Questão 1 (2 pontos): Vejamos os métodos!

```
%Função e intervalo:
f=@(x)x^3-x^2+10*x-5;
a=0;
b=1;

%Função de iteração para Ponto Fixo:
g=@(x)(5-x^3+x^2)/10;
```

	fzero	Bissecção	Posição Falsa	Ponto Fixo	Newton Raphson (proverDerivada=1)	Newton Raphson (proverDerivada=0)	Secante
n	6	16	4	5	2	2	3
r	0.5430	0.5430	0.5430	0.5430	0.5430	0.5430	0.5430
f(r)	-8.8818e-16	-2.0381e-07	1.6960e-05	-1.3029e-05	-3.4487e-09	-4.0417e-08	-2.9298e-07

<u>Gráfico(s)</u>:

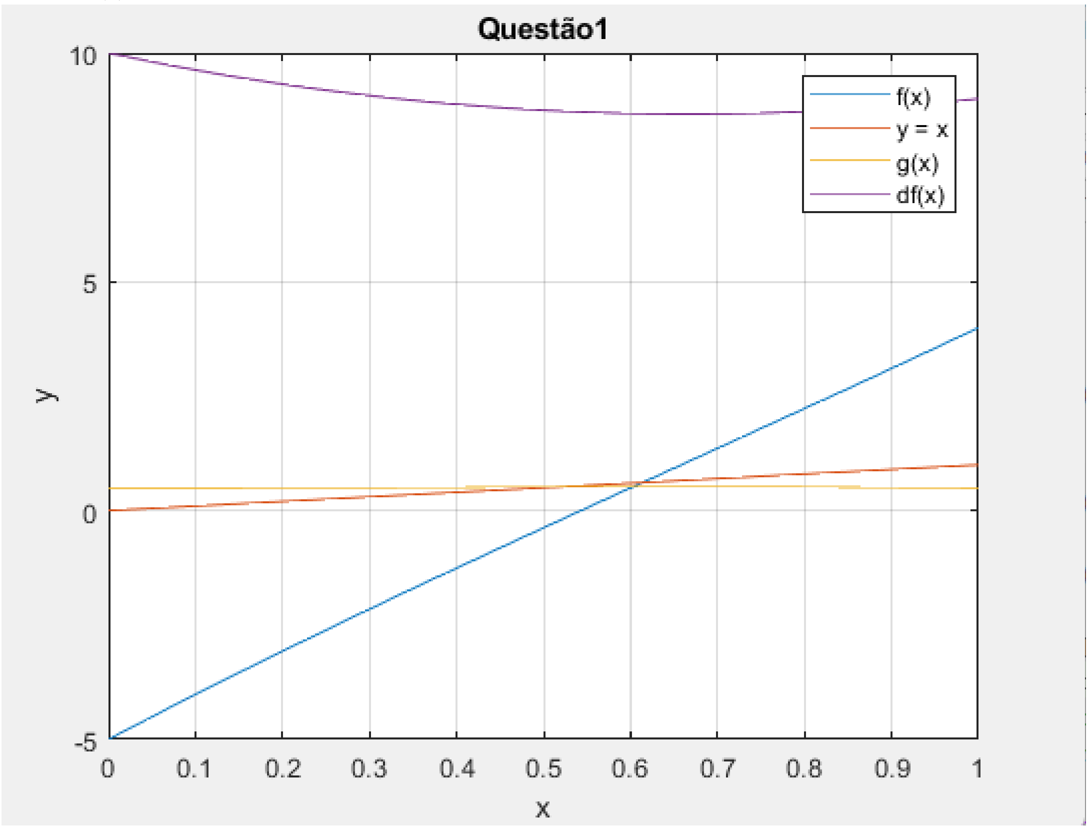


Figura 1.1: Gráficos solicitados

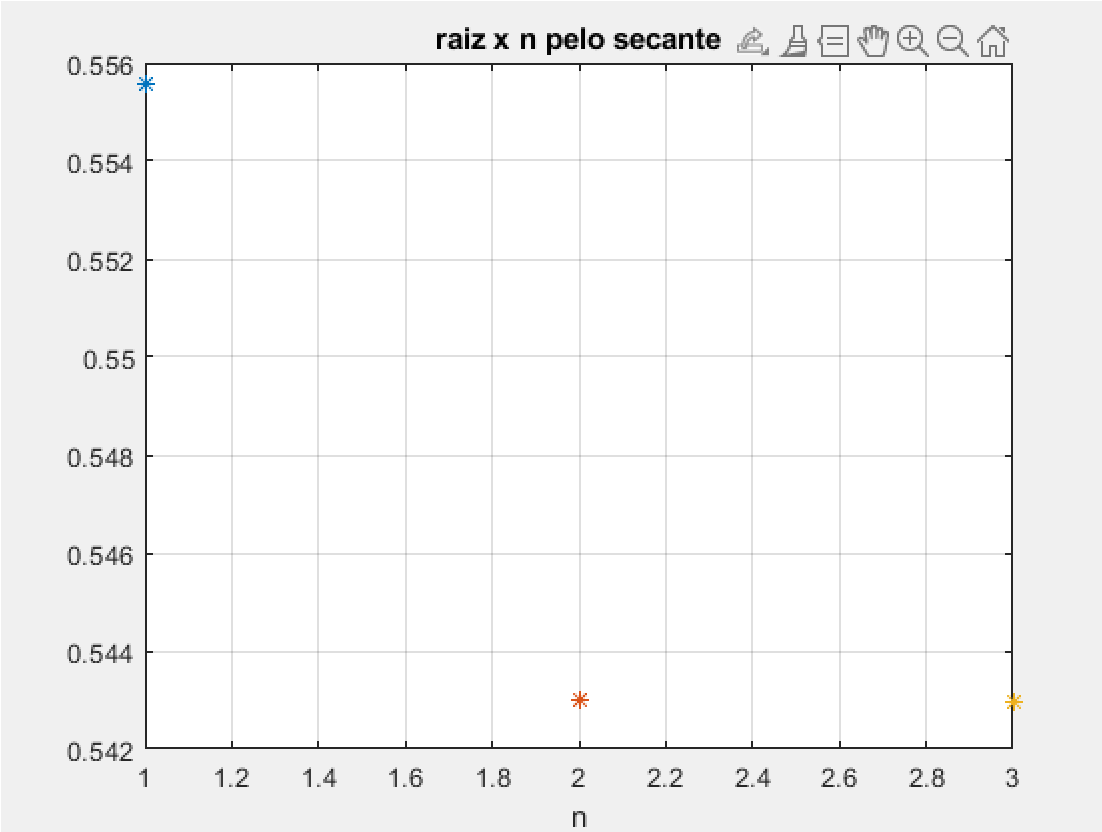


Figura 1.2: gráfico das raízes pelo método secante

<u>Comentários</u>: Como esperado, em termos de velocidade (menor número de iterações), a ordem dos melhores métodos foi Newton-Raphson, Secante, Posição Falsa, Ponto Fixo e Bissecção. O método fzero possui uma posição intermediária quanto à velocidade, porém f(r) deste método é muito mais próximo de zero do que f(r) dos outros métodos, embora os valores de r fornecidos por todos eles sejam os mesmos considerando-se apenas 4 casas decimais. Vale comentar ainda que, como a função g(x) é praticamente constante no intervalo dado, tem-se que o módulo de g'(x) é menor que 1 nesse intervalo e como tanto g(x) e g'(x) são contínuas, o método do ponto fixo converge e se chega rapidamente a uma boa aproximação para a raiz de f(x).

Questão 2 (2 pontos): Tem raiz aí?

```
%Função e intervalo:
f = @(x)(2.718281828^x + (sin(cos(x)))^2);
a=-1;
b=1;
```

	fzero	Bissecção	Newton Raphson (proverDerivada=0)	Secante
n	indefinido	1000	1000	145
r	indefinido	1	-1.5391	-67.5375240

`f(r)`	indefinido	2.9828	0.2155	4.5130e-05

<u>Gráfico(s)</u>:

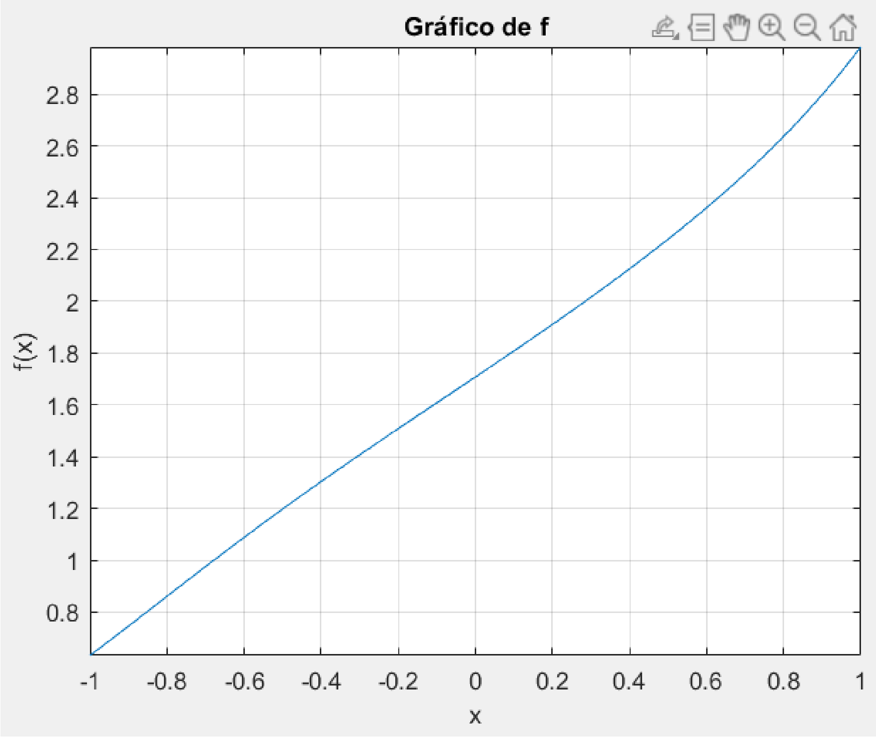


Figura 2.1 - Gráfico de f no intervalo [-1, 1].

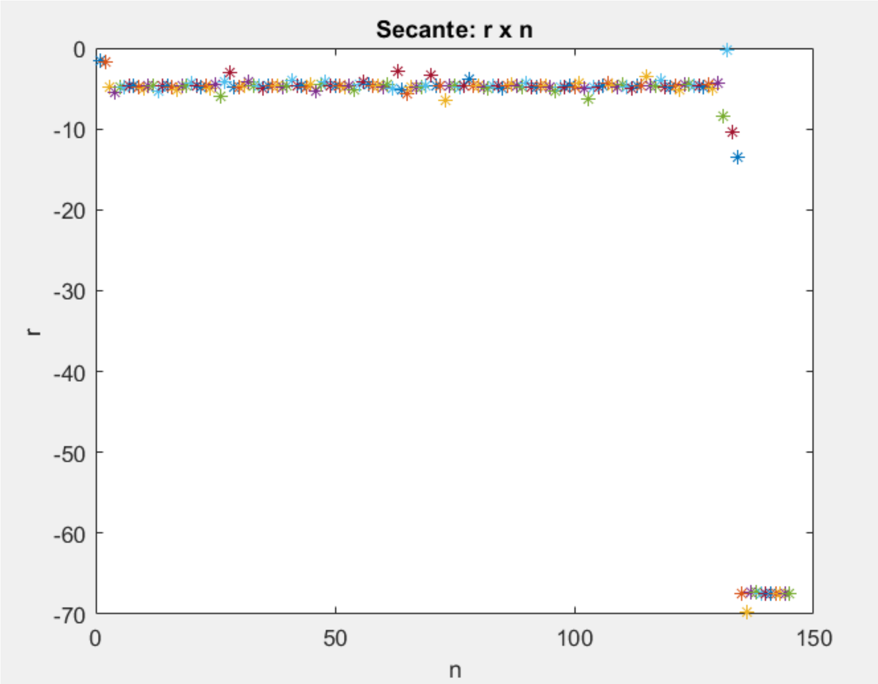


Figura 2.2 - Gráfico mostrando as sucessivas aproximações de r pelo método Secante.

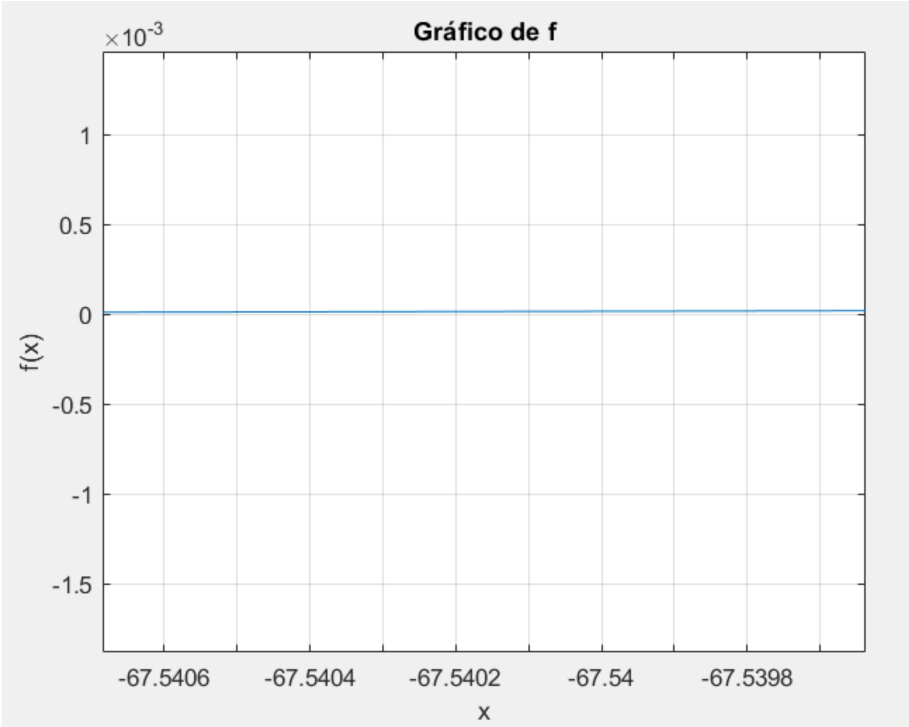


Figura 2.3 - Gráfico de f em intervalo contendo a pseudo-raiz encontrada pelo método Secante.

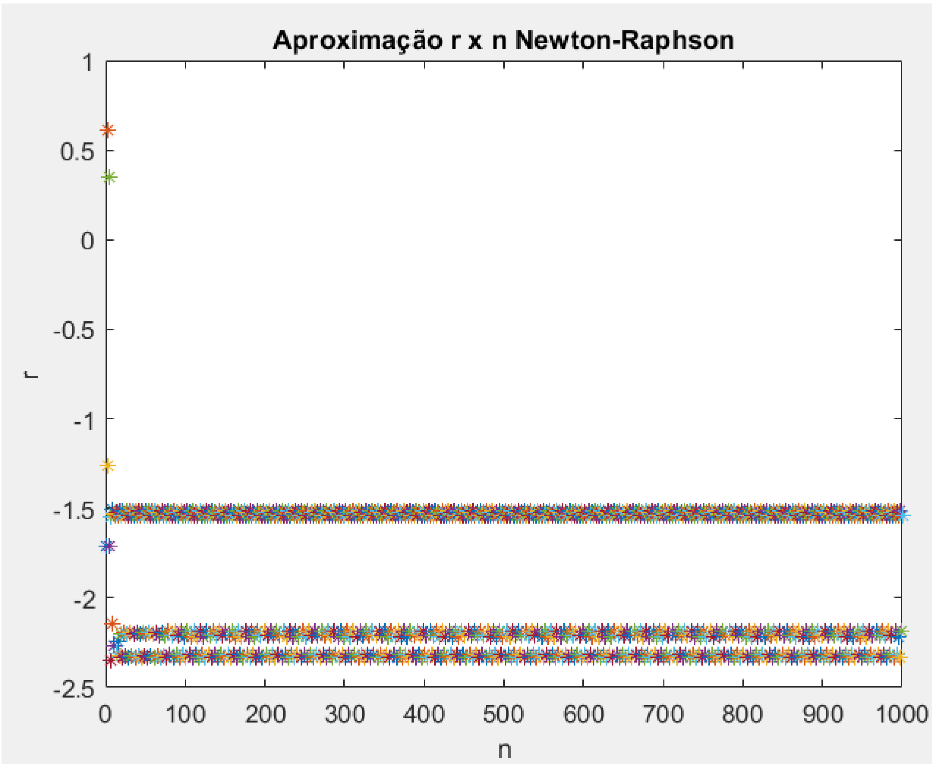


Figura 2.4 - Comportamento do método Newton-Raphson

<u>Comentários</u>:

Como se pode ver na Figura 2.1, não existe raíz para a função no intervalo [-1, 1]. Dessa forma, nem o método da bissecção nem o método Newton-Raphson foram capazes de encontrar uma raíz com a exatidão exigida. O método da bissecção não obteve êxito pois ele só funciona quando se insere um intervalo que possua uma raiz. Já o método Newton-Raphson não funciona pois, como se vê na Figura 2.4, os valores de r encontrados pelo método são periódicos, *i. e.,* ele entra em um loop infinito.

O método fzero retorna erro caso utilizado, visto que é necessário fornecer um intervalo que contenha uma raiz, ou um valor x0 dentro desse intervalo, como parâmetro para a função fzero. Já o método da secante foi capaz de encontrar uma pseudo-raiz já que ele pode acabar procurando fora dos limites estabelecidos pelo intervalo [a, b]. Verifica-se, assim, que quando não se fornece um intervalo com raízes, o único método que pode acabar encontrando uma raiz é o método da secante, mas isso não é garantido. Como se pode ver na Figura 2.3, o gráfico de f não corta o eixo x quando x se aproxima do valor indicado como raiz pelo método da secante. Por esse motivo, ao se aplicar o método fzero ao ponto -67.5 a função retorna erro.

Questão 3 (1.5 ponto): Cada um para um lado!

```
f = @(x)(sin(x));
a=-0.5;
b=7;
```

	fzero	Bissecção	Newton Raphson (proverDerivada=0)	Secante
n	4	13	2	38
r	3.1416	6.2831	3.1416	2.8997e+03
f(r)	1.2246e-16	-4.3217e-05	3.7030e-08	-2.0981e-10

Gráfico(s):

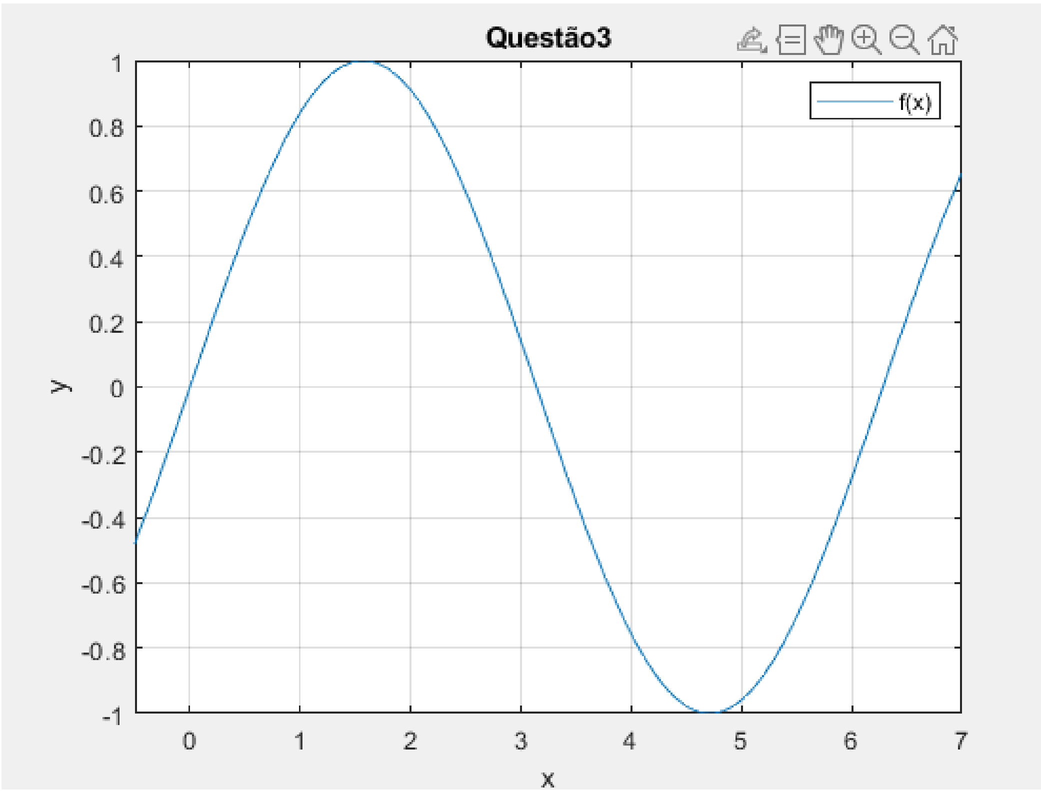


Figura 3.1 - Gráfico de f no intervalo proposto.

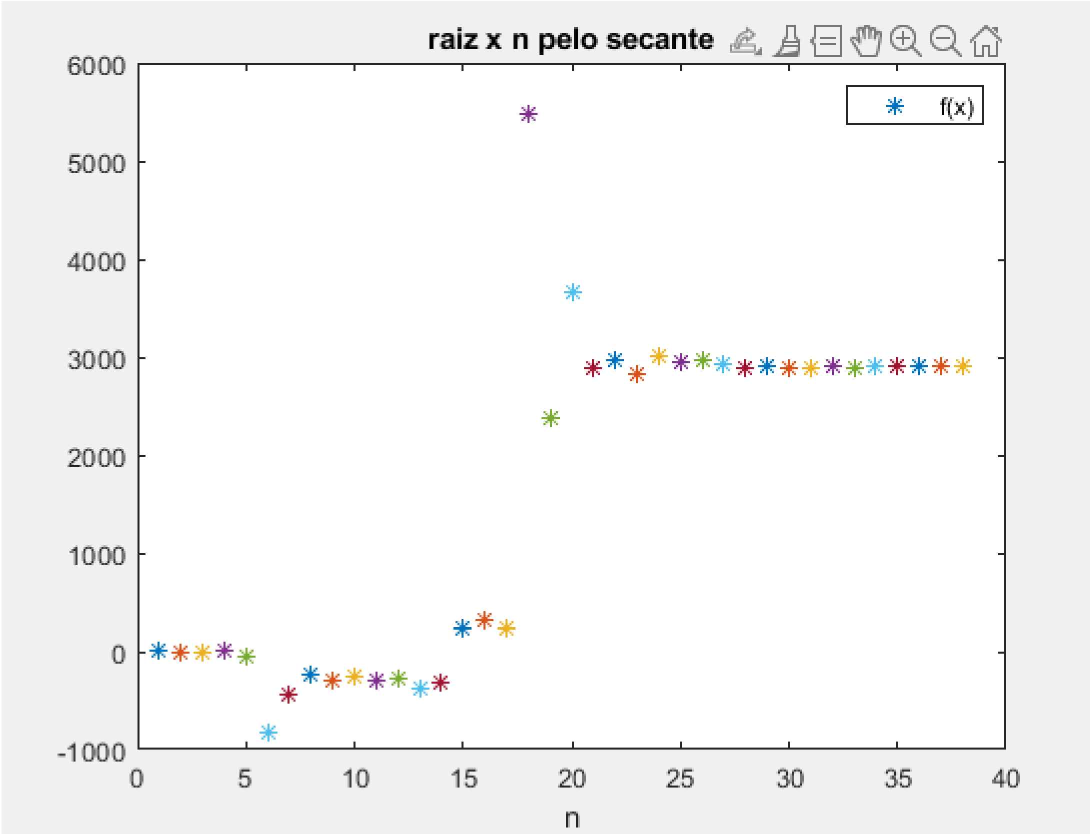


Figura 3.2 - Gráfico da secante

Comentários: Haviam várias respostas possíveis para r nessa questão. Alguns algoritmos tenderam à uma resposta mais à direita e outros a uma mais à esquerda. De modo especial, o método da secante acabou por achar uma raiz fora do domínio proposto [a,b], como era possível a até de se esperar da função, já que a derivada da função seno pode ser alta, fazendo com que haja um desvio do intervalo inicial. Tal comportamento pode ser visto no gráfico da secante, pois aproximadamente depois de n = 20, os valores da raiz passam a ser maiores que 2000 de súbito. Já o método de newton-raphson mostrou-se o mais rápido, pois sua derivada é calculada com precisão, necessitando apenas de 2 iterações. O método fzero foi o mais preciso, achando a raiz mais próxima do zero.

Questão 4 (1.5 ponto): Enganando o critério de parada das funções...

```
f = @(x)(x^2+1e-7);
a = -1;
b = 0;
```

	fzero	**Bissecção**	**Newton Raphson (proverDerivada=0)**	**Secante**
n	indefinido	7	6	1
r	indefinido	-0.0078125	-0.0078081	1.0000e-07
f(r)	indefinido	6.1135e-05	6.1067e-05	1.0000e-07

Gráfico(s):

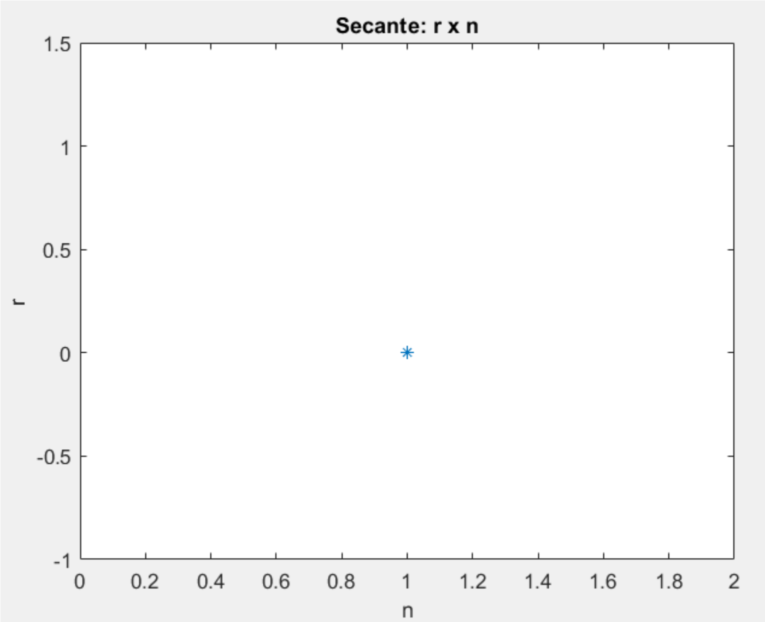


Figura 4.1 - Raiz encontrada pelo método Secante.

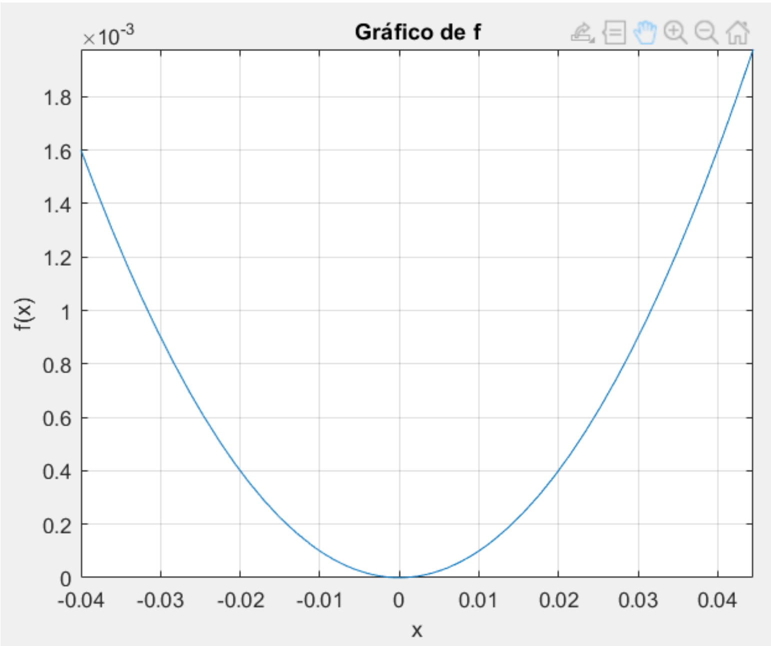


Figura 4.2 - Gráfico de f em escala maior

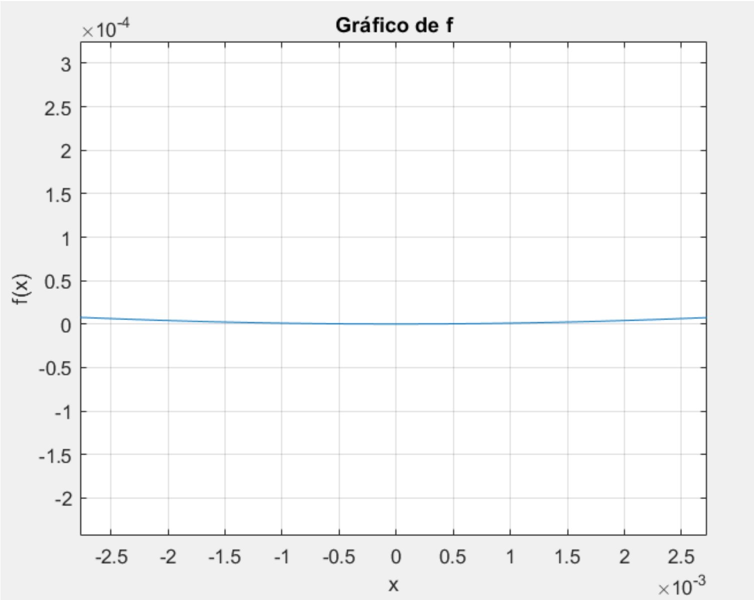


Figura 4.3 - Gráfico de f em escala menor

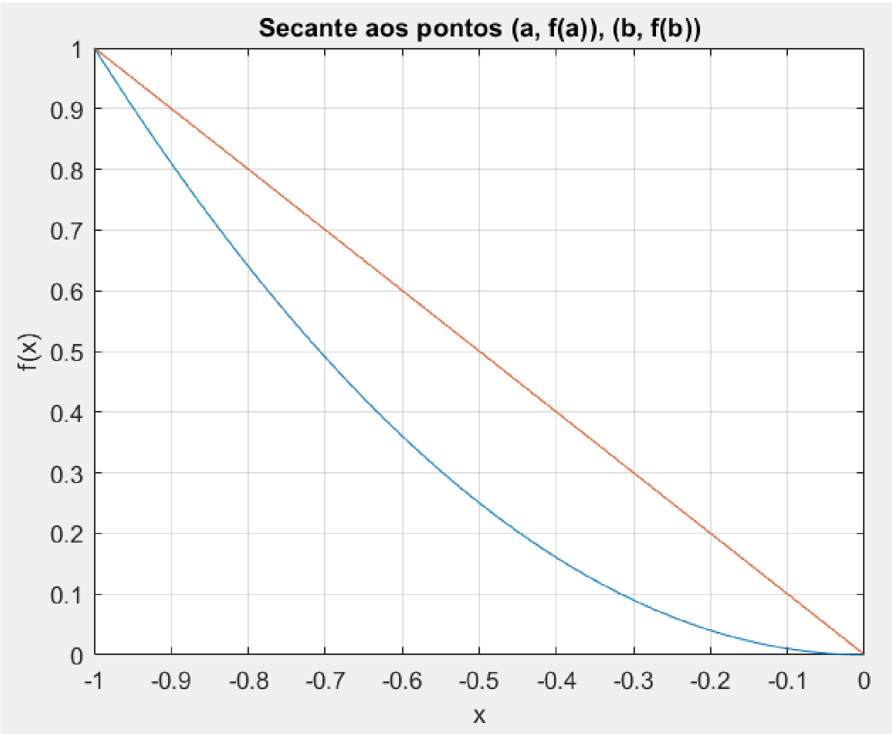


Figura 4.4 - Reta secante utilizada no método Secante

<u>Comentários</u>:

Esta é uma função que na verdade não possui raiz real. É fácil notar que, dado $x^2 >= 0$, então a função é sempre maior que 1.e-7. No entanto, o critério de parada é uma exatidão de 1.e-4 para a raiz, o que faz com que os métodos de Bissecção, Newton-Raphson e Secante parem as iterações de busca de raiz assim que f(r) se torna menor que 1.e-4. Dessa forma, os métodos são "enganados" e encontram uma raiz para uma função que não possui raízes. Para se evitar isso, poder-se-ia alterar o valor de epsilon para 1.e-8, o que faz com que os três métodos cheguem em 1000 iterações sem encontrar uma raiz.

Vale notar que o método da Secante apresenta melhor desempenho pois ao se pegar um valor muito próximo da raiz (nesse caso, raiz falsa), a reta secante toca o eixo x muito próximo do valor da raiz, e assim encontra um valor de r com poucas iterações, como se pode observar na Figura 4.4.

Novamente, o método fzero retorna um erro se for usado, já que ele exige a inserção de um intervalo [a, b] ou de um valor x0 nesse intervalo como parâmetro, respeitando-se a condição f(a).f(b) < 0;

Questão 5 (1.5 ponto): Secante é sempre rápido?

```
f = @(x)(x^7 + atan(90*x)/10);
a = -1;
b = 1.01;
```

	fzero	**Bissecção**	**Newton Raphson (proverDerivada=0)**	**Secante**
n	5	16	2	610
r	-4.9594e-19	7.6294e-07	1.4805e-06	1.2623e-06
f(r)	-4.4634e-18	6.8665e-06	1.3325e-05	1.1361e-05

Gráfico(s):

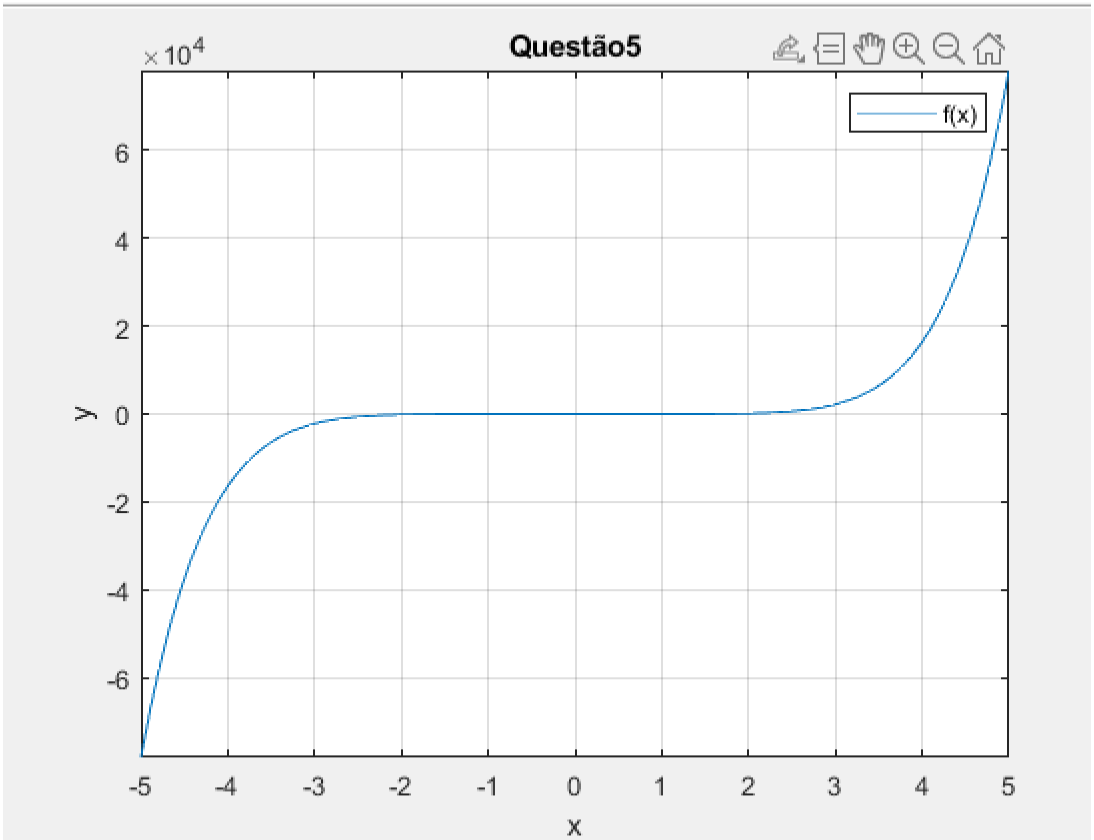


Figura5.1: gráfico da função f(x)

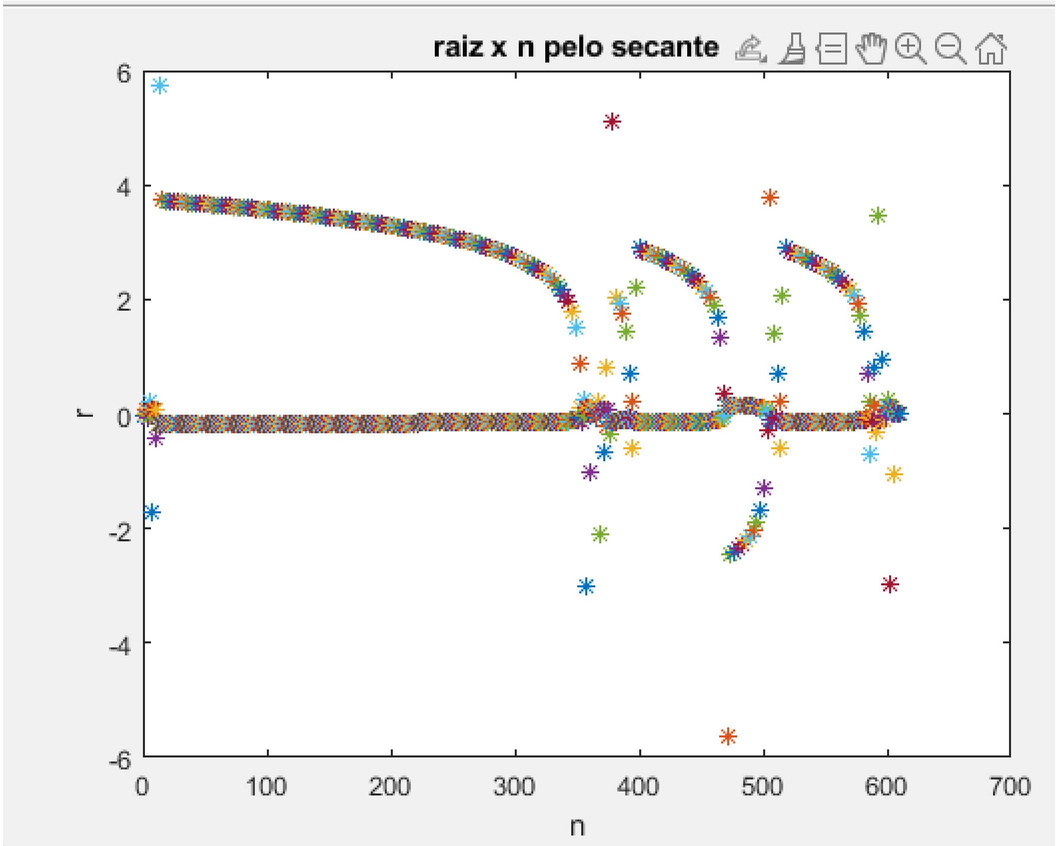


Figura 5.2: Raízes do método secante em função do número de iterações

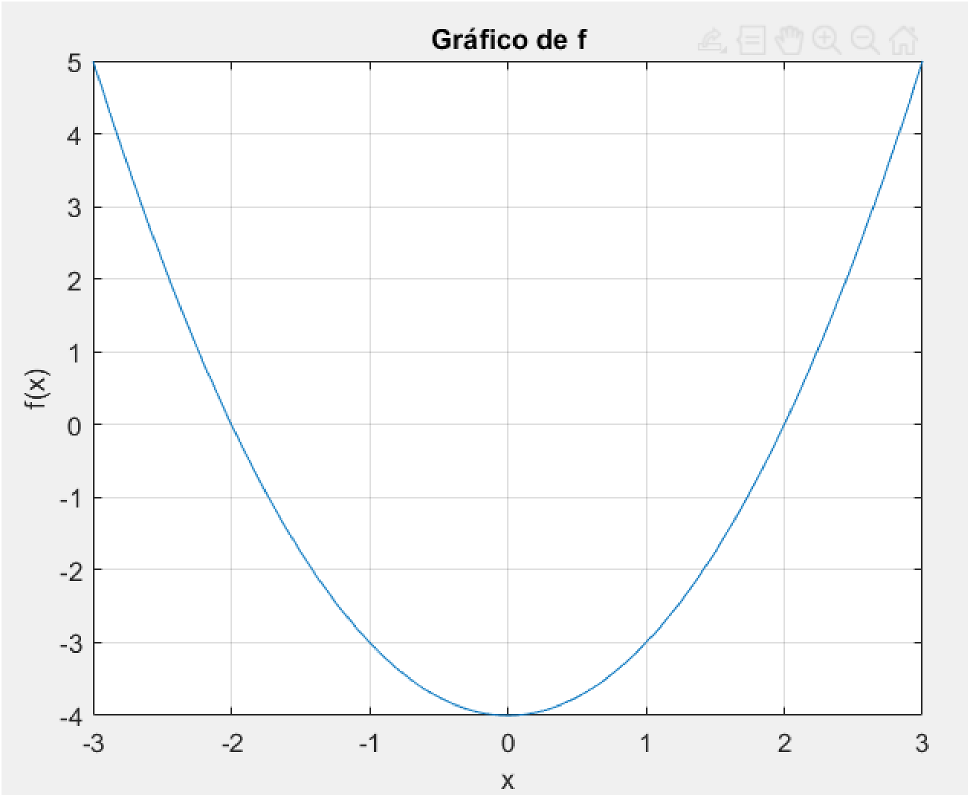
Comentários: O método secante nem sempre é muito rápido, pois, como vemos nesse caso, quando no intervalo requerido a derivada da função se aproxima muito de 0, há uma grande mudança no r a cada iteração, fazendo com que sejam necessárias muitas iterações para encontrar o r. Vê-se que as raízes entram num certo loop amortecido, fazendo com que elas oscilem entre zero e um valor próximo de 4 que vai decaindo. Isso em certa medida é uma limitação do método, que pode facilmente entrar em loop ou loop amortecido de grande amplitude quando o valor da derivada no intervalo proposto é muito pequeno, fazendo-o avançar muito a coordenada x. Vemos também que que a secante depende muito dos parâmetros a e b, pois se variarmos o b para 1 o processo resume-se a uma única iteração. Os outros métodos também resumiram-se a uma iteração quando b =1.

Questão 6 (1.5 ponto): Os melhores métodos podem ter problemas...

```
f = @(x)(x^2-4);
a = -3;
b = 3;
```

	fzero	Bissecção	Newton Raphson (proverDerivada=0)	Secante
n	6	17	1000	1000
r	2	-2.000015	indefinido	indefinido
f(r)	0	6.1035	indefinido	indefinido

Gráfico(s):



Comentários:

Tanto o método Newton-Raphson quanto o método Secante falham em encontrar a raiz para essa função com os parâmetros a e b dados. Isso ocorre porque, para ambos os métodos, ocorre uma divisão por zero na primeira iteração e o valor de r assume +inf ou -inf; No método Newton-Raphson em específico, a derivada é zero no ponto 0, e portanto, essa reta só toca o eixo x em +inf ou -inf.; já no método da secante, a secante na primeira iteração é horizontal e, logo, também só toca o eixo x em +inf ou -inf. Dessa forma, com os valores de a e b utilizados, nenhum destes métodos funciona. Trocando ligeiramente o valor de a ou o de b, os métodos já passam a funcionar.
Nota-se que o método da bissecção e a função fzero do MatLab não tiveram problemas para funcionar. É claro que a função fzero apresentou melhor desempenho, por encontrar o valor exato da raiz e com menos iterações.

Portanto, para se encontrar uma raiz nessa função, o mais adequado é utilizar o método da Bissecção ou fzero, com o intervalo [a, b] dado, ou então alterar um dos limites do intervalo para conseguir utilizar os métodos Newton-Raphson ou Secante.

Avaliação da dupla

Aluno(a)	Atividade realizadas	Percentual merecido da nota
Andrei Albani	*Fez questões pares*	*100%*
Vinicius Pereira	*Fez questões ímpares*	*100%*