RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES NÃO LINEARES PELO MÉTODO DE NEWTON

MÉTODOS COMPUTACIONAIS EM ENGENHARIA

Douglas Silva, Gutembergue da Silva, Jeffet da Silva e Roger Conceição

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
NOVEMBRO DE 2022

RESUMO

O presente trabalho terá como objetivo principal a obtenção da solução numérica de um sistema de equações não lineares utilizando o Método de Newton. Com base nisso, é importante destacar que este referido método consiste em estimar as raízes de uma função, a partir de uma aproximação inicial. Os detalhes desta aplicação serão apresentados conforme desenvolvimento desse afazer. Vale enfatizar, também, que a modelagem e a resolução numérica deste problema foram desenvolvidas através de um algoritmo na linguagem $C++$, o qual resolverá o problema e apresentará suas devidas respostas, no que concerne à proposta de atividade disponibilizada para o nosso grupo.

Sumário

1 Introdução

Para introduzir este trabalho, começaremos explicando o que seriam as equações não lineares e o que diferenciam essas das equações lineares. As equações não lineares são aquelas que os valores dos expoentes de suas variáveis são diferentes de 1, ou seja, maiores ou menores que 1, o que implica dizer que as equações lineares, por sua vez, têm o valor dos expoentes das suas variáveis igual a 1. Seguem os exemplos para melhor entendimento. Em 1 está mostrado um exemplo de equação linear, e em 2 um exemplo de equação não linear.

$$F(x) = 3x - 4 = 0 \tag{1}$$

$$F(x) = 5x^2 - 5 = 0 \tag{2}$$

Além disso, é imprescindível ter conhecimento de que um sistema de equações não lineares pode ser formado não tão somente com equações não lineares, ou seja, mesmo que em um sistema de equações contenham diversas equações lineares, mas que haja, ali, pelo menos uma equação não linear, esse sistema ainda será não linear, independentemente do seu tamanho.

$$\begin{cases} 3x - 4y + 2z = 0 \\ 4x - 3y + 5z = 0 \\ 5x^2 - 5y + 9z = 0 \end{cases}$$

Sistema 1: Exemplo de sistema não linear.

Por conseguinte, o objetivo deste estudo consiste na resolução numérica de um sistema não linear constituído por cinco equações e consequentemente cinco incógnitas desejadas para o vetor solução. Para isso, será utilizado o Método de Newton para sistemas de equações não lineares que, como já mencionado anteriormente, é bastante reconhecido e um dos mais utilizados para solucionar problemas análogos a esse. Logo, após a implementação do referido método, que será explicado mais a frente, ele nos fornece como resposta um sistema de equações linear, o qual será solucionado a partir da implementação do método de Gauss-Seidel, que possibilita uma convergência mais rápida a uma resposta final, quando comparado a outros métodos como o de Jacobi. Nos tópicos a seguir, serão apresentados o desenvolvimento e construção do problema, as dificuldades e limitações encontradas durante o processo e por fim, os resultados obtidos com a implementação dos métodos.

2 Apresentação do problema

Um sistema de equações não linear pode possuir uma ou mais soluções. Para encontrar essas soluções geometricamente, se faz necessário achar o ponto de interseção entre as curvas geradas. A obtenção da solução de um sistema não linear é de grande importância no âmbito físico e de engenharia, já que boa parte dos fenômenos podem ser descritos por eles. Em concordância, esse trabalho visa desenvolver a solução numérica do sistema não linear apresentado a seguir, composto por cinco equações. O objetivo final do estudo é a implementação do Método de Newton para obter as raízes que o satisfaçam.

$$\begin{cases} f_1 = -(3+\alpha x_1)x_1 + 2x_2 - \beta \\ f_2 = x_1 - (3+\alpha x_2)x_2 + 2x_3 - \beta \\ f_3 = x_2 - (3+\alpha x_3)x_3 + 2x_4 - \beta \\ f_4 = x_3 - (3+\alpha x_4)x_4 + 2x_5 - \beta \\ f_5 = x_4 - (3+\alpha x_5)x_5 - \beta \end{cases}$$

Em que $\alpha = -0.1$ e $\beta = 1$.

Dessa forma, todo o procedimento que será descrito nos tópicos a seguir, será trabalhado no sistema acima: cálculo da matriz jacobiana, aplicação do processo iterativo e aplicação do método da matriz inversa.

3 Desenvolvimento

O objetivo geral deste trabalho é comparar as soluções obtidas do sistema não linear citado acima pelo método de Newton, em que para um dos caminhos será utilizado a matriz jacobiana para encontrar a solução do sistema e no outro será utilizado a inversa da matriz jacobiana como forma de comparação dos resultados. No entanto, antes de se aprofundar nessa análise o método de Newton será conceituado. No tópico posterior será apresentado os programas desenvolvidos para resolver o sistema, e por fim serão analisados os resultados obtidos pelos dois caminhos diferentes.

3.1 Método de Newton

O Método de Newton é um dos métodos mais utilizados na resolução de problemas que envolvem sistemas de equações não-lineares e de duas ou mais variáveis. Este método utiliza a expressão:

$$JF(x_n)x = -f(x_n) \tag{3}$$

que permite obter, a cada nova iteração, um resultado aproximado melhor do que aquele obtido na iteração anterior. Sendo $JF(x_n)$ a matriz Jacobiana do sistema:

$$JF(x_n) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_n) & ... & \frac{\partial f_1}{\partial x_k}(x_n) \\ ... & ... & ... \\ \frac{\partial f_k}{\partial x_1}(x_n) & ... & \frac{\partial f_k}{\partial x_k}(x_n) \end{pmatrix}$$

S_n a matriz de passo, dada genericamente por:

$$S_n = \begin{bmatrix} x_1 \\ ... \\ x_n \end{bmatrix}$$

e $f(x_n)$ o conjunto de funções envolvidas.

Contudo, objetivando uma melhor compreensão da origem da expressão em 3 – e do método, por conseguinte -, consideremos o caso especial em que queremos obter aproximações para a solução de uma função não-linear de uma variável, por intermédio do Método de Newton. Para tanto, vejamos o seguinte gráfico da função não linear , em que é um chute inicial relativamente próximo ao valor suposto para a solução, e é o valor aproximado obtido na primeira iteração.

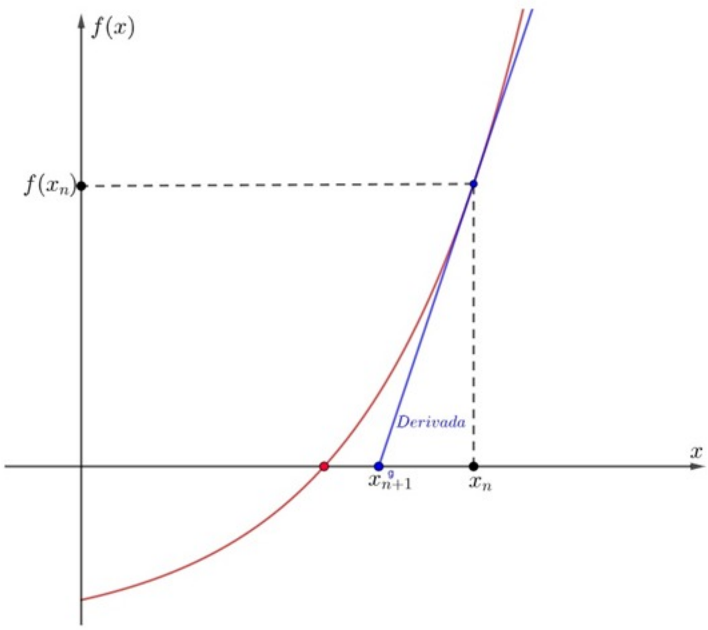

Figura 1: Método de Newton para função de uma variável.
Fonte: cn.ect.ufrn.br

Este valor é obtido traçando uma reta tangente à curva de $f(x)$ no ponto $R = (x_n, f(x_n))$, isto é, no ponto no qual a coordenada x é o valor do chute inicial. Desse modo, obtém-se x_{n+1} como sendo a raiz da reta tangente no ponto P, por meio da expressão a seguir.

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

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

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4}$$

Agora, se traçarmos uma reta tangente à curva no ponto $S = (x_{n+1}, f(x_{n+1}))$, obteremos x_{n+2} a raiz desta reta, sendo este um resultado mais próximo ao valor correto do que o obtido anteriormente. Desta maneira, repetindo o procedimento, é possível obter valores cada vez mais próximos da solução correta.

Podemos modelar matematicamente a reta tangente à curva em um determinado ponto. Para um ponto $Q = (x, y)$ qualquer pertencente à curva, temos que a reta tangente à curva em Q será modelada pela seguinte expressão:

$$tan(\theta) = a = \frac{y - f(x_n)}{x - x_n} \implies y - f(x_n) = a(x - x_n) \tag{5}$$

Seja $Q = (x, y) = (x_{n+1}, f(x_{n+1}))$. Então,

$$f(x_{n+1}) - f(x_n) = a(x_{n+1} - x_n) \implies 0 - f(x_n) = a(x - x_n)$$

$$-f(x_n) = a(x - x_n) \tag{6}$$

Podemos fazer,

$$-f(x_n) = aS \tag{7}$$

Vejamos que obtivemos uma maneira de determinar qualquer aproximação para a solução da equação, desde que se conheça apenas o valor da função no ponto calculado anteriormente. A expressão obtida acima pode ser generalizada para várias funções com várias variáveis, ou seja, ela pode ser estendida para ser aplicada a resolução de sistemas de equações não-lineares de várias variáveis. Para isso, percebamos que o coeficiente a, do lado direito de 7, é a derivada da função – o coeficiente angular, como mostra 5 -, que pode ser substituído pela matriz Jacobiana para o caso de várias variáveis, tendo em vista que esta consiste na matriz das derivadas parciais das equações. Ademais, consideremos não somente como uma equação que expressa a função, mas como um sistema de equações, bem como que, neste caso, passa a representar uma matriz de diferenças de variáveis independentes. Diante destas considerações, chegamos a

$$JF(x_n)S_n = -f(x_n)$$

que define o Método de Newton, conforme apresentado anteriormente pela equação 3.

3.2 Procedimento prático

A partir da explanação da origem da expressão utilizada no Método, abordada no subtópico anterior, pode-se inferir que o Método de Newton consiste em estabelecer sucessivas aproximações lineares para as funções nos pontos de interesse, por meio da reta tangente à curva da função, obtendo, desta forma, soluções de equações lineares que aproximam as soluções das equações não-lineares. É válido ressaltar que tal aproximação é mais precisa quanto maior for o número de iterações realizadas.

Fundamentado nessa teoria, o procedimento foi aplicado no sistema inicialmente proposto. No entanto, foi definida, de forma inicial, a expressão constituinte da matriz jacobiana para que fosse possível a construção do método iterativo. Assim, abaixo será relembrado o sistema de equações que se deseja resolver e a expressão gerada pela derivação dos termos para cada uma das cinco variáveis.

$$\begin{cases} f_1 = -(3+\alpha x_1)x_1 + 2x_2 - \beta \\ f_2 = x_1 - (3+\alpha x_2)x_2 + 2x_3 - \beta \\ f_3 = x_2 - (3+\alpha x_3)x_3 + 2x_4 - \beta \\ f_4 = x_3 - (3+\alpha x_4)x_4 + 2x_5 - \beta \\ f_5 = x_4 - (3+\alpha x_5)x_5 - \beta \end{cases}$$

$$JF(x_n) = \begin{pmatrix} -3-2\alpha x_1 & 2 & 0 & 0 & 0 \\ 1 & -3-2\alpha x_2 & 2 & 0 & 0 \\ 0 & 1 & -3-2\alpha x_3 & 2 & 0 \\ 0 & 0 & 1 & -3-2\alpha x_4 & 2 \\ 0 & 0 & 0 & 1 & -3-2\alpha x_5 \end{pmatrix}$$

Matriz 1: Matriz Jacobiana para o sistema desejado.

Com essa duas expressões, é possível iniciar o método iterativo de forma computacional para a resolução do sistema.

4 Algoritmos desenvolvidos

Os códigos que modelam o sistema foram desenvolvidos em linguagem C++ e no software Scilab. Esses visam solucionar o sistema de equações não lineares pelo método de Newton. No tópico anterior foi apresentado como encontrar as raízes de uma equação não linear pelo método de Newton, assim como as alterações na fórmula de Newton para tornar as equações do sistema linear.

4.1 Algoritmo em C++

A partir de um chute inicial, utilizando o método de Newton o sistema não linear é transformado em um linear, no entanto as soluções desse novo sistema não são iguais as do sistema anterior, mas sim uma aproximação mais próxima que chute inicial. Quanto mais perto o chute inicial estiver da solução do sistema não linear, mais aproximada estará a solução do sistema linear daquela desejada. Desta forma, é possível encontrar uma solução melhorada para o sistema não linear, que será somada com o chute inicial e será considerado o novo chute, que gerará um novo sistema linear que tem como solução uma melhor aproximação que a anterior, de forma geral o princípio de funcionamento dos códigos é esse, percorrendo esse *loop* de acordo com o número de iterações desejados. No entanto, existe também a verificação da precisão que verifica se as raízes do sistema já possuem, um erro mínimo em relação a resposta anterior de forma a encerrar o *loop* e apresentar as raízes, ou se será necessário mais um ciclo para obter a precisão necessária, o diagrama de blocos abaixo deixa mais claro o funcionamento do algoritmo, o código não será exposto neste trabalho devido ao seu tamanho.

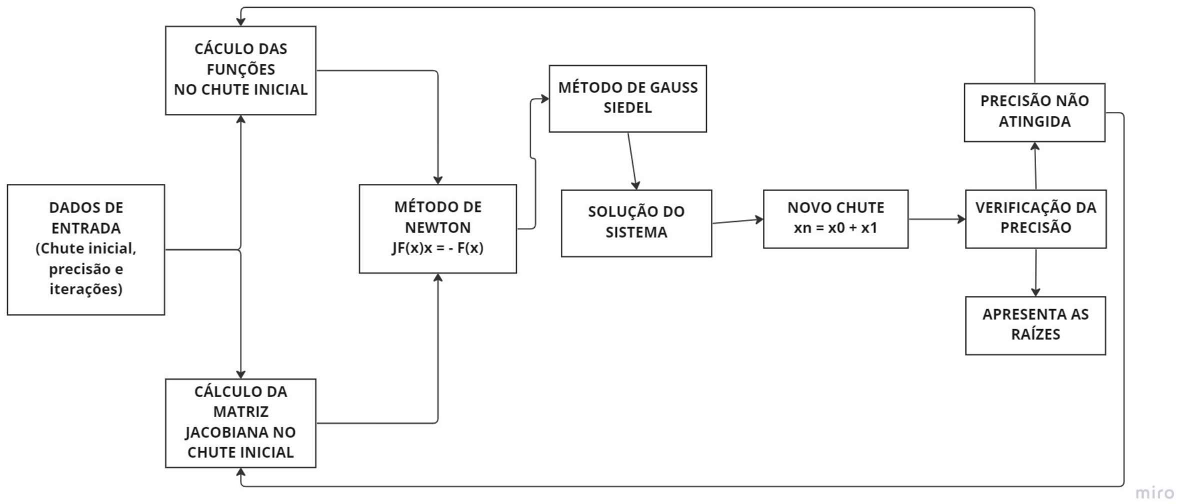

Figura 2: Construção do algoritmo.

Como pode-se observar acima, o algoritmo requer do usuário apenas um chute inicial, que pela teoria do método deve ser dado próximo às raízes desejadas, o número de iterações que ele deseja e a precisão requerida para a solução. A partir de então o programa inicia seu *loop*, até fornecer as raízes, em caso de convergência.

Juntamente à aplicação do Método de Newton, foi implementado o Método Iterativo de Gauss-Seidel para a resolução do sistema linear criado ao se calcular a jacobiana e as funções nos chutes. Essa escolha pelo método se deu pela sua convergência mais rápida quando comparada com outros métodos, como o de Jacobi. Vale ressaltar que o número de iterações fornecido na entrada não é o mesmo realizado pelo método de Gauss.

Assim, com a combinação dos dois processos iterativos, a solução numérica do sistema pode ser encontrada de forma rápida e precisa, a depender dos dados de entrada fornecidos pelos usuários.

O método de Newton é muito eficiente pois em poucas iterações ele é capaz de encontrar a solução do problema, isso o torna rápido. No entanto, por ser um método local, se o chute inicial se distancia muito da solução da questão provavelmente as respostas não serão encontradas, pois as soluções não convergiram para a resposta. Essas limitações serão discutidas em tópicos posteriores.

4.2 Comparação com Scilab

Além da implementação do algoritmo em C++, foi utilizado o Scilab como ferramenta de comparação dos resultados. No entanto, sua implementação se deu pelo método da matriz inversa, já que o próprio software permite algumas funções intrínsecas, como a inversão de matrizes. A expressão usada está mostrada abaixo e a implementação se deu de forma bem semelhante: um processo iterativo, que será encerrado quando uma certa tolerância (precisão) for atingida. Ainda assim, o chute inicial deve ser próximo ao resultado esperado.

$$x = [JF(x_n)]^{-1} f(x_n) \tag{8}$$

5 Resultados obtidos

Com a implementação da teoria e dos algoritmos descritos, foi possível a obtenção da solução numérica do sistema não linear desejado. O intervalo de chutes para a convergência da solução vem de valores negativos até valores positivos próximos e menores que 4. Acima disso, os valores de solução não convergem. Além disso, com poucas iterações, como já citado, o método apresenta uma solução com uma precisão excelente.

Na tabela abaixo, estão os valores obtidos como raízes do sistema para cada uma das variáveis com precisão de 15 casas decimais.

x_n	Solução Numérica
x_1	-1.540482047736029
x_2	-1.920991198446519
x_3	-1.790569232444309
x_4	-1.383208262809483
x_5	-0.774412294405729

Tabela 1: Raízes obtidas para o sistema.

Além da tabela, foi plotado um gráfico de variação da solução de acordo com o número de iterações digitados na entrada. Nele, o eixo horizontal mostra as variáveis x_n e o vertical o valor numérico da raiz. Essa verificação foi realizada usando uma precisão de 10^{-16} já que todo o processo foi trabalhado com precisão dupla e essa é a maior quantidade passível de representação.

Todos os teste mostrados na imagem 3 abaixo foram realizados utilizando o vetor nulo como chute inicial e seu intuito é apenas mostrar a rapidez de convergência do Método de Newton, que com apenas 8 iterações já atinge uma precisão tão boa já citada. Essa rapidez é característica do método e quando associada ao método de Gauss-Seidel, torna-se ainda mais eficaz.

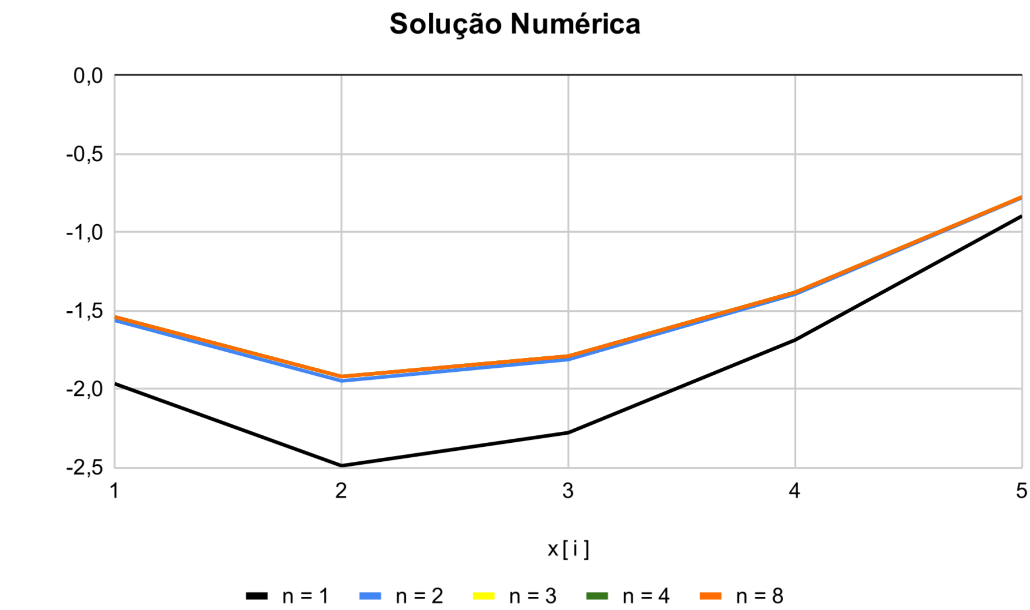

Figura 3: Solução em função das iterações.

Nota-se, em 3, que com uma iteração o processo ainda está relativamente distante da solução, porém da segunda iteração em diante, praticamente não nota-se diferença entre as curvas. Até atingir a precisão de 10^{-16} na oitava iteração. Abaixo, está mostrado o gráfico expandido no intervalo entre 1 e 3, no eixo horizontal, mostrando que a partir da iteração número 3, as curvas praticamente se sobrepõem.

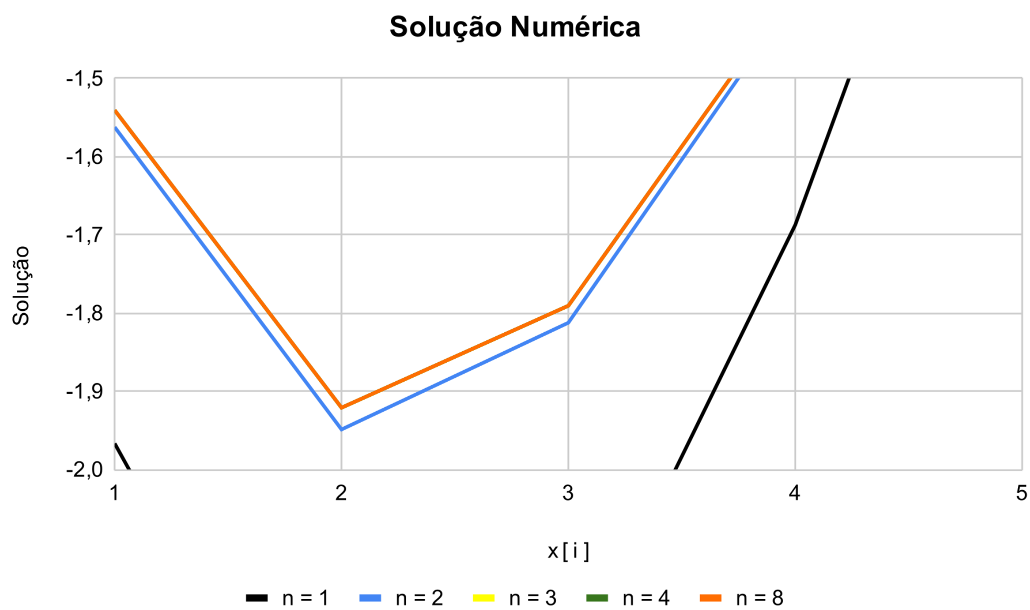

Figura 4: Gráfico expandido.

A aplicação do método da inversa no Scilab coincide com as raízes encontradas pelo método já citado, o que confirma a teoria e a eficácia do algoritmo desenvolvido em C++. Assim como, sofrendo suas mesmas limitações que serão descritas no tópico consecutivo.

6 Problemas encontrados

Inicialmente, podemos citar, como dificuldades encontradas, a definição dos valores referentes ao chute inicial. Conforme informa a literatura e as referências relativas ao Método de Newton, uma condição para a utilização do Método é a proximidade relativa entre os valores do chute inicial e das soluções buscadas. De fato, computacionalmente, ao executar o programa desenvolvido em C++, verificou-se que, ao fornecer valores iniciais relativamente distantes dos valores da solução como dados de entrada do programa, ocorre divergência da solução, conforme são feitas as iterações. Ademais, verificou-se, também, que este problema poderia ser solucionado com um número muito grande de iterações realizadas, o que requer maior esforço computacional, quanto maior for o número de iterações, tornando a solução inadequada. Por outro lado, valores próximos aqueles conhecidos como os valores da solução convergem para o valor esperado em pouquíssimas iterações. Desta maneira, a dificuldade encontrada se deve ao fato de que, para o êxito do método, é necessário que se conheça um valor em torno da solução correta. Vejamos que, para o caso de uma variável, isto pode ser feito mediante uma análise gráfica. No entanto, para mais de uma variável, este valor aproximado por meio da visualização gráfica torna-se inviável.

Em segundo lugar, foram encontradas dificuldades para implementação em código da segunda parte do problema, que consistia em admitir a inversa da Matriz Jacobiana para obter a solução desejada. Tais dificuldades têm relação com o fato de que as execuções dos programas não retornaram valores próximos aos esperados para matriz inversa . Nesse sentido, foram buscadas duas alternativas diferentes a primeira procurar um código pronto na internet que conseguisse realizar essa operação entretanto o mesmo apresentou problemas, a outra alternativa foi a implementação da biblioteca LAPACK (Linear Algebra Package) que é capaz de realizar esse procedimento no entanto não foi possível devido a dificuldade de instalação no ambiente Windows e a falta de conteúdo que explicasse seu funcionamento. Dessa forma, não adquiriu-se êxito nestas tentativas. Sendo assim, optou-se pela implementação desta parte do problema em um programa elaborado em Scilab apenas para fins comparativos.

7 Conclusão

O método de Newton é um método iterativo amplamente utilizado para resolução de sistemas não-lineares com mais de uma variável. Sabe-se que, nos campos da Física e da Engenharia, diversos problemas possuem um modelo matemático mais adequado a partir da aplicação de sistemas não-lineares, bem como da aplicação de funções de várias variáveis, visto que, muitas vezes, as grandezas envolvidas nos fenômenos físicos não possuem relação proporcional e, ainda, dependem de várias outras grandezas. Nessa ótica, o presente trabalho apresentou um programa capaz de tomar um sistema não-linear de várias variáveis e obter sua solução aproximada, mediante a implementação computacional do método de Newton.

O desenvolvimento da solução computacional se deu a partir de um algoritmo em C++, tal que a solução obtida foi posteriormente comparada com a solução apresentada por um algoritmo elaborado em Scilab, que utiliza uma pequena alteração no método apresentado. Sendo assim, a concordância entre as soluções para as variáveis do sistema não-linear obtidas pelos dois programas evidenciam a eficácia do software elaborado. Nessa lógica, o algoritmo desenvolvido neste trabalho mostra-se suficientemente confiável e aplicável para fins de modelagem matemática de situações-problema que envolvem sistemas de equações não-lineares, podendo, assim, ser útil para diversas finalidades práticas.

Por fim, conclui-se este trabalho com a satisfação advinda da possibilidade do contato, da utilização e da obtenção de conhecimentos acerca de um assunto que, indubitavelmente, acrescenta na formação discente em Engenharia Elétrica, propiciando, desta maneira, um engrandecimento intelectual, conforme o que se conjectura sobre a finalidade deste trabalho.

Referências bibliográficas

[1] ENGENHARIAS, Pet. Sistema de equações não lineares: Método de Newton - Cálculo Numérico. Disponível em: https://www.youtube.com/watch?v=4CFw7pXcw00. Acesso em: 03 nov. 2022.

[2] BLASS, Prof. Dr. Leandro. Método de Newton para Sistemas não lineares. Disponível em: https://www.youtube.com/watch?v=u4PaoDUuSgg&t=350s. Acesso em: 03 nov. 2022.

[3] VALLE, Marcos Eduardo. Sistemas Não Lineares e o Método de Newton. Campinas SP: Unicamp, 2016.

[4] UFSCAR. Derivadas e Retas Tangentes. Novas regras de derivação. Acesso em: 11 nov. 2022.

[5] UFRN, Ect. Ambiente Virtual de Aprendizagem para Computação Numérica. Disponível em: https://cn.ect.ufrn.br/index.php#. Acesso em: 11 nov. 2022.

[6] CORTIVO, Zaudir dal. Sistemas não lineares - Método de Newton. Disponível em: https://www.youtube.com/watch?v=4OnWKQRqM-Q. Acesso em: 11 nov. 2022.

[7] SAKATE, Canal Mathemathike - Prof. Thaty. Método de Newton para sistemas não lineares. Disponível em: https://www.youtube.com/watch?v=lno5xyQbIOU. Acesso em: 11 nov. 2022.