Trabalho 1

Introdução à Otimização

Alunos:
Cayo Valsamis
Gabriel Pelielo
Rafael Accácio
Rodrigo Moysés

Professor:
Afonso Celso del Nero

2 de Fevereiro de 2016

Conteúdo

1 Introdução

O trabalho detalhado neste relatório se baseia na busca de mínimos de funções escalares. Foram implementados quatro métodos diferentes de localização de mínimos, que foram:

- Método de Fibonacci
- Método da Divisão Áurea
- Método da interpolação Polinomial

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 de Fibonacci

O método de Fibonacci é um método iterativo utilizado para a localização do mínimo de funções. Esse método utiliza um intervalo I_k = [a b], escolhemos então dois pontos simétricos em relação ao centro. Chamando x_1=a e x_4=b os pontos escolhidos serão x_2 e x_3. como pode ser visto na 1

Figura 1: Segmento de reta $\overline{ab}$

O mecanismo de redução do intervalo é bem simples, usa-se o critério dos 2 pontos. Para determinar o próximo intervalo $I_k + 1$ calcula-se o valor da função f em x_2 e x_3, resultando em f_2 e f_3. Se f_2 ¡ f_3 então [x_1 x_3] será o novo intervalo (2) e se f_2 ¿ f_3, então retemos [x_2 x_4] (3). No caso de igualdade entre f_2 e f_3 a escolha é indiferente.

Figura 2: Intervalo antigo e novo

Figura 3: Intervalo antigo e novo

Criamos então o novo intervalo $I_k + 1$, caso ele seja:

x_1 x_3 $\rightarrow$ x_1 permanece x_1, x_3 se torna o novo x_4, x_2 se torna o novo x_3 e um novo x_2 é colocado simetricamente ao novo x_3.

x_2 x_4 $\rightarrow$ x_4 permanece x_4, x_2 se torna o novo x_1, x_3 se torna o novo x_2 e um novo x_3 é colocado simetricamente ao novo x_2.

Só nos resta agora explicar como definir a escolha inicial de x_2 e x_3. Esses valores são escolhidos utilizando um fator 0,7 ¡ α ¡ 0,8, então algebricamente calculamos $x_2 = \alpha_1 x_1 + (1-\alpha_1)x_4$ e $x_3 = (1-\alpha_1)x_1 + \alpha_1 x_4$. O valor de α é proveniente da fórmula utilizada para determinar o k-ésimo número da frequência de Fibonacci, portanto o método ganha esse nome. Sendo k o número de reduções necessárias em determinado problema, podemos determinar o valor de α através de: $\alpha = \dfrac{2}{1+\sqrt{5}} \times \dfrac{1-p^k}{1-p^{k+1}}$ onde $p = \dfrac{1-\sqrt{5}}{1+\sqrt{5}}$

Agora, com todas essas informações, podemos finalmente demonstrar o algoritmo que realmente será utilizado em sua forma computacional iterativa: **Dados:**

- $f : \mathbb{R} \rightarrow \mathbb{R}$, uma função suave e unimodal em
- $I_1 = [ab] \subset \mathbb{R}$, intervalo enquadrante inicial
- $n \in \mathbb{Z}$ número desejado de reduções
- $k \in \mathbb{Z}$ índice de Fibonacci $(= n + 1)$

Objetivo: Encontrar $I_n \subset I_1$ que enquadre um mínimo de f **Operações:**

1. $p = \dfrac{1-\sqrt{5}}{1+\sqrt{5}}$, $\alpha = \dfrac{2}{1+\sqrt{5}} \times \dfrac{1-p^k}{1-p^{k+1}}$
2. $i = 1$
3. $x_1 = a;\ x_4 = b;\ L_{ini} = b - a;$
4. $x_2 = \alpha x_1 + (1-\alpha)x_4;\ f_2 = f(x_2)$
5. $x_3 = \alpha x_4 + (1-\alpha)x_1;\ f_3 = f(x_3)$
6. $f_2 < f_3$
    - $a = x_1;\ b = x_3;\ L_{fin} = b - a;$
    - $i = n \rightarrow I_n = [ab] \rightarrow FIM$
    - $\alpha = \dfrac{(L_{ini} - L_{fin})}{L_{fin}};\ i = i + 1$
    - volta a 3.

    - $f_2 \geq f_3$
    - $a = x_2;\ b = x_4;\ L_{fin} = b - a;$
    - $i = n \rightarrow I_n = [ab] \rightarrow FIM$

- $\alpha = \dfrac{(Lini - L_{fin})}{L_{fin}};\ i = i + 1$
- volta a 3.

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

- $f_1(x) = 3x^2 + 20x - 8$
- $f_2(x) = xsin(x)cos(x)$
- $f_3(x) = 5x$
- $f_4(x) = -e^{-|x|}$

Os resultados para a função $f_1(x)$ foram:

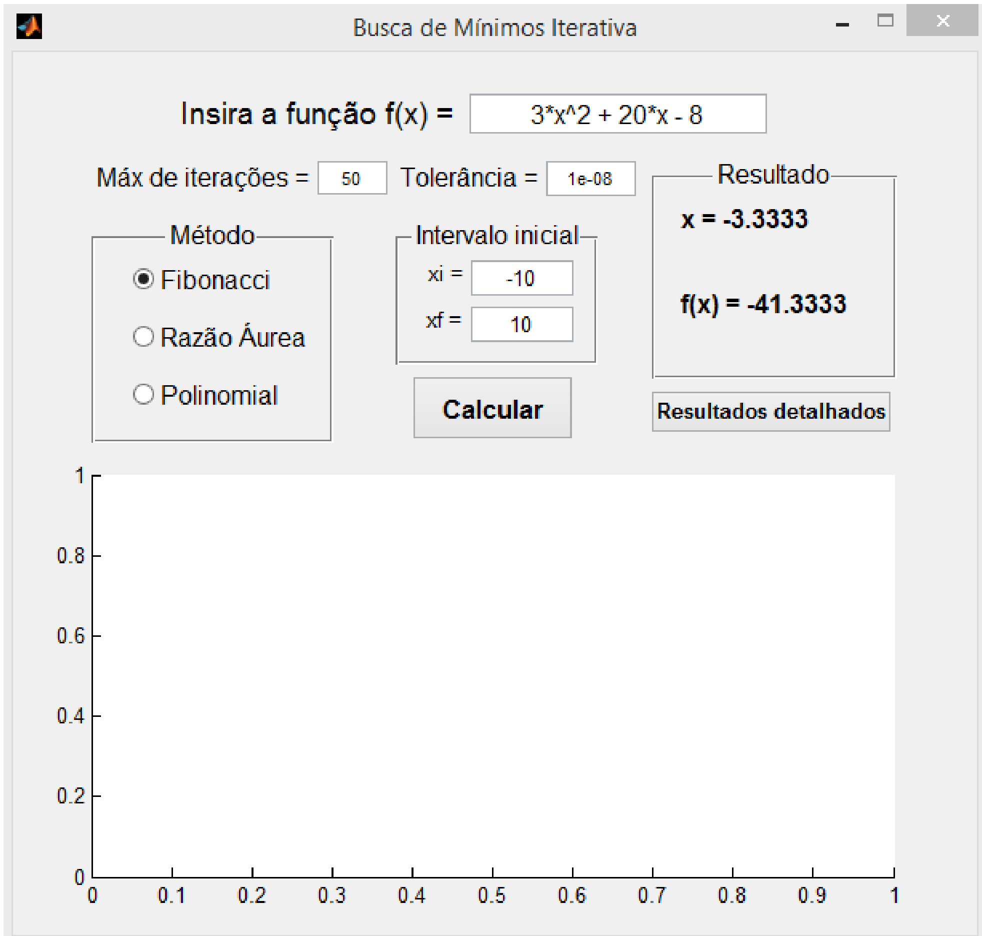


Figura 4: Janela de inicialização de $f_1(x)$

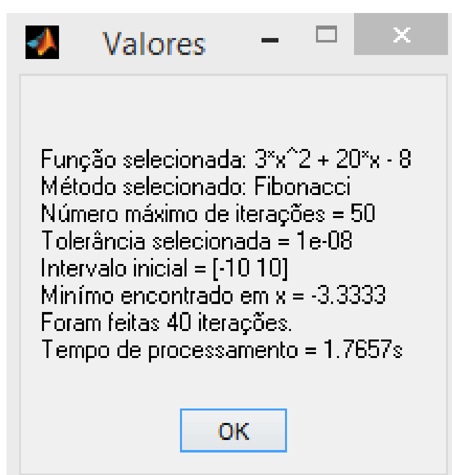


Figura 5: Resultados detalhados de $f_1(x)$

Os resultados para a função $f_2(x)$ foram:

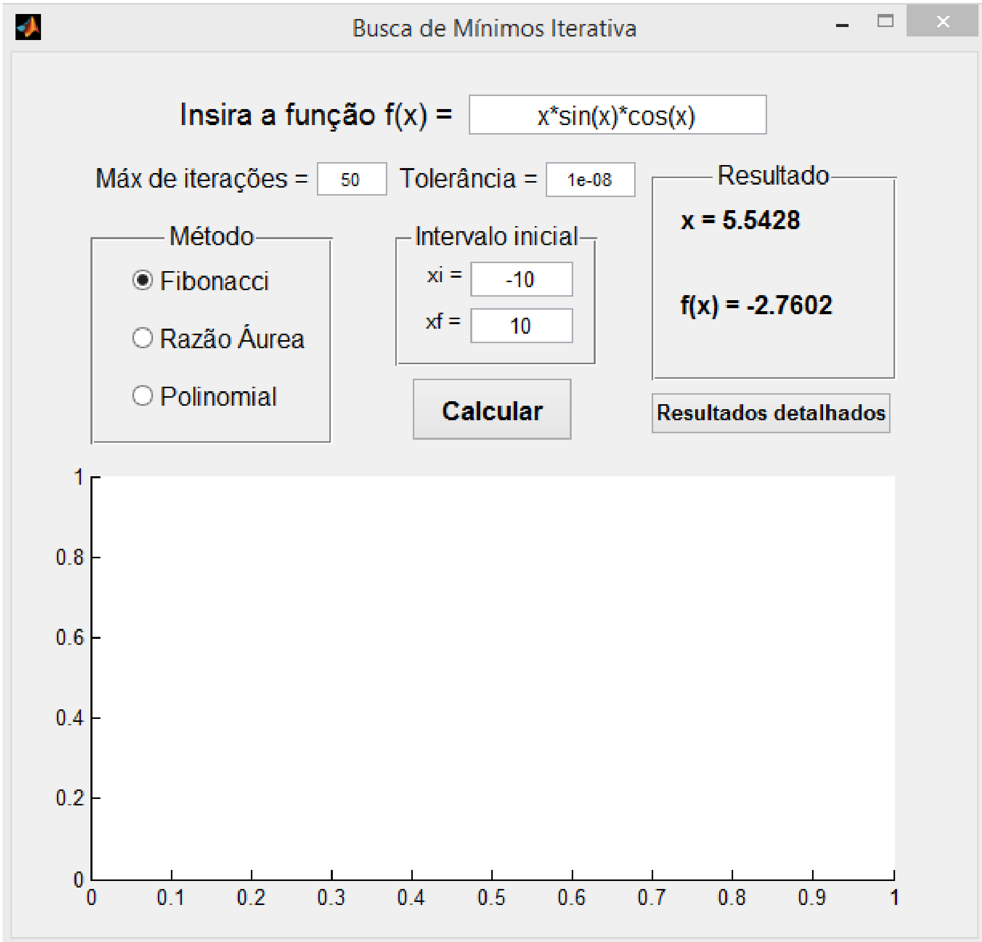


Figura 6: Janela de inicialização de $f_2(x)$

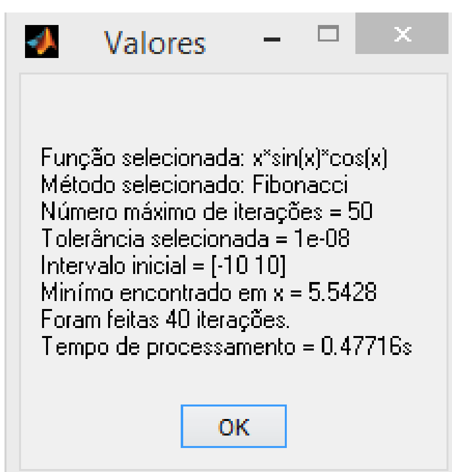


Figura 7: Resultados detalhados de $f_2(x)$

Os resultados para a função $f_3(x)$ foram:

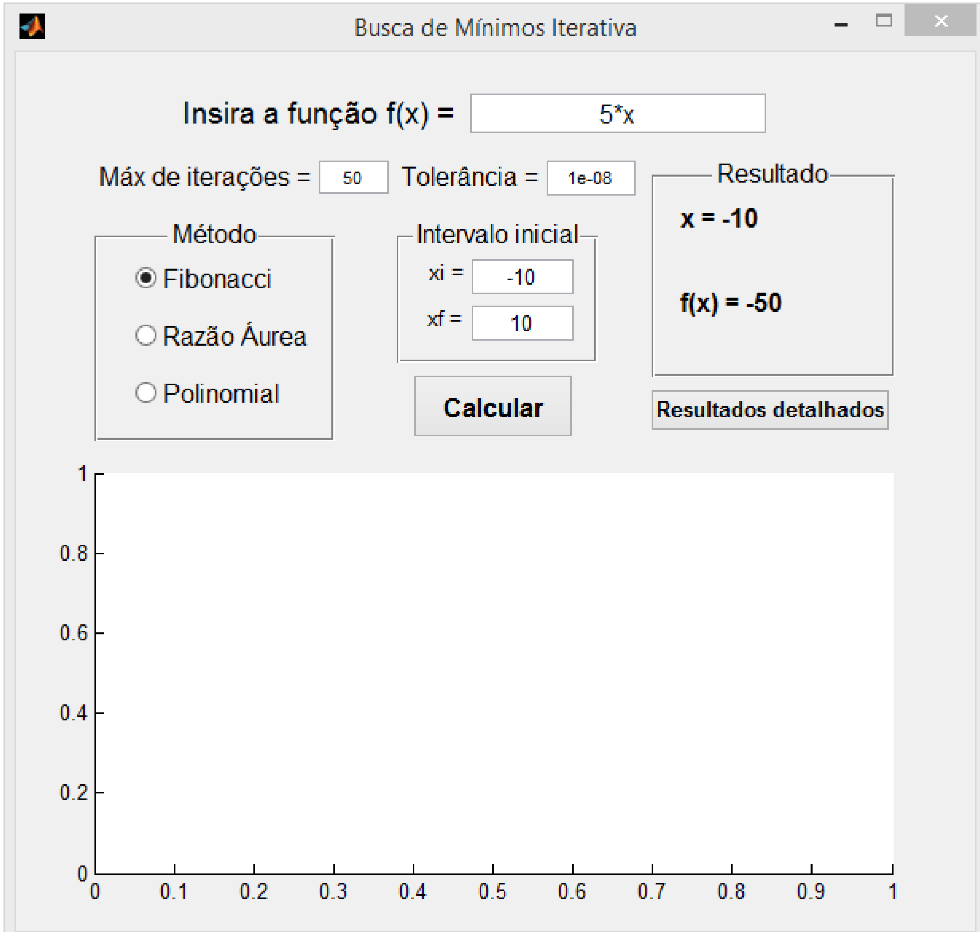


Figura 8: Janela de inicialização de $f_3(x)$

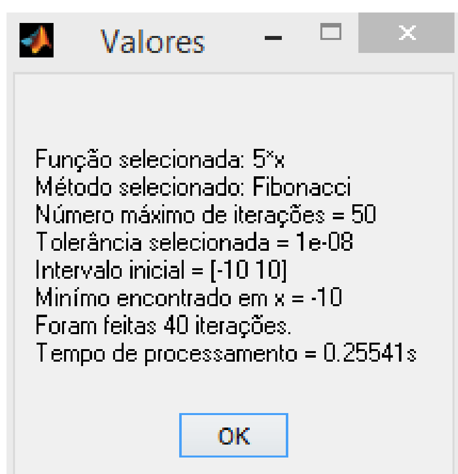


Figura 9: Resultados detalhados de $f_3(x)$

Os resultados para a função $f_4(x)$ foram:

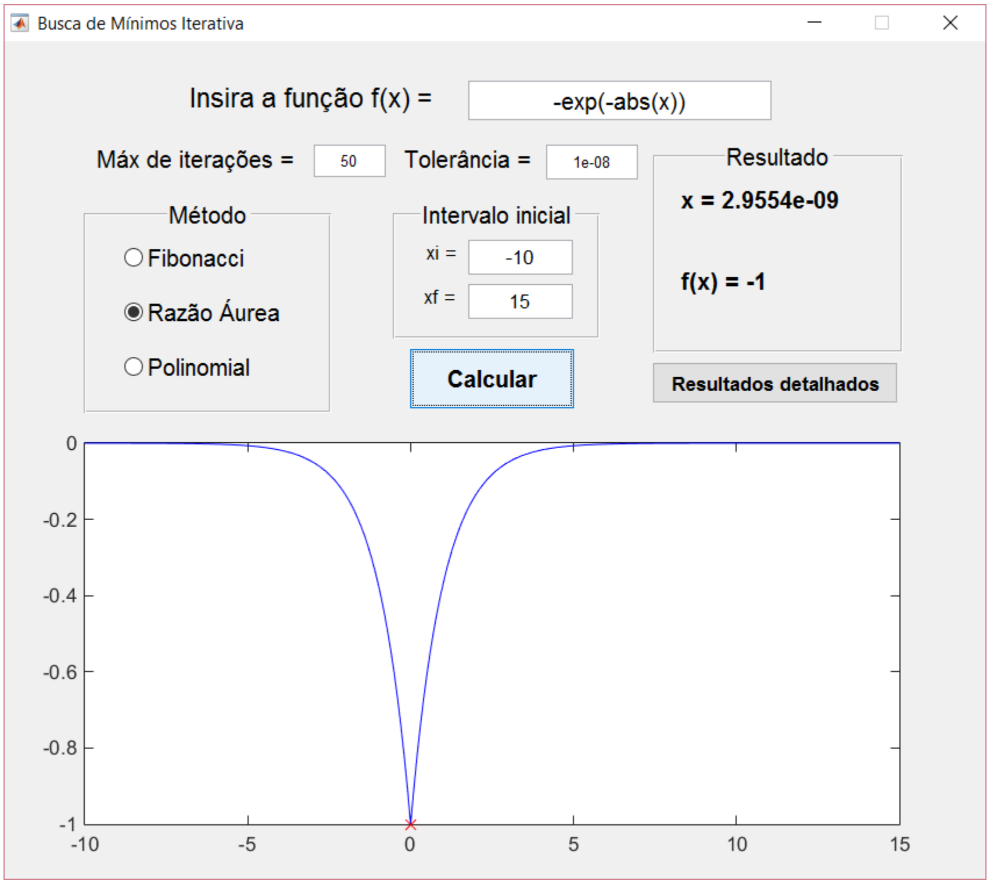


Figura 10: Janela de inicialização de $f_4(x)$

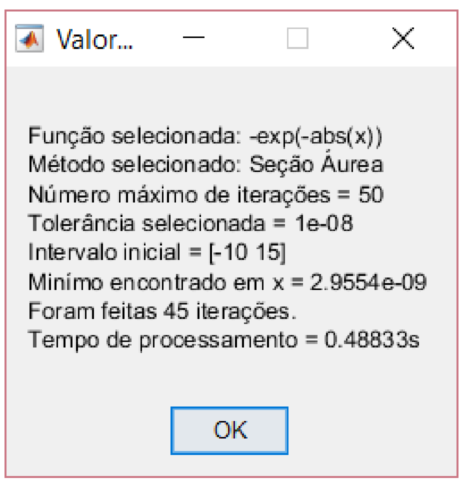


Figura 11: Resultados detalhados de $f_4(x)$

3 Método da Divisão Áurea

O método da Divisão áurea é outro método de localização de mínimos de funções escalares. Foi criado como uma avanço para o método de Fibonacci e se baseia na razão áurea. Por definição, para dividir o segmento de reta da figura 12 na razão áurea, é necessário que

$$\frac{\overline{AC}}{\overline{AB}} = \frac{\overline{AB}}{\overline{BC}}$$

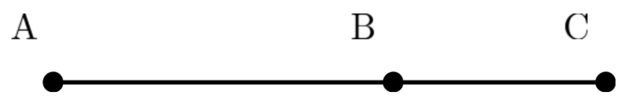


Figura 12: Segmento de reta $\overline{ABC}$

Dessa relação, podemos deduzir que $\overline{AB} = 0.618\overline{AC}$. Assim, o segmento de reta fica dividido segundo a razão áurea, como mostra a figura 13.

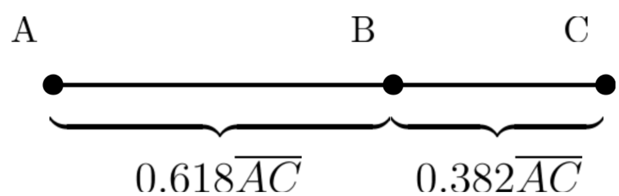


Figura 13: Divisão áurea do segmento de reta

A sequência de Fibonacci, base para a construção do método de Fibonacci, tem uma semelhança com a razão áurea: A divisão entre dois números consecutivos dessa sequência converge para a razão áurea. Ou seja, o método da divisão áurea tenta aprimorar o método de Fibonacci porque desde a primeira iteração, já faz a divisão do intervalo para o melhor possível.

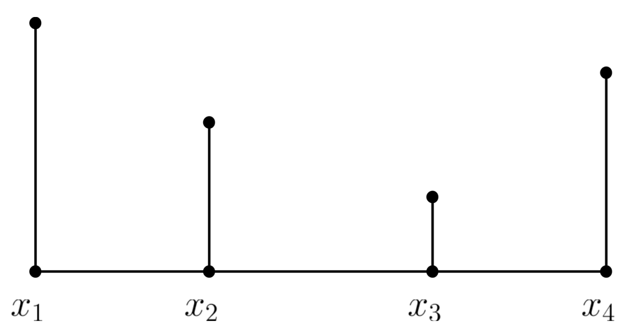


Figura 14: Divisão áurea do segmento de reta

A figura 14 ilustra a execução do método. É definido um intervalo que vai de x_1 até x_4 e dentro desse intervalo são escolhidos mais dois pontos, x_2 e x_3, porém dessa vez com a regra de divisão áurea. Depois disso, são calculados os valores da função objetivo que se deseja minimizar para os quatro pontos.

Tendo os dois valores $f(x_2)$ e $f(x_3)$, é selecionado o maior deles e o intervalo adjacente à esse é retirado. No caso da figura 14, o ponto x_1 seria retirado. Para continuar o método, é necessário que mais um ponto seja adicionado, novamente com a regra da razão áurea. Agora que temos novamente 4 pontos, o método pode ser repetido até que uma tolerância escolhida tenha sido atingida, e assim é localizado o mínimo da função.

Com o algoritmo da divisão áurea construído, foram testadas quatro funções para efeitos de comparação:

- $f_1(x) = 3x^2 + 20x - 8$
- $f_2(x) = xsin(x)cos(x)$
- $f_3(x) = 5x$
- $f_4(x) = -e^{-|x|}$

Os resultados para a função $f_1(x)$ foram:

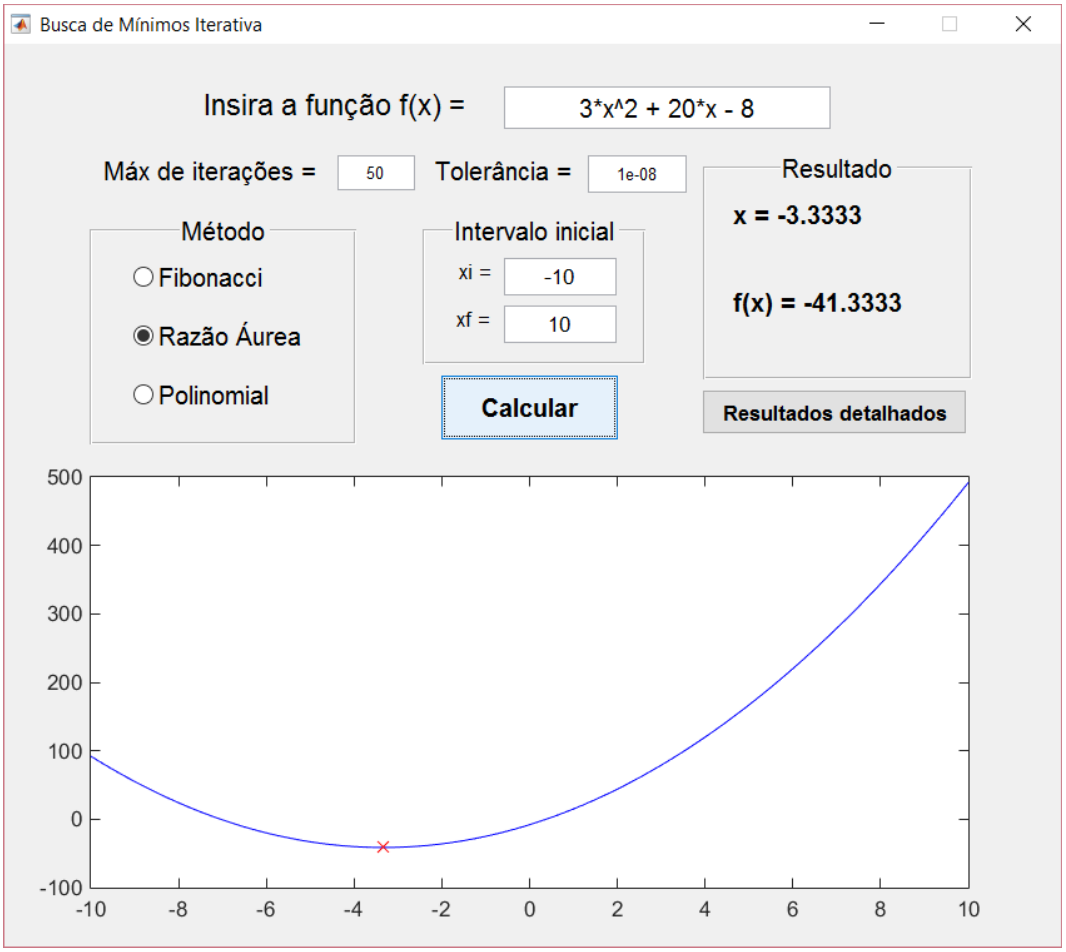


Figura 15: Janela de inicialização de $f_1(x)$

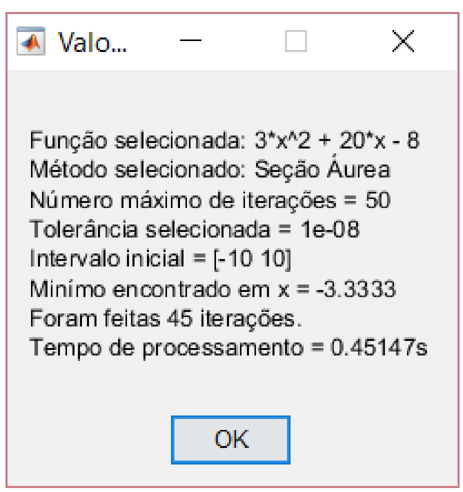


Figura 16: Resultados detalhados de $f_1(x)$

Os resultados para a função $f_2(x)$ foram:

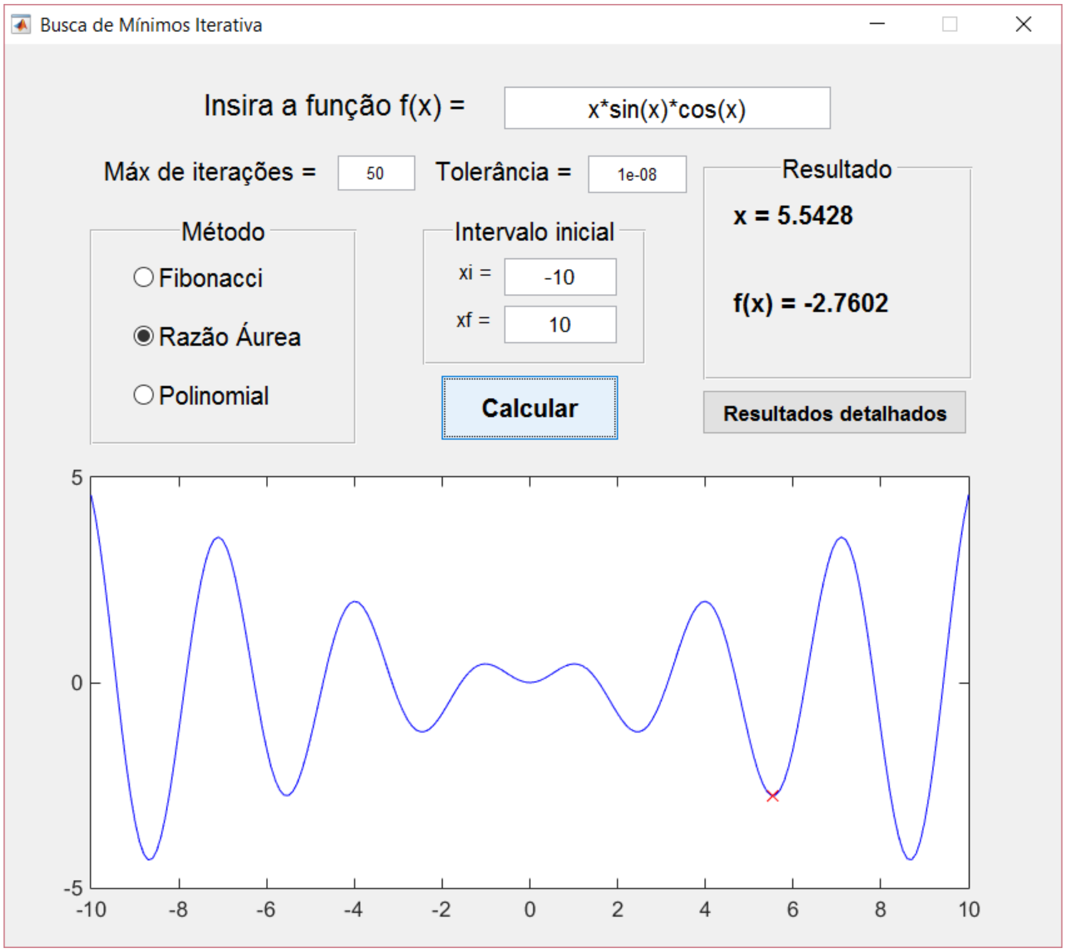


Figura 17: Janela de inicialização de $f_2(x)$

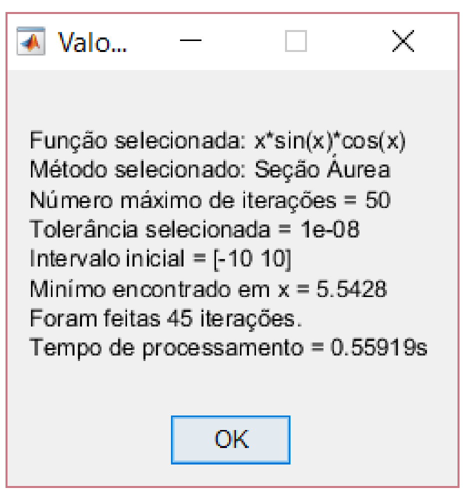


Figura 18: Resultados detalhados de $f_2(x)$

Os resultados para a função $f_3(x)$ foram:

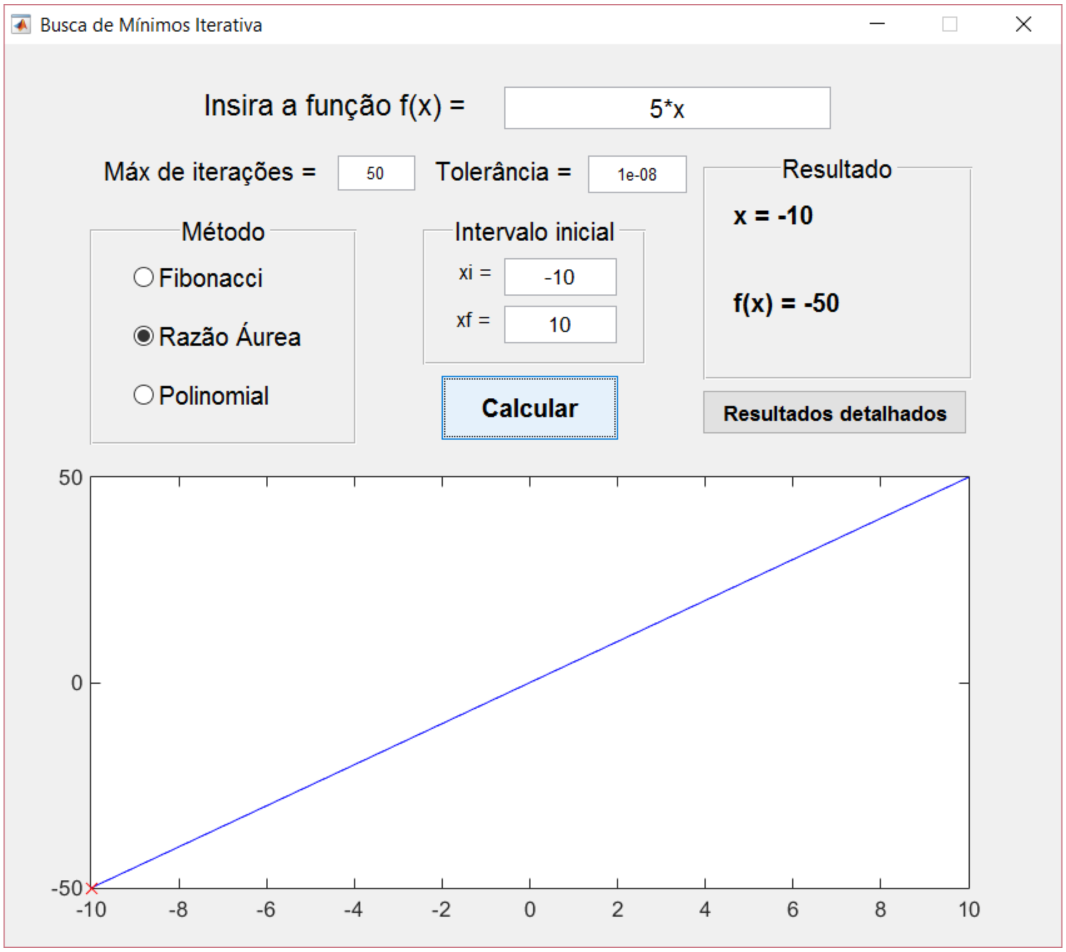


Figura 19: Janela de inicialização de $f_3(x)$

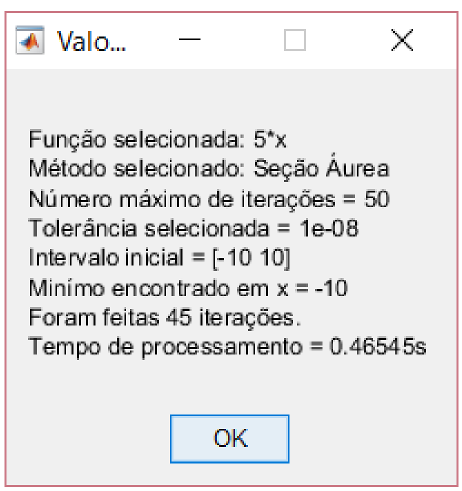


Figura 20: Resultados detalhados de $f_3(x)$

Os resultados para a função $f_4(x)$ foram:

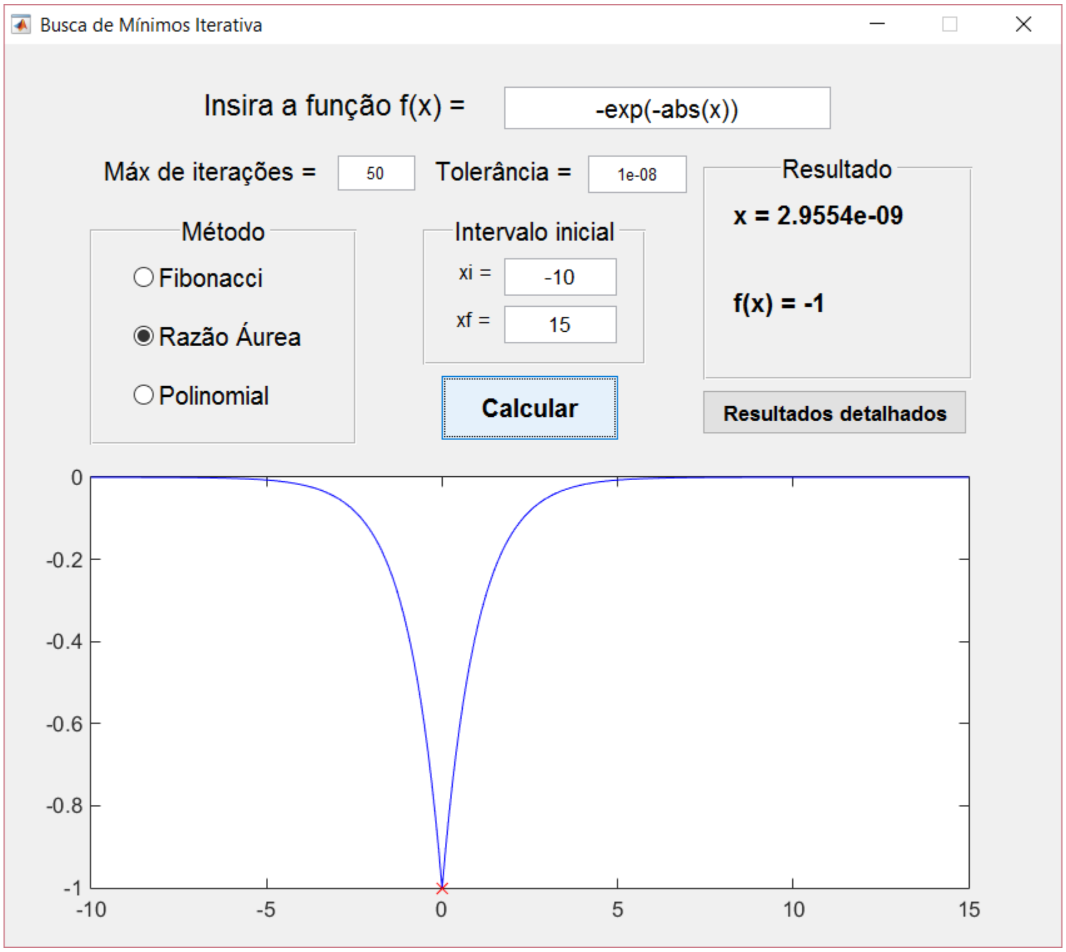


Figura 21: Janela de inicialização de $f_4(x)$

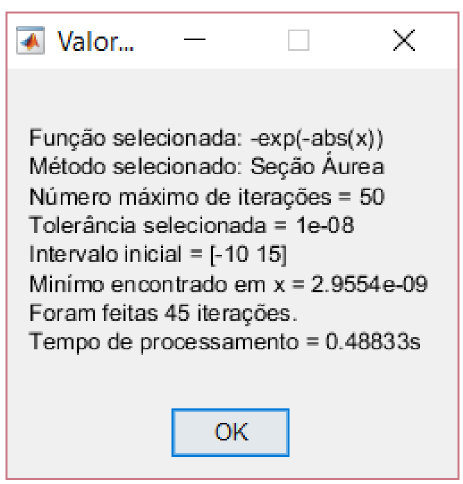


Figura 22: Resultados detalhados de $f_4(x)$

4 Método da Interpolação Polinomial

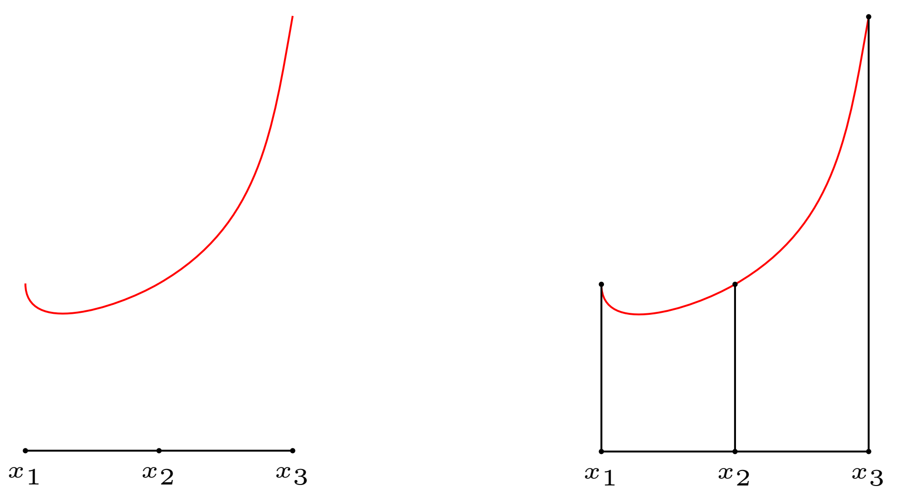
O método de Interpolação Polinomial é mais um método numérico iterativo para minimização de funções unidimensionais. O mesmo consiste em interpolar polinômios de segundo grau (ax^2+bx+c) a partir de 3 pontos da função, e achar o ponto mínimo dessa curva interpolada.

Digamos que queremos minimizar a função mostrada na figura 23a, escolhem-se dois pontos que definem um intervalo onde sabe-se previamente que possui um mínimo, na figura x_1 e x_3, o ponto x_2 é escolhido na metade do intervalo. A partir destes pontos encontra-se o valor da função nesses pontos e resolve-se o sistema $A\vec{x} = \vec{b}$, onde A é a matriz com os valores x^2, x e 1 para cada valor de x (x_1, x_2 e x_3), $\vec{x}$ são os valores dos coeficientes do polinômio interpolado e $\vec{b}$ os valores da função para cada x marcados em 23b.

Após interpolado, vemos a curva azul na figura 24a que representa a interpolação. E a partir dos coeficientes do polinômio sabemos que o mínimo encontra-se no valor em que $x = \frac{-b}{2a}$, ponto x_4 na figura 24b. Com os 4 pontos devemos escolher os três pontos que resultem no menor valor de função para prosseguir com o algoritmo. Escolhendo os pontos x_1, x_2 e x_2 realizamos novamente a interpolação como vemos na figura 25.

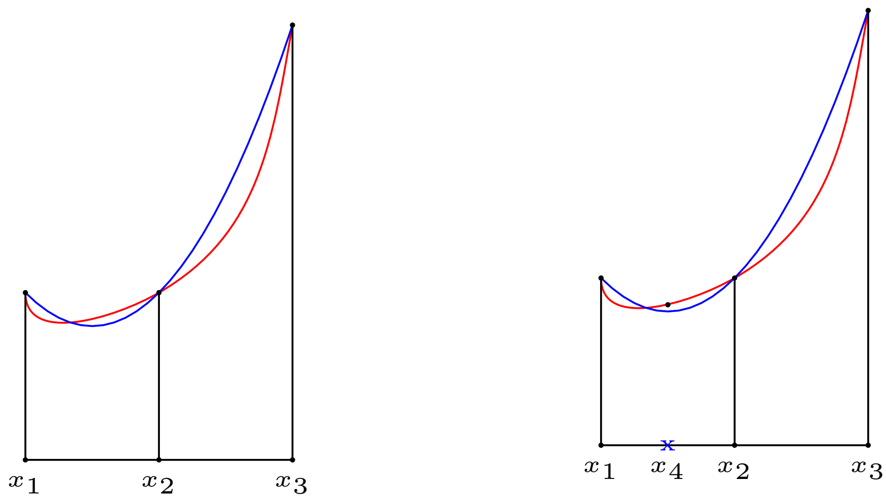
Uma das coisas que podem acontecer é o valor encontrado de mínimo do polinômio interpolado, coincidir com um dos valores de x já usados, assim como acontece na figura 25, uma maneira de contornar este problema é encontrar dois pontos já utilizados e recomeçar o algoritmo, neste caso podemos ver que foi escolhido o intervalo entre x_1 e x_4, em verde na mesma figura.

O algoritmo de interpolação puro é capaz de encontrar diversas dificuldades, devido a necessidade de inverter matrizes e de alguns valores de funções serem muito próximos para todos os três pontos, tornando a matriz mal condicionada. A fim de mostrar algumas dessas dificuldades, foi elaborado o algoritmo puro, sem as correções conhecidas dos métodos de Brent e afins.



(a) Função a ser minimizada **(b)** Valor da função em x_1 x_2 e x_3

Figura 23: Aquisição dos pontos



(a) Curva interpolada em azul

(b) x do mínimo da interpolação em azul e valor da função nele

Figura 24: Interpolação e minimização

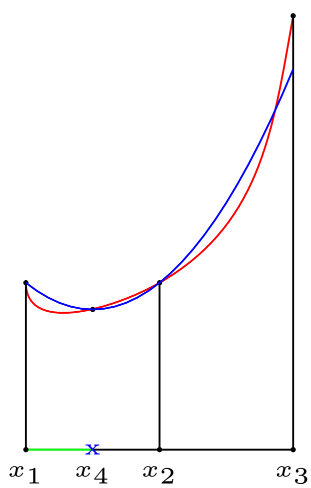


Figura 25: Segunda interpolação e próxima região para interpolação em verde

Depois de construído o algoritmo foram feitos testes com as mesmas funções:

- $f_1(x) = 3x^2 + 20x - 8$
- $f_2(x) = xsin(x)cos(x)$
- $f_3(x) = 5x$
- $f_4(x) = -e^{-|x|}$

E para fim de comparação com os outros métodos, foram usados exatamente os mesmos parâmetros, os resultados podem ser visto nas figuras de número 26 a 33.

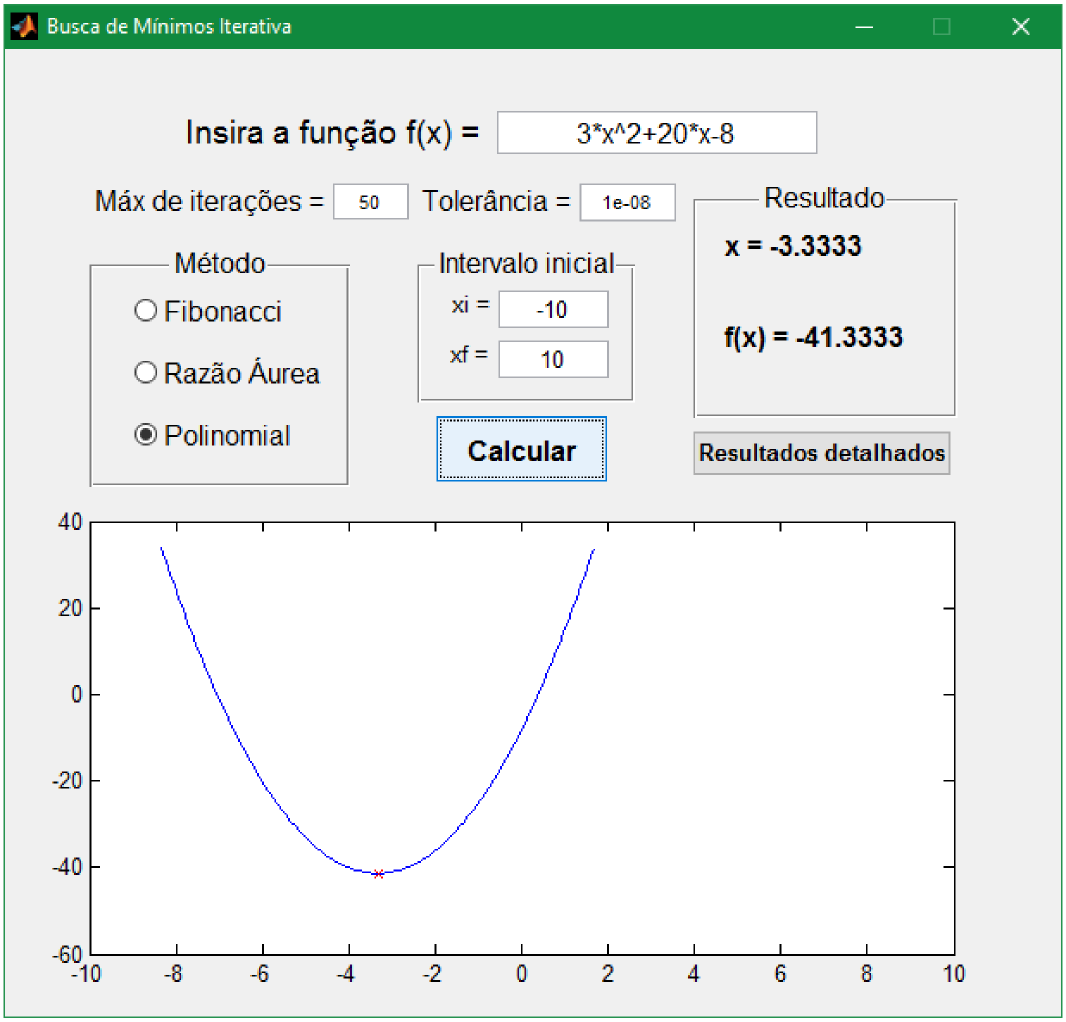


Figura 26: Resultados para a função f_1

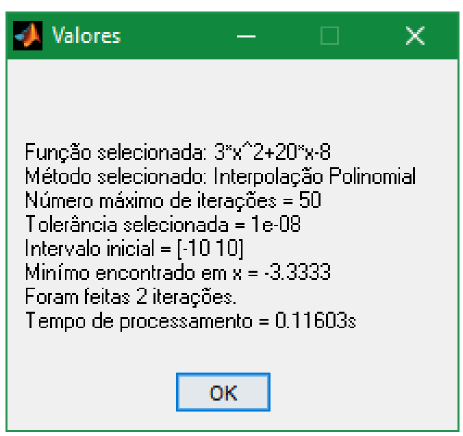


Figura 27: Resultados para a função f_1

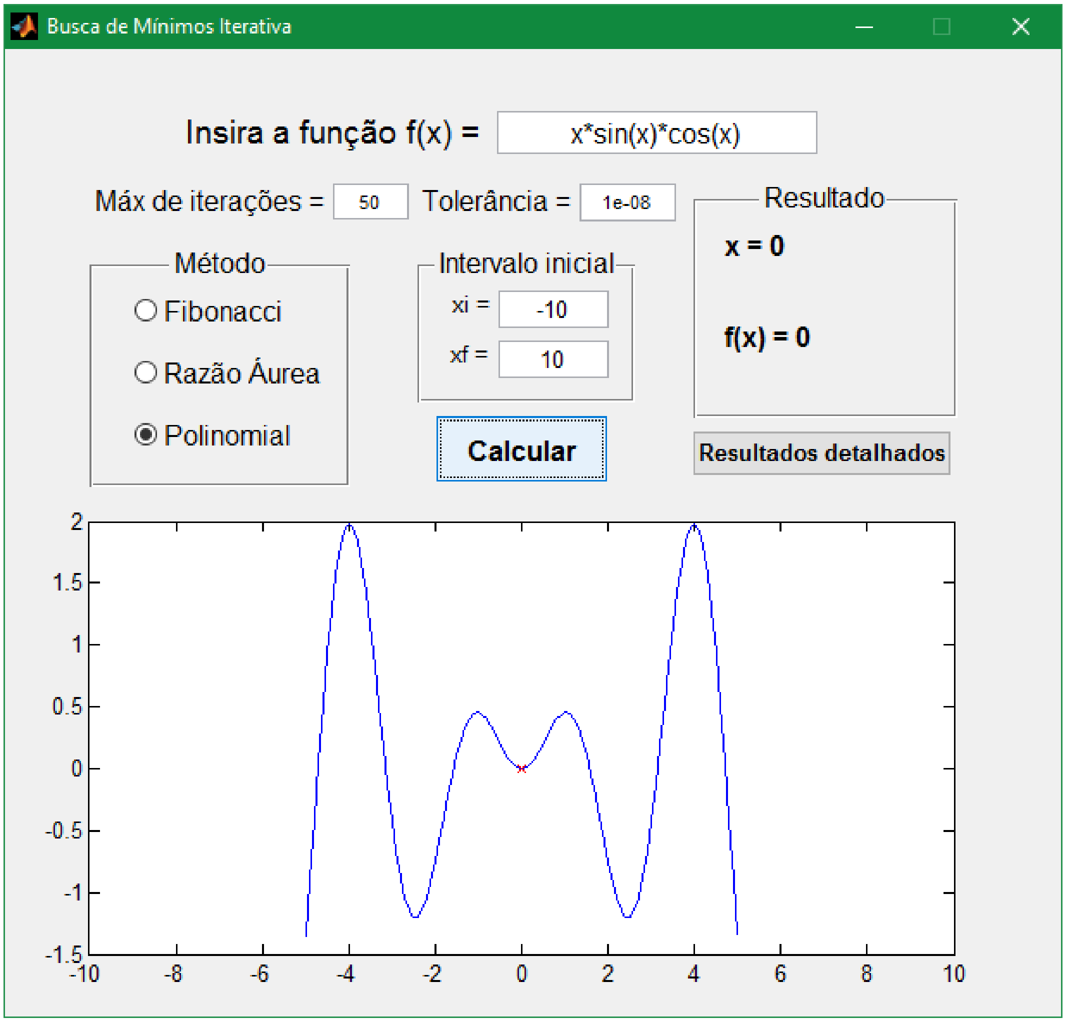


Figura 28: Resultados para a função f_2

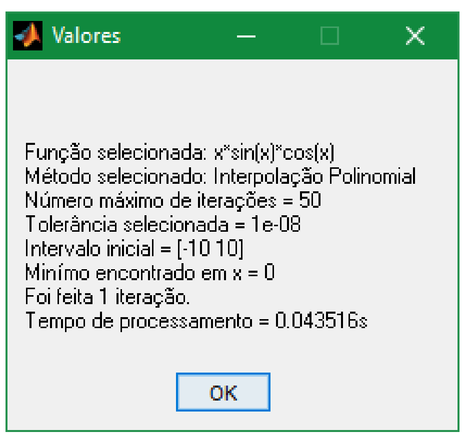


Figura 29: Resultados para a função f_2

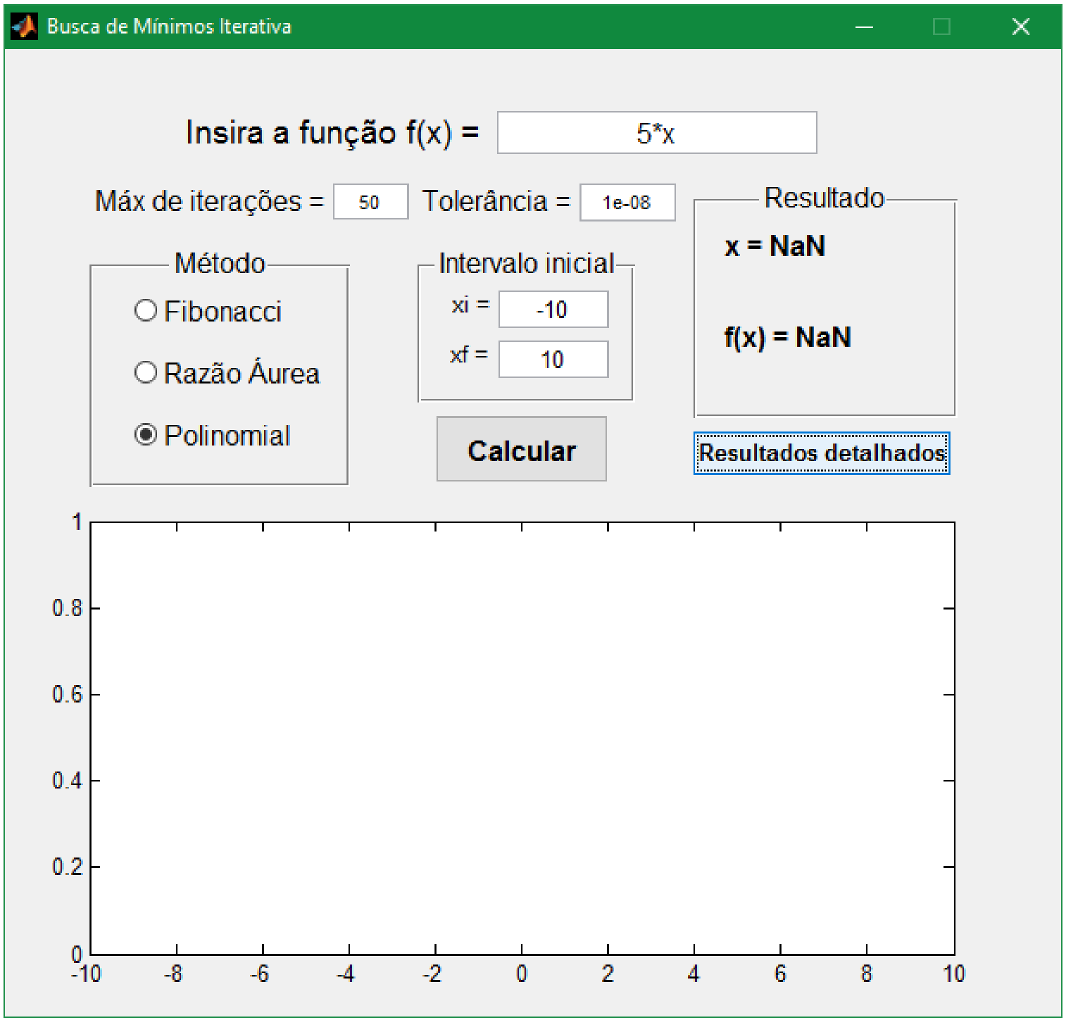


Figura 30: Resultados para a função f_3

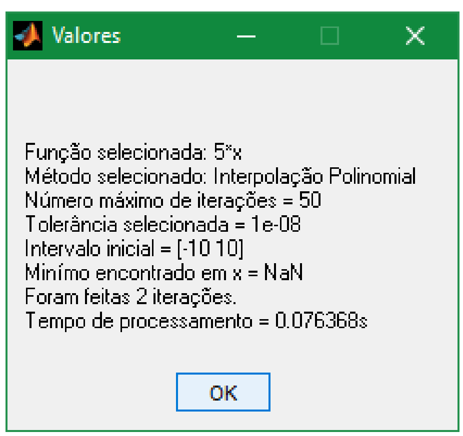


Figura 31: Resultados para a função f_3

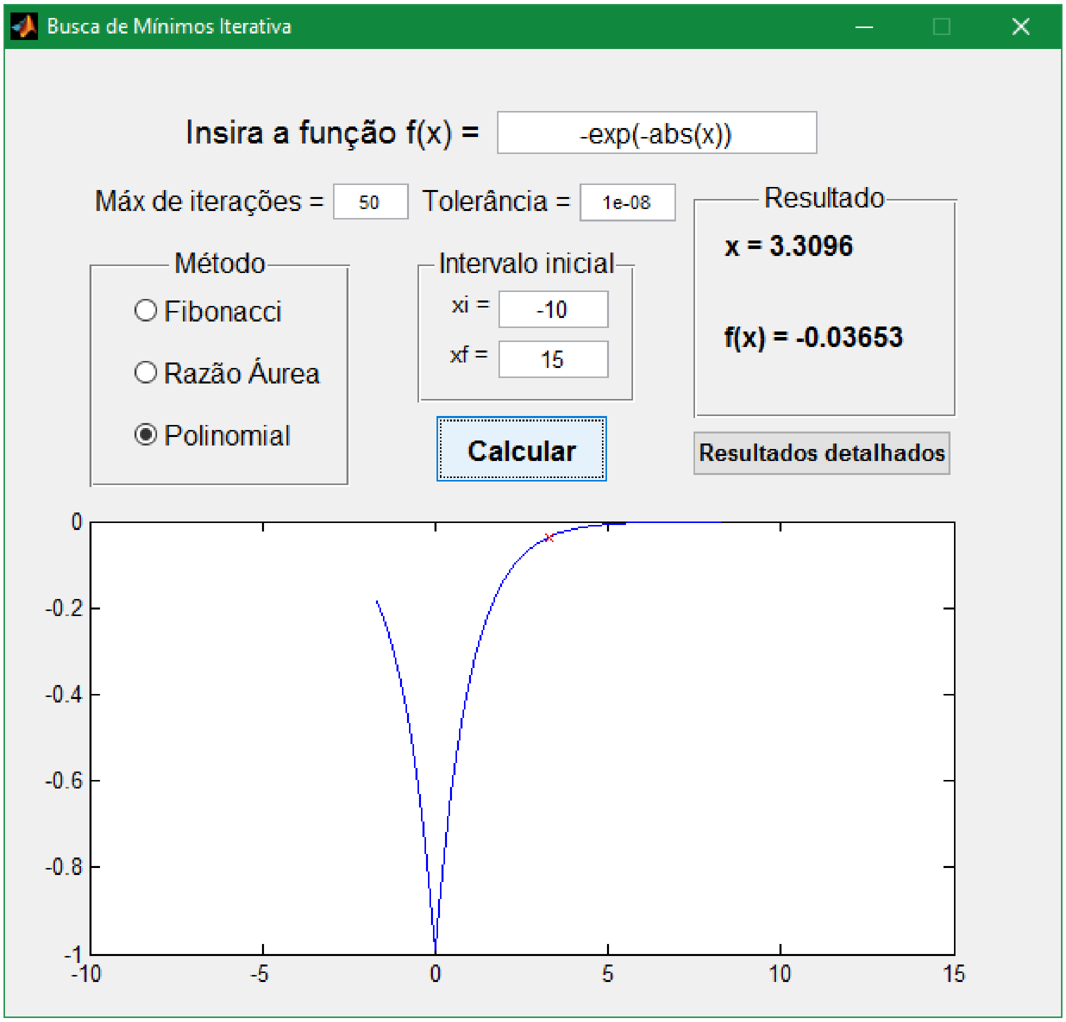


Figura 32: Resultados para a função f_4

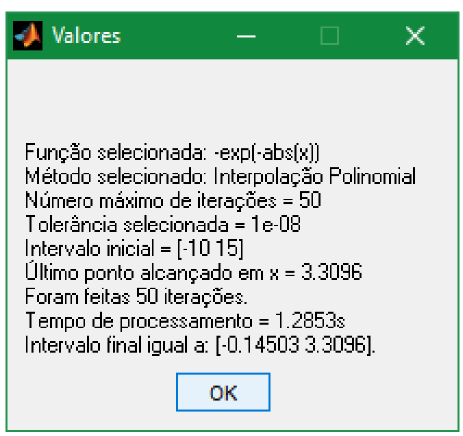


Figura 33: Resultados para a função f_4

5 Interface Gráfica

Após desenvolver os métodos descritos nas seções 2, 3 e 4 como funções do MATLAB, foi implementada uma interface gráfica, tornando a utilização desses métodos mais amigável ao usuário.

Essa interface foi desenvolvida no próprio MATLAB, em um ambiente chamado *GUIDE* (Graphical User Interface Development Environment) que permite a criação da janela do programa com poucos cliques, conforme ilustrado na figura 34. A programação dos elementos da interface é feita em um arquivo *.m que está ligado a interface gráfica, armazenada em uma figura do matlab (*.fig).

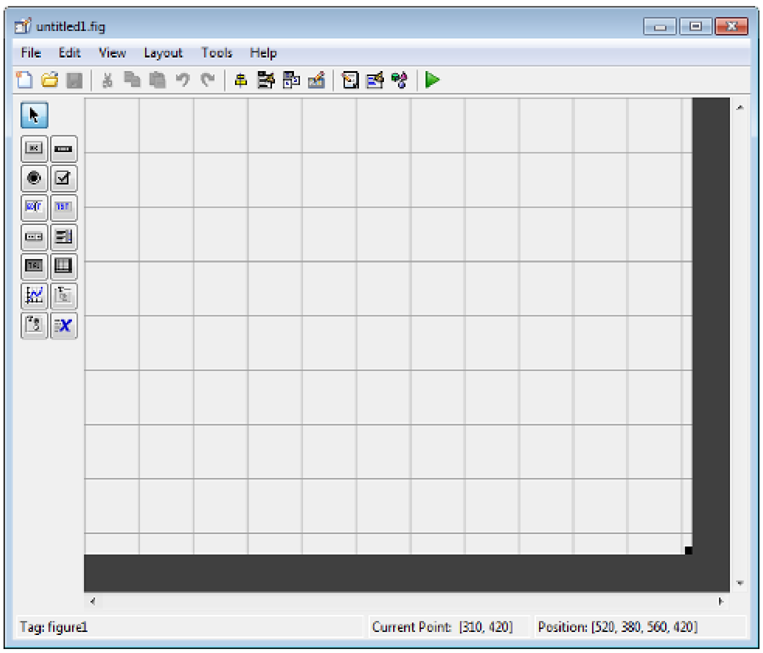


Figura 34: Ambiente de desenvolvimento de interface gráfica *GUIDE*.

O programa desenvolvido foi denominado **Busca de Mínimos Iterativa** e pode ser visto sua janela inicial na figura 35.

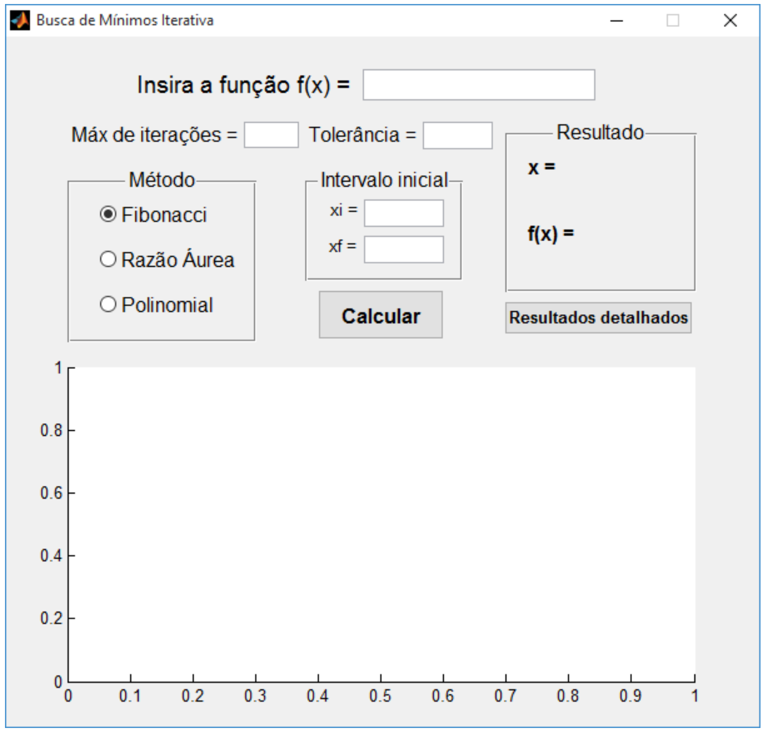


Figura 35: Janela do programa de Busca de Mínimos Iterativa.

O usuário começa colocando a expressão da função que ele deseja minimizar, lembrando de pontuar as multiplicações com um asterisco (*). Deve se definir então, **o método** a ser utilizado; o **máximo de iterações**, para que o programa não rode indefinidamente em casos em que não haja convergência; a **tolerância** desejada para o resultado procurado; e o **intervalo** onde o mínimo será buscado.

Feito isso, basta clicar no botão **Calcular** e os resultados serão exibidos ao lado, o valor de x que dá o mínimo da função e o valor minímo nesse ponto. Além disso, também é exibido o **gráfico da função** dentro do intervalo pedido e o mínimo é destacado com um asterisco vermelho (*).

6 Conclusão

Como podemos ver pelos resultados da busca de mínimos em cada seção, não existe um método melhor que os outros em todos os aspectos, mas para certas funções, os métodos podem ser qualificados.

Para a função $f_1(x) = 3x^2 + 20x - 8$, podemos dizer que todos os métodos convergiram para o valor de mínimo, porém o método da interpolação foi o mais rápido e com menos iterações utilizadas. Entre Fibonacci e Áurea, o segundo foi um pouco mais rápido.

Para a função $f_2(x) = xsin(x)cos(x)$, os resultados foram semelhantes aos da $f_1(x)$, com a interpolação sendo a mais rápida. Os três métodos possuem algoritmos de construção diferentes, e por isso, podemos ver que com uma função com mais de um mínimo, como a $f_2(x)$, cada método pode encontrar um mínimo diferente.

Para a função $f_3(x) = 5x$, podemos ver que o método da interpolação falha em encontrar um mínimo, porque a função escolhida só possui mínimo na restrição escolhida, e assim, o algoritmo da interpolação não consegue construir uma parábola para localizar este valor. Os outros dois métodos conseguem encontrar um valor, sendo o método da Razão Áurea um pouco mais rápido.

Para a função $f_4(x) = -e^{-|x|}$, o método da interpolação fez todas as iterações e não conseguiu chegar perto do valor do mínimo, isso se deve porque o intervalo inicial dado ao método foi não-simétrico, e por isso, o algoritmo de interpolação falhou em encontrar um valor satisfatório para esse número de iterações. Já os métodos de Fibonacci e Áurea convergiram, sendo o primeiro ligeiramente mais rápido.

Podemos concluir que nenhum método é ideal para todas as funções existentes, mas pudemos perceber que, para certas funções, existem métodos melhores que outros. Além disso, o método da interpolação, que parece ser mais rápido que os outros dois, nem sempre converge, fazendo assim com que os métodos da Razão Áurea e de Fibonacci sejam mais seguros.