UNIVERSIDADE ESTADUAL DE GOIÁS
Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo
Curso de Matemática

Métodos Numéricos para Aproximação da Raiz de uma Função: Teoria e Aplicações

TAYNÁ ELLEN NOÉ MARTINS

Anápolis
2018

TAYNÁ ELLEN NOÉ MARTINS

Métodos Numéricos para Aproximação da Raiz de uma Função: Teoria e Aplicações

Trabalho de Curso (TC) apresentado a Coordenação Adjunta de TC, como parte dos requisitos para obtenção do título de Graduado no Curso de Matemática da Universidade Estadual de Goiás.

Orientador: M.e Tiago de Lima Bento Pereira

Anápolis

2018

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UEG
com os dados fornecidos pelo(a) autor(a).

N M386
m

Noé Martins, Tayná Ellen
Métodos Numéricos para Aproximação da Raiz de uma Função: Teoria e Aplicações / Tayná Ellen Noé Martins; orientador Tiago de Lima Bento Pereira. -- Anápolis, 2018.
47 p.

Graduação - Matemática -- Câmpus-Anápolis CET, Universidade Estadual de Goiás, 2018.

1. Solução de equação. 2. Método da Bisseção. 3. Método de Newton-Raphson. 4. Excel. 5. MATLAB. I. de Lima Bento Pereira, Tiago, orient. II. Título.

À minha mãe e minha avó, sem elas eu não seria metade do que sou.

AGRADECIMENTOS

Agradeço a Deus por me conceder saúde para lutar pela realização dos meus sonhos e à Nossa Senhora, a quem sempre recorro em minhas orações

Agradeço à minha mãe, por todo amor, carinho e broncas nos momentos necessários, por ser meu porto seguro e por seus esforços incontáveis para proporcionar a melhor educação que eu poderia ter. À minha avó por suas orações e ajuda nos momentos difíceis. E para minha família em geral, irmã, sobrinho, madrinhas, padrinho, tias e tios, primas e primos, que sempre torceram pelo meu sucesso.

Aos meus amigos, pela paciência e por não me abandonar durante esse período que estive afastada. Às minhas colegas de turma pelo companheirismo, choros e risos durante essa jornada.

Aos professores do curso de Matemática que compartilharam conhecimentos e valores que irei levar por toda a minha vida. À Jô que, mesmo estando ausente durante a realização deste trabalho, sempre teve um cuidado de mãe com todos à sua volta.

Um especial agradecimento ao meu orientador professor M.e Tiago pela paciência, dedicação e amparo para a realização deste trabalho. E aos membros da banca examinadora que aceitaram fazer parte desse momento tão importante para minha formação.

"Grandes realizações não são feitas por impulso, mas por uma soma de pequenas realizações"

Vincent Van Gogh

RESUMO

Em diversos problemas das ciências exatas há a necessidade de determinar a solução de uma equação, ou a raiz real (zero real) de uma função. Utiliza-se métodos numéricos para solução de problemas cujas técnicas analíticas são de difícil aplicação e em alguns casos até mesmo inviáveis. Valendo-se de uma revisão bibliográfica apresenta-se os métodos numéricos da Bisseção e de Newton-Raphson para a busca de um valor aproximado da raiz real de uma função. Explana-se os conceitos relacionados a ambos os métodos e suas justificativas matemáticas. Para auxílio nos cálculos valeu-se de recursos computacionais, utilizando os so*ftwares* Microsoft Office Excel e MATLAB®, apresentando para ambos uma introdução de como aplicá-los. Como emprego da busca de solução de equações apresenta-se, pelos métodos da Bisseção e Newton-Raphson, uma solução para a determinação de variáveis psicrométricas, em especial a temperatura de bulbo úmido, com base em uma equação e dados utilizados em Engenharia Agrícola e Agronomia.

Palavras-chave: Solução de equação. Método da Bisseção. Método de Newton-Raphson. Excel. MATLAB®.

Métodos Numéricos para Aproximação da Raiz de uma Função: Teoria e Aplicação

TAYNA ELLEN NOÉ MARTINS

Trabalho de Curso de Matemática apresentado à Banca Examinadora como parte dos requisitos para a obtenção do grau de graduado em Licenciatura em Matemática.

Banca Examinadora do Trabalho de Curso de Matemática do Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo da Universidade Estadual de Goiás, Anápolis, sexta-feira, 30 de novembro de 2018.

M.e Tiago de Lima Bento Pereira
Presidente da Banca Examinadora

Leonardo Antonio Souto

M.e Leonardo Antonio Souto
1º Membro da Banca Examinadora

M.e Selma Marques de Paiva
2º Membro da Banca Examinadora

SUMÁRIO

INTRODUÇÃO

Métodos numéricos são ferramentas utilizadas para obter soluções aproximadas de problemas matemáticos por meio de calculadoras ou computador. Tais métodos tem aplicação em problemas que são difíceis de serem resolvidos através de métodos analíticos. Segundo Chapra (2011, p.02):

> Os métodos numéricos são técnicas pelas quais os problemas matemáticos são formulados de modo que possam ser resolvidos com operações aritméticas. Embora existam muitos tipos de métodos numéricos, eles têm uma característica em comum: invariavelmente envolvem grande número de cálculos aritméticos tediosos.

Os procedimentos efetuados antes dos computadores normalmente eram realizados por meio de métodos analíticos ou exatos, que são encontrados através de fórmulas, desenvolvidas manualmente, mas que abrangem apenas uma parte dos problemas mais simples, tendo assim um valor limitado, como afirma Chapra (2011, p. 2). O Método Numérico por sua vez, é uma opção que fornece soluções de maneira relativamente rápida, que abrangem um número maior de problemas, como equações mais complexas que por meio dos métodos analíticos se tornam inviáveis pelo grande número de cálculos e tempo de realização.

Para desenvolvimento dos cálculos, são utilizados processos de iteração. Chama-se de processo de iteração a repetição de um determinado procedimento que gera uma sequência $x_1, x_2, \dots, x_n, \dots$ dos resultados procurados, como afirma Barroso (1987, p.49). A repetição consiste em utilizar o resultado da iteração anterior para realizar um novo cálculo e obter um novo elemento da sequência. Assim é obtida uma solução aproximada a partir de um valor inicial. Espera-se que a sequência, $x_1, x_2, \dots, x_n, \dots$ de resultados obtidos nas iterações aproxime-se (convirja) para o resultado procurado. Para determinar o término dos cálculos das iterações são utilizados critérios de parada. Por exemplo: erro relativo, erro absoluto, entre outros.

Um problema frequente é a busca de um valor para x que satisfaça a equação $f(x) = 0$ ou zero de uma função. Os métodos analíticos oferecem cobertura apenas em casos mais simples, mas quando essa equação tem um grau maior que quatro ou é uma equação transcendental[1], os métodos numéricos se tornam mais viáveis e em alguns casos a única opção. Dentre os métodos numéricos que facilitam a busca por raízes de uma equação podemos citar o Método da Bisseção e o Método de Newton-Raphson.

O objetivo deste trabalho é estudar os Métodos da Bisseção e de Newton-Raphson sem o auxílio de recursos computacionais, buscando compreender o comportamento de cada um deles. Após essa etapa busca-se apresentar opções rápidas e viáveis para a aplicação desses métodos com o uso de programas computacionais. Para isso foi realizada uma revisão bibliográfica com o intuito de estudar cada método e os programas que foram utilizados para a resolução e por último realizar uma aplicação de tais métodos.

No primeiro capítulo são estudados os Métodos Numéricos da Bisseção e Newton-Raphson, com os cálculos realizados manualmente. No segundo e terceiro capítulos são apresentadas opções para a resolução de tais métodos utilizando respectivamente os programas computacionais Microsoft Office Excel e MATLAB®. O quarto capítulo busca estimar valores, utilizando dos programas desenvolvidos no terceiro capítulo, para a temperatura de um termômetro de bulbo úmido por meio de uma equação e dados utilizados em Engenharia Agrícola e Agronomia, considerando as informações de Miranda et al. (2006).

[1] “Entende-se por equação transcendental aquela que envolve funções transcendentais como $sen\ x, e^x, ln\ x$.” (SPERANDIO, et al. 2003, p. 15).

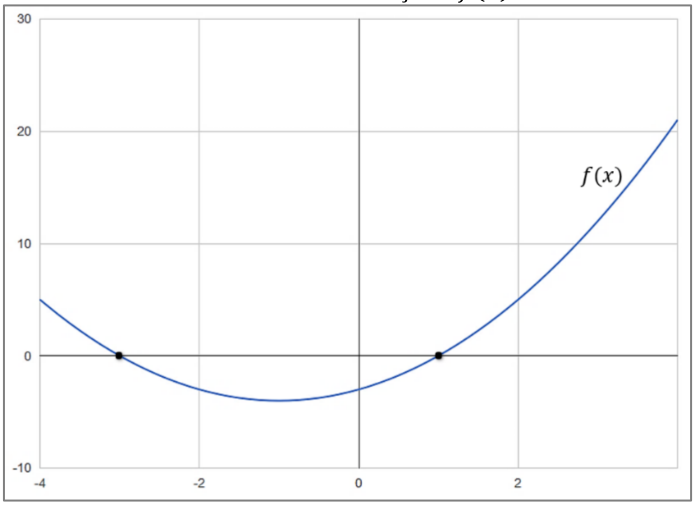
1 MÉTODOS NUMÉRICOS PARA DETERMINAR ZEROS REAIS DE FUNÇÕES

Nos mais diversos campos de estudo da Matemática existem problemas que envolvem a busca por um determinado número β que seja a solução de equações do tipo $f(x) = 0$. Em outras palavras, um número β tal que $f(\beta) = 0$. Tal solução geralmente é denominada raiz da equação $f(x) = 0$ ou zero da função $f(x)$. A determinação de tais raízes é especialmente importante em áreas da engenharia, em determinados projetos, nos quais parecem ser impossíveis resolver determinadas equações (CHAPRA, 2011, p. 4).

Analisando o gráfico de uma função, podemos verificar que os zeros desta são os valores nos quais o gráfico da função intercepta (corta) o eixo das abscissas. Em alguns casos é possível encontrar a solução exata da equação $f(x) = 0$ por meio de artifícios algébricos simples, como ocorre em certas funções polinomiais de grau menor que 4.

Exemplo 1.0.1. Na função $f(x) = x^2 + 2x - 3$ é possível observar que -3 e 1 são os zeros da função, ou seja, $f(-3) = 0$ e $f(1) = 0$. A Figura 1.1 exibe o gráfico de $f(x)$ onde destacam-se os zeros da função.

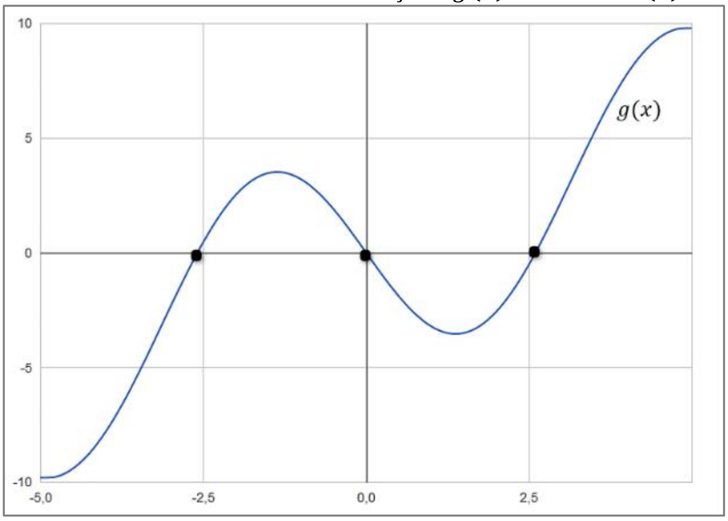
FIGURA 1.1 – GRÁFICO DA FUNÇÃO $f(x) = x^2 + 2x - 3$.



FONTE: Autoria própria (2018).

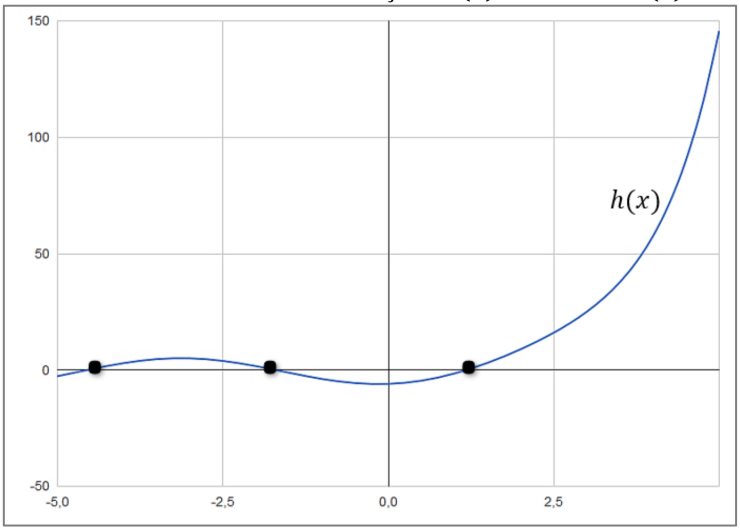
Exemplo 1.0.2. As funções $g(x) = x - 5sen(x)$ e $h(x) = e^x - 6\cos(x) - 1$ são exemplos de equações transcendentais e analisando as Figuras 1.2 e 1.3 podemos afirmar que as funções $g(x)$ e $h(x)$ tem cada uma ao menos 3 zeros reais.

FIGURA 1.2 – GRÁFICO DA FUNÇÃO $g(x) = x - 5sen(x)$.



FONTE: Autoria própria (2018).

FIGURA 1.3 -GRÁFICO DA FUNÇÃO $h(x) = e^x - 6\cos(x) - 1$.



FONTE: Autoria própria (2018).

Para determinar as raízes das equações citadas nos exemplos anteriores pode-se recorrer a métodos numéricos, que podem não fornecer a raiz exata, mas apresentar uma aproximação com nível de exatidão previamente determinado. O objetivo principal desses métodos é ao partir de uma primeira aproximação, determinar novas aproximações através de processos iterativos.

Os métodos para a obtenção de raízes são divididos em duas fases:

1. Isolamento das raízes;
2. Refinamento.

Nas próximas seções, descrevemos essas fases.

Destaca – se como principais referências, nesse aspecto, Barroso (1987), Chapra (2011), Ruggiero (1996) e Sperandio (2003).

1.1 Isolamento das Raízes

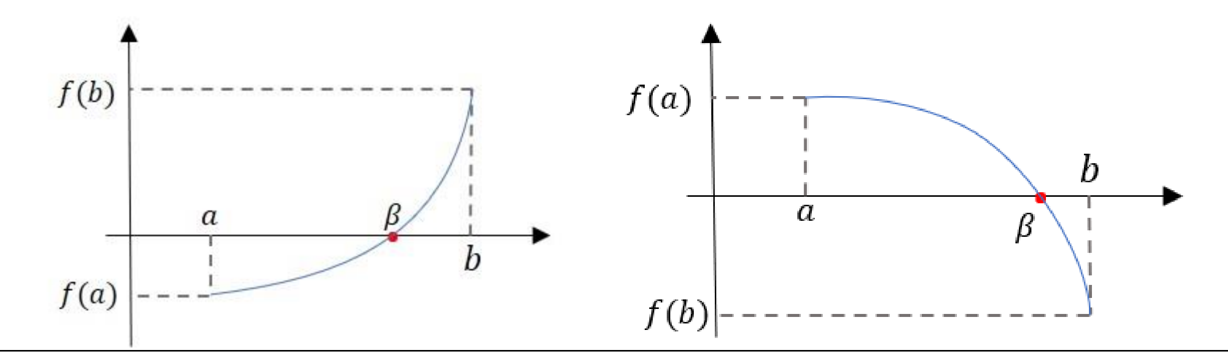
Nessa etapa do processo busca-se determinar um intervalo que contenha ao menos uma raiz da função. Para tal são feitas análises teóricas e ou gráficas da função. Para a análise teórica pode-se utilizar, dentre outros artifícios, o Teorema do Anulamento de Bolzano (caso particular do Teorema do Valor Intermediário) como suporte para os métodos em um intervalo $[a, b]$.

Teorema do Anulamento de Bolzano: Seja $f(x)$ uma função contínua num intervalo $[a, b]$. Se

$$f(a) \cdot f(b) < 0$$

então existe pelo menos um ponto $x = \beta$ entre a e b, onde $f(x) = 0$.

FIGURA 1.4 – REPRESENTAÇÃO GRÁFICA DO TEOREMA DO ANULAMENTO DE BOLZANO.



FONTE: Autoria própria (2018).

Assim uma das maneiras de encontrar um intervalo para a função é supor valores de x, para calcular $f(x)$, tabelando os resultados e analisando as mudanças de sinal da função.

Exemplo 1.1: Isolar as raízes de $f(x) = x^2 + 4\,sen(2x) + 1$.

TABELA 1.1: ESTUDO DE SINAL DA FUNÇÃO $f(x) = x^2 + 4sen(2x) + 1$.

x	-3	-2	-1	0	1	2	3
$f(x)$	$11,1177$	$8,0272$	$-1,6372$	$1,0000$	$5,6372$	$1,9728$	$8,8823$

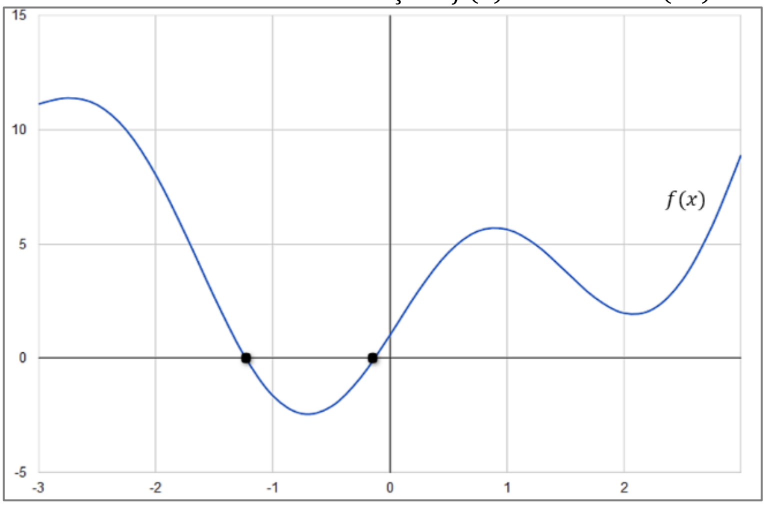
FONTE: Autoria própria (2018).

Pelo Teorema do Anulamento de Bolzano e com base na Tabela 1.1, ao analisar a troca de sinais em $f(x)$ observa-se que a função possui zeros nos intervalos $[-2,-1]$ e $[-1,0]$.

Outra maneira de encontrar um intervalo que contenha a raiz é por meio da análise do seu gráfico. Utilizando o esboço do gráfico, pode-se verificar onde há interseção com o eixo das abscissas (eixo x), nesses pontos de interseção estão localizados os zeros da função.

Esse método se torna mais viável quando a função é conhecida e o esboço de seu gráfico pode ser feito de maneira prática com o uso de programas para a construção de gráficos, tais como: Calculadora Online, Geogebra, MATLAB e Scilab.

FIGURA 1.5 – GRÁFICO DA FUNÇÃO $f(x) = x^2 + 4\,sen(2x) + 1$.



FONTE: Autoria própria (2018).

Analisando a Figura 1.5, é possível observar que existe uma raiz de $f(x) = x^2 + 4\,sen(2x) + 1$, no intervalo $[-2,-1]$ e uma outra raiz no intervalo $[-1,0]$.

Uma terceira possibilidade de encontrar intervalos que contenham um zero da função é a partir da equação $f(x) = 0$, separarando $f(x)$ em duas funções equivalentes e substituí-las na equação, ou seja, tornar $f(x) = g(x) - h(x)$, e assim obter a seguinte equação

$$g(x) - h(x) = 0,$$

que implica $g(x) = h(x)$. Desse modo, os valores de x para os quais $g(x) = h(x)$ são os valores para os quais $f(x) = 0$. De fato, tomando $x = \beta$ tem-se que:

$$g(\beta) = h(\beta)$$

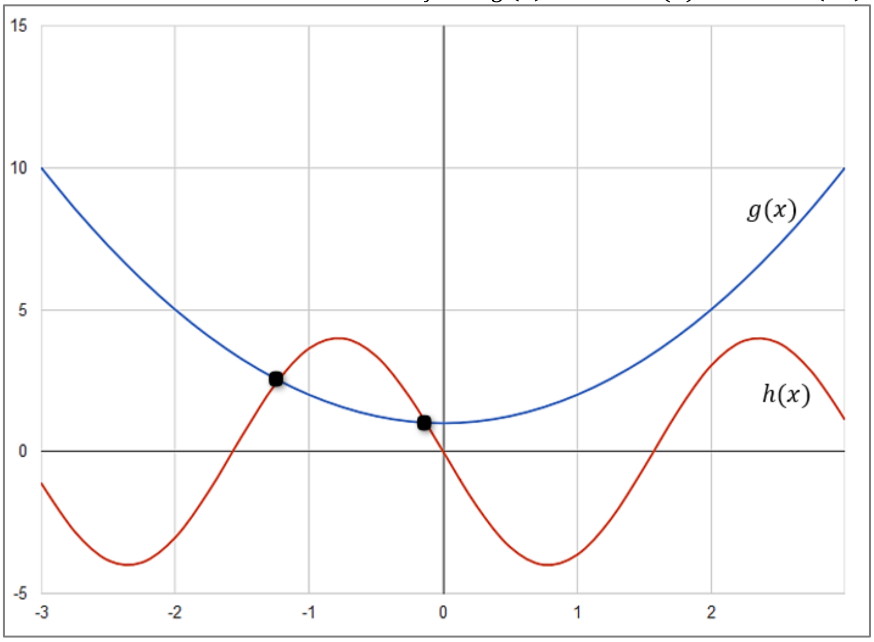
ou seja,

$$f(\beta) = g(\beta) - h(\beta) = 0$$

concluindo assim que, β é uma raiz. Logo, ao esboçar os gráficos de $g(x)$ e $h(x)$ no plano cartesiano, o ponto em que eles se interceptam é o local onde encontra-se um valor para a raiz de $f(x)$.

Considerando a equação $f(x) = x^2 + 4\,sen(2x) + 1 = 0$, pode-se reescreve-la transformando-a em $g(x) = h(x)$, onde, $g(x) = x^2 + 1$ e $h(x) = -4\,sen(2x)$. Os gráficos das funções $g(x)$ e $h(x)$ são mais simples de serem esboçados quando comparados ao gráfico da função $f(x)$. As raízes são dadas pelos pontos em que $g(x)$ intercepta $h(x)$.

FIGURA 1.6 - GRÁFICOS DAS FUNÇÕES $g(x)x^2 + 1$ e $h(x) = -4\,sen(2x)$.



FONTE: Autoria própria (2018).

Analisando a Figura 1.6, percebe-se a existência de raízes nos intervalos, nos intervalos $[-2, -1]$ e $[-1; 0]$, intervalos que possuem pontos nos quais os gráficos das funções $g(x)$ e $h(x)$ se encontram.

Depois de isoladas as raízes, é realizada a segunda fase: o refinamento.

1.2 Refinamento

Nessa fase, após determinado o intervalo ou a aproximação inicial, pode-se utilizar de diversos métodos numéricos para o refinamento ou aprimoramento da raiz. Esses métodos consistem em processos de iteração, que são sequências de passos, os quais são repetidos até chegar à solução aproximada desejada.

Como o processo iterativo pode ser feito uma quantidade de vezes indefinida é necessário um momento de cessar tal procedimento. Para tal são considerados critérios de parada dentro de uma precisão, ε, previamente estabelecida, normalmente expressa por uma potência 10^{-n}.

Entendemos como critério de parada as seguintes expressões:

Erro Absoluto: $|x_{n+1} - x_n| \leq \varepsilon$;

Erro relativo: $\frac{|x_{n+1}-x_n|}{|x_n|} \leq \varepsilon$.

Um terceiro critério de parada, é reduzir a amplitude do intervalo $[a, b]$ em que a raiz está contida, tal que: Se $\beta \in [a, b]$ e $(b - a) < \epsilon$ então $\forall\, x \in [a, b], |x - \beta| < \epsilon$. Portanto, $\forall\, x \in [a, b]$ pode ser tomado como uma raiz aproximada (RUGGIERO, 1996, p. 39).

Nas próximas seções serão estudados os Métodos da Bisseção e de Newton-Raphson.

1.2.1 Método da Bisseção

O método da bisseção, ou método de truncamento binário, ou ainda método do meio intervalo consiste em achar a raiz de uma determinada equação $f(x) = 0$ através da divisão do intervalo ao qual a raiz pertence.

> [...] método de busca incremental no qual o intervalo é sempre dividido na metade. Se uma função muda de sinal em um intervalo, calcula-se o valor da função em seu ponto médio. A posição da raiz é então determinada como sendo o ponto médio do subintervalo no qual a mudança de sinal ocorre. (CHAPRA, 2011, p.101)

Assim, através de aproximações calculadas a partir do ponto médio do intervalo $[a, b]$ é calculada a primeira aproximação, dada por:

$$x_1 = \frac{a+b}{2}.$$

Após encontrado o ponto médio, podem ocorrer três situações:

1. $f(x_1) = 0$, assim x_1 é raiz da equação, finalizando os cálculos;
2. $f(a) \cdot f(x_1) < 0$, então a raiz está no intervalo $[a, x_1]$. Possibilitando realizar o processo novamente (uma nova iteração) a partir desse novo intervalo (obtido ao substituir b por x_1), tomando:

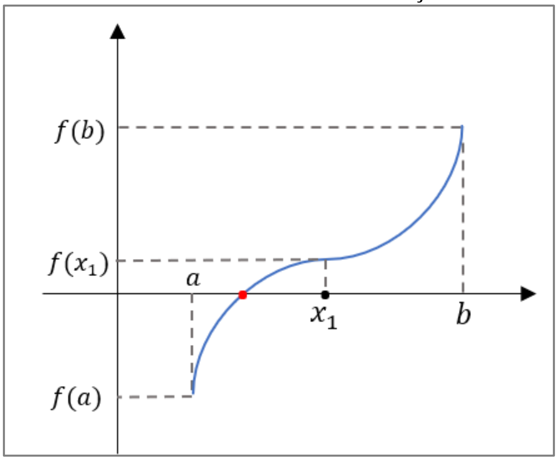
$$x_2 = \frac{a+x_1}{2}.$$

3. $f(b) \cdot f(x_1) < 0$, portanto a raiz está no intervalo $[x_1, b]$, obtido ao substituir a por x_1 assim para a nova iteração:

$$x_2 = \frac{x_1+b}{2}.$$

A situação (2) descrita acima, é exposta na Figura 1.7.

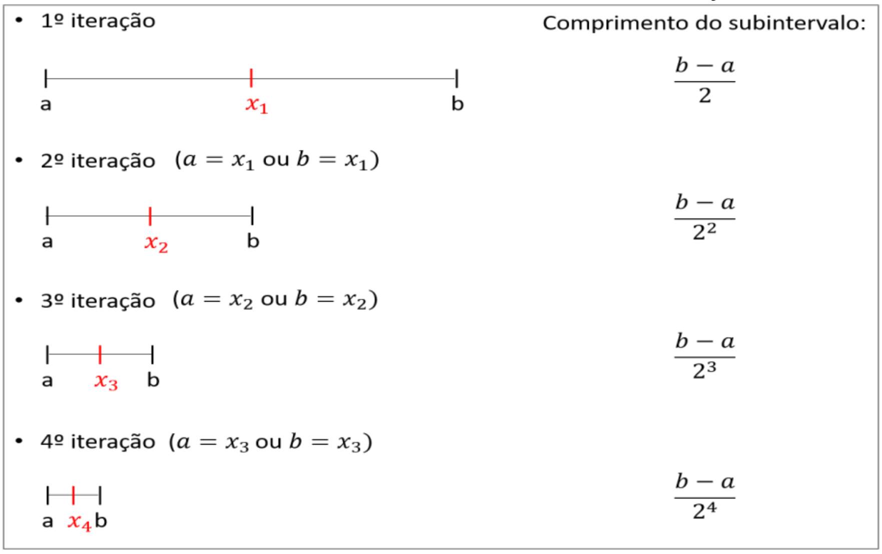
FIGURA 1.7 – INTERPRETAÇÃO GEOMÉTRICA MÉTODO DA BISSEÇÃO.



FONTE: Autoria própria (2018).

O processo é repetido sucessivamente através de novos intervalos que contenham a raiz, dividindo assim a amplitude de $[a, b]$ conforme a Figura 1.8.

FIGURA 1.8 – DIVISÃO INTERVALAR MÉTODO DA BISSEÇÃO.



FONTE: Autoria própria (2018).

A divisão é repetida até a enésima aproximação x_n, onde o tamanho do intervalo será dado por:

$$|x_n - x_{n-1}| = \frac{b-a}{2^n}.$$

Pelo critério do erro absoluto, $|x_n - x_{n-1}| \leq \varepsilon$. Assim:

$$\left|\frac{b-a}{2^n}\right| \leq \varepsilon.$$

Aplicando as propriedades adequadas e trabalhando essa fórmula, temos:

$$\frac{b-a}{2^n} \leq \varepsilon$$

$$\ln\left(\frac{b-a}{2^n}\right) \leq \ln \varepsilon$$

$$\ln(b-a) - \ln 2^n \leq \ln \varepsilon$$

$$\ln(b-a) - n \ln 2 \ \leq \ln \varepsilon$$

$$n \geq \frac{\ln(b-a) - \ln \varepsilon}{\ln 2}$$

Assim, são necessárias n iterações para calcular a raiz β no intervalo $[a, b]$ considerando a precisão ε.

Exemplo 1.2: Utilizando do método da Bisseção, determinar uma aproximação para o zero da função, considerando $\varepsilon < 10^{-3}$:

$$f(x) = x^2 + 4\,sen(2x) + 1.$$

Na seção (1.1) determinou-se os intervalos em que a equação $f(x) = 0$, tomando o intervalo $[-2, -1]$ e sabendo que $f(-2) = 8{,}0272$ e $f(-1) = -1{,}6378$.

Calculando o número de iterações:

$$n \geq \frac{\ln\big(-1 - (-2)\big) - \ln 10^{-3}}{\ln 2} \geq 9{,}9658$$

Assim o número de iterações a serem realizadas será igual a 10.

Intervalo 1: $[a, b] = [-2, -1]$, primeira iteração (x_1),

$$x_1 = \frac{a+b}{2} = \frac{(-2)+(-1)}{2} = -1{,}5$$
$$f(x_1) = f(-1{,}5) = 2{,}6855$$
$$f(a) \cdot f(x_1) > 0$$
$$f(b) \cdot f(x_1) < 0.$$

Como o produto de $f(b)$ e $f(x_1)$ é negativo, o novo intervalo será $[-1{,}5; -1]$. Substituindo a por x_1.

Intervalo 2: $[a, b] = [-1{,}5; -1]$, segunda iteração (x_2),

$$x_2 = \frac{a+b}{2} = \frac{(-1{,}5)+(-1)}{2} = -1{,}25$$
$$f(x_2) = f(-1{,}25) = 0{,}1686$$
$$f(a) \cdot f(x_2) > 0$$
$$f(b) \cdot f(x_2) < 0$$
$$\varepsilon = |x_2 - x_1| = |1{,}25 - 1{,}5| = 0{,}25.$$

Como o produto de $f(b)$ e $f(x_2)$ é negativo, o novo intervalo será $[-1{,}25; -1]$. Substituindo a por x_2.

[...]

Intervalo 9: $[a, b] = [-1{,}234375; -1{,}23046875]$, nona iteração (x_9),

$$x_9 = \frac{a+b}{2} = \frac{(-1{,}234375)+(-1{,}23046875)}{2} = -1{,}232421875$$
$$f(x_9) = f(-1{,}23242) = 0{,}013816712$$
$$f(a) \cdot f(x_9) > 0$$
$$f(b) \cdot f(x_9) < 0$$

$$\varepsilon = |x_9 - x_8| = |1{,}232421875 - 1{,}232421875| = 0{,}001953125.$$

Como o produto de $f(b)$ e $f(x_9)$ é negativo, o novo intervalo será $[-1{,}23242; -1{,}23047]$. Substituindo a por x_9.

Intervalo 10: $[a, b] = [-1{,}232421875; -1{,}23046875]$, décima iteração ($x_{10}$),

$$x_{10} = \frac{a+b}{2} = \frac{(-1{,}232421875) + (-1{,}23046875)}{2} = -1{,}231445313$$

$$f(x_{10}) = f(-1{,}23145) = 0{,}005324648.$$

Erro: $\varepsilon = |x_{10} - x_9| = |1{,}231445313 - 1{,}232421875| = 0{,}000976563.$

Assim, utilizando o método da bisseção, o valor aproximado para a raiz da função $f(x) = x^2 + 4\,sen(2x) + 1$ no intervalo $[-2;\ -1]$ é -1,2314.

A Tabela 1.2 apresenta os resultados de todas as dez iterações.

TABELA 1.2: RESULTADOS DE $f(x) = x^2 + 4sen(2x) + 1$ PELO MÉTODO DA BISSEÇÃO.

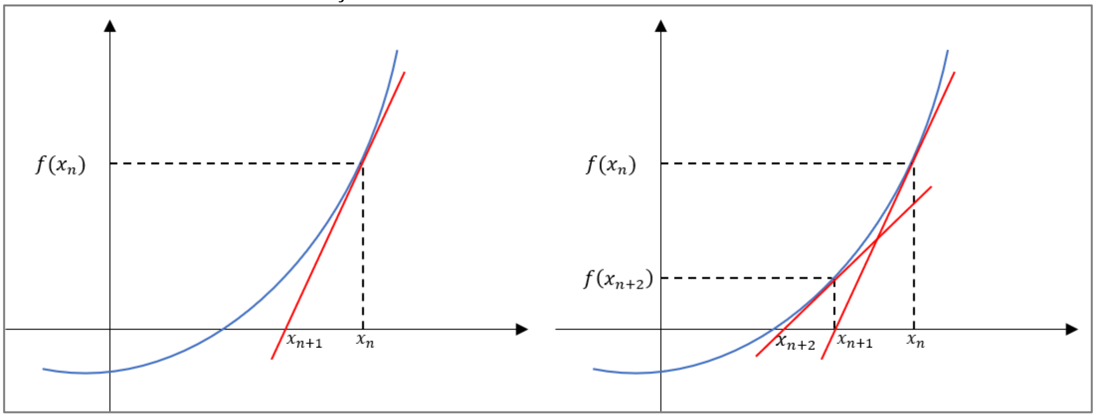
***Itera**ção*	$\boldsymbol{a}$	$\boldsymbol{b}$	$\boldsymbol{x_n}$	$\boldsymbol{f(a)}$	$\boldsymbol{f(b)}$	$\boldsymbol{f(x_n)}$	***Erro***
1	−2	−1	−1,5	8,027209981	−1,637189707	2,685519968	−
2	−1,5	−1	−1,25	2,685519968	−1,637189707	0,168611424	0,25
3	−1,25	−1	−1,125	0,168611424	−1,637189707	−0,846667788	0,125
4	−1,25	−1,125	−1,1875	0,168611424	−0,846667788	−0,364583878	0,0625
5	−1,25	−1,1875	−1,21875	0,168611424	−0,364583878	−0,104018507	0,03125
6	−1,25	−1,21875	−1,234375	0,168611424	−0,104018507	0,030835207	0,015625
7	−1,234375	−1,21875	−1,2265625	0,030835207	−0,104018507	−0,036962911	0,0078125
8	−1,234375	−1,2265625	−1,23046875	0,030835207	−0,036962911	−0,00315593	0,00390625
9	−1,234375	−1,23046875	−1,232421875	0,030835207	−0,00315593	0,013816712	0,001953125
10	−1,232421875	−1,23046875	−1,231445313	0,013816712	−0,00315593	0,005324648	0,000976563

FONTE: Autoria própria (2018).

1.2.2 Método de Newton-Raphson

O método de Newton-Raphson tem como objetivo encontrar uma aproximação que seja a raiz de uma equação através de uma aproximação inicial, a partir dela é calculada a reta tangente que passa pelo ponto dado pela aproximação inicial. Esse método busca, por meio dos pontos de interseção da reta tangente da função e o eixo das abscissas, estabelecer uma sequência de valores convergentes para a solução aproximada da raiz.

FIGURA 1.9 – INTERPRETAÇÂO GEOMÉTRICA DO MÉTODO DE NEWTON – RAPHSON.



FONTE: Autoria própria (2018).

Através da análise dos gráficos da Figura 1.9, verifica-se que a aproximação x_n é o ponto em que a reta tangente intercepta o eixo das abscissas. Ao deduzir a fórmula de Newton, fica mais claro o comportamento do método, para isso deve-se tomar como aproximação inicial x_n, e a partir do ponto $\left(x_n, f(x_n)\right)$ traçar uma reta tangente que cruze o eixo das abscissas no ponto $(x_{n+1}, 0)$, representando a primeira aproximação da raiz. A partir da equação geral da reta, segue que:

$$(f(x_n) - f(x_{n+1})) = a(x_n - x_{n+1}),$$

onde a é a inclinação da reta tangente no ponto $\left(x_n, f(x_n)\right)$.

Sabendo que a é o coeficiente angular da reta tangente, e que este é a derivada da função f no ponto $\left(x_n, f(x_n)\right)$, conclui-se que:

$$(f(x_n) - 0) = f'(x_n)(x_n - x_{n+1})$$

$$\frac{f(x_n)}{f'(x_n)} = (x_n - x_{n+1})$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

conhecida como fórmula de Newton-Rapshon.

Para a escolha de x_n dependerá da função que será calcula a aproximação da raiz. De acordo com Chapra (2011, p. 124):

> não existe nenhum critério de convergência geral para o Newton-Raphson. Sua convergência depende da natureza da função e da precisão da aproximação inicial. O único remédio é ter uma aproximação inicial que esteja "suficientemente" próxima da raiz. E para algumas funções, nenhuma aproximação funcionará! Boas aproximações dependem em geral do conhecimento das condições do problema físico ou dos artifícios tais como os gráficos que fornecem informação sobre o comportamento da solução.

Exemplo 1.3: Calcular as raízes da função $f(x) = 2x - sen(x) + 4$, considerando o erro absoluto $\varepsilon \leq 10^{-3}$.

Calculando a derivada de primeira ordem, obtém-se:

$$f'(x) = 2 - \cos(x)$$

Para a primeira iteração, de acordo com o Teorema do Anulamento de Bolzano, há uma raiz no intervalo $[-3, -2]$, considera-se $x_1 = -2$, por ser um ponto próximo a raiz. Assim,

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = -2,37634$$

Erro: $|x_2 - x_1| = |-2,37634 - (-2)| = 0,37634$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = -2,35430$$

Erro: $|x_3 - x_2| = |2,35430 - (-2,37634)| = 0,02204$

$$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)} = \ -2,35424$$

Erro: $|x_4 - x_3| = |-2,35424 - (-2,35424)| = 0,00001$

Portanto, utilizando o método de Newton-Raphson, o valor aproximado para a raiz da função $f(x) = 2x - sen(x) + 4$ com aproximação inicial de $x_1 = -2$ é $-2{,}3542$.

A Tabela 1.3 apresenta os resultados de todas iterações:

TABELA 1.3: RESULTADOS DE $f(x) = 2x - sen(x) + 4$ PELO MÉTODO DE NEWTON-RAPHSON.

Iterações	x_n	$f(x_n)$	$f'(x_n)$	***Erro***
1	-2	0,909297427	2,416146837	-
2	$-2{,}376341956$	$-0{,}059966087$	2,721208718	0,376341956
3	$-2{,}354305393$	$-0{,}000169475$	2,705769727	0,022036563
4	$-2{,}354242759$	$-1{,}38967E-09$	2,705725353	$6{,}26346E-05$

FONTE: Autoria própria (2018).

1.3 Comparação entre os métodos

As aproximações determinadas pelos métodos da Bisseção e de Newton-Raphson, assim como a quantidade de iterações necessárias em cada método, considerando uma mesma precisão, ε, podem apresentar diferenças. Conforme pode ser observado na Tabela 1.4, a função $f(x) = x^2 + 4\,sen(2x) + 1$ pelo Método da Bisseção foram efeituadas 10 iterações e para o Método de Newton-Raphson 8 iterações, para a função $f(x) = 2x - sen(x) + 4$ a quantidade de iterações foi a mesma para o Método da Bisseção, e ao utilizar o Método de Newton-Raphson para aproximar a raiz foram necessárias apenas 6 iterações.

TABELA 1.4: COMPARAÇÂO ENTRE OS MÉTODOS.

	$f(x) = x^2 + 4sen(2x) + 1$		$f(x) = 2x - sen(x) + 4$	
	Método da Bisseção	*Método de Newton-Rapshon*	*Método da Bisseção*	*Método de Newton-Rapshon*
Iterações	10	8	10	6
Aproximação	$-1{,}2309$	$-1{,}2308$	$-2{,}3540$	$-2{,}3542$

FONTE: Autoria própria (2018).

O método da Bisseção é realizado através de cálculos mais simples, com o intuito de apenas diminuir o intervalo que contém a raiz, e normalmente apresenta um número maior de iterações. Já o método de Newton-Raphson requer o conhecimento de técnicas de derivação, porém possui uma convergência mais rápida (BARROSO, 1987, p. 140).

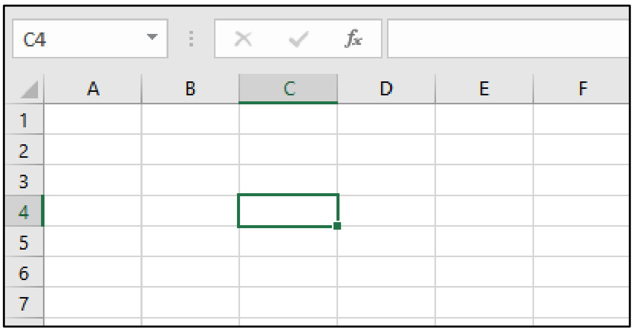
Conforme Ruggiero (1996, p. 78) "a escolha do método está diretamente relacionada com a equação que se quer resolver, no que diz respeito ao comportamento da função na região da raiz exata, às dificuldades com o cálculo de $f'(x)$". Assim, deve-se considerar as características de cada método, é notável que a escolha para a utilização depende das circunstâncias encontradas em cada problema, bem como a preferência de cada indivíduo.

2 RESOLUÇÃO COM O AUXÍLIO DO EXCEL.

O Microsoft Office Excel, ou simplesmente Excel, é um *software* da Microsoft que através de planilhas possibilita o controle de dados, realização de cálculos e a construção de gráficos. Com planilhas organizadas em formato matricial, ele possui células organizadas em linhas e colunas, onde é permitido ao usuário inserir textos e números (ANDRADE, 2011, p. 21).

As colunas e linhas do *software* são representadas por letras e números, respectivamente, permitindo que cada célula tenha um endereço específico, representado pelo o nome da coluna seguido do número da linha. Na Figura 2.1 está representada a célula de endereço C4.

FIGURA 2.1 - TABELA COM A CÉLULA C4.



FONTE: Autoria própria (2018).

Cada conteúdo inserido nas células pode ser manipulado através de fórmulas. Segundo Andrade (2011, p. 145) “As fórmulas são equações que executam cálculos sobre valores na planilha. A fórmula deve possuir uma lógica matemática que deve ser seguida para resultar os valores corretamente.”.

Por ser um *software* de fácil acesso, as Tabelas 1.2 e 1.3 (exemplos 1.2 e 1.3) dos métodos numéricos citados no Capítulo 1 foram construídas primeiramente por meio de planilhas no Excel.

Neste Capítulo é descrito como construir tabelas que auxiliam na resolução das equações utilizando os métodos da Bisseção e Newton-Raphson, e os procedimentos para os cálculos das iterações.

2.1 Método da Bisseção

Como visto no Capítulo 1, o método da bisseção consiste em dividir os intervalos ao meio até chegar à aproximação desejada. Ao iniciar o programa, deve-se inserir os dados que cada coluna irá receber, e as linhas as iterações que serão realizadas, seguindo o exemplo da Figura 2.2.

FIGURA 2.2: TABELA PARA O MÉTODO DA BISSEÇÃO.

	A	B	C	D	E	F	G	H
1	Método da Bisseção							
2	**Iterações**	**a**	**b**	**x_n**	**f(a)**	**f(b)**	**f(x)**	**erro**
3	1							
4	2							
5	3							
6	...							
7	n							

FONTE: Autoria própria (2018).

Primeira Iteração:

1° passo: Os Primeiros valores de a e b serão inseridos na tabela, que correspondem respectivamente as células B3 e C3;

2° passo: Para os valores de x_n, temos que somar a e b e em seguida dividir por 2, ou seja, $x_n = \frac{a+b}{2}$, assim para a célula D3, é inserida a seguinte fórmula:

=(B3+C3)/2,

considerando o fato que o *software* recebe apenas os valores informados em cada célula, o que facilita no cálculo das novas células.

3° passo: Na célula correspondente ao primeiro $f(a)$, inserimos a função substituindo o x pela célula correspondente ao valor de a, ou seja, B3. O mesmo vale para $f(b)$ e $f(x)$, porém substituindo o x por C3 e D3 respectivamente.

Na primeira linha o erro não é calculado.

Segunda Iteração:

Na segunda linha que representa a segunda iteração do método da bisseção (linha 4 no Excel) são inseridos comandos nas células que correspondem aos valores de a e b.

1° passo: Atribuir comandos a B4.

Sabemos que se o produto de $f(a) \cdot f(x_1) < 0$ o valor de a se mantém o mesmo, e que se $f(a) \cdot f(x_1) > 0$, o valor de a será substituído pelo o valor de x_1.

Usando essa lógica e a função "=SE()" do Excel, vamos montar o seguinte comando em B4:

=SE(E3*G3<0;B3;D3).

Que equivale dizer ao *software*: se o produto de E3 e G3 for menor que zero, usar o valor de B3, se não, usar o valor de D3.

2° passo: Atribuir comandos a C4.

Usando a mesma lógica do anterior. Sabemos que se o produto de $f(b) \cdot f(x_1) < 0$ o valor de b permanece o mesmo, e se $f(b) \cdot f(x_1) > 0$, o valor de b será substituído pelo o valor de x_1.

A função "=SE()" na célula C4 terá o seguinte comando:

=SE(F3*G3<0;C3;D3).

Significado: se o produto de F3 e G3 for menor que zero, usar o valor de C3, se não, usar o valor de D3.

3° passo: Cálculo do erro absoluto.

A partir da segunda iteração o erro absoluto é calculado, assim é verificado se este é menor que a precisão desejada, trabalhando o erro absoluto (módulo da iteração atual menos a anterior), isto é,

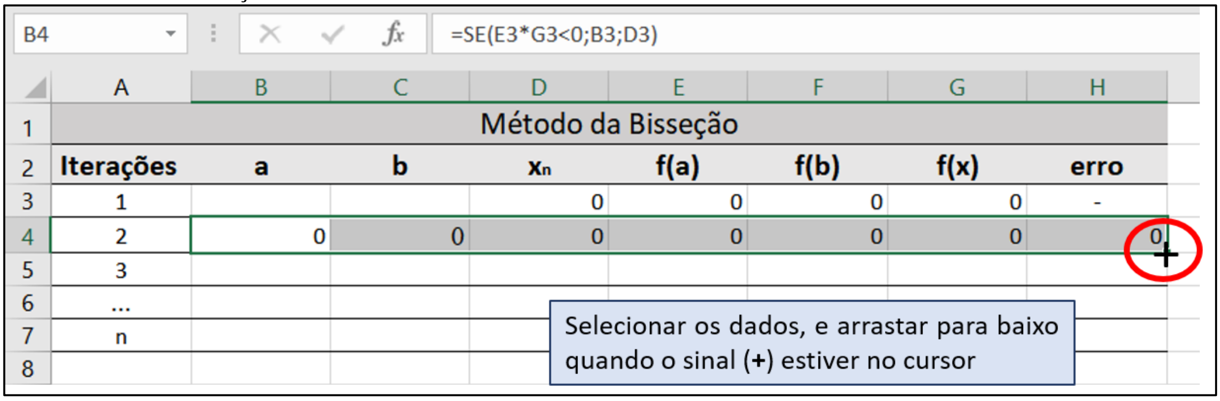
$$|x_n - x_{n-1}| \leq \varepsilon.$$

Como o erro é calculado pelo seu valor absoluto, é utilizado o comando "=ABS()". A célula H4 ficará da seguinte maneira:

=ABS(D4 - D3).

Nas células D4, E4, F4 e G4, os comandos utilizados nas células D3, E3, F3 e G3, irão se repetir, mudando apenas o número correspondente a linha. Ao final da segunda linha de dados da tabela, basta selecionar toda a linha, e arrastar para que os comandos sejam inseridos nas próximas linhas. As Figuras 2.3 e 2.4 exibem a planilha do Excel a primeira com o preenchimento apenas da linha 4, ou a segunda iteração, a segunda Figura exibe o preenchimento de toda a tabela.

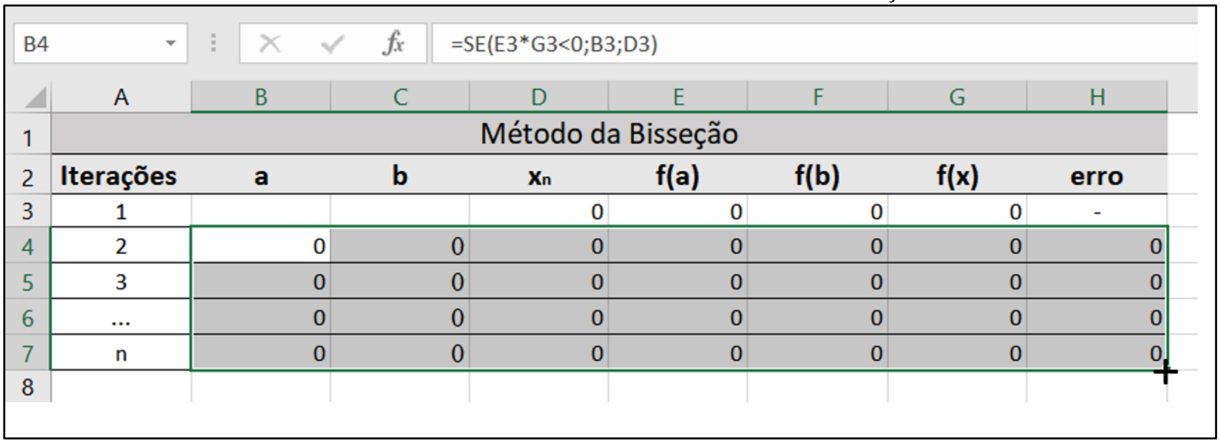
FIGURA 2.3 - DADOS SELECIONADOS NA SEGUNDA LINHA DE ITERAÇÃO MÉTODO DA BISSEÇÃO.


B4 =SE(E3*G3<0;B3;D3)

	A	B	C	D	E	F	G	H
1	Método da Bisseção							
2	Iterações	a	b	x_n	f(a)	f(b)	f(x)	erro
3	1			0	0	0	0	-
4	2	0	0	0	0	0	0	0
5	3							
6	...							
7	n							
8								

FONTE: Autoria própria (2018).

FIGURA 2.4 - ARRASTANDO/COPIANDO OS DADOS: MÉTODO DA BISSEÇÃO.


B4 =SE(E3*G3<0;B3;D3)

	A	B	C	D	E	F	G	H
1	Método da Bisseção							
2	Iterações	a	b	x_n	f(a)	f(b)	f(x)	erro
3	1			0	0	0	0	-
4	2	0	0	0	0	0	0	0
5	3	0	0	0	0	0	0	0
6	...	0	0	0	0	0	0	0
7	n	0	0	0	0	0	0	0
8								

FONTE: Autoria própria (2018).

Exemplo2.1: Calcular uma raiz da função $f(x) = 2x - \cos(x) + 4$ considerando o intervalo $[-3, -2]$ com um erro menor que 10^{-3} no Excel, por meio do método da bisseção.

Trabalhando com o intervalo inicial $[-3, -2]$, digita-se nas células B3 e C3 os valores -3 e -2. Nas células correspondentes ao cálculo da função em x, como por exemplo a célula E3 onde tem-se $f(a)$, a expressão a ser digitada é:

=2*B3 - COS(B3) + 4.

A Figura 2.5 apresenta um esquema resumido da construção das fórmulas no Excel e os resultados das iterações. Observando ainda a Figura 2.5 vemos que, na décima iteração o erro absoluto é menor que a precisão estabelecida e o valor aproximado para a raiz é -2,3525.

FIGURA 2.5 – RAIZ DA FUNÇÃO $f(x) = 2x - \cos(x) + 4$ NO EXCEL, PELO MÉTODO DA BISSEÇÃO.

=(B3+C3)/2

=2*B3-COS(B3)+4

	A	B	C	D	E	F	G	H
1	Método da Bisseção							
2	Iterações	a	b	x_n	f(a)	f(b)	f(x)	erro
3	1	-3	-2	-2,5	-1,0100075	0,41614684	-0,1988564	-
4	2	-2,5	-2	-2,25	-0,1988564	0,41614684	0,12817362	0,25
5	3	-2,5	-2,25	-2,375	-0,1988564	0,12817362	-0,0297215	0,125
6	4	-2,375	-2,25	-2,3125	-0,0297215	0,12817362	0,05054504	0,0625
7	5	-2,375	-2,3125	-2,34375	-0,0297215	0,05054504	0,01075267	0,03125
8	6	-2,375	-2,34375	-2,359375	-0,0297215	0,01075267	-0,0093978	0,015625
9	7	-2,359375	-2,34375	-2,3515625	-0,0093978	0,01075267	0,0006989	0,0078125
10	8	-2,359375	-2,351563	-2,3554688	-0,0093978	0,0006989	-0,0043441	0,00390625
11	9	-2,3554688	-2,351563	-2,3535156	-0,0043441	0,0006989	-0,0018212	0,00195313
12	10	-2,3535156	-2,351563	-2,3525391	-0,0018212	0,0006989	-0,0005608	0,00097656

=SE(E3*G3<0;B3;D3)

=ABS(D4-D3)

FONTE: Autoria própria (2018).

2.2 Método de Newton-Raphson

O Método de Newton-Raphson, como visto no Capítulo 1, consiste em escolher uma aproximação inicial e, através da reta tangente à função, verificar onde ela intercepta o eixo das abscissas e o ponto na função.

Para iniciar a tabela para o cálculo da raiz pelo método de Newton-Raphson, deve-se inserir os dados fixos que cada coluna irá receber, como os títulos de cada coluna e as iterações correspondentes a cada linha, seguindo o exemplo da Figura 2.6.

FIGURA 2.6 – TABELA MÉTODO DE NEWTON-RAPHSON.

	A	B	C	D	E
1	Método Newton-Raphson				
2	**Iterações**	**x_n**	**$f(x_n)$**	**$f'(x_n)$**	**erro**
3	1				
4	2				
5	3				
6	...				
7	n				

FONTE: Autoria própria (2018).

Primeira iteração:

1º Passo: Inserir o valor de x_n na célula B3;

2º Passo: Para calcular $f(x_n)$ basta inserir a função na célula C3 substituindo a variável por B3, que é o valor da aproximação inicial. O mesmo procedimento e realizado em $f'(x_n)$ na célula D3. O erro não é calculado na primeira linha.

Segunda iteração:

1º Passo: Na segunda linha da tabela (terceira no Excel), apenas o valor de x_n passará a ser o resultado obtido a partir da fórmula de Newton-Raphson, assim a célula B4 ficará com os seguintes dados:

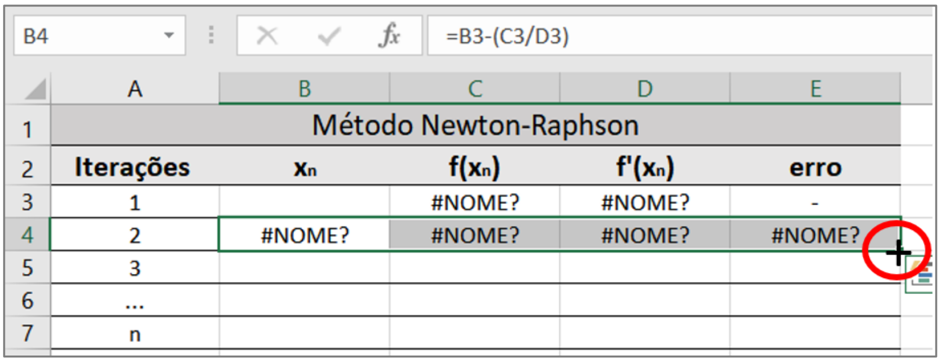
=B3 – (C3/D3).

2º Passo: O erro será calculado pelo módulo, assim a célula E4 terá o comando:

=ABS(B4 - B3).

As células C4 e D4 repetirão os comandos. A partir da segunda iteração os dados irão se repetir, então basta selecionar toda a linha 4 e arrastar para que os comandos sejam inseridos nas próximas linhas, da mesma forma que foi feito no método da bisseção na seção 2.1. A Figura 2.7 mostra a linha 4 selecionada, já a Figura 2.8 mostra a tabela completamente preenchida.

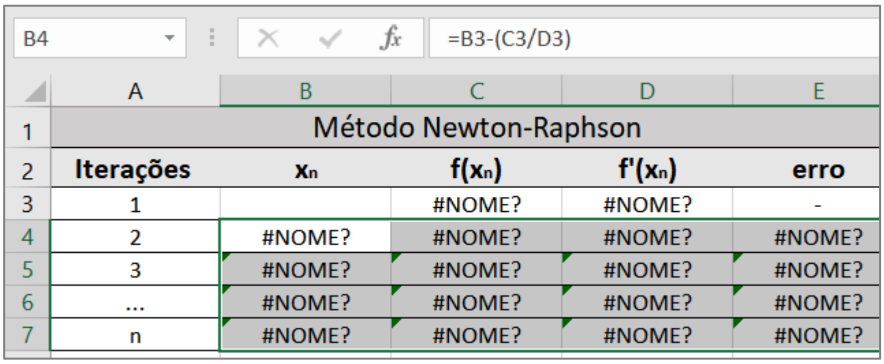
FIGURA 2.7 - DADOS SELECIONADOS NA SEGUNDA LINHA DE ITERAÇÃO MÉTODO DE NEWTON-RAPHSON.



	A	B	C	D	E
1	Método Newton-Raphson				
2	Iterações	x_n	$f(x_n)$	$f'(x_n)$	erro
3	1		#NOME?	#NOME?	-
4	2	#NOME?	#NOME?	#NOME?	#NOME?
5	3				
6	...				
7	n				

FONTE: Autoria própria (2018).

FIGURA 2.8 - ARRASTANDO/COPIANDO OS DADOS MÉTODO DE NEWTON-RAPHSON.



	A	B	C	D	E
1	Método Newton-Raphson				
2	Iterações	x_n	$f(x_n)$	$f'(x_n)$	erro
3	1		#NOME?	#NOME?	-
4	2	#NOME?	#NOME?	#NOME?	#NOME?
5	3	#NOME?	#NOME?	#NOME?	#NOME?
6	...	#NOME?	#NOME?	#NOME?	#NOME?
7	n	#NOME?	#NOME?	#NOME?	#NOME?

FONTE: Autoria própria (2018).

Exemplo 2.2: Calcular uma raiz da função $f(x) = 2x - \cos(x) + 4$ considerando a aproximação inicial $x_1 = -1$ com um erro menor que 10^{-3} no Excel, pelo método de Newton-Raphson.

Trabalhando com a aproximação inicial $x_1 = -1$, digitada na célula B3. Na célula C3, tem-se o seguinte comando, que corresponde ao cálculo da função em x_1:

=2*B3 - COS(B3) + 4.

Na célula D3 é calculada a derivada em relação a x, seu comando ficará da seguinte forma:

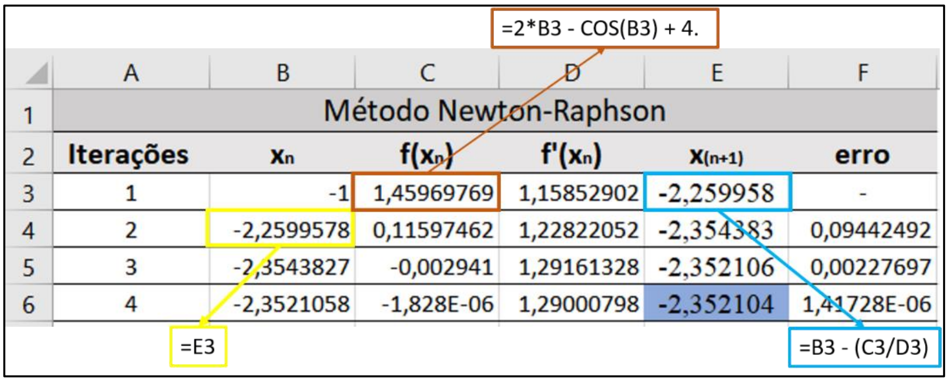
=SEN(B3) + 2.

Já na célula E3 é inserida a fórmula de Newton-Raphson e seu comando é dado por:

=B3 - (C3/D3).

A Figura 2.9 apresenta um esquema resumido da construção das fórmulas no Método de Newton-Raphson e os resultados das iterações. Observando ainda a Figura 2.9 se vê que, na quarta iteração o erro absoluto é menor que a precisão estabelecida e o valor aproximado para a raiz é -2,3521.

FIGURA 2.9 – RAIZ DA FUNÇÃO $f(x) = 2x - \cos(x) + 4$ NO EXCEL, PELO MÉTODO DE NEWTON RAPHSON.



	A	B	C	D	E	F
1	Método Newton-Raphson					
2	**Iterações**	**x_n**	**$f(x_n)$**	**$f'(x_n)$**	**$x_{(n+1)}$**	**erro**
3	1	-1	1,45969769	1,15852902	-2,259958	-
4	2	-2,2599578	0,11597462	1,22822052	-2,354383	0,09442492
5	3	-2,3543827	-0,002941	1,29161328	-2,352106	0,00227697
6	4	-2,3521058	-1,828E-06	1,29000798	-2,352104	1,41728E-06

FONTE: Autoria própria (2018).

A utilização do Microsoft Office Excel é uma alternativa para simplificar os cálculos, considerando o fato de que não há a necessidade de executar os cálculos manualmente ou conhecer alguma linguagem de programação. Porém requer a mudança nas células a cada nova função ou equação, o que pode ocasionar erros durante esse processo.

3 IMPLEMENTAÇÃO DOS MÉTODOS NO MATLAB®.

MATLAB® é um software desenvolvido pela MathWorks que permite a manipulação de matrizes, construção de gráficos e implementação de algoritmos. A sua utilização abrange áreas especificas de estudo como: Álgebra Linear, Analise Numérica, Equações Diferencias Ordinárias, Estatística e Engenharia. Segundo Gilat (2012, p.01):

> O nome MATLAB® vem da elisão das palavras MATrix LABoratory. Isto se deve à base operacional do software, que são as matrizes. O MATLAB® é bastante versátil em cálculos matemáticos, modelagens e simulações, análises numéricas e processamentos, visualização e gráficos, desenvolvimentos de algoritmos, etc.

No estudo do cálculo numérico o MATLAB® permite a implementação de diversos métodos através da construção de algoritmos que facilitem a realização de seus respectivos cálculos e aproximações.

Neste capítulo são descritas características importantes do *software* para a implementação dos métodos estudados anteriormente. Da mesma forma, são apresentados os algoritmos utilizados nas implementações dos métodos.

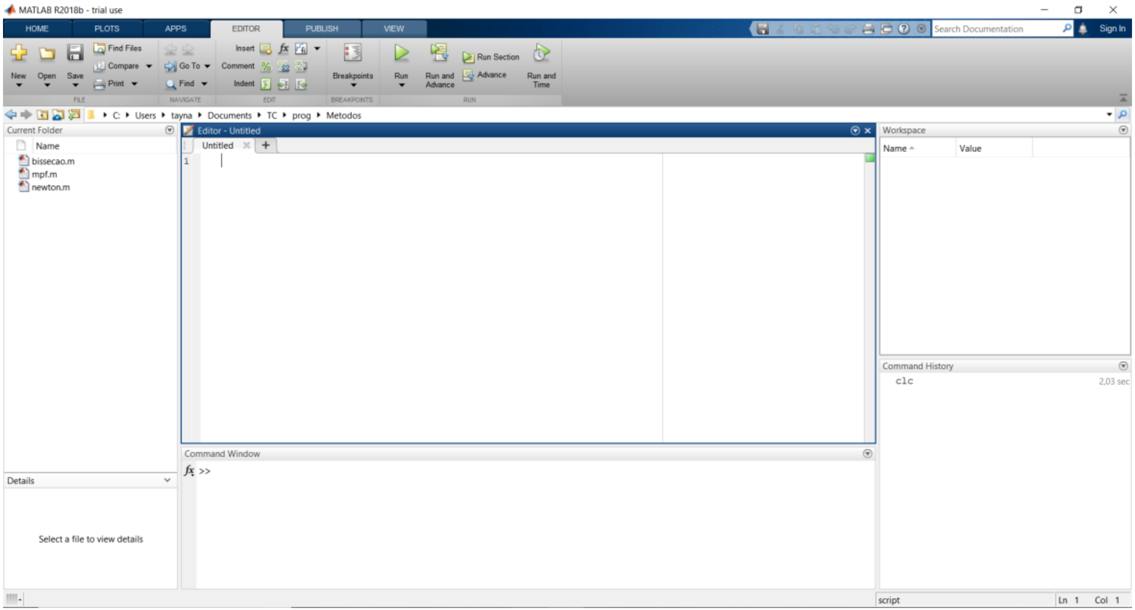
A versão do MATLAB® utilizada na realização do presente trabalho foi a versão R2018b, *software* de avaliação (*Trial Software*) disponível no site oficial da MathWorks Inc. (*www.mathworks.com*) havendo a necessidade de realizar um cadastro para adquirir a licença gratuita de 30 dias para utilização.

3.1 O MATLAB®

O *software* MATLAB® pode ser operado inserindo cada comando por vez, ou editando arquivos na janela de edição. Os arquivos-M (M-files) construídos através da janela de edição, são programas no MATLAB® com uma série de instruções que são executadas de uma só vez. Estes arquivos são salvos com extensão .m e podem ser do tipo *scripts* ou funções. Assim, pode-se utilizar do software através da janela de comando (*command windown*) onde os dados são inseridos um de cada vez, ou pela janela de edição (*edit windown*) utilizada para criação de arquivos-M. A implementação dos métodos numéricos foi realizada através da janela de edição.

Ao iniciar o software, a primeira visão que se tem são as janelas: *Command Window, Current, Folder* e *Workspace* como é mostrado na Figura 3.1. Sua tela inicial é dividida em blocos que permitem a visualização simultânea do que está sendo executado. Através do *workspace* é possível visualizar as variáveis e resultados que foram obtidos na Janela de comando (*Command Window*).

FIGURA 3.1 – TELA DE INÍCIO DO SOFTWARE MATLAB®.



FONTE: Autoria própria (2018).

Na Tabela 3.1 estão listadas as janelas citadas e suas respectivas funções.

TABELA 3.1 – FUNÇÕES DAS JANELAS DO MATLAB®.

Janela	Propósito
Command Window	Janela principal, inicialização de variáveis e execução de programas.
Figure Window	Apresenta o(s) resultado(s) dos comandos gráficos.
Edit Window	Permite a edição e a depuração de programas (script files) e funções.
Help Window	Ajuda na utilização do programa.
Command History Window	Apresenta o histórico dos comandos mais recentes digitados na janela Command Window.
Workspace Window	Disponibiliza informação sobre as variáveis que estão em uso.
Current Folder Window	Exibe os arquivos presentes no diretório ou pasta atual.

FONTE: GILAT (2012).

Há alguns comandos simples que devem ser observados durante a utilização da janela de comando, principal janela do MATLAB®, utilizada para executar comandos de maneira mais direta e rápida, e na janela de edição, que permite a edição dos algoritmos. Como os comandos *clc* e *clear*, que limpa a janela de comando e a memória respectivamente.

O *software* também possui um número extenso de funções matemáticas nativas, as quais são definidas pelo nome acompanhadas do argumento entre parênteses (GILAT, 2012, p. 14). Presentes em boa parte dos problemas matemáticos, as funções seno, cosseno e tangente são um exemplo delas. Abaixo são listadas algumas das funções matemáticas disponíveis no MATLAB®.

Funções matemáticas elementares:

- `sqrt(x)` – raiz quadrada
- `exp(x)` – e^x
- `abs(x)` – valor absoluto
- `log(x)` – logaritmo natural (base e)

Funções Trigonométricas:

- `sin(x)` – seno (x em radianos)
- `cos(x)` – cosseno
- `tan(x)` – tangente

Existem casos onde se tem que declarar variáveis para serem utilizadas em funções e que, ao se executar o algoritmo são substituídas. Para isso tem-se o comando *syms*, que ao ser declarado é acompanhado da variável, em sua maioria no início do algoritmo.

Há outras inúmeras funcionalidades e comandos presentes no MATLAB® que podem ser encontradas em livros, como do Amos Gilat de 2012 (MATLAB® com aplicações em engenharia, 4ºedição) usado como referência para o presente trabalho. Outra opção é a utilização do comando *Help*, que encaminha para o site da MathWords, com dicas para a sua utilização, porém esse site não é traduzido para o Português.

3.1.1 Programação no MATLAB®.

A janela de edição (*Edit Window*) permite construir programas (arquivos-M) com determinadas regras que possibilitam a implementação dos métodos numéricos. De acordo com Chapra (2013, p. 49) "um programa de MATLAB® consiste em uma série de instruções ou comandos que podem ser executados todos de uma vez.". No MATLAB® há dois tipos de programas: os programas comuns (*Scripts files*) e as funções (*function files*), ambos são salvos com a extensão .m, através dela é possível editar o arquivo caso seja necessário.

Os programas que constituem os métodos numéricos são do tipo *script file*. Para Matsumoto (2013, p. 69) esses arquivos reúnem um conjunto de comandos que tornam automático o processo de repetição. Ao implementar um método, eles recebem os argumentos de entrada necessários para a execução, como os intervalos e erros, e retornam uma saída, a raiz. Esses resultados são executados e exibidos na *command window*.

Cada programa é executado, linha por linha, e ao final ele retorna o resultado, que nos métodos numéricos nem sempre é o que desejamos. Ao resolver problemas numericamente há a necessidade da repetição para uma aproximação mais precisa, assim seria inevitável uma nova execução do programa.

Ao desenvolver um programa, há a necessidade de incluir estruturas que facilitem a escolha de qual caminho seguir, tais declarações são classificadas como Decisões (seleção) e Laços (repetição). A primeira permite a divisão do código com base em uma única sentença, a segunda possibilita que operações possam ser executadas mais de uma vez. (CHAPRA, 2011, p. 57).

As decisões possibilitam a tomada de medidas de acordo com uma sentença, julgando-a verdadeira ou falsa. Esta condição é representada pelo comando *if*, e podem ser usadas das seguintes maneiras: *if* - *end*, *if* – *else* - *end* e *if* – *elseif* – *else* - *end*. Para a implementação dos métodos foram utilizadas as *if* - *end* e *if* - *else* - *end*.

A condição expressa por *if* - *end* julga a sentença de forma que, caso ela seja verdadeira executa as próximas linhas, do contrário, ele pula a sequência de comandos e executa as linhas após o *end*, se encaminhado diretamente para os comandos após a decisão.

```
if      Condição
        Comandos
end
```

Entretanto a *If* – *else* - *end*: O programa executa a estrutura com a possibilidade de escolha entre dois caminhos, dependentes do resultado da sentença, se verdadeiro, tem-se a opção de seguir para um grupo de comando, se falso, há um segundo grupo de comandos diferentes. Assim, caso a sentença seja falsa, o programa não é direcionado diretamente para o fim, mas sim para um outro grupo de comandos.

if *Condição*

Comandos

else

Comandos

end

Os laços são semelhantes as estruturas de decisões, porém ao excuta-los é permitido que os comandos sejam executados mais de uma vez, até a sua interrupção “Cada sequência de execução (repetição) do laço é denominada passo. A cada passo, ao menos uma variável é modificada dentro do laço.” (GILAT, 2012, p. 190). O caminho que cada programa irá seguir é definido com o auxílio de operadores lógicos e relacionais.

For - end: neste laço o comando é repetido por um número de vezes previamente determinado. *While - end*: os comandos são repetidos um número de vezes indefinido, ou seja, enquanto a expressão é verdadeira ele executa os comandos. Na Tabela 3.2 é exemplificado a estrutura de cada laço.

TABELA 3.2 – LAÇOS.

Sintaxe típica	Descrição
while <Expr.Log> <Codigo> **end**	Executa o trecho de código <Codigo> enquanto a expressão lógica <Expr.Log> for verdadeira.
for V=<Inic>:<Incr>:<Fim> <Codigo> **end**	Para cada valor da variável V (de <Inic> até <Fim>, com incremento <Incr>), executa o trecho de código <Codigo>.

FONTE: MATSUMOTO (2013).

3.2 Implementação em MATLAB® para determinação de zeros reais de funções.

Nesta seção, são apresentadas as implementações dos métodos da Bisseção e Newton-Raphson com seus respectivos códigos. O método da Bisseção possui dois programas diferentes, um mais simples, com uma quantidade menor de linhas, mas que apresenta apenas o resultado final e um segundo código que requer a utilização de mais linhas, porém exibe uma tabela com os resultados mais detalhados das iterações. O método de Newton possui apenas um programa que exibe os dados em tabela da mesma forma que o segundo programa para o método da Bisseção.

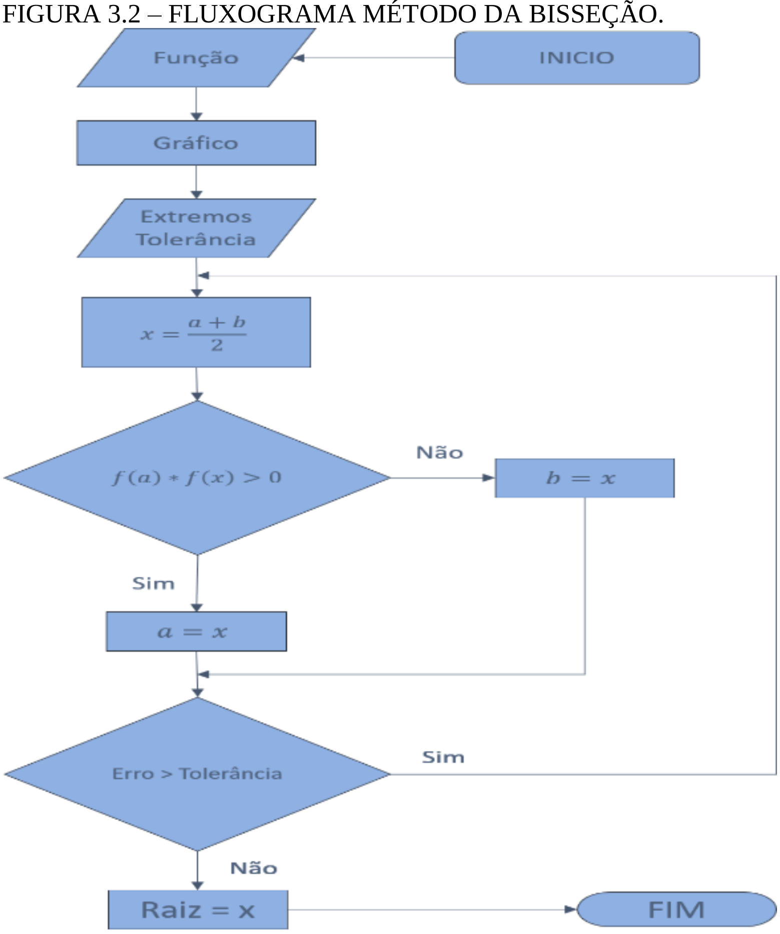
Para ambos os métodos, o processo de refinamento e escolha do intervalo em que a raiz está contida foi feito através de análise gráfica – os gráficos já foram programados para serem exibidos durante a execução dos programas.

3.2.1 Método da Bisseção

Como visto no primeiro capítulo, o Método da Bisseção consiste em reduzir a amplitude do intervalo que contém a raiz até que a sua dimensão seja menor que a precisão desejada. Assim, são realizadas divisões sucessivas do intervalo.

A Figura 3.2 representa o fluxograma do método da Bisseção. Neste algoritmo são inseridos a função, os extremos do intervalo $[a, b]$ onde a raiz está localizada e a precisão para a realização dos cálculos. Após realizar todos as etapas do processo, encontra-se a raiz da equação.

FIGURA 3.2 – FLUXOGRAMA MÉTODO DA BISSEÇÃO.



FONTE: Autoria Própria (2018).

Para a implementação foram construídos dois programas: o primeiro programa (salvo com o nome MetodoBissecao.m) é mais simples e gera o resultado desejado; o segundo programa (salvo com o nome MetodoBissecao2.m) organiza os dados de todas as iterações em tabelas exigindo uma quantidade maior de códigos que o primeiro, além do erro ser calculado pela amplitude do intervalo, ou seja, $erro < abs(b - a)$.

Ambos os códigos estão comentados, para a explicação de cada linha, os comentários são as sentenças após o símbolo de porcentagem (%).

Programa 1: Salvo com o nome MetodoBissecao.m

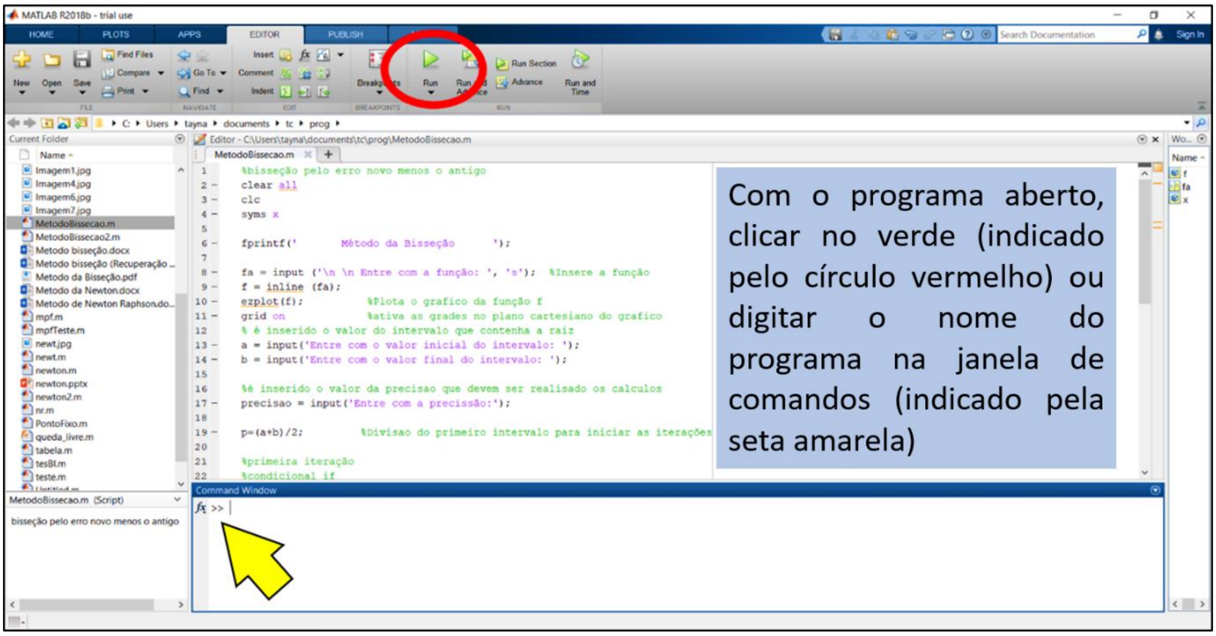
```
%Bisseção (iteração nova menos a antiga)
clear all
clc
syms x
fprintf('       Método da Bisseção      ');
fa = input ('\n \n Entre com a função: ', 's');  %Insere a função
f = inline (fa);
ezplot(f);            %Plota o gráfico da função f
grid on               %ativa as grades no plano cartesiano do gráfico
% é inserido o valor do intervalo que contenha a raiz
a = input('Entre com o valor inicial do intervalo: ');
b = input('Entre com o valor final do intervalo: ');
%é inserido o valor da precisão que devem ser realizado os cálculos
precisao = input('Entre com a precisão:');
p=(a+b)/2;           %Divisão do primeiro intervalo para iniciar as iterações
%primeira iteração
%condicional if
if f(a)*f(b)<0
        a=p;
    else
        b=p;     %caso contrário f(a)*f(x)<0 substitui-se por p
end     %fim da condicional if
 x=(a+b)/2;           %divisão do novo intervalo
 %calculo do erro (valor atual(novo) menos o valor antigo de x
 erro=abs(x-p);
 %Laço while
 while abs(x-p)>precisao    %se o valor do erro calculado for maior
                               que a precisão
         p=x;                 %faz o valor de p igual  a x para a    substituição
         na função

     if f(a)*f(x)>0   %se f(a)*f(x)>0 substitui a por x
        a=x;
     else
        b=x;          %caso contrario f(a)*f(x)<0 substitui-se b por p
     end              %fim da condicional if

     x=(a+b)/2;       %divisão do novo intervalo

     erro = abs(x-p);     %calculo do novo erro
end    %fim do laço while

%Imprimir o valor da Raiz
fprintf('O valor de x é %f \n', x);
```

Programa 2: Salvo com o nome MetodoBissecao2.m

```
%Bisseção pela amplitude do intervalo
clear all
clc

fprintf('%s\n','       Método da Bisseção      ');


fb = input ('Entre com a função:', 's'); %Insere a função
f = inline (fb);
ezplot(f);           %Plota o grafico da função f
grid on              %ativa as grades no plano cartesiano do grafico


% é inserido o valor do intervalo que contenha a raiz
a=input('Entre com o valor inicial do intervalo:');
b=input('Entre com o valor final do intervalo:');

%é inserido o valor da precisao que devem ser realisado os calculos
precisao = input('Entre com a precissão:');

k=0;

%Construção da TAbela
fprintf('%s \n',' ');
fprintf('%s \n','-------------------------------------------------');
fprintf('%s \n',' iteração     a       b      x       f(x)      c');
fprintf('%s \n','-------------------------------------------------');

x=(a+b)/2; %Divisao do primeiro intervalo para iniciar as iterações

while abs(b-a)>precisao     %while, comando utilizado para a execução de
uma
                  %serie de comandos
%Calcula o erro(pela amplitude do intervalo),
%caso o valor absoluto (abs) do intervalo seja
                  %maior que a precisão,
%calcula-se o proximo passo

if f(a)*f(x)>0  %if = condicional
%o comando subs serve para substituir um valor
%na função
a=x;       %se f(a)*f(x)>0 substitui a por x

else            %else = condicional
b=x;       %caso contrario f(a)*f(x)<0 substitui-se b por x

end             %fim da condicional if
x=(a+b)/2;  %é realizado a divisão do novo intervalo

erro=abs(b-a);  % e calculado o erro no novo intervalo
k = k+1;

%Resultados Tabelados
fprintf(' %2d  %10.9f  %10.9f  %10.9f  %11.8f  %10.9f\n' ,k, a, b, x, f(x),
erro);

end         %fim do comando while

fprintf('\nRaiz aproximada => n(%d) = %10.8f\n', k, x);
```

Para iniciar um programa, o mesmo deve estar aberto no MATLAB® e ao clicar na seta verde, indicada pelo círculo vermelho na Figura 3.3, ele irá começar exibindo os dados na janela de comando. Também pode ser iniciado na janela de comando, sempre que houver duas setas (>>), na janela de comandos (*Command Windown*).

FIGURA 3.3 – COMO INICIAR UM PROGRAMA.

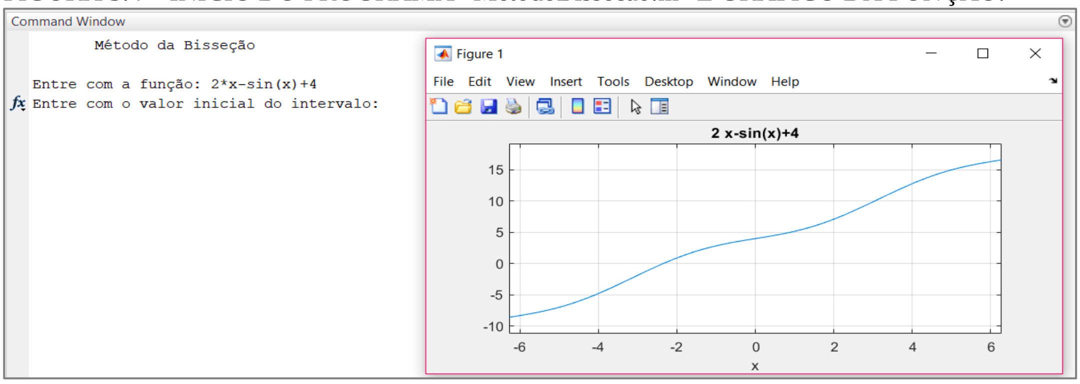


FONTE: Autoria Própria (2018)

Exemplo 3.1: Cálculo da equação da função $f(x) = 2x - \text{sen}(x) + 4$ no MATLAB®, considerando o erro de 10^{-3}.

Abrir o arquivo MetodoBissecao.m, clicar em *Run/* e fornecer os dados conforme as instruções que são dadas na janela de comando (*command window*). Primeiramente será solicitada a função (que deve ser digitada em função da variável x), como ilustrado na Figura 3.4.

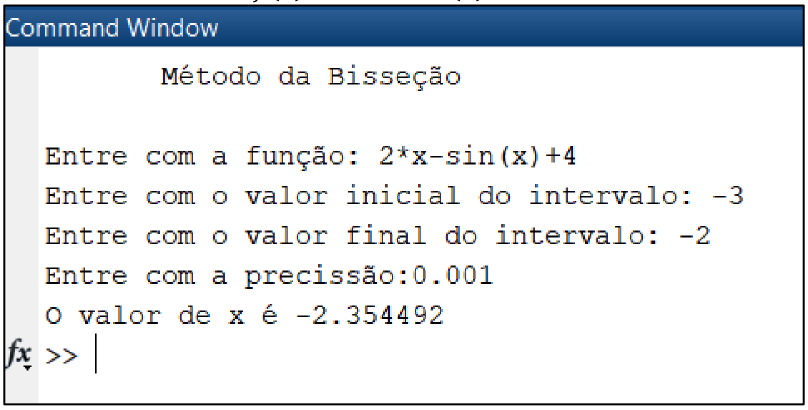
FIGURA 3.4 – INÍCIO DO PROGRAMA “MetodoBissecao.m” E GRÁFICO DA FUNÇÃO.



FONTE: Autoria Própria (2018)

Observando o gráfico gerado e exibido na janela *Figure* 1, então se preenche os próximos dados que se referem ao intervalo que contém a raiz $[-3, -2]$.

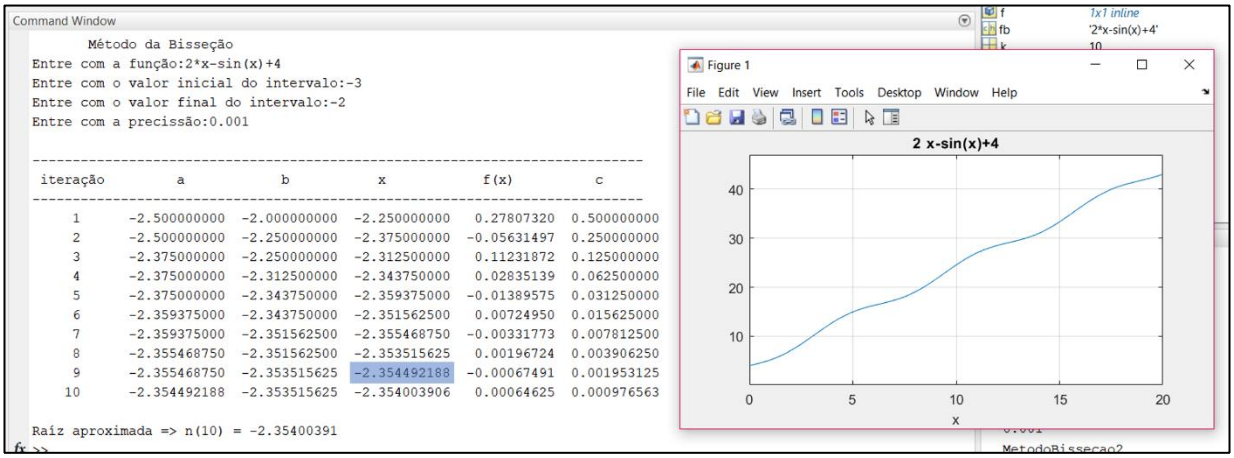
FIGURA 3.5 – RESULTADO DO PROGRAMA “MetodoBissecao.m” PARA $f(x) = 2x - sen\ (x) + 4$.



```
Command Window
        Método da Bisseção

Entre com a função: 2*x-sin(x)+4
Entre com o valor inicial do intervalo: -3
Entre com o valor final do intervalo: -2
Entre com a precissão:0.001
O valor de x é -2.354492
>>
```

FONTE: Autoria Própria (2018).

FIGURA 3.6 – RESULTADO DO PROGRAMA “MetodoBissecao2.m”: TABELA COM AS ITERAÇÕES, RAIZ E GRÁFICO.



```
Command Window
        Método da Bisseção
Entre com a função:2*x-sin(x)+4
Entre com o valor inicial do intervalo:-3
Entre com o valor final do intervalo:-2
Entre com a precissão:0.001
```

iteração	a	b	x	f(x)	c
1	-2.500000000	-2.000000000	-2.250000000	0.27807320	0.500000000
2	-2.500000000	-2.250000000	-2.375000000	-0.05631497	0.250000000
3	-2.375000000	-2.250000000	-2.312500000	0.11231872	0.125000000
4	-2.375000000	-2.312500000	-2.343750000	0.02835139	0.062500000
5	-2.375000000	-2.343750000	-2.359375000	-0.01389575	0.031250000
6	-2.359375000	-2.343750000	-2.351562500	0.00724950	0.015625000
7	-2.359375000	-2.351562500	-2.355468750	-0.00331773	0.007812500
8	-2.355468750	-2.351562500	-2.353515625	0.00196724	0.003906250
9	-2.355468750	-2.353515625	-2.354492188	-0.00067491	0.001953125
10	-2.354492188	-2.353515625	-2.354003906	0.00064625	0.000976563

```
Raíz aproximada => n(10) = -2.35400391
```


FONTE: Autoria Própria (2018).

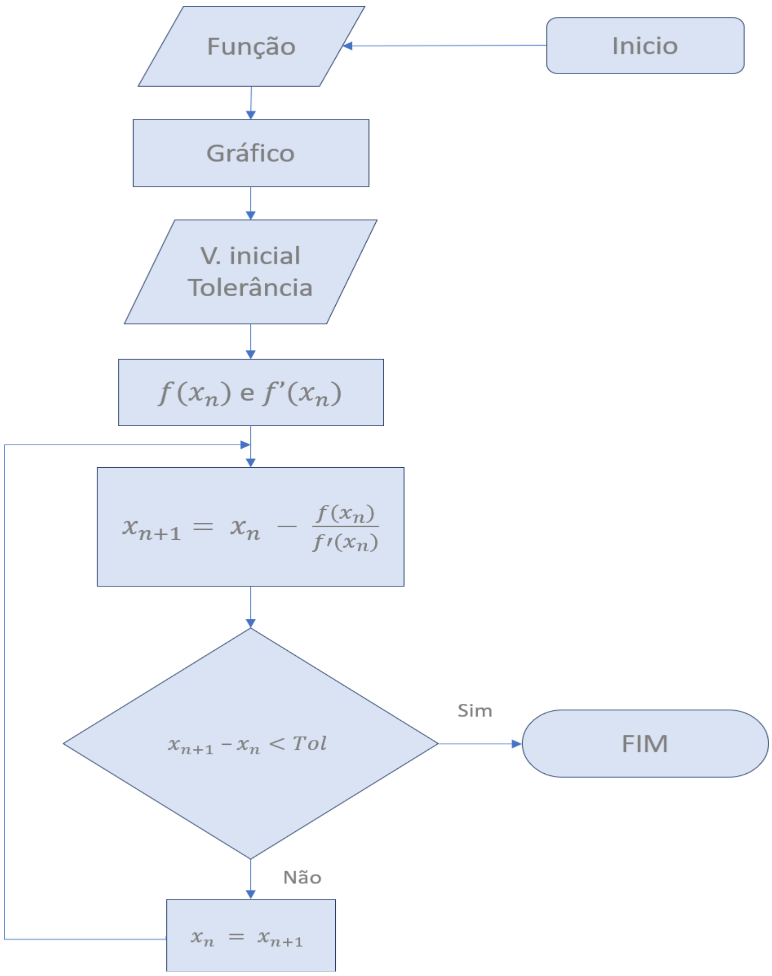
Observando as Figuras 3.5 e 3.6, nota-se que a Figura 3.5 resultou em uma raiz a aproximadamente $-2{,}3544$, enquanto a Figura 3.6 forneceu um resultado igual a $-2{,}3540$. Tal diferença deve-se ao fato de que no programa MetodoBissecao.m utiliza-se $|x_n - x_{n-1}|$ para o cálculo do erro e no MetodoBissecao2.m utiliza-se $|b - a|$ como erro, isto gera mais uma iteração no segundo programa. Note que o valor de x_9 na Figura 3.6 é igual à $-2{,}3546$ resultado do programa MetodoBissecao.m.

3.2.2 *Método de Newton-Raphson*

No primeiro capítulo, vimos que o método de Newton-Raphson consiste em obter um valor aproximado para a raiz de uma função através da interseção da reta tangente a curva com o eixo das abscissas.

A Figura 3.7 representa o fluxograma do método de Newton-Raphson,. Neste algoritmo são inseridos a função, o valor inicial x_n próximo a raiz, a precisão para a realização dos cálculos e a quantidade de iterações. Após realizar todos as etapas do processo encontra-se a raiz da equação.

FIGURA 3.7 – FLUXOGRAMA MÉTODO DE NEWTON-RAPHSON



FONTE: Autoria Própria (2018).

Para a implementação do método de Newton-Raphson foi construído um programa (salvo com o nome: Newton.m) que organiza os dados em tabelas e exige um número maior de linhas para ser desenvolvido e, a precisão é calculada pelo erro absoluto, ou seja, $|f(x_n) - f(x_{n-1})|$.

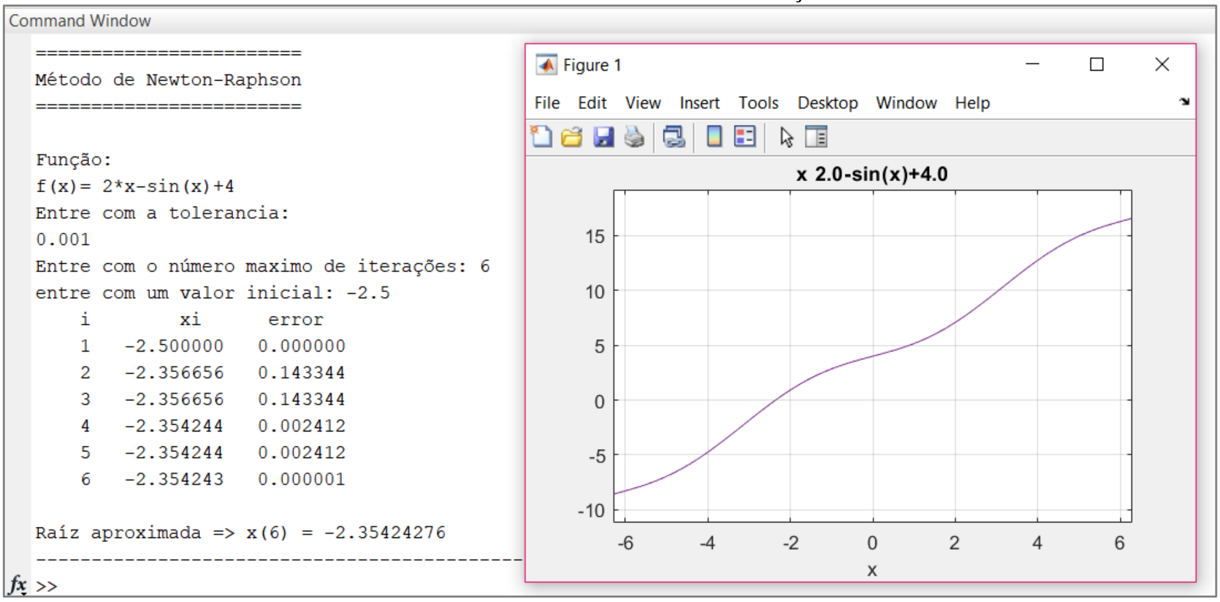
Programa 3: Salvo com o nome Newton.m

```
%Metodo de Newton
clear,
clc
syms x %declarando a variavel x

fprintf('%s\n','========================');
fprintf('%s\n','Método de Newton-Raphson');
fprintf('%s\n','========================');

%Recebe a funçao
    fprintf('\nFunção:\n');
    cf = input('f(x)= ');
    f = inline (cf);

    ezplot(f);  %grafico da Funcão
    grid on     %ativa as grades do grafico
    hold on

%Derivada
    derivada = diff (cf, x);    %o comando diff calcula a derivada
    df = inline(derivada);


%Recebe a precisão
    disp ('Entre com a tolerancia: ');
    tol = input ('','s');
    error = 0;

%Recebe o numero maximo de iterações
    max = input ('Entre com o número maximo de iterações: ');

%valor inicial
    x = input ('entre com um valor inicial: ');

    i = 1;      %contagem da primeira iteração

    %construindo a tabela
    disp('    i        xi       error')

    %comando while
    while (i <= max)     %se o numero de iterações for menor que o
                           %max de iterações

        fprintf('   \t%i \t%4f  \t%f\n', i, x, error);

        p = x;                      %p torna-se o valor de x, facilitando no
                                         %calculo do erro
        x = x - (f(x)/df(x));       %novo valor de x, formula do método
                                   de newton
        error = abs(x - p);          %calculo do erro, valor de x atual menos
                                         %o valor antigo de x (p)

        % if  error < tol
        if tol > error              %condicional if, se o tolerancia
                                        %for maior que o
        i = i + 1;
        fprintf('\t%i \t%4f \t%f\n', i, x, error);
        end
        i = i+1;
    end

fprintf('\nRaiz aproximada => x(%d)  = %10.8f\n', max, x);
fprintf('%s\n','-------------------------------------------');
```

Exemplo: Cálculo da raiz da função $f(x) = 2x - \text{sen}(x) + 4$ no Excel, considerando o erro de 10^{-3}, utilizando o método de newton.

Ao executar o arquivo MetodoNewton.m, o primeiro dado a ser inserido conforme as instruções será a função (digitada em função da variável x). O código já está programado para calcular a derivada, então resta inserir os outros dados. Observando o gráfico na janela *figure 1*, preenche-se o dado referente à aproximação inicial, que será igual a $-2{,}5$.

FIGURA 3.8 – PROGRAMA Newton.m COM OS GRÁFICOS DA FUNÇÃO.



FONTE: Autoria Própria (2018)

Na Figura 3.8 nota-se que o valor aproximado da raiz para a função $f(x) = 2x - sen\,(x) + 4$ é de $-2{,}3542$, com uma precisão menor que 10^{-3} e em 6 iterações.

Durante a programação dos métodos é notável a semelhança entre eles, os dados são inseridos através de comandos semelhantes, como o *fprintf* ou *imput*, as principais diferenças são apresentadas quando o laço é construído, onde é respeitado as características dos métodos e suas respectivas formas de calcular as aproximações. A principal complicação durante a implementação dos métodos no MATLAB® foi estabelecer o critério de parada conforme o tipo de erro a ser considerado. Nota-se que é mais simples utilizar o critério de parada por meio da amplitude do intervalo. No geral a programação dos métodos é bem semelhante, e após se familiarizar com o *software* MATLAB® a implementação dos métodos fica mais intuitiva.

4 APLICAÇÃO: ESTIMATIVA DE VARIÁVEIS PSICROMÉTRICAS

Como visto nos capítulos anteriores, métodos numéricos podem ser usados para encontrar raízes de funções, sua aplicabilidade se dá principalmente nos ramos das ciências exatas. Neste capítulo apresenta-se uma aplicação dos Métodos Numéricos que pode ser utilizada no ramo da Engenharia Agrícola e Agronomia.

> O desenvolvimento de técnicas de manejo de irrigação requer o conhecimento de quando, quanto e como aplicar água, sendo dependentes das condições atmosféricas locais; o conhecimento dessas variações tem grandes aplicações no meio agrícola, principalmente em se tratando das necessidades hídricas das culturas (CASTELLVÍ et al, 1996, apud MIRANDA et al, 2006, p.686).

Através do artigo “APLICAÇÃO DE MÉTODOS NUMÉRICOS PARA ESTIMATIVA DE VARIÁVEIS PSICROMÉTRICAS” (MIRANDA, et al. 2006), foram coletados os dados necessários para estimar valores da temperatura do termômetro de bulbo úmido (T_u) em função de algumas variáveis, por meio do método da Bisseção. Como o artigo utiliza do Método de Newton-Raphson para calcular as estimativas, este será usado apenas para comparação dos valores encontrados.

Os dados utilizados para o cálculo aproximado da temperatura do termômetro de bulbo úmido ($\mathrm{T_u}$) foram os mesmos utilizados no artigo, assim os resultados obtidos neste capítulo podem ser comparados aos originais.

A equação para determinar a temperatura do termômetro de bulbo úmido ($\mathrm{T_u}$) é obtida a partir de outras duas equações.

A primeira determina a pressão exercida pelo teor saturante de vapor d’água ($\mathrm{e_s}$). Segundo Pereira (2001, p.46)

$$\mathrm{e_s} = 0{,}6108 \cdot 10^{\frac{7{,}5\mathrm{T_{AR}}}{237{,}3+\mathrm{T_{AR}}}}$$

onde T_{AR} é a temperatura do ar em $°C$. Considerando a temperatura do psicrômetro de bulbo úmido ($\mathrm{T_u}$), em ºC, tem-se

$$\mathrm{e_{su}} = 0.6108 \cdot 10^{\frac{7.5\mathrm{T}_u}{237.3+\mathrm{T}_u}} \tag{1}$$

A segunda equação necessária calcula a pressão exercida pela massa atual de vapor d’agua ($\mathrm{e_a}$), é dada por:

$$e_a = e_{su} - cp \cdot P \cdot (T_s - T_u) \tag{2}$$

Onde cp é a constante psicrométrica com valor de $0.0008\ °C^{-1}$, "para o psicrômetro não ventilado (em abrigo com meteorológico com ventilação natural)" (PEREIRA et al, 2001, p. 48), P é a pressão atmosférica local e T_s é a temperatura do psicrômetro de bulbo seco. Substituindo e_{su}, definida na equação (1), na equação (2), é obtém-se a seguinte expressão:

$$e_a = 0.6108 \cdot 10^{\frac{7.5T_u}{237.3+T_u}} - cp \cdot P \cdot (T_s - T_u) \tag{3}$$

A partir da equação (3), pode-se obter o valor de T_u, levando em consideração os que são fornecidos os seguintes dados: a pressão atual de vapor (e_a), a temperatura do termômetro de bulbo seco (T_s), a constante psicrométrica (cp) e a pressão atmosférica local (P). Desta forma, resta apenas encontrar uma aproximação para a temperatura do termômetro de bulbo úmido (T_u).

Logo, obtém-se a seguinte equação (4) que será aplicada nos programas desenvolvidos no capítulo 3:

$$0.6108 * 10^{\frac{7.5T_u}{237.3+T_u}} - cp\ P\ (T_s - T_u) - e_a = 0, \tag{4}$$

considerando a função $f(T_u) = 0.6108 * 10^{\frac{7.5T_u}{237.3+T_u}} - cp\ P\ (T_s - T_u) - e_a$.

Na próxima seção aplica-se a equação (4) em ambos os métodos desenvolvidos. Destaca-se o método da bisseção, considerando que o artigo realiza os cálculos utilizando o método de Newton-Raphson.

4.1 Calculo da $(\mathbf{T_u})$ no MATLAB®.

Nesta seção o cálculo da temperatura do termômetro de bulbo úmido através da equação (4) foi realizado por meio dos programas desenvolvidos no Capítulo 3. Para a sua realização foram consideradas as seguintes variáveis[2]:

$cp = 0.0008\ ºC^{-1}$;

$P = 94.2\ kPa$;

$T_s = 15.03\ °C$;

[2] Dados obtidos segundo Miranda (2006).

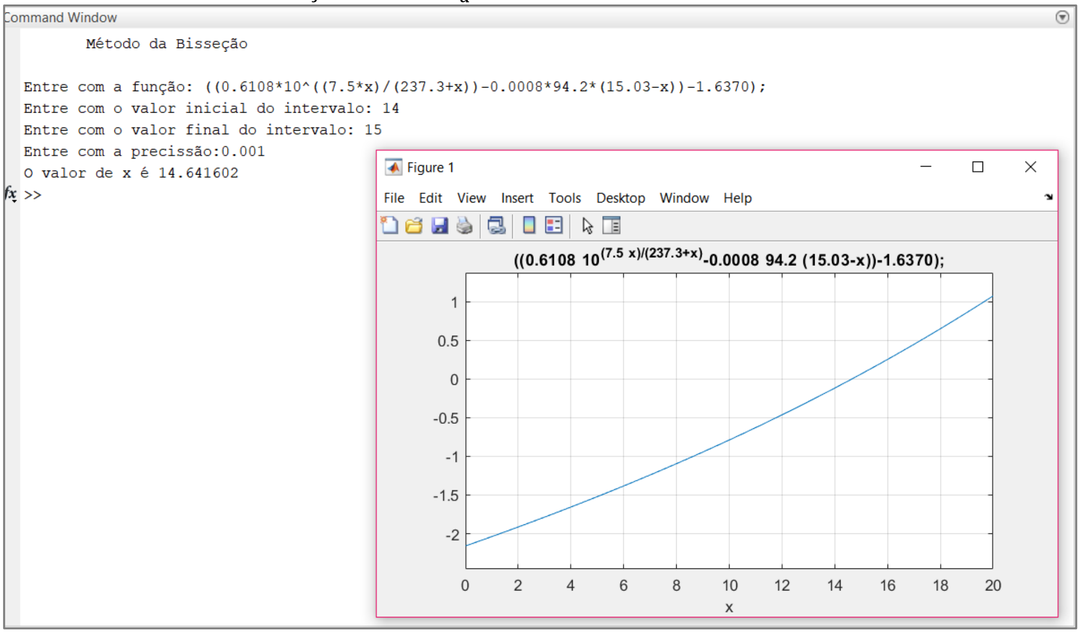
$e_a = 1.6370\ \text{kPa}.$

Substituindo na equação (4), temos que:

$$0.6108 \cdot 10^{\frac{7.5T_u}{237.3+T_u}} - 0.0008 \cdot 94.2 \cdot (15.03 - T_u) - 1.6370 = 0 \qquad (5)$$

A equação (5) foi aplicada nos programas MetodoBissecao.m e MetodoBissecao2.m. Nas Figuras 4.1 e 4.2 é possível observar que foram obtidas as aproximações de 14.5416 e 14.6411 para T_u respectivamente. Considerados o intervalo $[14, 15]$ e uma precisão $\epsilon < 10^{-3}$.

FIGURA 4.1 – APROXIMAÇAO PARA T_u PELO “MetodoBissecao.m”.



FONTE: Autoria Própria (2018)

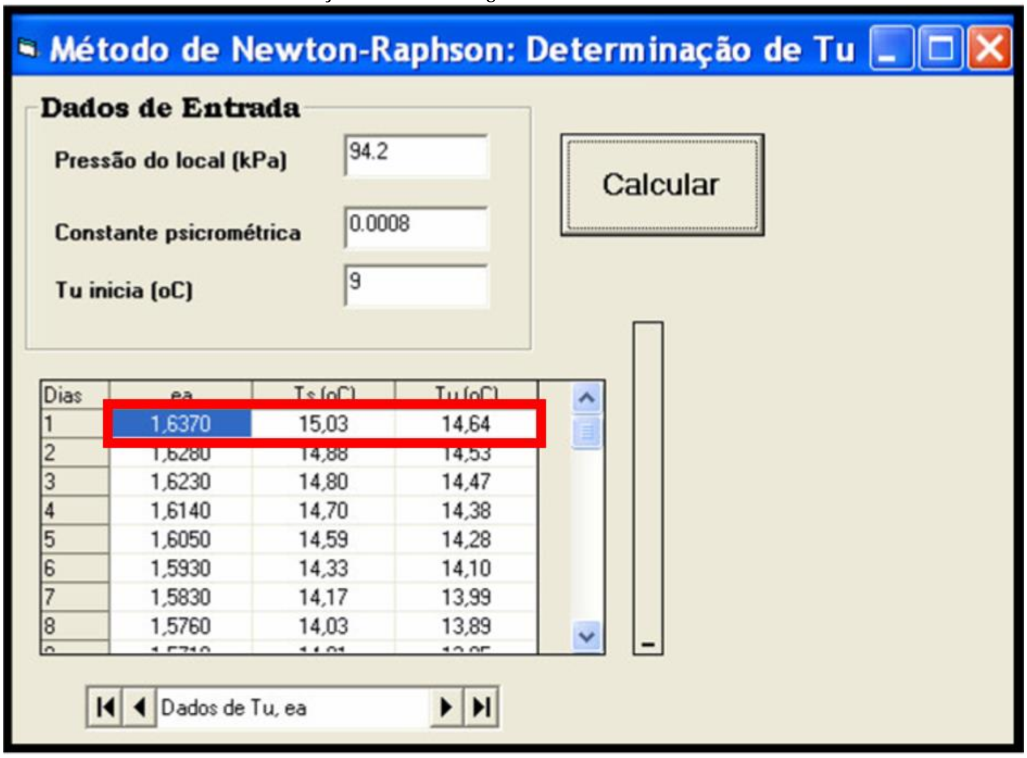
FIGURA 4.2 – APROXIMAÇAO PARA T_u PELO “MetodoBissecao.m”.

```
Command Window
       Método da Bisseção
   Entre com a função:((0.6108*10^((7.5*x)/(237.3+x))-0.0008*94.2*(15.03-x))-1.6370)
   Entre com o valor inicial do intervalo:14
   Entre com o valor final do intervalo:15
   Entre com a precissão:0.001

   ---------------------------------------------------------------------------
    iteração         a            b            x             f(x)          c
   ---------------------------------------------------------------------------
        1       14.500000000  15.000000000  14.750000000   0.01993043   0.500000000
        2       14.500000000  14.750000000  14.625000000  -0.00297224   0.250000000
        3       14.625000000  14.750000000  14.687500000   0.00846717   0.125000000
        4       14.625000000  14.687500000  14.656250000   0.00274449   0.062500000
        5       14.625000000  14.656250000  14.640625000  -0.00011462   0.031250000
        6       14.640625000  14.656250000  14.648437500   0.00131475   0.015625000
        7       14.640625000  14.648437500  14.644531250   0.00060001   0.007812500
        8       14.640625000  14.644531250  14.642578125   0.00024268   0.003906250
        9       14.640625000  14.642578125  14.641601563   0.00006403   0.001953125
       10       14.640625000  14.641601563  14.641113281  -0.00002530   0.000976563

   Raíz aproximada => n(10) = 14.64111328
fx >>
```

FONTE: Autoria Própria (2018).

Nota-se que ambos chegam a valores aproximados e, quando comparado ao cálculo Método de Newton-Raphson, realizado por Miranda et al (2006) conforme Figura 4.3, verifica-se a eficiência dos programas criados no Capítulo 3.

FIGURA 4.3 – APROXIMAÇÕES PARA T_U PELO MÉTODO DE NEWTON-RAPHSNON.



FONTE: MIRANDA et al. (2006).

Através do cálculo da temperatura do termômetro de bulbo úmido (T_u) pode-se comparar os resultados obtidos pelo autor do artigo com os dados obtidos através dos programas para o cálculo do Método da Bisseção, o que proporcionou um bom desempenho durante a aplicação.

CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho possibilitou a realização de um breve estudo acerca dos métodos numéricos da Bisseção e de Newton-Raphson, que objetivam a busca pela raiz real (zero real) de uma função, dando bases para o desenvolvimento de técnicas opcionais para as resoluções. O Método da Bisseção apresenta cálculos simples para serem executados quando comparado ao Método de Newton-Raphson. Porém o número de iterações do segundo método e menor em relação ao primeiro.

É perceptível a eficiência dos métodos utilizados, estes apresentam aproximações convenientes da raiz procurada, contudo, dada a grande quantidade de cálculos, mostrou-se necessário o desenvolvimento de opções que possibilitam maior agilidade na execução dos métodos. Para tal, o uso dos *softwares* Microsoft Office Excel e MATLAB® mostrou-se como recurso eficiente para obter resultados satisfatórios, com uma demanda menor de tempo e menor possibilidade de erros nos cálculos se comparados as resoluções manuais.

Os códigos desenvolvidos no MATLAB® mostraram ser eficientes para o cálculo de uma estimativa da Temperatura do Termômetro de Bulbo Úmido, o que pôde ser comprovado por meio da comparação com o resultado obtido por Miranda et al. (2006).

Para a continuidade deste trabalho, pode-se ampliar o estudo através de outros métodos, como o Método do Ponto Fixo, Método da Secante e o Método de Pégaso. Para ambos os métodos podem ser estudas as possibilidades para a sua resolução e verificação se os mesmos podem ser executados através dos *softwares* mencionados anteriormente . Outras aplicações podem ser realizadas, como por exemplo em problemas econômicos que envolvem a geração térmica de energia elétrica ou na área da Engenharia Ambiental, para o cálculo da concentração de oxigênio em um rio.

REFERÊNCIAS

ANDRADE, D. F. **Excel 2010: controlando dados.** 1. ed. Santa Cruz do Rio Pardo, SP: Editora Viena, 2011.

BARROSO, L. C. et al. **Cálculo Numérico (com Aplicações)**. 2. ed. São Paulo: HARBRA, 1987.

CHAPRA, S. C.; Canale R. P. **Métodos Numéricos para Engenharia**. 5. ed. Porto Alegre: AMGH, 2011.

CHAPRA, S. C. **Métodos Numéricos Aplicados com o MATLAB® para Engenheiros e Cientistas.** 3º ed. Porto Alegre: AMGH, 2013.

GILAT, A. **MATLAB com Aplicações em Engenharia.** 4. ed. Porto Alegre, 2012.

MATSUMOTO, E. Y.; **MATLAB® R2013a: teoria e programação: guia prático.** 1. ed. São Paulo, 2013.

MIRANDA, J. H. de et al. **Aplicação de métodos numéricos para estimativa de variáveis psicrométricas.** Eng. Agríc., Jaboticabal, v. 26, n. 3, p. 686-694, dez. 2006. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-69162006000300004&lng=pt&nrm=iso>. acessos em 07 out. 2018.

PEREIRA, A. R.: et al. **Agrometeorologia: fundamentos e aplicações práticas.** Guaíba: Livraria e Editora agropecuária, 2001.

PINTO, J. C.; LAGE, P. L. C. **Métodos Numéricos em Problemas de Engenharia Química.** Rio de Janeiro: E-papers 2001.

RUGGIERO, M. A. G.; LOPES, V. L. R. **Cálculo Numérico: aspectos teóricos e computacionais.** 2. ed. São Paulo: Makron Books, 1996.

SPERANDIO, D.; MENDES, J. T.; E SILVA, L. H. M. **Cálculo Numérico: características matemáticas e computacionais dos métodos numéricos.** São Paulo: Prentice Hall, 2003.