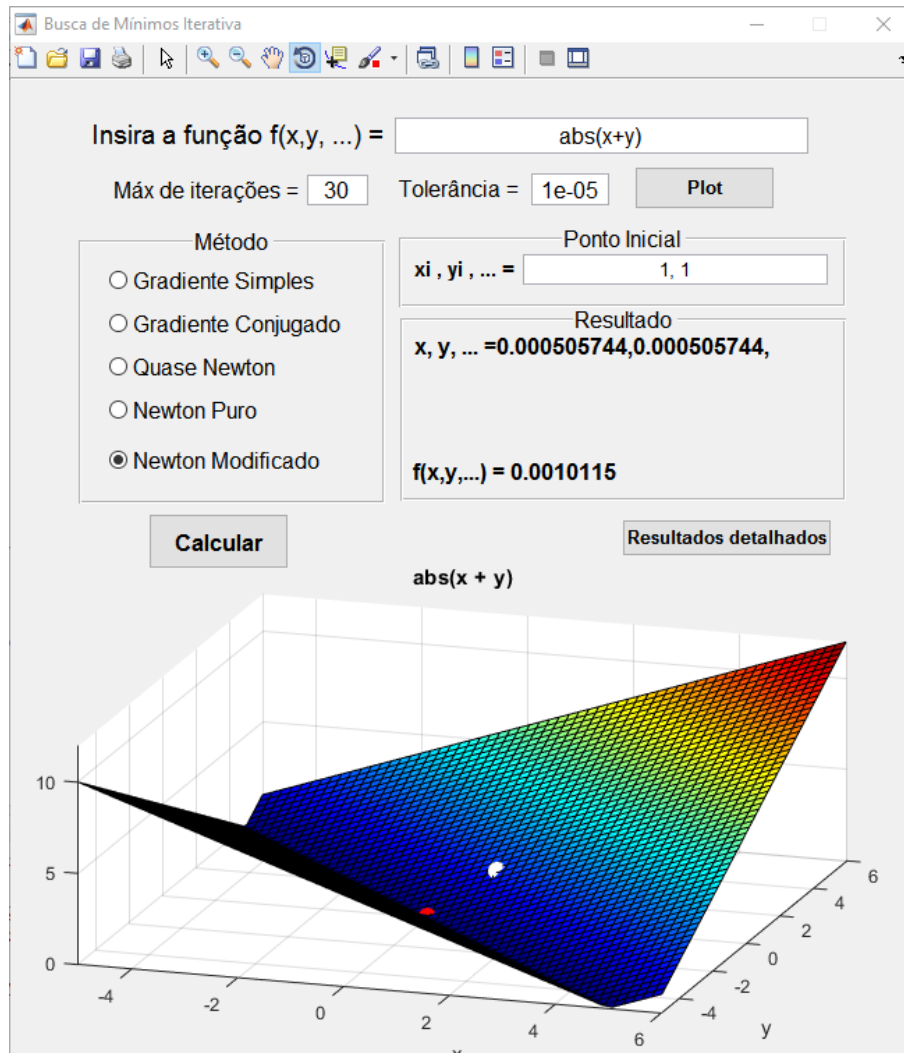
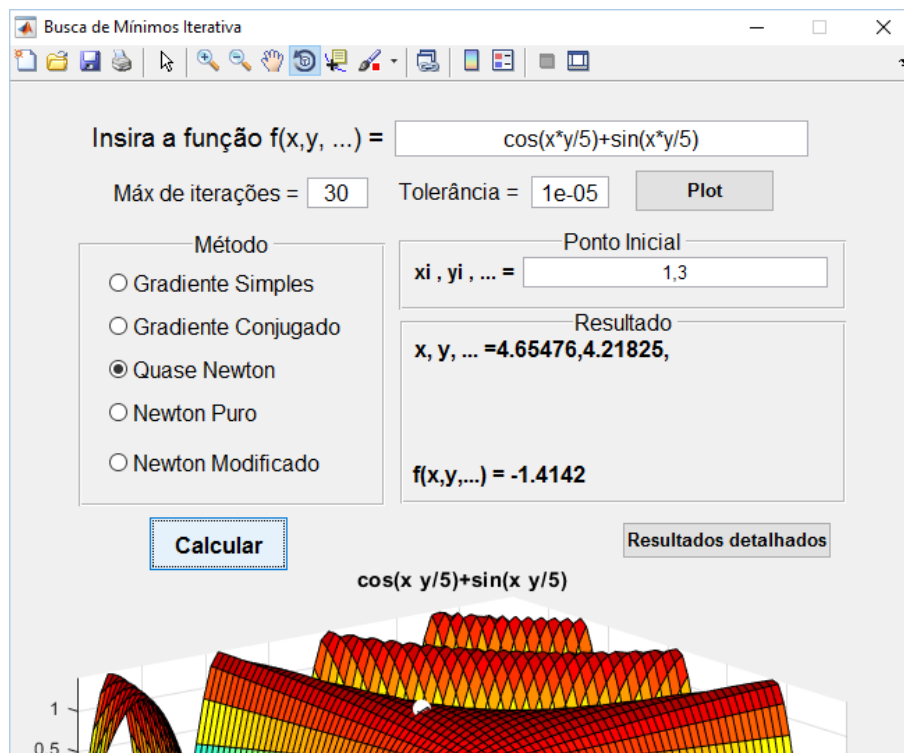
Figura 10: Janela de inicialização de $f_1(x, y)$

- $f_3(x, y) = \cos\left(\frac{xy}{5}\right) + \sin\left(\frac{xy}{5}\right)$
- $f_4(x, y) = |x + y|$

Os resultados foram os seguintes:

Figura 11: Janela de inicialização de $f_2(x, y)$ 

Figura 12: Janela de inicialização de $f_3(x, y)$ 

Figura 13: Janela de inicialização de $f_4(x, y)$ 

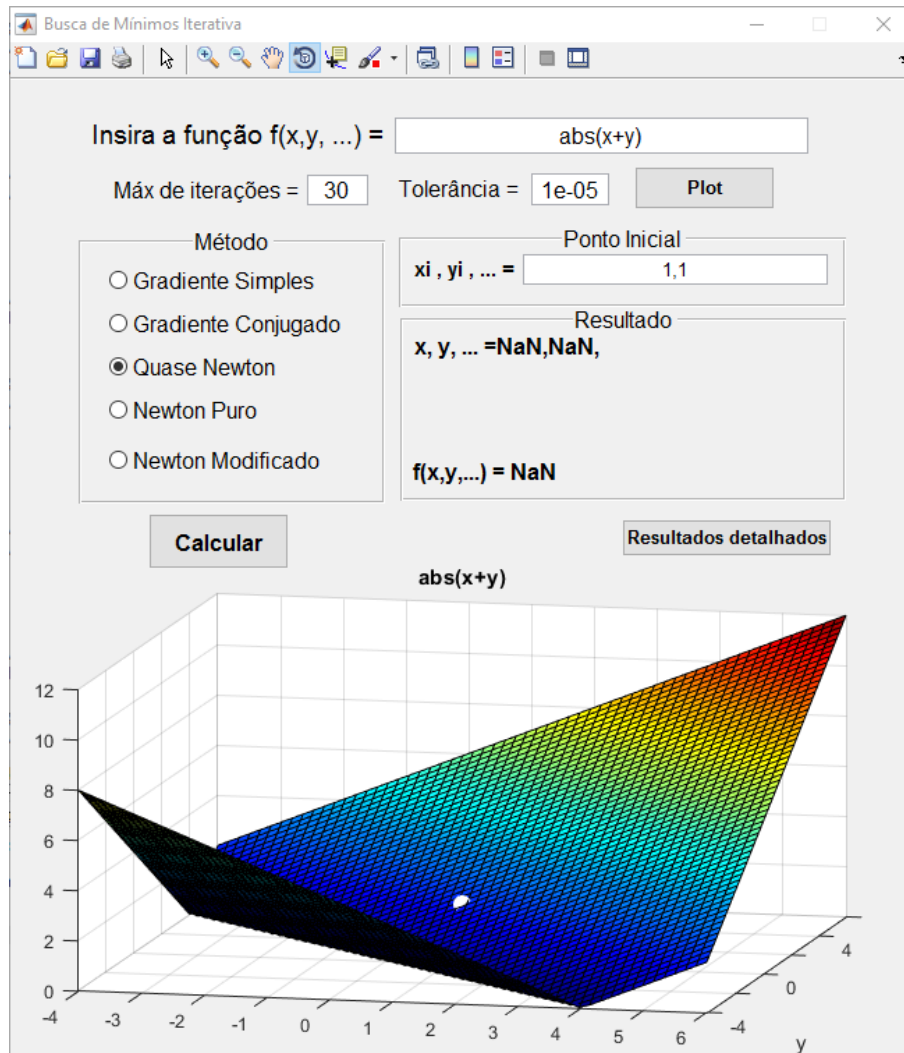


Figura 17: Janela de inicialização de $f_4(x,y)$

7 Conclusão

Para comparar os cinco métodos utilizados, testamos quatro funções com o objetivo de descobrir os pontos fracos e fortes de cada método. Para facilitar essa comparação, analisaremos os resultados obtidos por cada método no programa criado.

- Para o primeiro caso, foi utilizada a função $f_1(x, y) = x^2 + y^2$ com ponto inicial em (2,2)

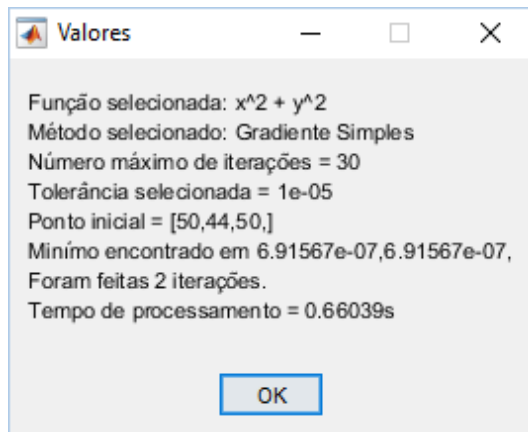


Figura 18: Resultados de f_1 (Gradiente)

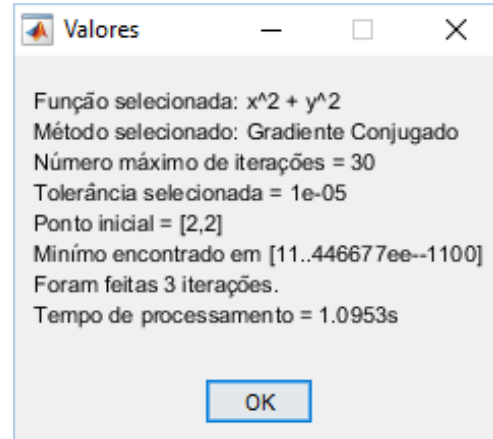


Figura 19: Resultados de f_1 (Gradiente Conjugado)

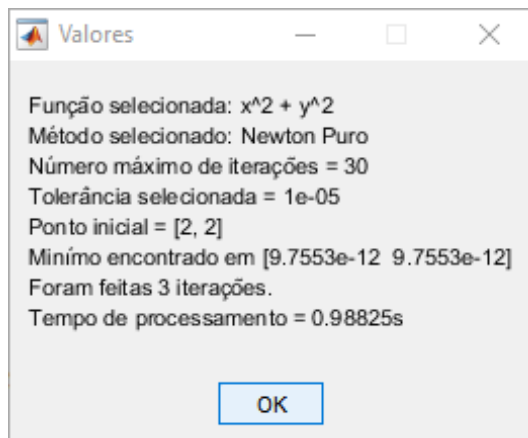


Figura 20: Resultados de f_1 (Newton)

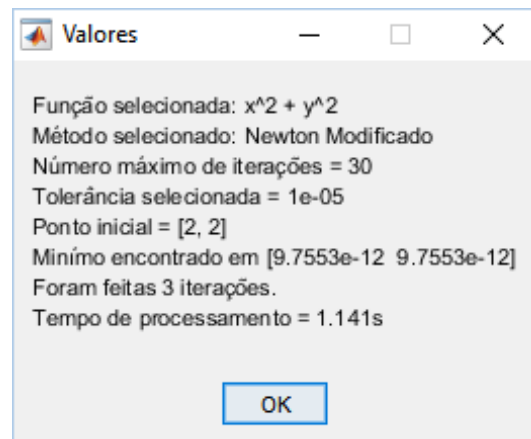


Figura 21: Resultados de f_1 (Newton Modificado)

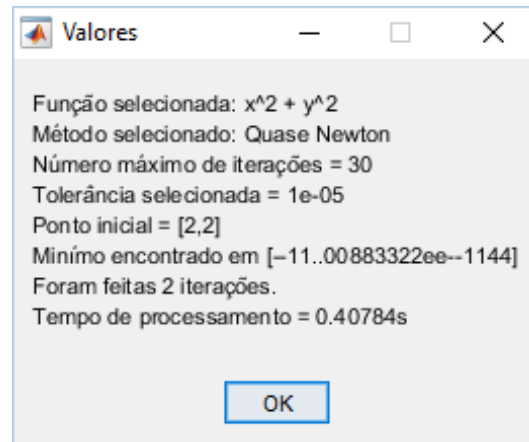


Figura 22: Resultados de f_1 (Quase Newton)

A função $f_1(\mathbf{x}, \mathbf{y})$ foi utilizada como controle para os outros testes, por ser uma função simples, e o seu mínimo é facilmente encontrado por todos os métodos, porém alguns se mostraram mais rápidos que outros. O método de Quase Newton se mostrou o mais rápido, além de convergir com apenas duas iterações.

- No segundo caso, usamos a função $f_2(\mathbf{x}, \mathbf{y}) = -e^{-x^2-y^2}$ com ponto inicial em (3,2)

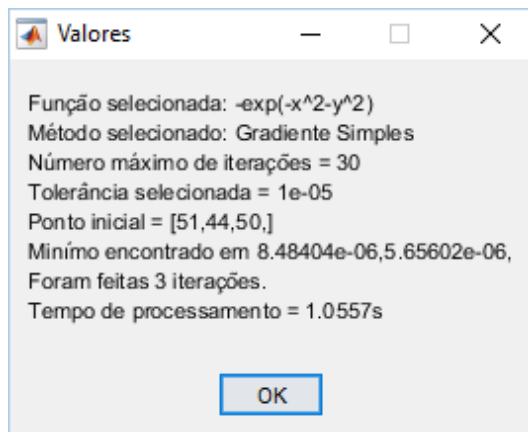


Figura 23: Resultados de f_2 (Gradiente)

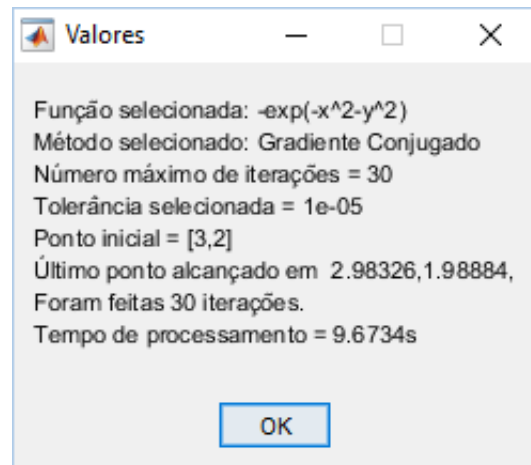


Figura 24: Resultados de f_2 (Gradiente Conjugado)

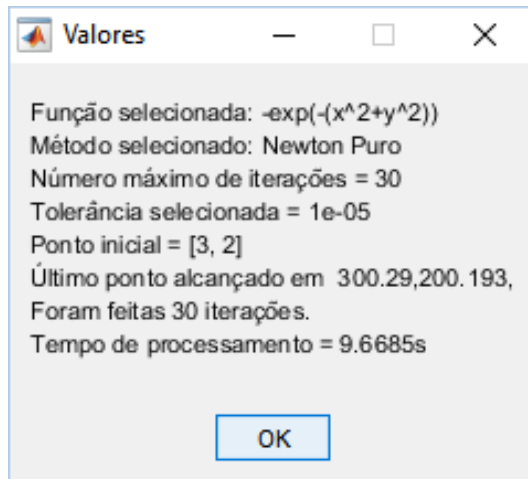


Figura 25: Resultados de f_2 (Newton)

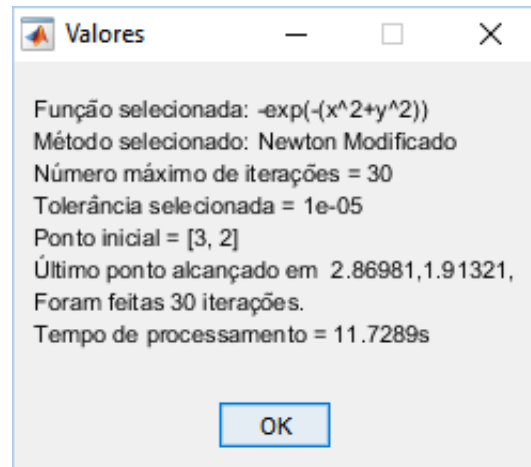


Figura 26: Resultados de f_2 (Newton Modificado)

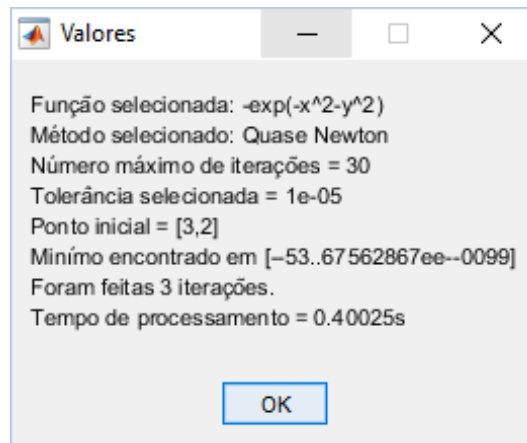
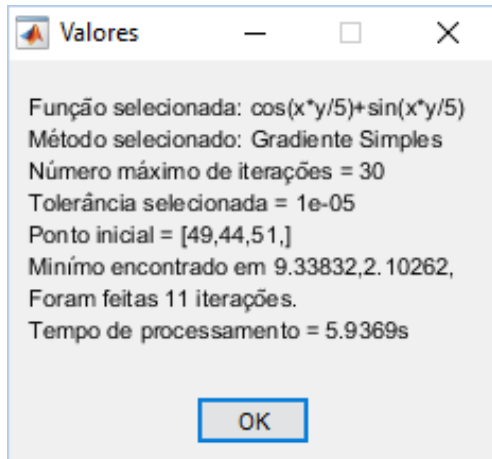
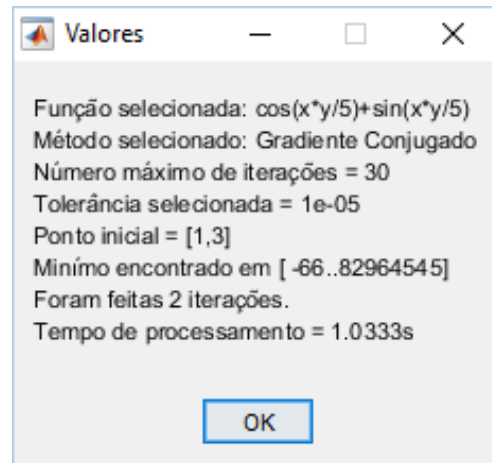
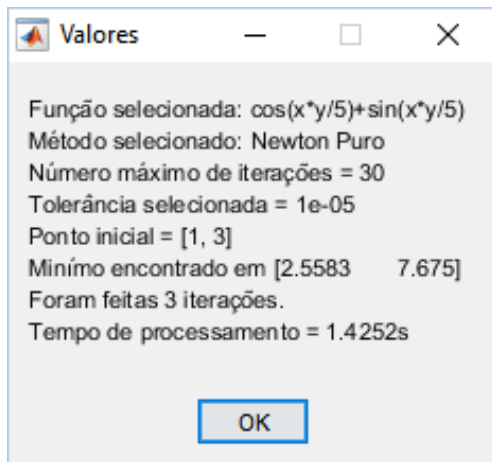
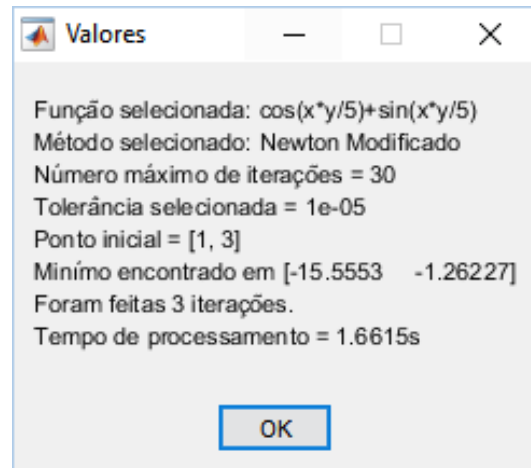
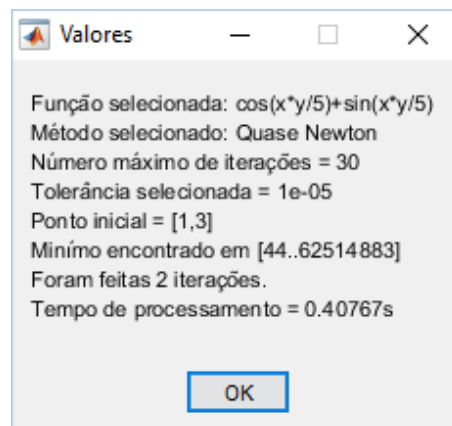


Figura 27: Resultados de f_2 (Quase Newton)

Por ser uma função com uma descida repentina e ter sido usado um ponto inicial longe do mínimo, os métodos de Newton não funcionam muito bem no geral para a função $f_2(\mathbf{x}, \mathbf{y})$, já que não é possível criar uma aproximação quadrática para a função escolhida. Por isso, os métodos de Newton e de Newton Modificado fizeram todas as iterações pedidas e não conseguiram alcançar o mínimo. Podemos ver que as melhorias do método de Quase Newton fizeram com que esse método contornasse o problema do método de Newton puro, achando o mínimo da função. O método do Gradiente Conjugado não conseguiu encontrar o mínimo mesmo depois de todas as iterações.

- Como terceiro teste, usamos a função $f_3(\mathbf{x}, \mathbf{y}) = \cos(\frac{xy}{5}) + \sin(\frac{xy}{5})$ com ponto inicial em (1,3)

**Figura 28:** Resultados de f_3 (Gradiente)**Figura 29:** Resultados de f_3 (Gradiente Conjugado)**Figura 30:** Resultados de f_3 (Newton)**Figura 31:** Resultados de f_3 (Newton Modificado)**Figura 32:** Resultados de f_3 (Quase Newton)

Podemos ver que, para a função $f_3(\mathbf{x}, \mathbf{y})$, os métodos de Newton, Newton Modificado e Quase Newton funcionaram bem, convergindo com poucas iterações e rapidamente. O método do Gradiente não funcionou tão bem, por ter sido aplicado em uma função com muitas ondulações. As alterações implementadas no método do Gradiente Conjugado fizeram com que esse método conseguisse encontrar o mínimo com relativa facilidade, em apenas duas iterações.

- Para a última comparação, foi testada a função $f_4(\mathbf{x}, \mathbf{y}) = |\mathbf{x} + \mathbf{y}|$ com ponto inicial em (1,1)

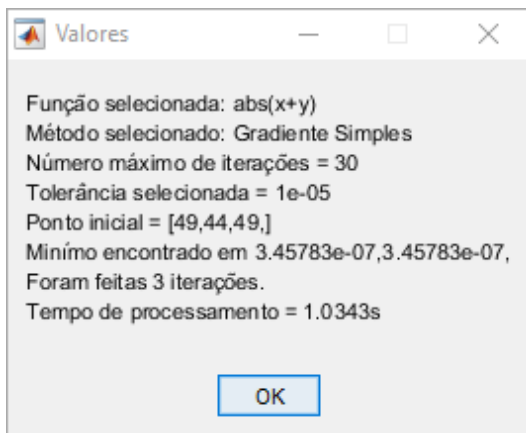


Figura 33: Resultados de f_4 (Gradiente)

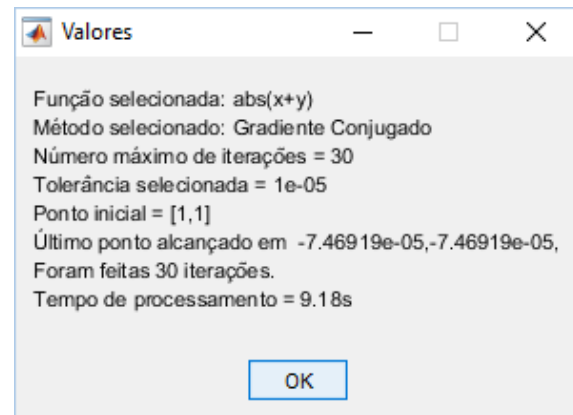


Figura 34: Resultados de f_4 (Gradiente Conjugado)

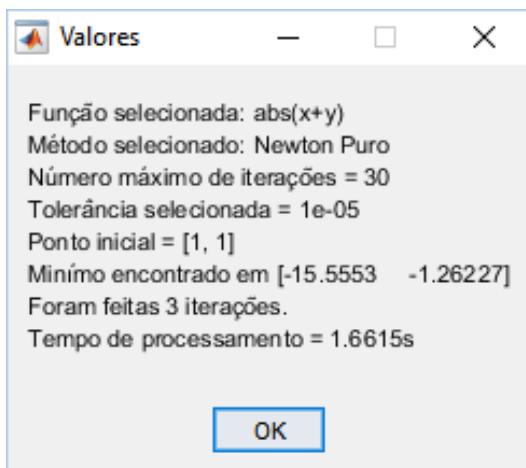


Figura 35: Resultados de f_4 (Newton)

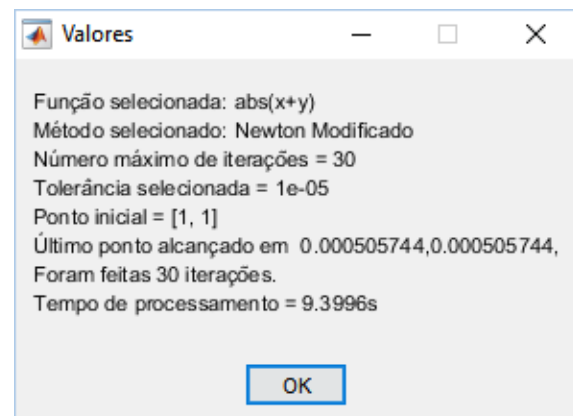


Figura 36: Resultados de f_4 (Newton Modificado)

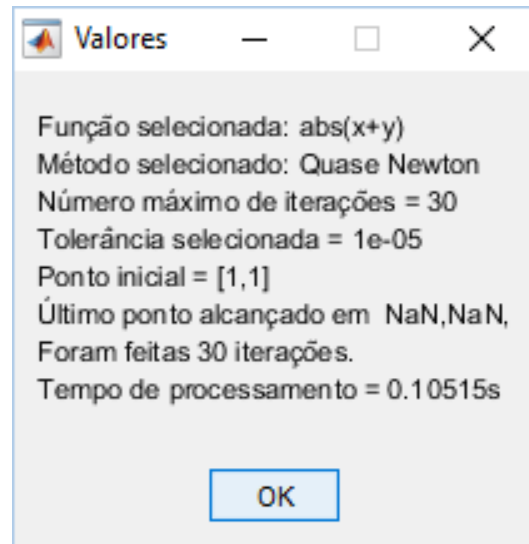


Figura 37: Resultados de f_4 (Quase Newton)

Alguns métodos tiveram dificuldades para encontrar o mínimo da função $f_4(\mathbf{x}, \mathbf{y})$, devido ao fato de que essa função possui um bico no seu mínimo, o que dificulta a ação dos métodos de Newton, por isso, o método de Quase Newton não conseguiu encontrar um mínimo, e o método de Newton Modificado fez todas as iterações pedidas, demorando quase 10 segundos para encontrar um valor. O método do Gradiente conseguiu encontrar facilmente o mínimo, em apenas 3 iterações, enquanto que o método do Gradiente Modificado teve que usar o número máximo de iterações para encontrar um valor.

Podemos então concluir que não existe um método absoluto para localização de mínimos de funções vetoriais, mas podemos classificar alguns métodos em seções. Os métodos de Newton costumam funcionar melhor para funções que admitem boas aproximações quadráticas, enquanto que os métodos do tipo Gradiente não costumam funcionar muito bem se a função objetivo oscila muito rapidamente.