Zero de Funções

Jhone Mendes de Almeida
Universidade Estadual de Feira de Santana
Curso de Engenharia de Computação
Departamento de Ciências Exatas

E-mail: jhonedarts@gmail.com

***Abstract*— the real roots of an equation, also called function zeroes, may become too complex to be calculated in the conventional way. In this scenario, so-called numerical methods are used that facilitate the resolution of such problems. These do not guarantee the accuracy of the result, but can be configured to bring approximations with negligible errors.**

The present work aims to demonstrate the solution adopted by the author to solve the problem of function zero of the equation f (x) = ln (2x) + cos (2x) -2 that was proposed. The methods used for this resolution were: Bisection, False Position, Newton-Raphson and Secants. The implementation of the 4 individual codes, one for each method addressed were done in the MatLab language, using the MATLAB R2016a IDE and function independently of each other.

Keywords*— *Métodos Numéricos, Bisseção, Iterativo Linear, Newton-Raphson.

I. Introdução

As raízes reais de uma equação, também chamadas de zeros da função, podem acabar se tornando complexas demais para serem calculadas da maneira convencional. Neste enredo, são utilizados os chamados métodos numéricos que facilitam a resolução de tais problemas. Estes, não garantem a exatidão do resultado, mas podem ser configurados para trazerem aproximações com erros desprezíveis.

Uma equação f(x) = 0 pode ter uma, múltiplas, infinitas, ou nenhuma solução. Antes de utilizar um método para calcular a solução, e conveniente determinar alguns limites ou aproximações iniciais de modo a convergir para o resultado. Como veremos, todos os métodos que serão apresentados neste relatório exigem uma ou mais aproximações iniciais, e quanto mais próximas da solução estejam, maiores as chances de determina-la com sucesso e rapidez.

O presente trabalho visa demonstrar a solução adotada pelo autor para solucionar o problema de zero de função da equação f(x)= ln(2x) + cos(2x)-2 que foi proposta. Os métodos utilizados para esta resolução foram: Bisseção, Falsa Posição, Newton-Raphson e Secantes. Estes códigos foram utilizados por meio de implementações em MATLAB respeitando suas condições e dando suporte a possibilidade de funcionar com outras funções.

II. Fundamentação Teórica

A. *Métodos Iterativos para zero de funções*

Existem diversos métodos numéricos que são processos iterativos. Observando o nome concluímos intuitivamente que esses processos se caracterizam pela repetição de uma determinada operação. A ideia nesse tipo de processo é repetir um determinado cálculo várias vezes, obtendo-se a cada repetição ou iteração um resultado mais próximo do resultado que aquele obtido na iteração anterior. E, a cada iteração utiliza-se o resultado da iteração anterior como parâmetro de entrada para o cálculo seguinte. Para os métodos abordados neste trabalho, seguem os seguintes aspectos:

1. Estimativa inicial: como um processo iterativo se caracteriza pela utilização do resultado da iteração anterior para o cálculo seguinte, logicamente é necessário que o problema lhe supra com valores para as aproximações iniciais para que o método comece. Esses valores podem ser conseguidos de diferentes formas, uma delas que inclusive foi a forma neste trabalho foi o estudo do gráfico da função.

2. Convergência: a fim de se obter um resultado próximo do resultado real, é preciso que a cada passo ou iteração, o resultado esteja mais próximo daquele esperado, isto é, é necessário que o método convirja para o resultado real. Essa convergência nem sempre está garantida em um processo numérico. Portanto, é muito importante se estar atento a isso e realizar- a verificação da convergência do método para um determinado problema antes de tentar resolvê-lo;

3. Critério de Parada: o método não pode ficar realizando iterações infinitamente. É preciso pará-lo em um determinado instante. Para isso, devemos utilizar um critério de parada, que vai depender do problema a ser resolvido e da precisão que precisamos obter na solução.

B. *Método da Bisseção*

É um método de quebra, ou seja, cada nova iteração é um quebra da iteração anterior. O método da Bisseção é um confinamento usado para se obter a solução de uma equação na forma f(x)=0 quando se sabe que, dentro de um dado intervalo [a,b], f(x) é continua e que a equação possui solução (GILAT, 2008).

Diante de intervalo que respeite as condições, o método divide este intervalo pela metade, caso o critério de parada não tenha sido alcançado, este valor intermediário do intervalo servirá para selecionar um novo intervalo mais próximo da solução. Este novo intervalo é entre o valor intermediário e [a] ou dele e [b], o que possuir a raiz. Para identificar qual subintervalo possui a raiz é verificado se a função com o valor de [a] possui sinal diferente a função com o valor intermediário, se sim, tem a raiz, se não, o intervalo do valor intermediário e [b] que possui a raiz e deve ser escolhido para a nova iteração.

Formula: x=(a+b)/2, onde x representa o valor intermediário de cada iteração.

C. *Método da Falsa Posição*

Também chamado de Cordas é muito similar ao método da Bisseção e necessita das mesmas condições de operação para o intervalo inicial. A diferença é a forma com que é calculado o valor intermediário, que ao invés de ser a metade do intervalo [a,b] é um valor ponderado. Isso faz com que o método convirja mais rapidamente para a solução exata.

Esse ponderamento que tende a solução exata é dado pela função que calcula o x, sendo este o valor intermediário:

Formula: x = a – ((b-a)*f(a))/(f(b)-f(a)).

D. *Método de Newton-Raphson*

O método de Newton-Raphson, também conhecido como método das Tangentes, ou apenas de Newton, é um esquema usado para achar a solução numérica de uma equação na forma de f(x)=0, onde f(x) é continua e diferençável.

Segundo Gilat (2008), o processo de solução começa com a escolha do ponto x_1 como primeira estimativa da solução. A segunda estimativa, x_2, é obtida a partir do cruzamento com o eixo x da tangente a f(x) no ponto $(x_1,f(x_1))$, para o x_3, repete-se o processo agora usando x_2 nos cálculos. O ciclo é repetido até que seja atingido o critério de parada.

Para decidir qual o melhor extremo do intervalo [a,b] a iniciar o método, basta verificar qual dos extremos possui função e segunda derivada com mesmo sinal: $f(x_k).f''(x_k) > 0$ (para k igual aos extremos do intervalo).

E. *Método das Secantes*

O método das secantes também é similar ao método de Newton-Raphson. Uma diferença é que a reta secante é substituída pela tangente em um dado ponto. A substituição da tangente no método de Newton pela secante torna o método relativamente mais simples, pois não é necessário calcular a derivada da função para de obter o coeficiente angular da reta. Diante do fato de que uma secante cruza com a curva da função em dois pontos, para o método das secantes precisa-se de duas aproximações iniciais $[x_1, x_0]$, ao invés de uma como no de Newton-Raphson.

Definido o intervalo, o próximo passo é traçar uma reta secante à curva, que passe pelos dois pontos. Esta reta intercepta o eixo das abcissas em um dado ponto x2. Este, por sua vez, configura um novo intervalo, definido por $[x_2, x_1]$.

F. *Erro absoluto Estimado*

É uma forma de mensurar o erro embutido na aproximação atual em comparação com a aproximação anterior, tratando-se da diferença entre elas. Formula: Ea = $|x_{i-1} - x_i|$, algumas referências mostram como Ea = $|x_i - x_{i+1}|$, porém retratam o mesmo cálculo.

III. Desenvolvimento

Foram implementados quatro códigos individuais, um para cada método abordado. Foi utilizada a linguagem MatLab, usando a IDE MATLAB R2016a e funcionam independentes entre si. Os códigos resolvem um problema sem a interação com o usuário durante suas execuções. Para que os algoritmos resolvam outro problema de zero de função deverão ser feitas alterações nos códigos mudando a função, critério de parada e intervalo ou aproximações iniciais. Sendo estes os “parâmetros de entrada” de cada um dos quatro algoritmos. Em todos eles é feito uma verificação inicial antes de entrar no laço de repetição verificando se os valores de entrada já são o suficiente no que diz respeito aos critérios de parada da execução do algoritmo.

Para todos os métodos foram utilizados o intervalo inicial de 7 a 7.5 que contém a quarta menor raiz positiva da função: f(x)= ln(2x)+cos(2x)-2. Esse intervalo foi definido por meio da análise do gráfico da função obtido pela ferramenta online Wolfram.

E para evitar que os algoritmos continuassem executando indefinidamente caso algo de errado tenha ocorrido, foi implementado em todos um gatilho de parada de 100 iterações, ou seja, cada algoritmo executa no máximo 100 iterações. Caso seja interessante mudar isso, assim como os outros parâmetros, a alteração deverá ser feita no código.

Para o método da Bisseção o critério de parada é uma amplitude do intervalo menor que 10^{-6}. Sua execução nos forneceu um resultado arredondado para 8 casas decimais igual a 7.45798779 com a amplitude de 0.00000095 após 19 iterações. O Arredondamento para os métodos restantes foi com 16 casas decimais.

Já o método da Falsa Posição usou o erro absoluto estimado de 10^{-6}. Sua execução foi mais rápida que a da Bisseção. Encontrou uma aproximação satisfatória com 9 iterações. Valor encontrado foi 7.4579885738987706 e o erro absoluto estimado ficou em 0.0000003736832124.

O método de Newton-Raphson foi o mais rápido, usando o erro absoluto estimado de 10^{-6} como critério de parada. Sua execução encontrou uma aproximação satisfatória após 4 iterações, com o valor da aproximação final igual a 7.4579884566707024 e o erro absoluto estimado = 0.0000000000307487.

O método das Secantes foi o segundo mais rápido, usando o erro absoluto estimado de 10^{-6} como critério de parada. Sua execução encontrou uma aproximação satisfatória após 5 iterações, com o valor da aproximação final igual a 7.4579884566706767 com o erro absoluto estimado =

0.0000000037667967.

IV. Conclusão

O trabalho desenvolvido explorou implicitamente um comparativo entre os métodos: Bisseção, Falsa posição, Newton-Raphson e Secantes. Esse comparativo é dado pela tabala 1.

Metodo	I	Erro/Amplitude	Valor encontrado
Bisseção	19	0.00000095	7.45798779
Falsa Posição	9	0.0000003736832124	7.4579885738987706
Newton-Raphson	4	0.0000000000307487	7.4579884566707024
Secantes	5	0.0000000037667967	7.4579884566706767

Vemos com clareza que de longe o método da Bisseção é o mais ineficiente perante os demais. Os mais rápidos foram o de Newton-Raphson e o das Secantes, possuindo também os resultados com menores erros. Diante de todas essas informações é previsível concluir que o melhor método pra a resolução da quarta menor raiz real da equação proposta é o de Newton-Raphson.

Referências

[1] A. Gilat V. Subramaniam, “Métodos Numéricos para engenheiros e cientistas uma introdução com aplicações usando o MATLAB,” tradução Alberto Resende de Conti – Porto Alegre, Bookman, 2008.

[2] Patrício Torres Costa e Luciane Sobrale Silva. (2009, Junho). *Métodos Numéricos para Zeros de Funções*. [Online]. Available: https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/107647/MTM0038-M.pdf?sequence=1&isAllowed=y

[3] Ruggiero, M.A.G. e Lopes, V.L.R. Cálculo Numérico, Aspectos Teóricos e Computacionais. São Paulo. McGraw-Hill, 1988.