UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULADE DE ENGENHARIA MECÂNICA

RELATÓRIO DE ALGORITMO IMPLEMENTADO

CÁLCULO DE SOLUÇÕES DE EQUAÇÕES DE UMA VARIÁVEL

MÉTODO DA SECANTE

Aluno: Diego Fernando Luque Martin

Matrícula: 191606

Campinas

2017

SUMÁRIO

1. INTRODUÇÃO

Este relatório tem por objetivo demonstrar por meio de um algoritmo implementado em linguagem de programação C, o cálculo de soluções de equações com uma variável utilizando o método da Secante com a apresentação numérica dos resultados de raízes de uma função.

2. MÉTODO

O método de Newton estudado anteriormente é uma técnica altamente eficiente, mas tem uma desvantagem importante: a necessidade de conhecer o valor da derivada de f(x) em cada aproximação. Com frequência, o cálculo de f'(x) é muito mais difícil e requer mais operações aritméticas que o f(x).

$$f'(p_{n-1}) = \lim_{x \to p_{n-1}} \left(\frac{f(x) - f(p_{n-1})}{x - p_{n-1}} \right)$$

se p_{n-2} estiver próximo de p_{n-1}, então:

$$f'(p_{n-1}) \approx \frac{f(p_{n-2}) - f(p_{n-1})}{p_{n-2} - p_{n-1}} = \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}$$

Utilizar essa aproximação para $f'(p_{n-1})$ na formula de Newton fornece:

$$p_n = p_{n-1} - \frac{f(p_{n-1}) \cdot (p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$$

2.1. Algoritmo

O livro Analise Numérica (Burden, 2015), fornece o seguinte algoritmo:

ENTRADA: aproximação p_0 e p_1; tolerância TOL; número máximo de iterações n_0.

SAÍDA: solução aproximada p ou mensagem de erro

Passo 1: Faça i = 2;
$q_0 = f(p_0)$;
$q_1 = f(p_1)$;

Passo 2: Enquanto $i \leq n_0$, execute os Passos 3 a 6.

Passo 3: Faça $p = p_1 - q_1 (p_1 - p_0) / (q_1 - q_0)$ (Calcule p_i.)

Passo 4: Se $| p - p_1 | < TOL$, então
SAÍDA (p); (Procedimento concluído com sucesso.)
PARE.

Passo 5: Faça i = i + 1.

Passo 6: Faça $p_0 = p_1$. (Atualize p_0, p_1, q_0, q_1.)

$q_0 = q_1$;

$p_1 = p$;

$q_1 = f(p)$;

Passo 7: SAÍDA (‘O método falhou após n_0 iterações, n_0= ‘, n_0);

(O procedimento não foi bem-sucedido.)

PARE.

Com base no exemplo dado acima, o algoritmo deverá exigir informações do usuário:

- Estimativa Inicial p_0;
- Estimativa Inicial p_1;
- Tolerância ou Erro desejado;
- Número máximo de iterações.

Para fins de facilitação para o usuário, decidiu-se que o valor de tolerância e o número máximo de iterações serão parâmetros internos do algoritmo cabendo ao usuário apenas informar e estimativa dos valores p_0 e p_1.

Após realizadas as iterações de cálculo, o algoritmo deverá apresentar ao final de sua execução uma das seguintes alternativas:

- Resultado da raiz com base nas estimativas iniciais;
- Erro devido a inexistência de raízes ou o número máximo de iterações atingido.

Para execução do programa em linguagem de programação, os seguintes parâmetros foram adotados:

- Linguagem de Programação: C;
- Compilador: Code::Blocks 16.01

2.2. Teste

Para teste do código fonte escrito em linguagem de programação C, foi usado exemplo de cálculo

$$f(x) = x^3 + 4x^2 - 10; \quad p_0 = 1, p_1 = 2; \quad tol. = 10^{-16}$$

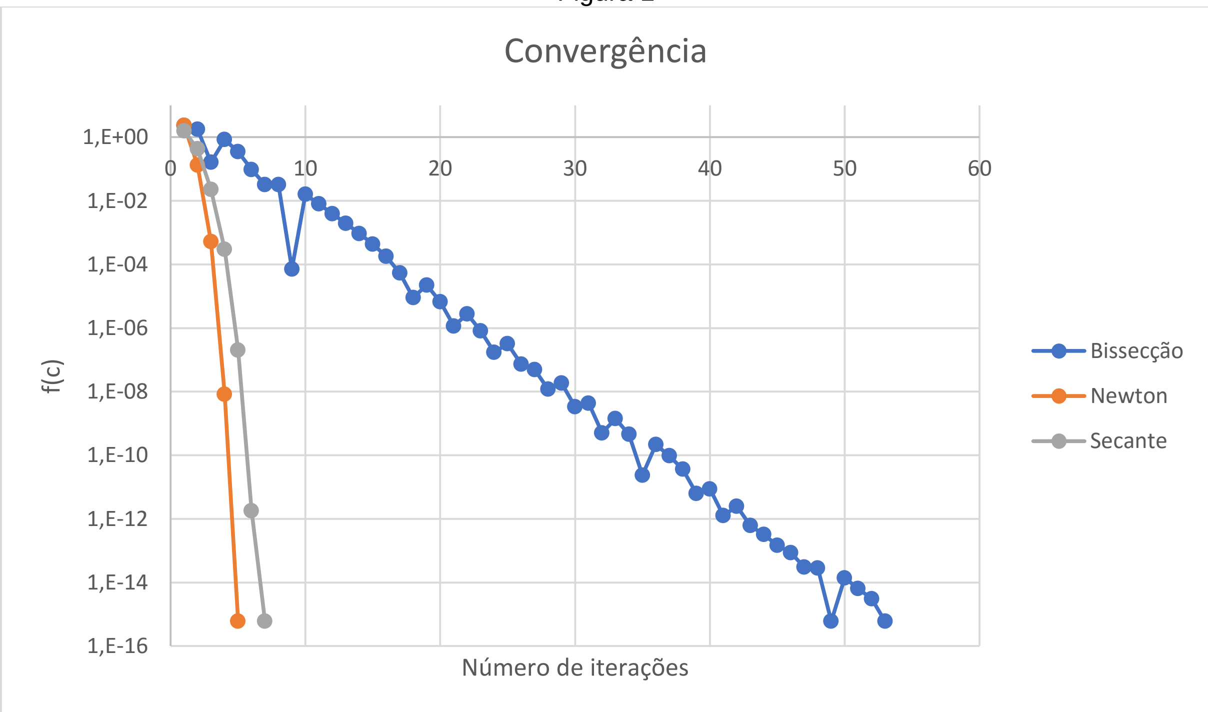
Resultados:

```
Digite um número para a Aproximação P0: 1
Digite um número para a Aproximação P1: 2
i: 1 p0: 1,0000000000000000 p1: 2,0000000000000000 p2: 1,2631578947368420 f(p2): -1,6022743840209952
i: 2 p0: 2,0000000000000000 p1: 1,2631578947368420 p2: 1,3388278388278387 f(p2): -0,4303647480045272
i: 3 p0: 1,2631578947368420 p1: 1,3388278388278387 p2: 1,3666163947193450 f(p2): 0,0229094307759479
i: 4 p0: 1,3388278388278387 p1: 1,3666163947193450 p2: 1,3652119026318565 f(p2): -0,0002990679193269
i: 5 p0: 1,3666163947193450 p1: 1,3652119026318565 p2: 1,3652300011108591 f(p2): -0,0000002031682731
i: 6 p0: 1,3652119026318565 p1: 1,3652300011108591 p2: 1,3652300134142061 f(p2): 0,0000000000018049
i: 7 p0: 1,3652300011108591 p1: 1,3652300134142061 p2: 1,3652300134140969 f(p2): 0,0000000000000006

Raiz encontrada para P0 = 1,0000000000000000 e P1 = 2,0000000000000000: 1,3652300134140969
```

A título de comparação com outros métodos para cálculo de soluções de equações de uma variável, foi possível traçar um gráfico tendo o número de iterações no eixo horizontal e o valor de f(p) , no eixo vertical em escala logarítmica de base 10. Desta maneira pode se observar a característica de Convergência do resultado do método de cálculo como demostrado na figura 1:

Figura 1



Considerações sobre o valor de tolerância:

Após alguns testes realizados, convencionou-se que o valor de tolerância definido como um parâmetro interno do algoritmo não deve ultrapassar 10^{-16}, pois quando este valor é ultrapassado para baixo, os cálculos exigidos da máquina ou computador apresentam erro e valores não representativos.

Considerações sobre o valor do número máximo de iterações:

Devido ao valor mínimo de tolerância definido acima, nos exemplos estudados ente relatório, o número máximo de iterações não ultrapassou $\mathrm{i} = 12$. Por termos um estudo com poucos exemplos devemos definir também uma margem de segurança para que este valor de iterações não seja impeditivo para outros casos. Com base nessas informações convencionou-se o número máximo de iterações em 100.

2.3. Validação

Para validação do algoritmo escrito foi utilizado o exemplo de cálculo demonstrado no livro Análise Numérica, (Burden, 2015), página 78, onde o autor demostra a resolução de uma função (raiz) para valor inicial e tolerância como dado abaixo:

$$\mathrm{f(x)} = \cos(\mathrm{x}) - \mathrm{x}\,; \quad \mathrm{p}_0 = 0{,}5, \mathrm{p}_1 = \frac{\pi}{4}; \quad \mathrm{tol.} = 10^{-16}$$

Os resultados obtidos pelo autor estão na tabela 1 abaixo:

Tabela 1 (Burden, 2011)

	Secant
n	p_n
0	0.5
1	0.7853981635
2	0.7363841388
3	0.7390581392
4	0.7390851493
5	0.7390851332

Utilizando os mesmos critérios de aproximação e tolerância do exemplo acima e aplicando ao algoritmo, obteve se os seguintes resultados:

```
Digite um número para a Aproximação P0: 0,5
Digite um número para a Aproximação P1: 0,785398163397448309615660845819881
i: 1 p0: 0,5000000000000000 p1: 0,7853981633974483 p2: 0,7363841388365822 f(p2): 0,0045177185221702
i: 2 p0: 0,7853981633974483 p1: 0,7363841388365822 p2: 0,7390581392138897 f(p2): 0,0000451772159638
i: 3 p0: 0,7363841388365822 p1: 0,7390581392138897 p2: 0,7390851493372764 f(p2): -0,0000000269821671
i: 4 p0: 0,7390581392138897 p1: 0,7390851493372764 p2: 0,7390851332150645 f(p2): 0,0000000000001609
i: 5 p0: 0,7390851493372764 p1: 0,7390851332150645 p2: 0,7390851332151607 f(p2): 0,0000000000000000

Raiz encontrada para P0 = 0,5000000000000000 e P1 = 0,7853981633974483: 0,7390851332151607
```

Com base nos resultados da iteração 5, o algoritmo foi considerado válido pois o valor de p_n, correspondentes ao valor do resultado do algoritmo p_2 tem igualdade em todas as casas decimais apresentadas pelo autor salvo arredondamentos.

3. EXERCÍCIOS RESOLVIDOS

Exercício 2.3 – 3a (Burden, 2015)

$$f(x) = x^2 - 6; \quad p_0 = 3, p_1 = 2 \quad ; \quad p_n = 3$$

```
Digite um número para a Aproximação P0: 3
Digite um número para a Aproximação P1: 2
i: 2 p0: 3,0000000000000000 p1: 2,0000000000000000 p2: 2,3999999999999999 f(p2): -0,2400000000000004
i: 3 p0: 2,0000000000000000 p1: 2,3999999999999999 p2: 2,4545454545454546 f(p2): 0,0247933884297523

Erro! O número máximo de iterações foi atingido.
Erro! Não encontrada raiz da função!
```

Exercício 2.3 – 8a (Burden, 2015)

$$f(x) = e^x - 2^{-x} + 2\cos x - 6; \quad p_0 = 1, p_1 = 2; \quad tol. = 10^{-5}$$

```
Digite um número para a Aproximação P0: 1
Digite um número para a Aproximação P1: 2
i: 1 p0: 1,0000000000000000 p1: 2,0000000000000000 p2: 1,6783084847673806 f(p2): -0,5456738336051021
i: 2 p0: 2,0000000000000000 p1: 1,6783084847673806 p2: 1,8081028770204484 f(p2): -0,0857386860365993
i: 3 p0: 1,6783084847673806 p1: 1,8081028770204484 p2: 1,8322984635189548 f(p2): 0,0119845510501582
i: 4 p0: 1,8081028770204484 p1: 1,8322984635189548 p2: 1,8293311729315336 f(p2): -0,0002150280912968
i: 5 p0: 1,8322984635189548 p1: 1,8293311729315336 p2: 1,8293834739842123 f(p2): -0,0000005247853521

Raiz encontrada para P0 = 1,0000000000000000 e P1 = 2,0000000000000000: 1,8293834739842123
```

Exercício 2.3 – 8b (Burden, 2015)

$$f(x) = \ln(x-1) + \cos(x-1); \quad p_0 = 1{,}3, p_1 = 2; \quad tol. = 10^{-5}$$

```
Digite um número para a Aproximação P0: 1,3
Digite um número para a Aproximação P1: 2
i: 1 p0: 1,3000000000000000 p1: 2,0000000000000000 p2: 1,5206070484982963 f(p2): 0,2147576421245230
i: 2 p0: 2,0000000000000000 p1: 1,5206070484982963 p2: 1,2043576628327315 f(p2): -0,6086920298206076
i: 3 p0: 1,5206070484982963 p1: 1,2043576628327315 p2: 1,4381284530202825 f(p2): 0,0803041040773659
i: 4 p0: 1,2043576628327315 p1: 1,4381284530202825 p2: 1,4108819229719018 f(p2): 0,0273195256748568
i: 5 p0: 1,4381284530202825 p1: 1,4108819229719018 p2: 1,3968332639158123 f(p2): -0,0019495178898838
i: 6 p0: 1,4108819229719018 p1: 1,3968332639158123 p2: 1,3977690004081287 f(p2): 0,0000436500400295
i: 7 p0: 1,3968332639158123 p1: 1,3977690004081287 p2: 1,3977485079374574 f(p2): 0,0000000680125771

Raiz encontrada para P0 = 1,3000000000000000 e P1 = 2,0000000000000000: 1,3977485079374574
```

Exercício 2.3 – 8f (Burden, 2015)

$$f(x) = sen(x) - e^{-x}; \quad [p_0, p_1] = [0{,}1], [3{,}4], [6{,}7]; \quad tol. = 10^{-5}$$

```
Digite um número para a Aproximação P0: 0
Digite um número para a Aproximação P1: 1
i: 1 p0: 0,0000000000000000 p1: 1,0000000000000000 p2: 0,6786141005751505 f(p2): 0,1203951835902913
i: 2 p0: 1,0000000000000000 p1: 0,6786141005751505 p2: 0,5690622514011431 f(p2): -0,0272136879976925
i: 3 p0: 0,6786141005751505 p1: 0,5690622514011431 p2: 0,5892596135984055 f(p2): 0,0010078002098714
i: 4 p0: 0,5690622514011431 p1: 0,5892596135984055 p2: 0,5885383580178513 f(p2): 0,0000077860740297

Raiz encontrada para P0 = 0,0000000000000000 e P1 = 1,0000000000000000: 0,5885383580178513

Digite um número para a Aproximação P0: 3
Digite um número para a Aproximação P1: 4
i: 1 p0: 3,0000000000000000 p1: 4,0000000000000000 p2: 3,1054103831646231 f(p2): -0,0086317508738427
i: 2 p0: 4,0000000000000000 p1: 3,1054103831646231 p2: 3,0953360036650501 f(p2): 0,0009803535237471
i: 3 p0: 3,1054103831646231 p1: 3,0953360036650501 p2: 3,0963635053476626 f(p2): 0,0000004073173139

Raiz encontrada para P0 = 3,0000000000000000 e P1 = 4,0000000000000000: 3,0963635053476626
```

```
Digite um número para a Aproximação P0: 6
Digite um número para a Aproximação P1: 7
i: 1 p0: 6,0000000000000000 p1: 7,0000000000000000 p2: 6,3005368623638853 f(p2): 0,0155153653078940
i: 2 p0: 7,0000000000000000 p1: 6,3005368623638853 p2: 6,2835947537593615 f(p2): -0,0014572317018477
i: 3 p0: 6,3005368623638853 p1: 6,2835947537593615 p2: 6,2850493677904664 f(p2): 0,0000000945836889

Raiz encontrada para P0 = 6,0000000000000000 e P1 = 7,0000000000000000: 6,2850493677904664
```

Exercício 2.3 – 13b (Burden, 2015)

$$f(x) = 230x^4 + 18x^3 + 9x^2 - 221x - 9; \quad p_0 = -1, p_0 = 1; \quad tol. = 10^{-6}$$

```
Digite um número para a Aproximação P0: -1
Digite um número para a Aproximação P1: 0
i: 1 p0: -1,0000000000000000 p1: 0,0000000000000000 p2: -0,0203619909502262 f(p2): -4,4963809278368672
i: 2 p0: 0,0000000000000000 p1: -0,0203619909502262 p2: -0,0406912564352419 f(p2): 0,0070874831622538
i: 3 p0: -0,0203619909502262 p1: -0,0406912564352419 p2: -0,0406592625776911 f(p2): -0,0000057062437739
i: 4 p0: -0,0406912564352419 p1: -0,0406592625776911 p2: -0,0406592883157251 f(p2): -0,0000000000074759

Raiz encontrada para P0 = -1,0000000000000000 e P1 = 0,0000000000000000: -0,0406592883157251

Digite um número para a Aproximação P0: 0
Digite um número para a Aproximação P1: 1
i: 1  p0: 0,0000000000000000 p1: 1,0000000000000000 p2: 0,2500000000000000 f(p2):-62,5078125000000000
i: 2  p0: 1,0000000000000000 p1: 0,2500000000000000 p2: 0,7737627651217597 f(p2):-83,8305202790104000
i: 3  p0: 0,2500000000000000 p1: 0,7737627651217597 p2:-1,2854177835209231 f(p2): 879,6389856119251400
i: 4  p0: 0,7737627651217597 p1:-1,2854177835209231 p2: 0,5945955204028854 f(p2):-104,6913894786802200
i: 5  p0:-1,2854177835209231 p1: 0,5945955204028854 p2: 0,3946411046833960 f(p2):-88,1289403624369640
i: 6  p0: 0,5945955204028854 p1: 0,3946411046833960 p2:-0,6693181355515908 f(p2): 183,7131597403886200
i: 7  p0: 0,3946411046833960 p1:-0,6693181355515908 p2: 0,0497143976160742 f(p2):-19,9610215816172140
i: 8  p0:-0,6693181355515908 p1: 0,0497143976160742 p2:-0,0207541508238155 f(p2):-4,4095742941073253
i: 9  p0: 0,0497143976160742 p1:-0,0207541508238155 p2:-0,0407353328963781 f(p2): 0,0168594733503808
i: 10 p0:-0,0207541508238155 p1:-0,0407353328963781 p2:-0,0406592282432061 f(p2):-0,0000133183513628
i: 11 p0:-0,0407353328963781 p1:-0,0406592282432061 p2:-0,0406592883155716 f(p2):-0,0000000000415126

Raiz encontrada para P0 = 0,0000000000000000 e P1 = 1,0000000000000000: -0,0406592883155716
```

Exercício 2.3 – 24 (Burden, 2015)

Utilizaremos o método da bissecção para termos uma boa estimativa do valor da raiz da função abaixo:

$$f(x) = \frac{1500}{x} \cdot [(1 + x)^{240} - 1] - 750000; \quad \text{Intervalo} = [-10, 20]; \quad tol. = 10^{-3}$$

```
Digite um número para o início do intervalo: -10
Digite um número para o fim do intervalo: 20
i: 15 a: 0,0048828125000000 b: 0,0067138671875000 c: 0,0057983398437500 f(c): 27418,1132068717020000
i: 16 a: 0,0048828125000000 b: 0,0057983398437500 c: 0,0053405761718750 f(c): -22373,3182413807630000
```

Com os valores de obtidos nas iterações 15 e 16 aplicados no método da Secante como p_0 e p_1 e tolerância de 10^{-16} temos:

```
Digite um número para a Aproximação P0: 0,0057983398437500
Digite um número para a Aproximação P1: 0,0053405761718750
i: 1 p0: 0,0057983398437500 p1: 0,0053405761718750 p2: 0,0055462680351597 f(p2): -489,0574751153636200
i: 2 p0: 0,0053405761718750 p1: 0,0055462680351597 p2: 0,0055508647242772 f(p2): 8,9767790729082435
i: 3 p0: 0,0055462680351597 p1: 0,0055508647242772 p2: 0,0055507818716173 f(p2): -0,0035187573866438
i: 4 p0: 0,0055508647242772 p1: 0,0055507818716173 p2: 0,0055507819040816 f(p2): -0,0000000183220550
i: 5 p0: 0,0055507818716173 p1: 0,0055507819040816 p2: 0,0055507819040817 f(p2): -0,0000000411004635
i: 6 p0: 0,0055507819040816 p1: 0,0055507819040817 p2: 0,0055507819040814 f(p2): -0,0000000539742473
i: 7 p0: 0,0055507819040817 p1: 0,0055507819040814 p2: 0,0055507819040827 f(p2): 0,0000000974193881
i: 8 p0: 0,0055507819040814 p1: 0,0055507819040827 p2: 0,0055507819040819 f(p2): -0,0000000073868591
i: 9 p0: 0,0055507819040827 p1: 0,0055507819040819 p2: 0,0055507819040819 f(p2): -0,0000000153027599
i: 10 p0: 0,0055507819040819 p1: 0,0055507819040819 p2: 0,0055507819040818 f(p2): -0,0000000001171543
```

```
i: 11 p0: 0,0055507819040819 p1: 0,0055507819040818 p2: 0,0055507819040818 f(p2): 0,0000000000443947
i: 12 p0: 0,0055507819040818 p1: 0,0055507819040818 p2: 0,0055507819040818 f(p2): 0,0000000000443947
```

A taxa mínima de juros compostos mensalmente para o atingimento da quantia é de 0,55507819040818 %.

Exercício 2.3 – 27 (Burden, 2015)

Utilizaremos o método da bissecção para termos uma boa estimativa do valor da raiz da função abaixo:

$$f(x) = \frac{1+x}{2} \cdot \left(\frac{x}{1-x+x^2}\right)^{21} - 0{,}5; \quad \text{Intervalo} = [0, 1]; \quad \text{tol.} = 10^{-2}$$

```
Digite um número para o início do intervalo: 0
Digite um número para o fim do intervalo: 1
i: 1   a: 0,0000000000000000  b: 1,0000000000000000  c: 0,5000000000000000  f(c): -0,4998496356700892
i: 2   a: 0,5000000000000000  b: 1,0000000000000000  c: 0,7500000000000000  f(c): -0,3370689630289616
i: 3   a: 0,7500000000000000  b: 1,0000000000000000  c: 0,8750000000000000  f(c): 0,1464692535436415
i: 4   a: 0,7500000000000000  b: 0,8750000000000000  c: 0,8125000000000000  f(c): -0,1276737613564455
i: 5   a: 0,8125000000000000  b: 0,8750000000000000  c: 0,8437500000000000  f(c): 0,0064349264939472

Raiz no intervalo [0,000000,1,000000]: 0,8437500000000000
```

Com os valores de obtidos nas iterações 4 e 5 aplicados no método da Secante como p_0 e p_1 e tolerância de 10^{-16} temos:

```
Digite um número para a Aproximação P0: 0,8125000000000000
Digite um número para a Aproximação P1: 0,8437500000000000
i: 1 p0: 0,8125000000000000 p1: 0,8437500000000000 p2: 0,8422505337375296 f(p2): -0,0002413113600136
i: 2 p0: 0,8437500000000000 p1: 0,8422505337375296 p2: 0,8423047316632186 f(p2): -0,0000002640719190
i: 3 p0: 0,8422505337375296 p1: 0,8423047316632186 p2: 0,8423047910380839 f(p2): 0,0000000000112000
i: 4 p0: 0,8423047316632186 p1: 0,8423047910380839 p2: 0,8423047910355658 f(p2): 0,0000000000000009

Raiz encontrada para P0 = 0,8125000000000000 e P1 = 0,8437500000000000: 0,8423047910355658
```

A probabilidade p calculada é de 0,842.

4. CONCLUSÃO

O método numérico de Secante é relativamente mais complexo se comparado com outros métodos de cálculo de raízes de funções. Requer conhecimento sobre operações matemáticas e pode ser utilizado com o uso de calculadora ou planilha de cálculo dependendo da complexidade da função estudada, porém não necessita da determinação direta da derivada da função tornando o mais simples do que o método de Newton - Raphson.

Para ser aplicado requer o conhecimento de dois valores estimados que devem ser muito próximos da raiz se comparado com os valores estimados de intervalo utilizados no método da bissecção, configurando assim um método mais utilizado para o refinamento do resultado de outros métodos como, por exemplo, o da bissecção.

Uma das desvantagens analisadas através dos exercícios resolvidos foi que o método exige uma estimativa dos valores p_0 e p_1 muito próxima da raiz da função e não garante a convergência.

Conhecendo esta estimativa e os limitantes de aplicação, o método converge para uma raiz de forma rápida se comparado ao método da bissecção (tabela 2), pois possui uma convergência de ordem quadrática, porém é ligeiramente mais lento que o método de Newton-Raphson por apresentar em sua definição uma aproximação da derivada da função presente no método anterior.

Tabela 2

$$\mathrm{f(x)} = \mathrm{x}^3 + 4\mathrm{x}^2 - 10; \quad \mathrm{Intervalo} = [1,2]; \quad \mathrm{tol.} = 10^{-16}$$

Método	**Convergência**	**Iterações**
Bissecção	Linear	53
Newton	Quadrática	5
Secante	Quadrática	7

5. REFERÊNCIAS

Burden, R. L. (2011). *Numerical Analysis* (9th Edition ed.). Boston: Cengage Learning.

Burden, R. L. (2015). *Análise Numérica* (Tradução da 10ª edição norte-americana ed.). São Paulo: Cengage Learning.

ANEXO A. CÓDIGO FONTE

```
/*********************************************
 * \brief   Programa realizado para a Aula de Métodos Númericos em Fenômenos de Transporte
 *          Análise Numérica - Burden, Richard - 10ª edição - Pág. 78
 *          Algoritmo 2.4 - Método da Secante
 *
 * \param   Aproximação Inicial P0
 * \param   Aproximação Inicial P1
 * \return  Raiz da função
 *
 *********************************************/

#include <stdio.h>
#include <math.h>
#include <locale.h>

#define tol 10e-16  /*valor mínimo de tolerância = 10e-16*/
#define n_max 100   /*valor máximo para iterações = 100*/

double f(double x){
    /*validação ex. 2 no ponto p0 = 0,5 e p1 = 0,785398*/   return (cos(x)-x);
    /*comparação com método da bissecção p0 = 1 e p1 = 2*/  /*return (pow(x,3)+4*pow(x,2)-10);*/
    /*exercicio 2.3-3a no ponto p0=3, p1=2 e n_max=3*/      /*return (pow(x,2) - 6);*/
    /*exercicio 2.3-8a no ponto p0=1 e p1=2 */              /*return (exp(x) + pow(2,-x) + 2 * cos(x) -
6);*/
    /*exercicio 2.3-8b no ponto p0=1,3 e p1=2 */            /*return (log(x - 1) + cos(x - 1));*/
    /*exercicio 2.3-8f em [p0,p1] = [0,1],[3,4],[6,7]*/     /*return (sin(x) - exp(-x));*/
    /*exercicio 2.3-13b em [p0,p1] = [-1,0], [0,1]*/         /*return (230 * pow(x,4) + 18 * pow(x,3) +
9 * pow(x,2) - 221 * x - 9);*/
    /*exercicio 2.3-18 em  [p0,p1] = [pi,pi/2]*/            /*return (1/2 + 1/4 * pow(x,2) - x * sin(x)
- 1/2 * cos(2 * x));*/
    /*exercicio 2.3-24 em  [p0,p1] = [0,0057,0,0053]*/      /*return ((1500/x) * (pow((1 + x),240) - 1)
- 750000);*/
    /*exercicio 2.3-27 em  [p0,p1] = [0,81,0,84]*/           /*return(((1 + x) / 2) * pow((x / (1 - x +
pow(x,2))),21) - 0.5);*/
}

double Secante(double p0, double p1){
    double p2;
    int i = 1;

    do{
        p2 = p1 - (f(p1) * (p1 - p0)) / (f(p1) - f(p0));

        /*printf("i: %i \t p0: %.16f \t p1: %.16f \t p2: %.16f \t f(p2): %.16f \n", i, p0, p1, p2,
f(p2));*/
        printf("i: %i p0: %.16f p1: %.16f p2: %.16f f(p2): %.16f \n", i, p0, p1, p2, f(p2));


        if (fabs(f(p2)) <= tol) {return p2;} else {p0 = p1; p1 = p2;}

        i++;
    }while( i <= n_max);

    printf("\nErro! O número máximo de iterações foi atingido.\n\n");
    return p2;
}

int main(void){

    setlocale(LC_ALL,"");

    double p, p0, p1;

    printf("Digite um número para a Aproximação P0: ");
    scanf("%lf", &p0);
    printf("Digite um número para a Aproximação P1: ");
    scanf("%lf", &p1);

    p = Secante(p0, p1);
```

```
    if (fabs(f(p)) <= tol)
        {printf("\nRaiz encontrada para P0 = %.16f e P1 = %.16f: %.16f \n\n", p0, p1, p);}
    else
        {printf("\nErro! Não encontrada raiz da função!\n\n");}

    return 0;
}
```