MÉTODOS NUMÉRICOS PARA calcular zeros de funções

Evandro Pedro Alves de Mendonçaa, Marcelino José de Lima Andradea.

a Núcleo de Tecnologia (NTI), Universidade Federal de Pernambuco (UFPE), Campus Acadêmico do Agreste (CAA), Rodovia BR-104, km 59, S/N, Nova Caruaru, CEP. 55.014-900, Caruaru-PE, Brasil, <http://www.ufpe.br/caa>

**Palavras Chave:** raízes de funções, zeros de funções, métodos numéricos, convergência, gráficos, aproximações.

**Resumo**. Na ciência, é muito comum encontrar valores para os quais uma determinada função é igual a zero. Estes valores são chamados de zeros da função, ou raízes da função. Muitas vezes, é complicado e dispendioso demais calculá-los analiticamente e, por isso, existem métodos numéricos que fornecem valores aproximados das raízes e que são aceitáveis, pois estão dentro de um limite tolerável de erro. Este trabalho se destina, portanto, a resolver alguns problemas utilizando, para isso, métodos numéricos para determinação de zeros de funções. Ao longo do trabalho será feita uma discussão em torno desses métodos, abordando vantagens e desvantagens de cada um e como é feita a implementação deles utilizando o MATLAB. Também serão feitas comparações entre os métodos implementados neste trabalhos e os que já são nativos do MATLAB.

1. INTRODUção

Em várias áreas das ciências exatas encontram-se constantemente situações onde é necessário resolver um problema com a seguinte forma:

(1)

Algumas vezes, a resolução de um problema desse tipo pode ser feita de forma analítica, como é o caso das equações polinomiais de 1º e 2º grau. Porém, ao lidar com equações polinomiais de 3º grau em diante, ou quando trabalha-se com funções mais complexas, como seno, cosseno, a função logarítmica, entre outras, torna-se mais complicada a resolução do problema.

Graficamente, a resolução do problema representado na equação (1) acima é ponto onde a função *f(x)* intercepta o eixo das abcissas, ou eixo *x*. Esses pontos são as chamadas raízes da função. Isso é mostrado na imagem abaixo, onde *x'* e *x''* são as raízes da função.

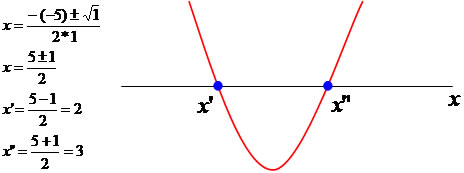


Figura 1: raízes de um polinômio de segundo grau.  
Adaptado de <http://mundoeducacao.bol.uol.com.br/upload/conteudo/Untitled-7(12).jpg>

Como já dito, em alguns casos, o cálculo analítico das raízes de uma função é inviável. Uma saída para essa problema é a utilização de métodos numéricos para a determinação das raízes. Inicialmente, procura-se dar uma estimativa para a raiz da função. Em seguida, é aplicado um método para aperfeiçoar essa estimativa inicial. É importante observar que, nos métodos numéricos, nem sempre é possível fazer uma estimativa exata para a raiz, diferente do cálculo analítico. Muitas vezes apenas será possível realizar uma boa estimativa da raiz.

Os métodos de estimativa de raízes dividem-se em dois grupos, são eles: os métodos de confinamento, e os métodos abertos. Os métodos de confinamento requerem um intervalo inicial em que a raiz da função se encontra em seu interior. Os métodos abertos consistem em, a partir de um palpite de solução inicial, são feitas manipulações numéricas para o cálculo da raiz. No método aberto, essa estimativa inicial deve ser próxima da raiz exata.

Independentemente do método utilizado, é necessário estabelecer uma tolerância no cálculo da raiz, visto que, em alguns casos, o cálculo exato é praticamente impossível, além de que, o custo computacional é extremamente alto. Em termos simples, a tolerância seria o quanto a solução encontrada pode desviar-se da solução exata. Obviamente, quanto menor for a tolerância permitida, menor será o erro associado. Porém, uma boa escolha do método a ser utilizado leva a um menor custo computacional, e a um menor erro associado. Alguns métodos são listados a seguir:

* Bisseção – método de confinamento
* Falsa posição – método de confinamento
* Ponto fixo – método aberto
* Newton-Raphson – método aberto
* Secante – método aberto

A escolha do método a ser utilizado depende do problema a ser resolvido e da precisão e exatidão necessárias. Além disso, existem casos específicos onde alguns desses métodos não funcionam, ou se distanciam muito da solução exata. Uma análise do procedimento usado por cada método leva a uma boa escolha de qual deve ser utilizado. O detalhamento de cada método é feito nos exercícios propostos deste trabalho.

1. Exercícios Propostos

Segue, abaixo, a solução dos exercícios propostos sobre o tema.

* 1. 1ª questão

Utilizando o algoritmo que consta no Anexo 1, foram encontrados os seguintes resultados.

* 1º intervalo [0,2]: raiz = 3,675460815429688 x 10-1;
* 2º intervalo [2,4]: raiz = 3,767829895019531;
* 3º intervalo [4,6]: raiz = 5,954948425292969;

Detalhes no Anexo 6.

* 1. 2ª questão

Utilizando o algoritmo que consta no Anexo 1, foram encontrados os seguintes resultados.

* Letra a.
  + Primeiro intervalo [-3/2, 5/2]: raiz = 0;
  + Segundo intervalo [-1/2, 12/5]: raiz = 2,861022949203437 x 10-6;
  + Terceiro intervalo [-1/2, 3]: raiz = 1,999997138977051;
  + Quarto intervalo [-3, -1/2]: raiz = -2,000005722045898;

Detalhes no Anexo 7.

* Letra b.

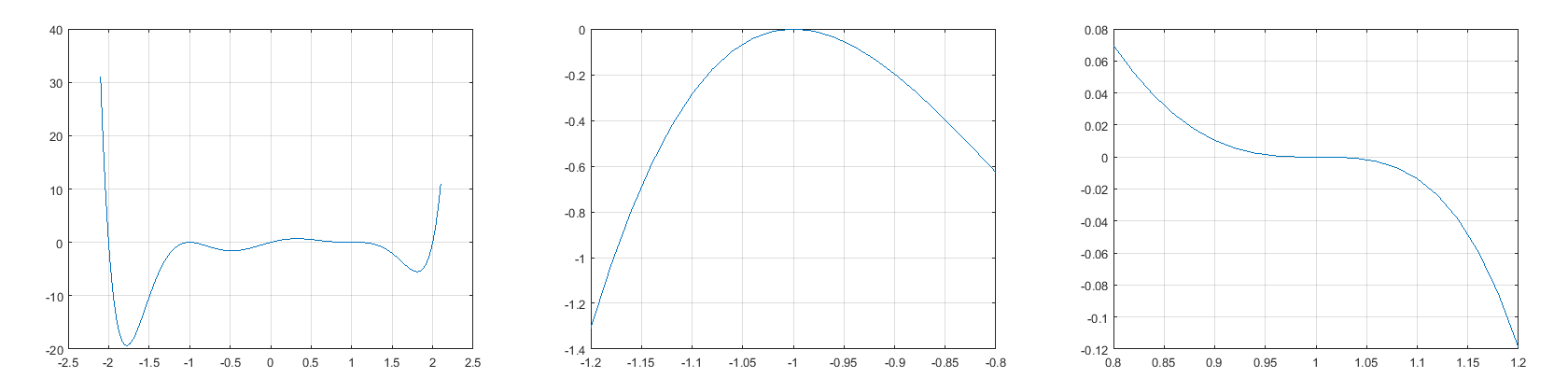


Figura 2: à esquerda, gráfico da função na vizinhança da raiz x = -1; à direita, da raiz x = 1.

Analisando o gráfico da função mostrado na Figura 2, não fica muito claro se é possível ou não aplicar o método da bisseção para as raízes -1 e 1. Portanto, foram gerados outros dois gráficos, dispostos na Figura 2, que mostram o que acontece na imagem da função na vizinhança dessas duas raízes. Dessa forma, é possível ver que na raiz x = -1 qualquer par de pontos tomados em sua vizinhança terá imagens negativas em ambos os pontos, e não será possível aplicar o método da bisseção, uma vez que ele exige um par de pontos com imagens de sinais contrários. Já na raiz x = 1, observa-se que as imagens terão sinais contrários, sendo, assim, possível aplicar o método da bisseção para encontrá-la.

* 1. 3ª questão

Primeiramente, vejamos o gráfico da função.

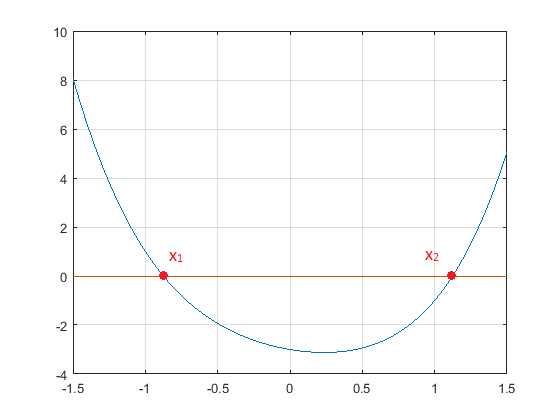


Figura 3: gráfico da função dada na questão entre os pontos x = -1,5 e x = 1,5. A raiz próxima ao ponto x = -1 será chamada de x1 e a raiz próxima ao ponto x = 1 será chamada de x2.

Para determinar a convergência da função de iteração em cada caso, fazemos uma plotagem dos gráficos de cada função em contraste com a função identidade (y = x) e observamos a interseção entre elas. O ponto de interseção determina o valor da raiz.

Primeiramente, a função da letra (A):

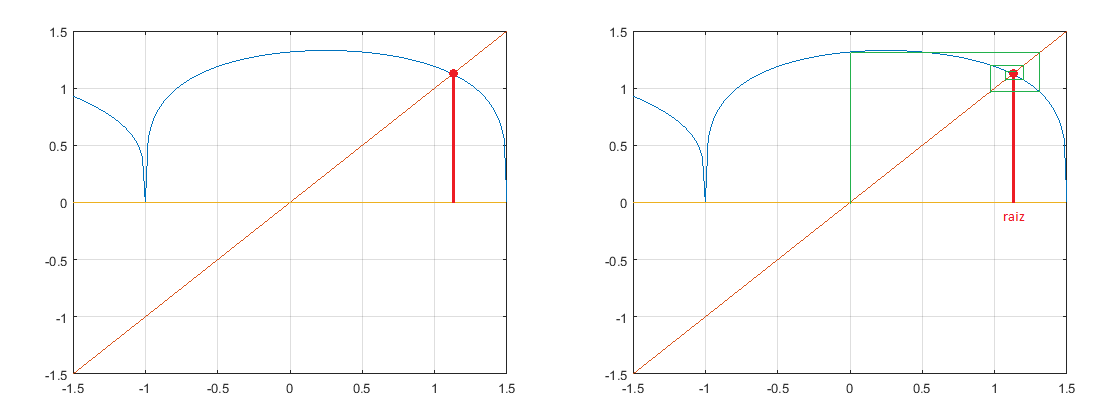


Figura 4: à esquerda, vemos o ponto de interseção entre a função de iteração e a reta y = x; à direita, o procedimento gráfico do método do ponto fixo para encontrar a raiz a partir de retas horizontais e verticais tendo como estimativa inicial o ponto xe = 0.

Com a análise dos gráficos, vemos que a raiz dessa função de iteração converge apenas para o ponto próximo a x = 1. Vejamos agora o que acontece na função de iteração da letra (B):

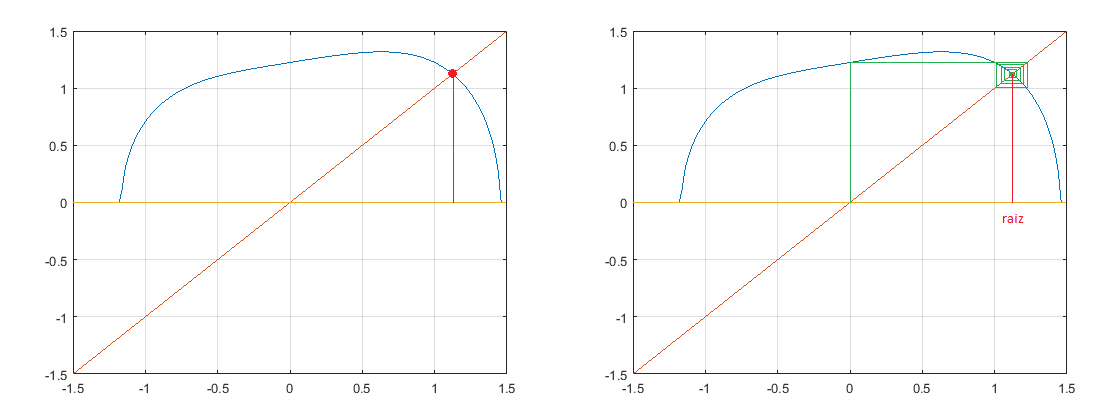


Figura 5: à esquerda, vemos o ponto de interseção entre a função de iteração e a reta y = x; à direita, o procedimento gráfico do método do ponto fixo para encontrar a raiz a partir de retas horizontais e verticais tendo como estimativa inicial o ponto xe = 0.

Percebemos que as duas funções de iteração têm comportamento semelhante. Ambas convergem apenas para a raiz próxima ao ponto x = 1. Assim, respondendo à pergunta do item 3.a), não utilizaria nenhuma delas para encontrar a raiz x1, pois nenhuma delas converge para esse ponto; já para a raiz x2, ambas convergem, mas a função de iteração da letra (A) converge muito mais rápido, então essa seria a escolha para encontrar a raiz x2.

* 1. 4ª questão

Para esta questão, foi utilizado o algoritmo que consta no Anexo 4.

* 4.a)
  + Intervalo: 2 ≤ x ≤ 3

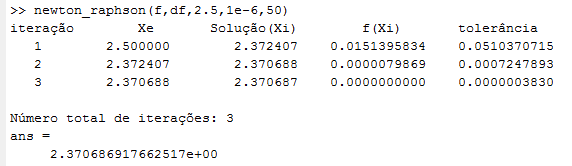


Figura 6: resultado para cálculo de raiz de f(x) utilizando método de Newton-Raphson com estimativa de 2,5.

* + Intervalo: 3 ≤ x ≤ 4

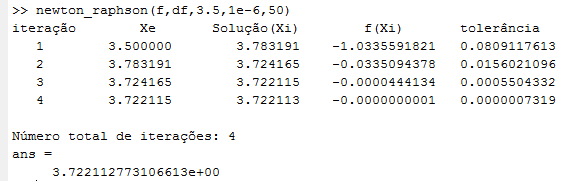


Figura 7: resultado para cálculo de raiz de f(x) utilizando método de Newton-Raphson com estimativa de 3,5.

* 4.b)
  + Intervalo: 0 ≤ x ≤ 1

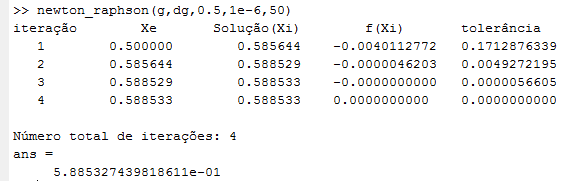


Figura 8: resultado para cálculo de raiz de g(x) utilizando método de Newton-Raphson com estimativa de 0,5.

* + Intervalo: 3 ≤ x ≤ 4

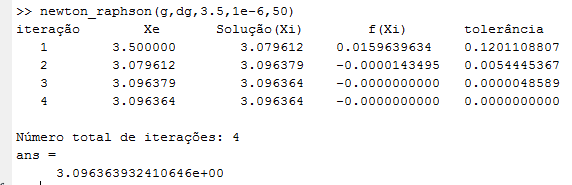


Figura 9: resultado para cálculo de raiz de g(x) utilizando método de Newton-Raphson com estimativa de 3,5.

* 4.c)
  + Intervalo: 1,3 ≤ x ≤ 2

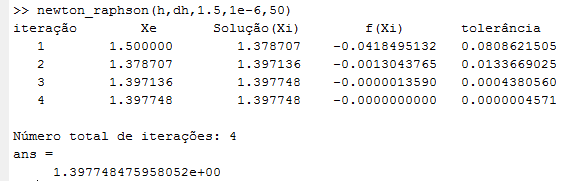


Figura 10: resultado para cálculo de raiz de h(x) utilizando método de Newton-Raphson com estimativa de 1,5.

A ordem de convergência do método de Newton-Raphson é quadrática, como pode ser observado nos gráficos que constam no Anexo 8, juntamente com outros detalhes mais.

* 1. 5ª questão

Para esta questão, foi utilizado o algoritmo que consta no Anexo 3. Mais detalhes, vide Anexo 9.

* 5.a) Intervalos: 2 ≤ x ≤ 3 e 3 ≤ x ≤ 4

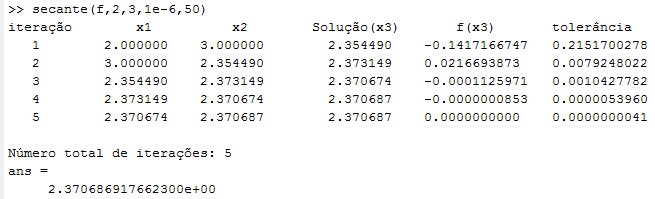


Figura 11: resultado para cálculo de raiz de f(x) utilizando método da Secante com pontos 2 e 3.

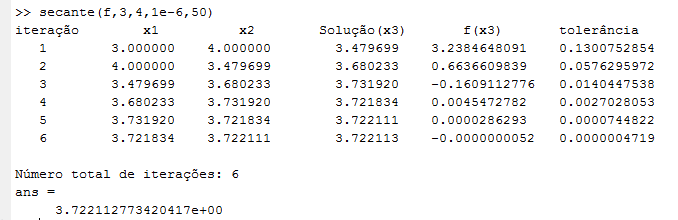


Figura 12: resultado para cálculo de raiz de f(x) utilizando método da Secante com pontos 2 e 3.

* 5.b) Intervalos: 0 ≤ x ≤ 1 e 3 ≤ x ≤ 4

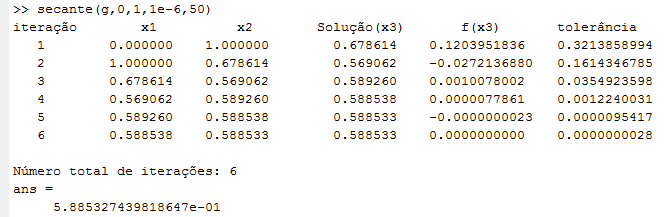


Figura 13: resultado para cálculo de raiz de g(x) utilizando método da Secante com pontos 0 e 1.

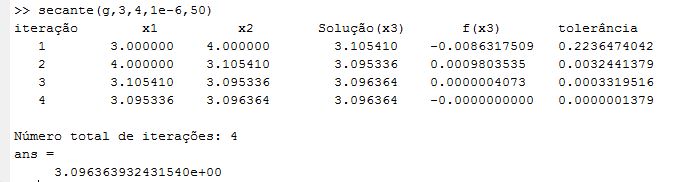


Figura 14: resultado para cálculo de raiz de g(x) utilizando método da Secante com pontos 3 e 4.

* 5.c) Intervalo: 1,3 ≤ x ≤ 2

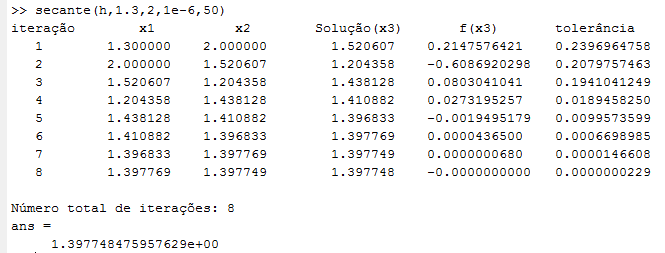


Figura 15: resultado para cálculo de raiz de h(x) utilizando método da Secante com pontos 1,3 e 2.

* 1. 6ª questão

Para esta questão, foram utilizados os algoritmos que constam nos Anexos 3 e 4, por terem o melhor desempenho na solução de funções não lineares. Mais detalhes, como gráfico e execução dos comandos no MATLAB, vide Anexo 10.

Função:

Derivada:

* 2.a)
  + 1ª raiz com o método de Newton-Raphson:
    - Raiz: 4,506567478899358 x 10-1;
    - Tempo de processamento: 0,032329s;
    - Iterações: 4.
  + 1ª raiz com o método da Secante:
    - Raiz: 4,506567478888424 x 10-1;
    - Tempo de processamento: 0,025429s;
    - Iterações: 5.
  + 2ª raiz com o método de Newton-Raphson:
    - Raiz: 1,744738053368827;
    - Tempo de processamento: 0,008068s;
    - Iterações: 3.
  + 2ª raiz com o método da Secante:
    - Raiz: 1,744738053368886;
    - Tempo de processamento: 0,020020s;
    - Iterações: 7.
  + 3ª raiz com o método de Newton-Raphson:
    - Raiz: 2,238319795074177;
    - Tempo de processamento: 0,004335s;
    - Iterações: 3.
  + 3ª raiz com o método da Secante:
    - Raiz: 2,238319795040133;
    - Tempo de processamento: 0,009471s;
    - Iterações: 7.
* 2.b) Para encontrar raízes de funções não lineares, o MATLAB fornece a função *fzero*, que utiliza o método da Bisseção.
  + 1ª raiz com a função *fzero*:
    - Raiz: 4,506567478899357 x 10-1;
    - Tempo de processamento: 0,010251s.
  + 2ª raiz com a função *fzero*:
    - Raiz: 1,744738053368827;
    - Tempo de processamento: 0,037282s.
  + 3ª raiz com a função *fzero*:
    - Raiz: 2,238319795074138;
    - Tempo de processamento: 0,006313s.
* 2.c)

Para fazer a comparação entre a letra A e B desse exercício, vimos que a função fzero só fornece a resposta final, sem mostrar o número de iterações nem informar os valores de cada iteração. Dessa forma, nossas funções fornecem mais informações e, por isso, tem um tempo de execução muito maior. Sendo assim, eliminamos as informações de cada iteração e deixamos apenas a apresentação do número de iterações e, com isso, o tempo de execução se tornou menor (para detalhes, vide Anexo 10). Portanto, temos a seguinte tabela de valores para comparação:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Função | Parâmetros | Raiz 1 | Raiz 2 | Raiz 3 |
| Newton-Raphson (antes da alteração) | Valor | 4,506567478899358 x 10-1 | 1,744738053368827 | 2,238319795074177 |
| Iterações | 4 | 3 | 3 |
| Tempo de processamento | 0,032329s | 0,008068s | 0,004335s |
| Newton-Raphson (depois da alteração) | Valor | 4,506567478899358 x 10-1 | 1,744738053368827 | 2,238319795074177 |
| Iterações | 4 | 3 | 3 |
| Tempo de processamento | 0,004730s | 0,009042s | 0,002354s |
| Secante (antes da alteração) | Valor | 4,506567478888424 x 10-1 | 1,744738053368827 | 2,238319795040133 |
| Iterações | 5 | 7 | 7 |
| Tempo de processamento | 0,025429s | 0,020020s | 0,009471s |
| Secante (depois da alteração) | Valor | 4,506567478888424 x 10-1 | 1,744738053368827 | 2,238319795040133 |
| Iterações | 5 | 7 | 7 |
| Tempo de processamento | 0,011602s | 0,002466s | 0,001808s |
| *fzero* | Valor | 4,506567478899357 x 10-1 | 1,744738053368827 | 2,238319795074138 |
| Iterações | Não informado. | Não informado. | Não informado. |
| Tempo de processamento | 0,010251s | 0,037282s | 0,006313s |

Tabela 1: comparação dos resultados obtidos para os 3 métodos testados.

Visualizando os dados da Tabela 1, vemos que, na maioria das vezes, as funções *newton\_raphson* e *secante* tiveram tempo de processamento menor que o de *fzero*. Este comportamento é ainda mais notável quando as funções *newton\_raphson* e *secante* foram modificadas (removendo a impressão de valores de cada iteração) Isto se deve ao fato de que os métodos da Secante e NR (Newton-Raphson) têm taxas de convergência muito maiores que a do método da Bisseção. Comparando agora NR à Secante, vemos que NR geralmente tem tempo de processamento menor que Secante, mas o que chama mais a atenção é que o número de iterações é muito menor. Para esse caso, não faz tanta diferença, mas para situações mais adversas, o número de iterações pode crescer muito e o método NR tende a ser mais vantajoso que o da Secante. A desvantagem, porém, do método NR é que o usuário deve fornecer além da função, sua derivada. O método da Secante não precisa da derivada.

1. conclusão

Após implementação dos códigos propostos e análise dos resultados obtidos, conclui-se que há uma variedade considerável de métodos numéricos disponíveis para se chegar ao mesmo resultado: calcular raízes de funções. Porém, para cada caso, haverá sempre um método melhor que outro, e as situações variam muito, pois isso depende do tipo de função que está sendo usada. É importante estar ciente de questões como tempo de processamento, número de iterações, convergência e precisão. Todos esses parâmetros podem ser ajustados conforme especificações do projetos com o qual se está trabalhando.

REFERÊNCiaS

Chapra, S. C., e Canale, R. P. *Métodos Numéricos para Engenharia*. 5ª edição. Porto Alegre: AMGH, 2011.

Gilat, A., e Subramaniam, V. *Métodos Numéricos para Engenheiros e Cientistas: uma introdução com aplicações usando o MATLAB*. Porto Alegre: Bookman, 2008.

anexo 1

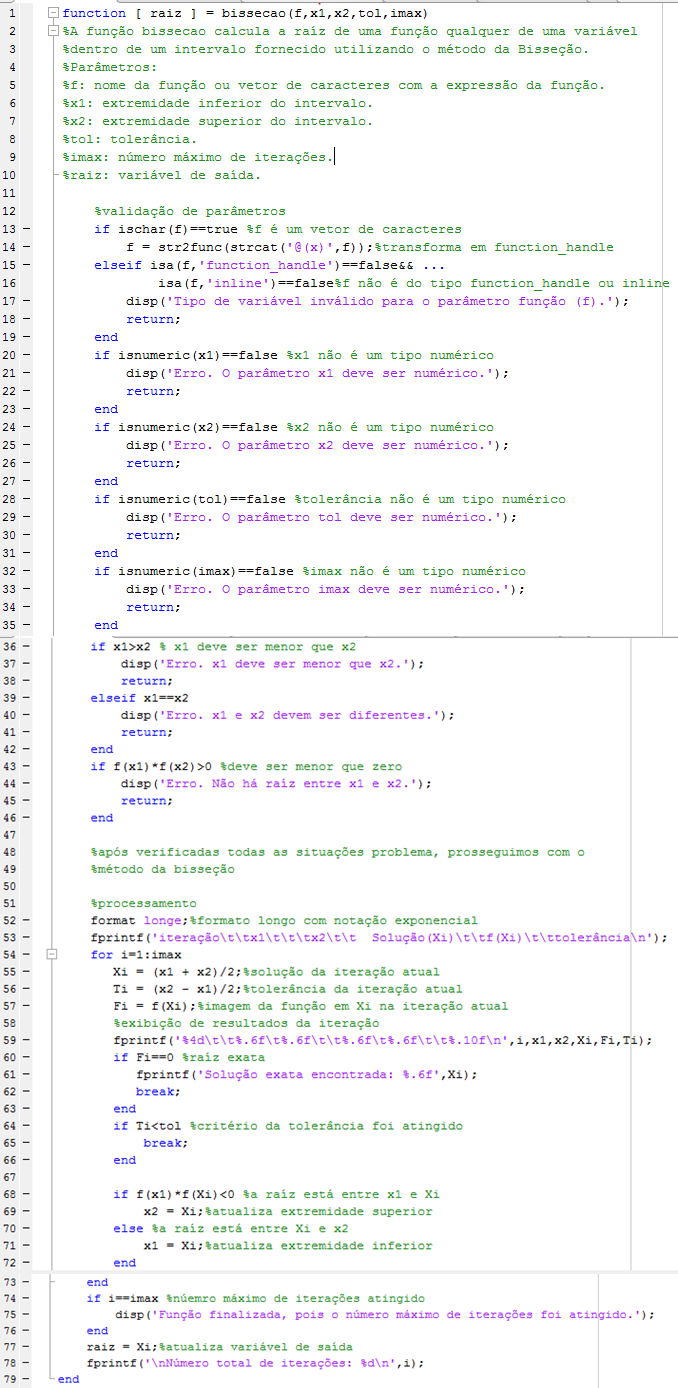


Figura 16: função criada em MATLAB que calcula uma raiz pelo método da Bisseção (parte 1).

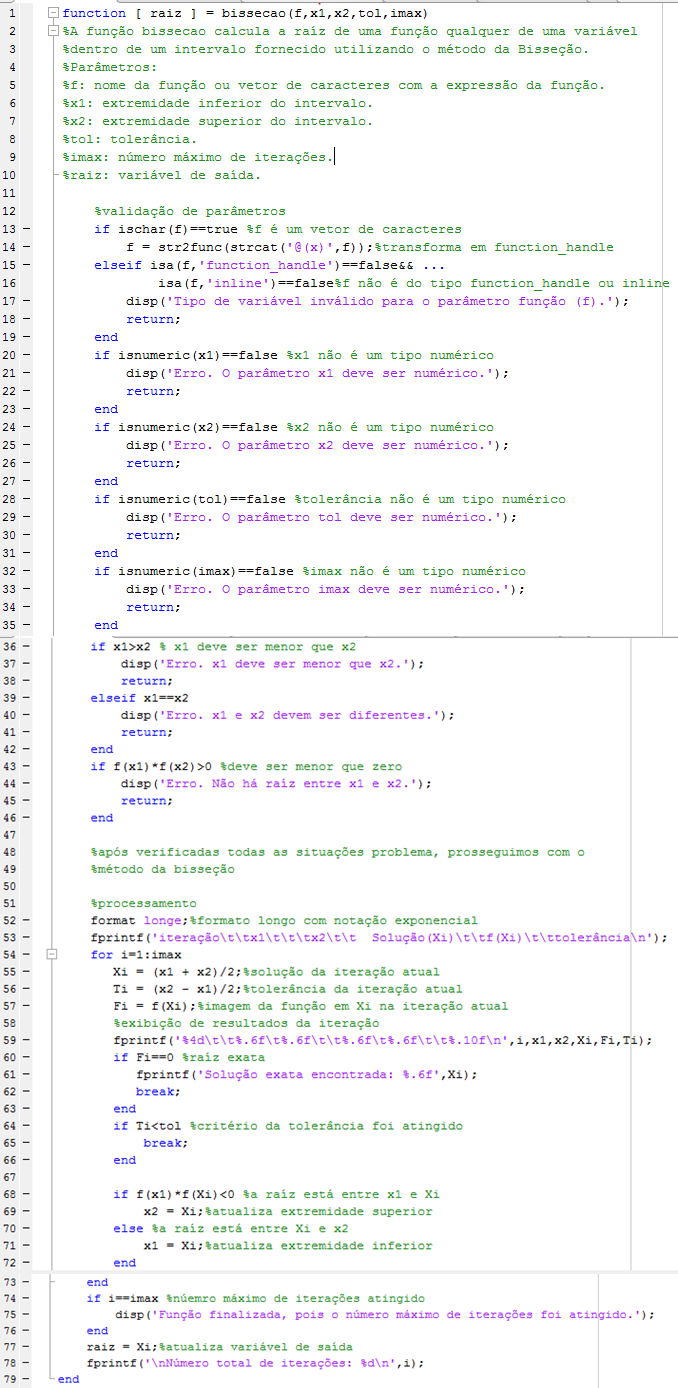


Figura 17: função criada em MATLAB que calcula uma raiz pelo método da Bisseção (parte 2).

anexo 2

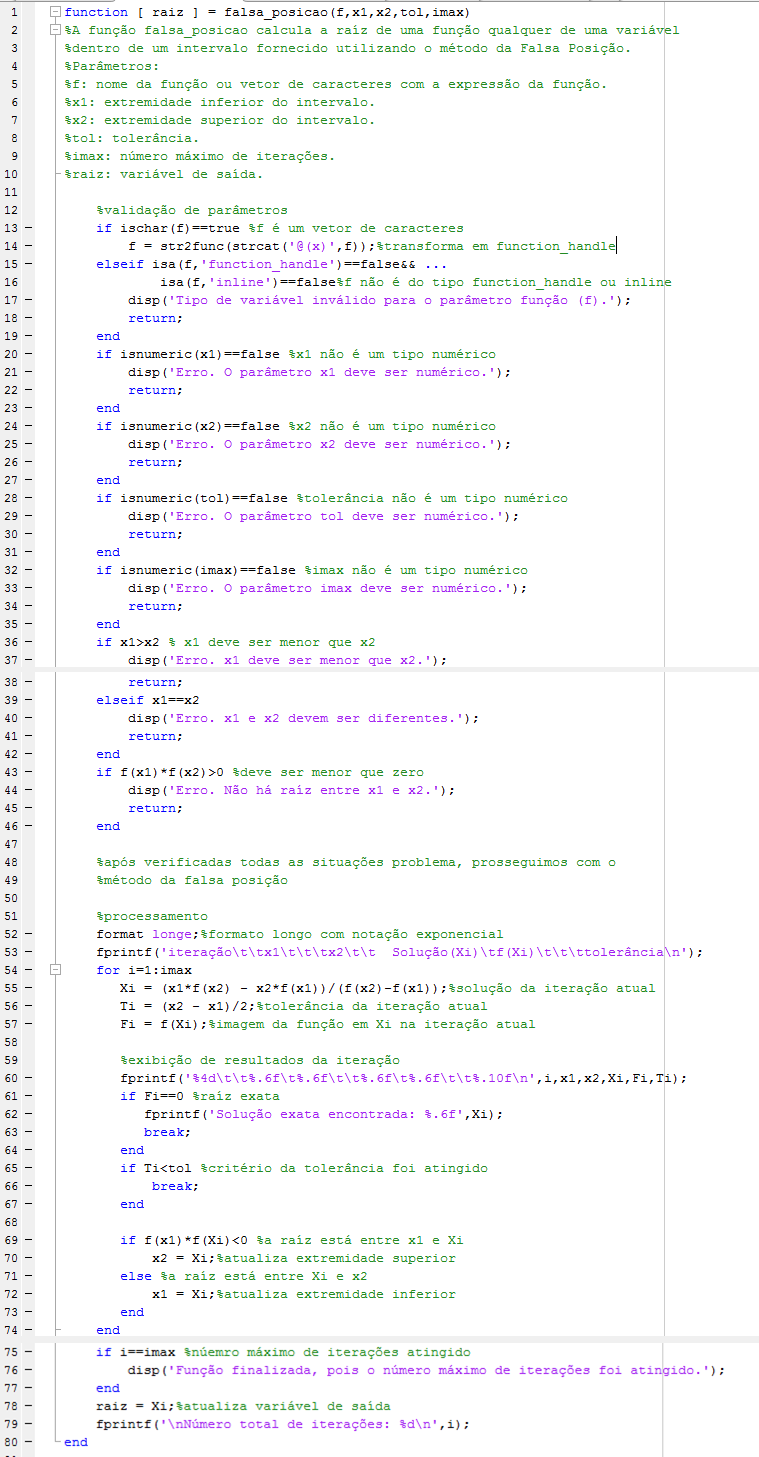


Figura 18: função criada em MATLAB que calcula uma raiz pelo método da Falsa Posição (parte 1).

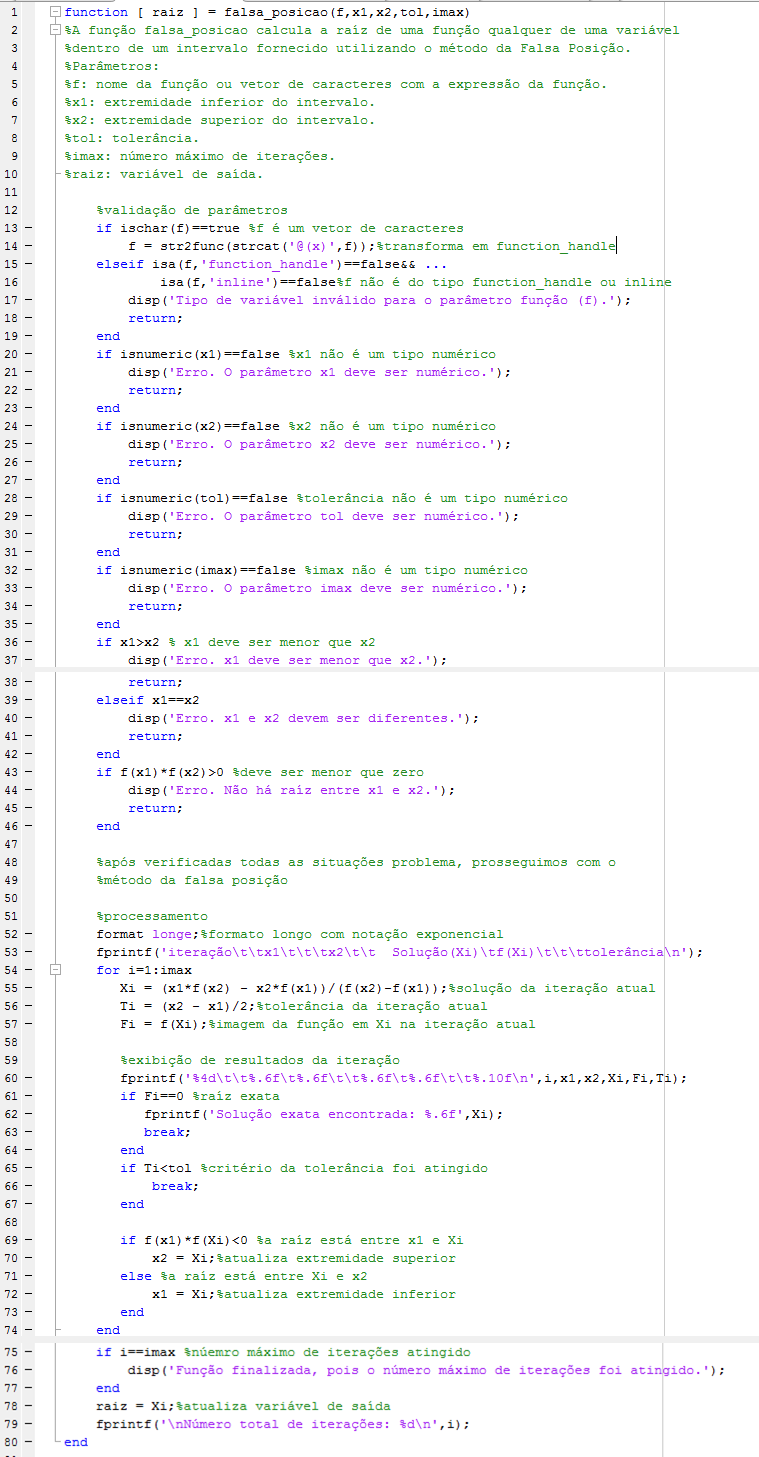


Figura 19: função criada em MATLAB que calcula uma raiz pelo método da Falsa Posição (parte 2).

anexo 3

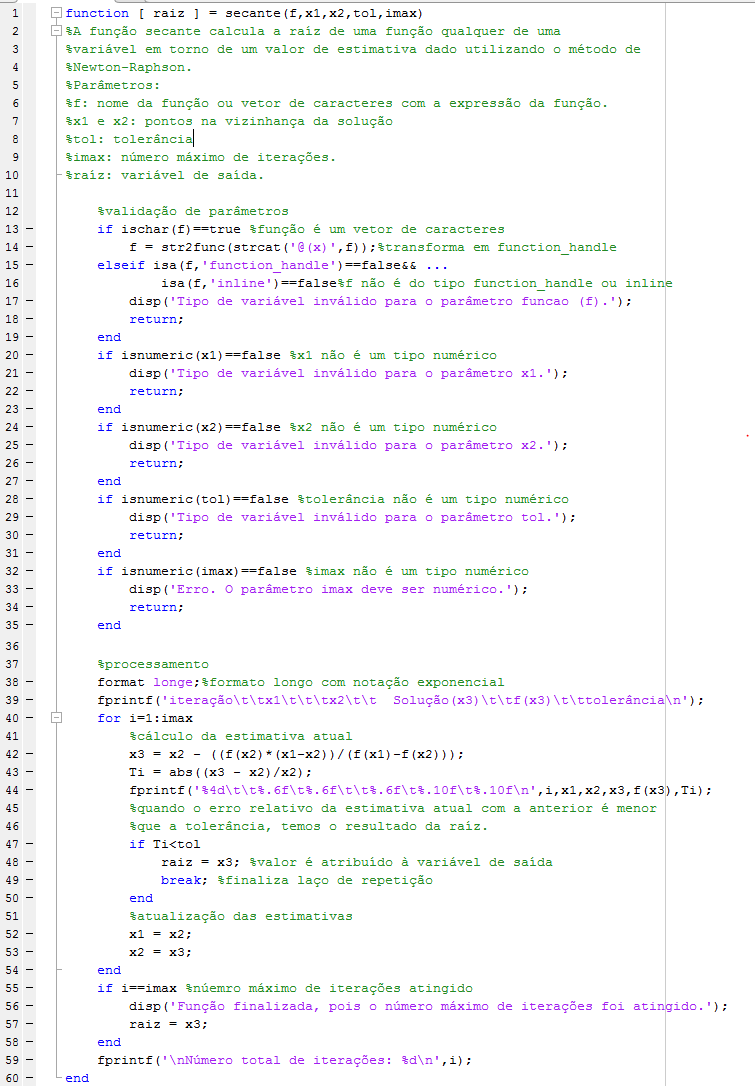


Figura 20: função criada em MATLAB que calcula uma raiz pelo método da Secante.

anexo 4

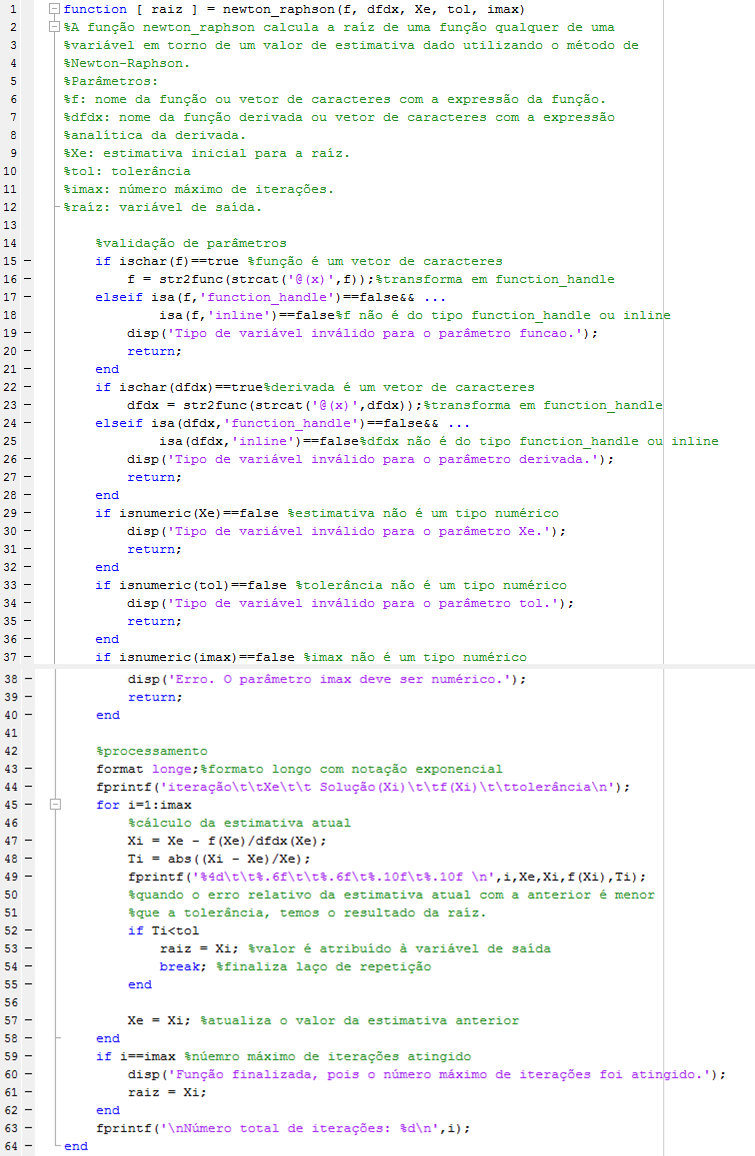


Figura 21: função criada em MATLAB que calcula uma raiz pelo método de Newton-Raphson.

anexo 5

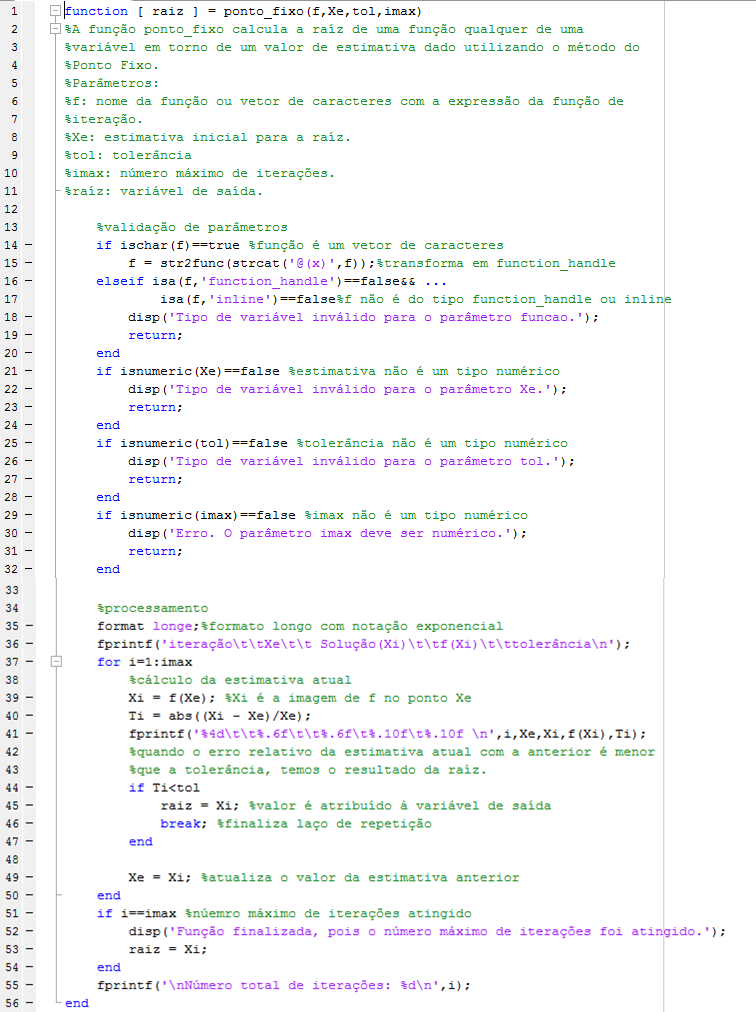


Figura 22: função criada em MATLAB que calcula uma raiz pelo método de Iterações de Ponto Fixo.

anexo 6 – Detalhes sobre a 1ª questão.

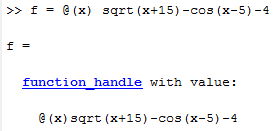


Figura 23: declaração da função conforme enunciado da questão.

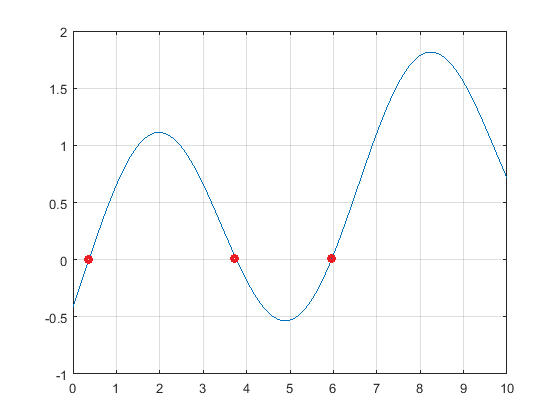


Figura 24: Gráfico da função evidenciando as raízes nos intervalos fornecidos.

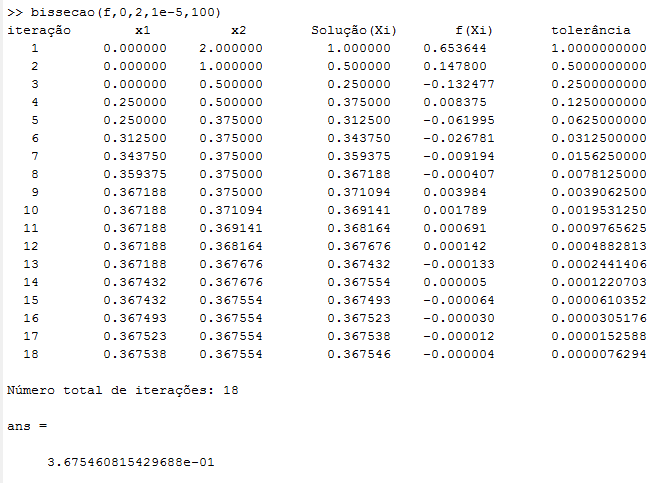


Figura 25: cálculo da primeira raiz utilizando o método da Bisseção.

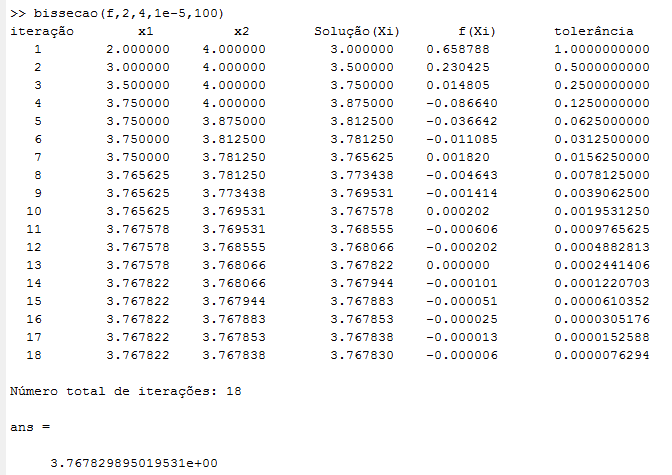


Figura 26: cálculo da segunda raiz utilizando o método da Bisseção.

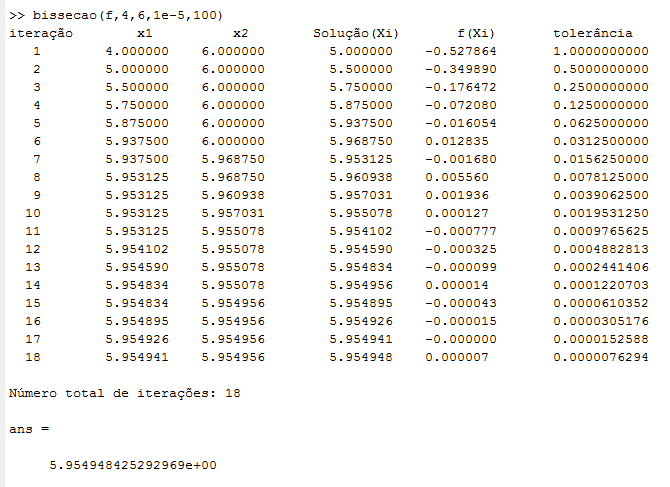


Figura 27: cálculo da terceira raiz utilizando o método da Bisseção.

anexo 7 – Detalhes sobre a 2ª questão.

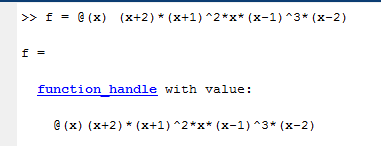


Figura 28

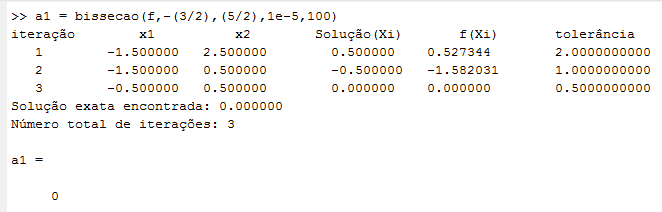


Figura 29: cálculo da primeira raiz utilizando o método da Bisseção.

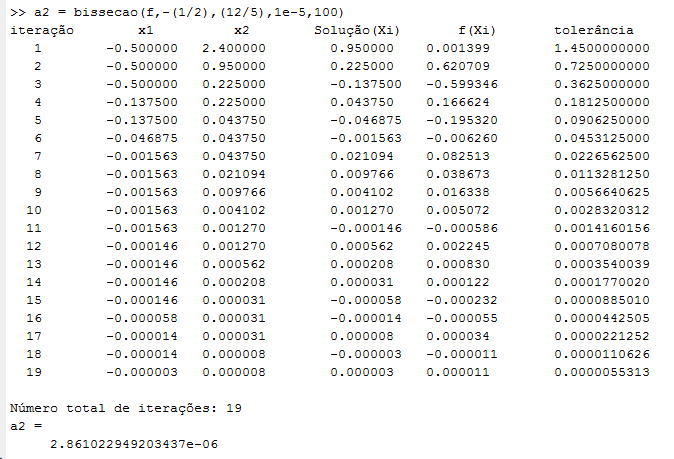


Figura 30: cálculo da segunda raiz utilizando o método da Bisseção.

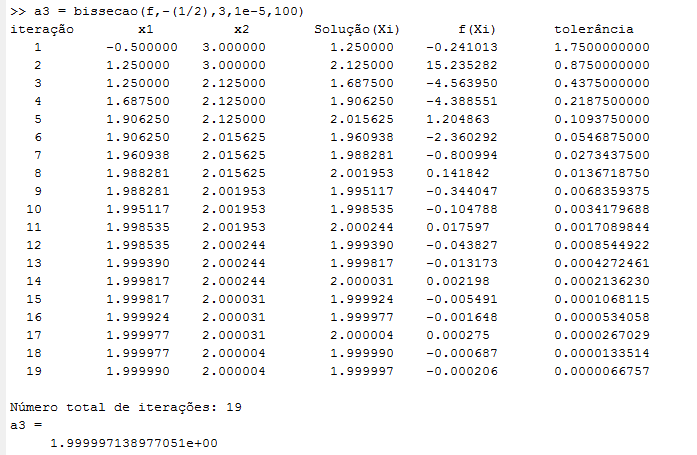


Figura 31: cálculo da terceira raiz utilizando o método da Bisseção.

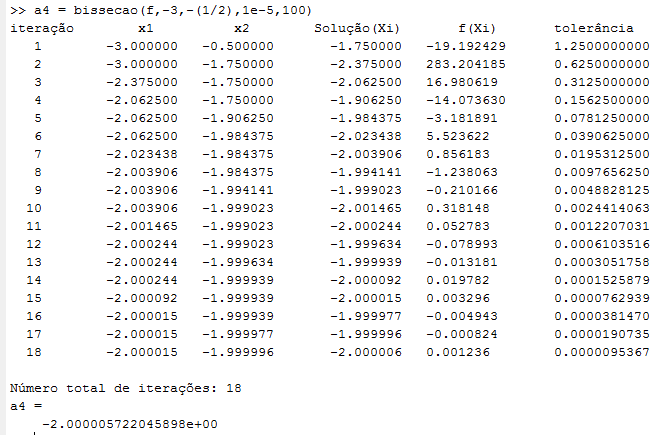


Figura 32: cálculo da quarta raiz utilizando o método da Bisseção.

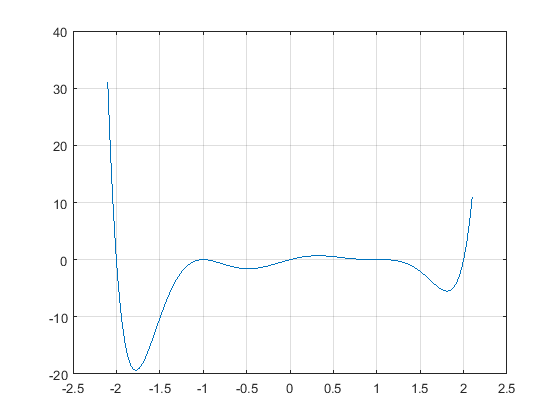


Figura 33: gráfico da função envolvendo todas as raízes apresentadas na questão.

anexo 8 – Detalhes sobre a 4ª questão.

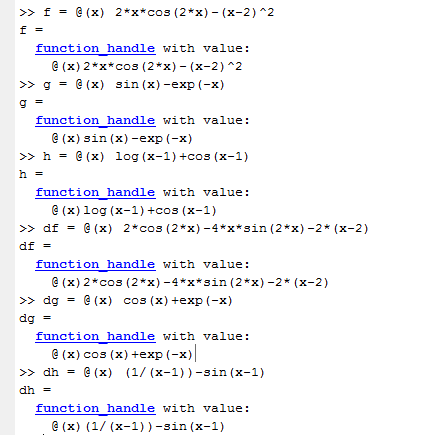


Figura 34: declaração das funções e de suas derivadas, conforme enunciado da questão.

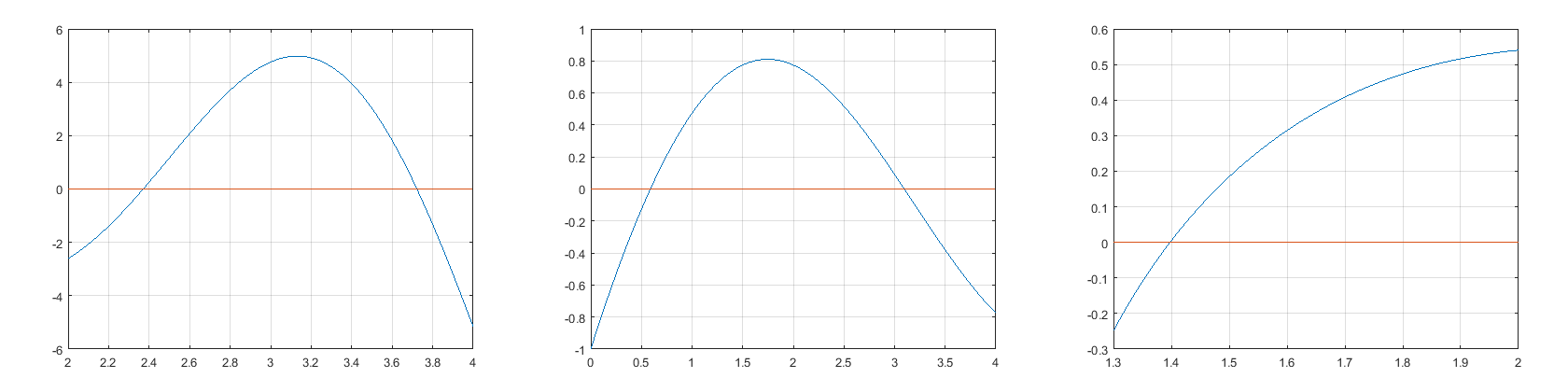


Figura 35: gráficos das funções envolvendo as raízes. À esquerda, f(x); no centro, g(x) e; à direita, h(x).

Aqui serão mostrados os gráficos das ordens de convergência das funções usando o método de Newton-Raphson.

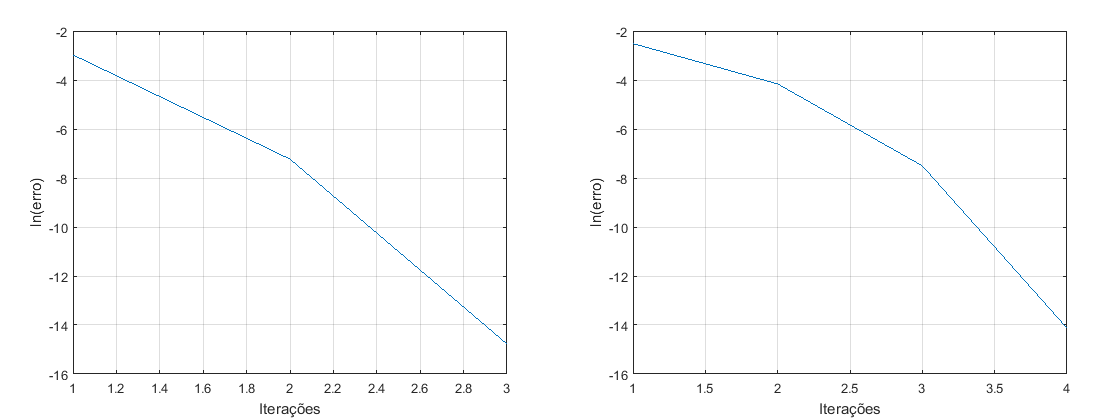


Figura 36: ordem de convergência da função f(x). À esquerda, intervalo 1; à direita, intervalo 2.

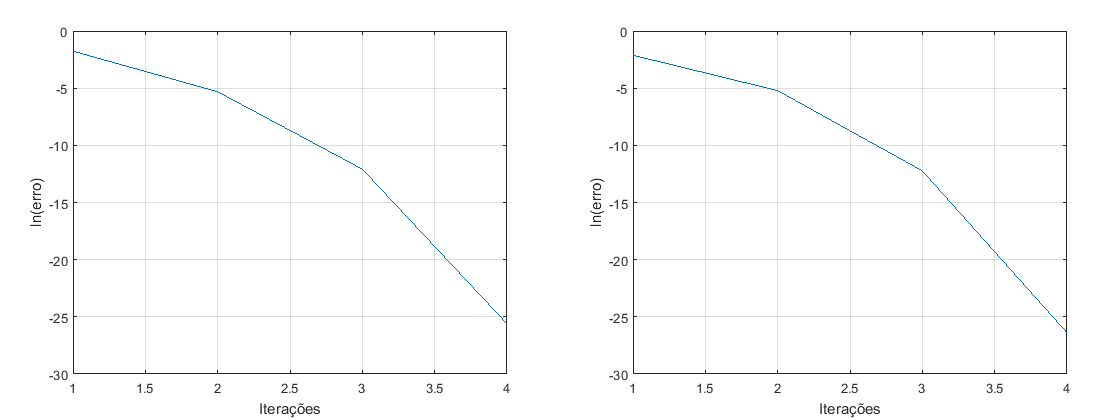


Figura 37: ordem de convergência da função g(x). À esquerda, intervalo 1; à direita, intervalo 2.

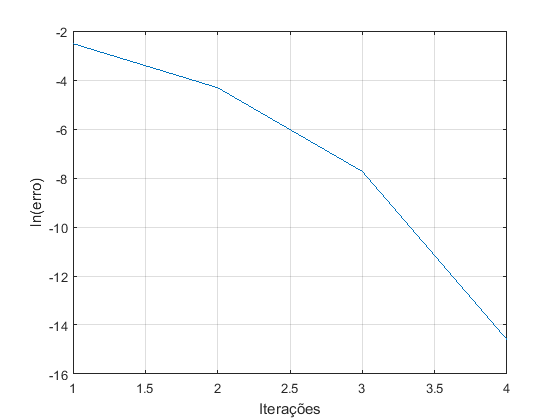


Figura 38: ordem de convergência da função h(x).

ANEXO 9 – DETALHES SOBRE A 5ª QUESTÃO.

Nesta questão, as funções, os gráficos e as declarações das funções no MATLAB são iguais aos da 4ª questão. Portanto, serão mostrados apenas os gráficos de ordem de convergência das funções usando o método da Secante.

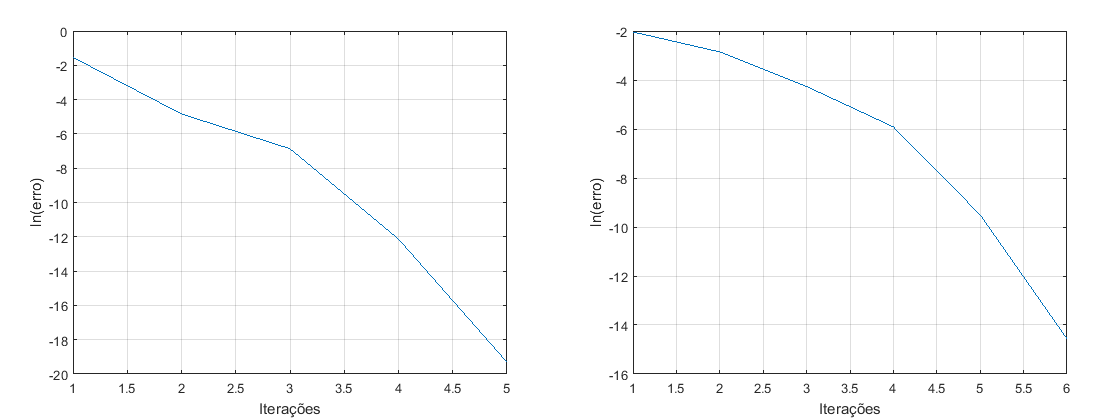


Figura 39: ordem de convergência da função f(x). À esquerda, intervalo 1; à direita, intervalo 2.

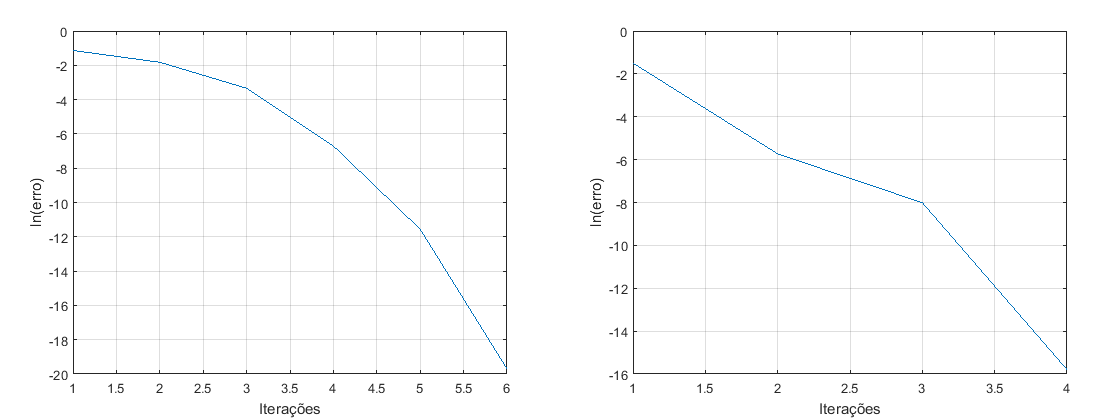


Figura 40: ordem de convergência da função g(x). À esquerda, intervalo 1; à direita, intervalo 2.

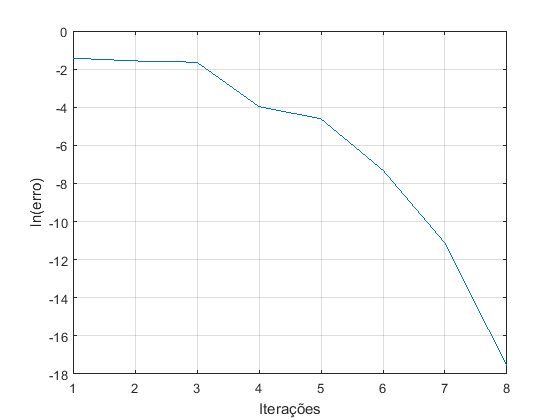


Figura 41: ordem de convergência da função h(x).

anexo 10 – Detalhes sobre a 6ª questão.

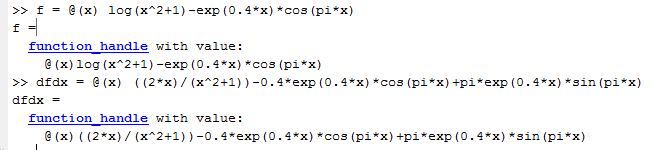


Figura 42: declaração da função e de sua derivada, conforme enunciado da questão.

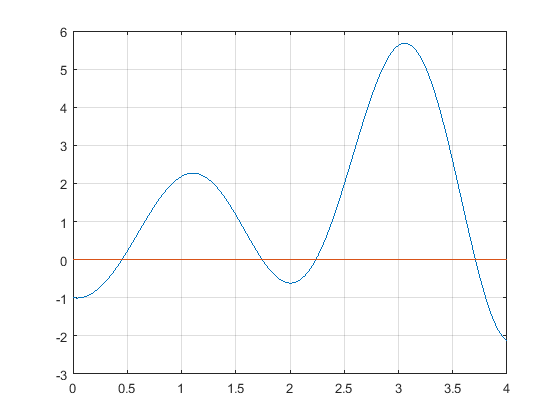


Figura 43: gráfico da função. Vemos que as três primeiras raízes positivas estão nos intervalos [0,1/2], [3/2,2] e [2,5/2]. Portanto, foram usados esses valores de intervalo para o método da Secante e o valor médio do intervalo para a estimativa do método de Newton-Raphson.

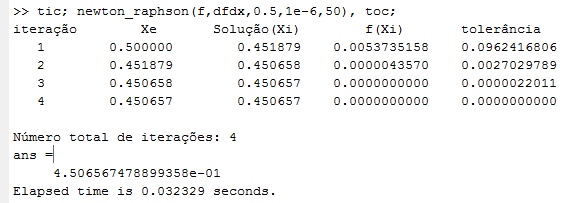


Figura 44: aplicando método NR (Newton-Raphson) para calcular a primeira raiz positiva (antes da alteração).

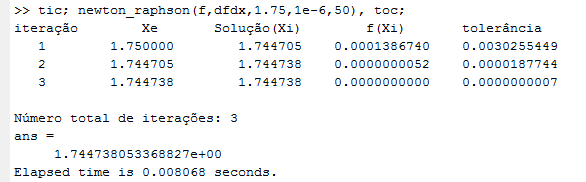


Figura 45: aplicando método NR para calcular a segunda raiz positiva (antes da alteração).

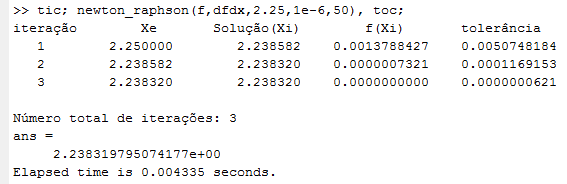


Figura 46: aplicando método NR para calcular a terceira raiz positiva (antes da alteração).

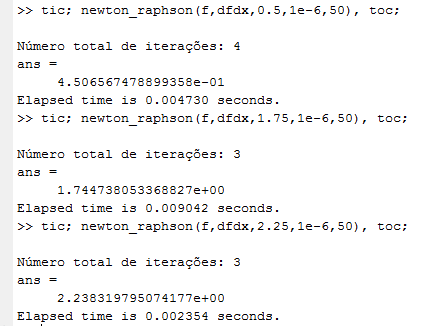


Figura 47: aplicando o método NR para calcular as três primeiras raízes positivas da função. Aqui foi alterada a função newton\_raphson para que não exibisse os dados de cada iteração e, assim, economizasse tempo de processamento.

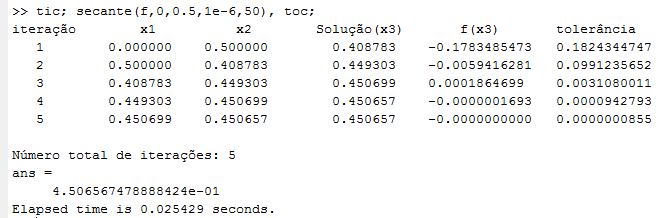


Figura 48: aplicando o método da Secante para calcular a primeira raiz positiva (antes da alteração).

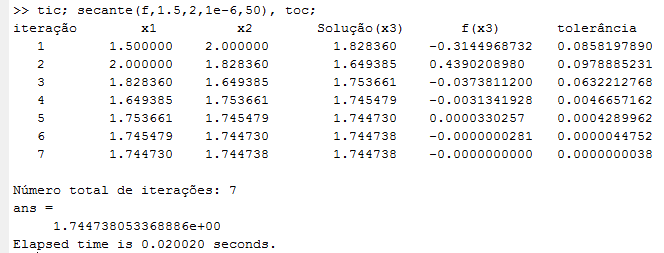


Figura 49: aplicando o método da Secante para calcular a segunda raiz positiva (antes da alteração).

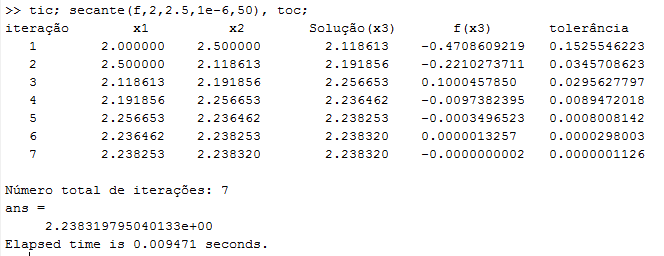


Figura 50: aplicando o método da Secante para calcular a terceira raiz positiva (antes da alteração).

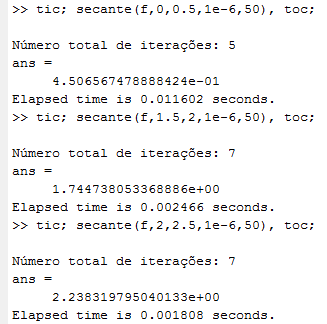


Figura 51: aplicando o método da Secante para calcular as três primeiras raízes positivas da função. Aqui foi alterada a função secante para que não exibisse os dados de cada iteração e, assim, economizasse tempo de processamento.

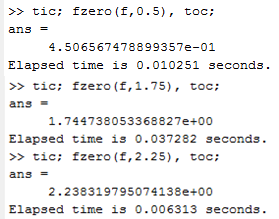


Figura 52: aplicando a função fzero do MATLAB, que utiliza o método da Bisseção, para calcular as três primeiras raízes positivas da função.