REVISTA BRASILEIRA DE CIÊNCIAS DA AMAZÔNIA

Edição especial:
II Simpósio Integrado UNIR | Ji-Paraná

Artigo

Aplicação do software Scilab no método da bissecção

Application of scilab software in the bissection method

Felipe Aparecido Batista da Silva[1*], Reginaldo Tudeia dos Santos[2], Enoque da Silva Reis[3], Carlos Alberto Almendras Montero[4]

1 Universidade Federal de Rondônia/ Departamento de Matemática e Estatística - Ji-Paraná – ORCID: https://orcid.org/0009-0003-4514-2758,
2 Universidade Federal de Rondônia/ Departamento de Matemática e Estatística - Ji-Paraná – ORCID: https://orcid.org/0000-0003-4986-2720, rtudeia@unir.br
3 Universidade Federal de Rondônia/ Departamento de Matemática e Estatística - Ji-Paraná – ORCID: https://orcid.org/0000-0001-6631-9688
4 Universidade Federal de Rondônia/ Departamento de Matemática e Estatística - Ji-Paraná – ORCID: https://orcid.org/0000-0003-0037-9500
* Correspondência: rtudeia@unir.br

Citação: Silva, F. A. B. da; Santos, R. T. dos; Reis, E. da S.; Montero, C. A. A. Aplicação do software Scilab no método da bissecção. *RBCA* **2024**, *13*, 3. p.123-142.

Editor de Seção: Dra. Karen Janones da Rocha
Recebido: 11/07/2024
Aceito: 17/08/2024
Publicado: 02/09/2024

Nota do editor: A RBCA permanece neutra em relação às reivindicações jurisdicionais em sites publicados e afiliações institucionais.

Abstract: Numerical Calculus is an area of mathematics dedicated to solving problems whose functions are non-analytical. In this sense, the objective of this work is to find the zero of a function using the SCILAB software, through the bisection method. The results have shown that the software was effective in obtaining an approximate solution to the proposed problem, from the implementation of the scripts related to the iterative procedures required for the resolution. The research emphasizes the importance of knowing both the software and the applicable numerical methods in SCILAB, enabling the teacher to teach the subject matter and the handling of the technological tool. It is worth noting, furthermore, that SCILAB is not limited to the application of the bisection method, as it can be used in various iterative methods, as well as in other areas of knowledge.

Keywords: Technological Tool; Numerical calculation; Iterative Method.

Resumo: Cálculo Numérico é uma área da matemática dedicada a resolver problemas cujas funções são não analíticas. Nesse sentido, o objetivo deste trabalho é encontrar o zero de uma função utilizando o *software SCILAB*, por meio do método da bissecção. Os resultados demonstraram que o *software* foi eficaz na obtenção de uma solução aproximada para o problema proposto, a partir da implementação dos *scripts* necessários para os procedimentos iterativos necessários para a resolução. A pesquisa enfatiza a importância de conhecer tanto o software quanto os métodos numéricos aplicáveis no *SCILAB*, o que possibilita ao professor ensinar o conteúdo da disciplina e o manuseio da ferramenta tecnológica. É importante salientar, ainda, que o *SCILAB* não se limita à aplicação do método da bissecção, uma vez que pode ser utilizado em diversos métodos iterativos, além de outras áreas do conhecimento.

Palavras-chave: Ferramenta Tecnológica; Cálculo Numérico; Método Iterativo.

1. Introdução

O presente trabalho tem como objetivo encontrar o zero de uma função que não possui solução analítica, bem como analisar a aplicabilidade do *software SCILAB* em certos problemas de Cálculo Numérico, empregando o método bissecção. A ideia principal é demonstrar a resolução de um problema matemático de maneira simplificada utilizando esse programa computacional. O *software SCILAB* pode ser utilizado tanto para problemas com soluções analíticas quanto, especialmente, para aqueles que não as possuem.

Ao considerarmos a matemática dividida em duas partes – o Cálculo Numérico e o Cálculo Algébrico –, podemos, então, afirmar que o Cálculo Numérico sempre requer métodos e ferramentas para obter uma solução de problema matemático de forma aproximada, ou seja, buscando um resultado aproximado para problemas que não possuem solução analítica. Por outro lado, quando o problema possui solução analítica, ela pode ser obtida de forma algébrica, às vezes, por fórmulas específicas.

O Cálculo Numérico, além de permitir a criação de métodos matemáticos para a solução numérica de um problema, permite, também, a utilização de *softwares* computacionais, como o *SCILAB*, que utiliza uma linguagem de programação robusta, com a finalidade de obter a solução aproximada do problema.

O *SCILAB* dispõe de um conjunto de linguagens de programação computacional disponíveis em sua biblioteca, ou que podem ser criadas para a aplicação de métodos numéricos no intuito de resolver problemas que necessitam de solução numérica. Ele é um *software* de código aberto, uma alternativa a outros *softwares* pagos, como o *MATLAB*, e sua história remonta a 1990, quando foi desenvolvido por pesquisadores franceses. Atualmente, é utilizado para a resolução de problemas simples e complexos que possuem solução analítica e/ou numérica.

Desta forma, o objetivo deste trabalho é encontrar o zero de uma função que não possui solução analítica, fazendo uso do *software SCILAB* e do método da bissecção.

2. Materiais e Métodos

A história da Matemática, na qual o Cálculo Numérico está inserido, remonta aos registros da época da matemática babilônica. Esta matemática engloba qualquer forma de matemática desenvolvida pelos povos da Mesopotâmia, desde os antigos Sumérios até a queda da Babilônia, em 539 a.C. (Sobral, 2015). Naquela época, os povos já desenvolviam soluções para os problemas geométricos de forma manual e, atualmente, essas soluções podem ser elaboradas em ambientes computacionais.

O Cálculo Numérico pode ser definido como um conjunto de ferramentas e métodos utilizados para obter soluções aproximadas de problemas matemáticos (Justo *et al.*, 2020; Lima *et al.*, 2023). Essas ferramentas podem ser utilizadas e executadas tanto para problemas que possuem solução analítica quanto àqueles que não as têm, ou seja, problemas que requerem métodos numéricos para resolução (Justo *et al.*, 2020; Lima *et al.*, 2023).

Ao solucionar um problema matemático que possui um grau de dificuldade maior, o mais recomendado é utilizar um *software* para obter a solução. Por isso, é preciso que se tenha conhecimento de como e quais procedimentos são necessários para resolver um problema de forma manual, para, posteriormente, adotar os recursos necessários para solucionar o problema por um método computacional (Lima *et al.*, 2023).

Para a resolução de um problema, é preciso observar alguns procedimentos, tais como:

- Se existem métodos disponíveis para solucionar o problema; se existirem, verificar se tais métodos são analíticos ou numéricos;
- Decidir qual o melhor caminho e método para a solução do problema; e,
- Comparar a solução obtida, tanto de forma manual quanto computacional.

Resolver o problema de solução não analítica, de forma manual, pode demandar uma quantidade maior de iterações, já que envolve menor precisão devido aos erros de arredondamento necessários em cada iteração, enquanto que, de forma computacional, a precisão pode ser maior, por usar mais casas decimais em cada procedimento. No entanto, o profissional ou interessado precisa conhecer os procedimentos necessários para resolver o problema e decidir se um determinado método é ou não adequado para a devida solução do problema.

Além disso, com o avanço da tecnologia computacional, os métodos numéricos tornaram-se mais eficazes para solucionar os mais diversos tipos de problemas. Uma parte dessa tecnologia permite, por exemplo, solucionar problemas matemáticos como de sistemas de equações lineares e não lineares, calcular derivadas e integrais, resolver equações diferenciais e equações algébricas, além de contribuir para o aprimoramento do aprendizado em matemática, etc.

> Os softwares educativos podem ser um notável auxiliar para o aluno adquirir conceitos em determinadas áreas do conhecimento, pois o conjunto de situações, procedimentos e representações simbólicas oferecidas por essas ferramentas é muito amplo e com um potencial que atende parte dos conteúdos das disciplinas. Estas ferramentas permitem auxiliar os alunos para que deem novos significados às tarefas de ensino e ao professor a oportunidade para planejar, de forma inovadora, as atividades que atendem aos objetivos de ensino (Bona, 2009, p. 36).

Portanto, o uso das ferramentas computacionais pode agregar no aprendizado da obtenção da solução de problemas matemáticos, desde que de forma planejada e com a metodologia adequada. Segundo Almeida (2016, p. 234), "as novidades tecnológicas e essa grande variedade de softwares educativos disponíveis na rede mundial de computadores podem contribuir de forma expressiva para facilitar o processo de ensino aprendizagem de Matemática". Para Lima *et al.* (2023), antes de ter algum domínio e entendimento sobre a ferramenta computacional, é importante ter conhecimento dos procedimentos necessários para obter uma solução manual do problema, para, depois, aprimorar os métodos computacionais no aprendizado matemático.

Existem diversos recursos tecnológicos e *softwares* que podem ser utilizados para contribuir no aprimoramento do aprendizado do aluno em Cálculo Numérico. Dentre eles, nessa pesquisa, será utilizado o *software SCILAB*, uma ferramenta gratuita e de fácil acesso que pode ser utilizada em diversas áreas das ciências, principalmente na Matemática.

2.1 Software SCILAB

O *SCILAB* é uma ferramenta desenvolvida para computação numérica, de código aberto, e pode ser utilizada para a resolução de problemas simples e complexos.

> O *Scilab* é um software livre de computação e programação numérica desenvolvido na França, em 1990, por pesquisadores do INRIA (Institut National de Recherche em Informatique et en Automatic) e do ENPC (École des Ponts ParisTech). Atualmente, o mesmo é mantido pelo *Scilab* Consortium. A aplicação é similar aos *softwares Matlab*, Octave e Gauss (GOMEZ, 1999). Atualmente o *Scilab* é muito utilizado no meio acadêmico como ferramenta para as mais diversas aplicações incluindo o ensino-aprendizagem de Cálculo Numérico (Amaral; Leite; Silva, 2013, p. 3).

Ainda segundo Amaral, Leite e Silva (2013, p. 3), o *software SCILAB*:

> É capaz de resolver problemas matemáticos de forma muito mais fácil do que utilizando as linguagens de programação *PASCAL*, *FORTRAN*

> ou C. Graças à grande variedade de funções, tem a capacidade de realizar cálculos relacionados à álgebra linear, processamentos de sinais, construção de gráficos em duas e três dimensões, entre outros.

Operar com o *SCILAB*, além de conhecimento em matemática, exige um pouco de conhecimento em computação, já que o *software* é implementado por códigos e comandos.

De acordo com Amaral, Leite e Silva (2013, p. 5), "o *software Scilab* exige conhecimento de programação, expressões aritméticas, comandos de entrada e saída de dados, elaboração de gráficos e estudo da linguagem de programação, para entender a construção de programas e funções dentro do *software Scilab*".

Além disso, o *SCILAB* permite definir funções matemáticas por meio de comandos computacionais, como, por exemplo, o comando *function*, para entrar com a função desejada ou, ainda, para definir o tipo de método que será utilizado. No entanto, ao digitar uma função no *SCILAB*, é necessário se atentar à sintaxe do *software*, como pode ser visto no Quadro 1, referente a algumas funções.

Quadro 1 – Algumas funções pré-definidas do *SCILAB*

Função no *SCILAB*	**Função**
abs(x)	Valor absoluto de x
sin(x)	Seno de x
asin(x)	Arco cujo seno é x
cos(x)	Cosseno de x
acos(x)	Arco cujo cosseno é x
tan(x)	Tangente de x
atan(x)	Arco cuja tangente é x
cotg(x)	Cotangente de x
exp(x)	Exponencial de x
log(x)	Logaritmo de x na base 2
sqrt(x)	Raiz quadrada de x
modulo(x,y)	Resto da divisão inteira de x por y
int(x.y)	Quociente da divisão de x por y

Fonte: Ferreira (2009, p. 9)

Dessa forma, o *software* permite um trabalho com diversos tipos de funções pré-definidas, fazendo com que seja atendida a finalidade do usuário.

2.2 Download do SCILAB

O *software SCILAB* pode ser obtido a partir do *download* na página de navegação da internet, em www.scilab.org (Figura 1).

Figura 1 – Site oficial do *SCILAB*.
Fonte*:* *Dassault Systemes* (*SCILAB*, 2023).

O próximo passo é clicar em *download* e escolher a versão e o sistema operacional. As opções disponíveis são *Windows*, *GNU/Linux* e *macOS*. Além da opção para *download* do *software*, o *SCILAB* também pode ser utilizado de forma *online*, no endereço https://cloud.scilab.in.

2.3 *Ambiente Scilab*

Após a instalação do *software*, a tela exibida será como na Figura 2.

Figura 2 – Ambiente do SCILAB.
Fonte: Autores (2024).

A tela inicial contém os seguintes elementos:

- Navegador de arquivos: permite navegar pelos arquivos disponíveis;
- Console: campo de execução para digitar os comandos;
- Navegador de variáveis: espaço que permite visualizar as variáveis salvas na memória;
- Histórico de comandos: lista os últimos comandos executados do console.

2.4. Alguns comandos do SCILAB

Dentre os comandos básicos disponíveis no *SCILAB* estão as operações de adição, subtração, entre outras, as quais podem ser visualizadas no Quadro 2.

Quadro 2 – Expressões aritméticas

Operação	**Operador**	**Exemplo**
Adição	+	a+b
Subtração	-	a-b
Multiplicação	*	a*b
Divisão	/	a/b
Potenciação	^	a^b

Além dos comandos básicos, o *SCILAB* permite criar variáveis para resolver diversos tipos de problemas, seja matemático ou não. As referidas variáveis são utilizadas como uma forma de armazenar informações temporárias na memória do computador, por exemplo, os números, os resultados de expressões ou os valores lógicos (Furtado, 2014). Porém, para criar variáveis, é preciso obedecer a algumas regras de sintaxe, conforme aponta Ferreira (2009):

a) Os nomes de variáveis começam com uma letra seguida de letras, algarismos ou sublinhados, como, por exemplo: notas, A1, B23 e matemática exata;
b) Não são permitidos acentos, espaços ou caracteres especiais (ç, !, ?, etc.), entretanto, o "_" é permitido; e,
c) Letras maiúsculas e minúsculas são diferentes; por exemplo, variável Beta é diferente de BETA e bEtA.

Ademais, de acordo com as regras acima, estes nomes de variáveis são inválidos: 4B, Nota [1], A/B e X@Z. Dessa forma, é necessário indicar qual a variável e qual a informação ela deverá armazenar. Um exemplo da aplicação das regras pode ser observado na Figura 3, em que a variável armazena o valor 1.

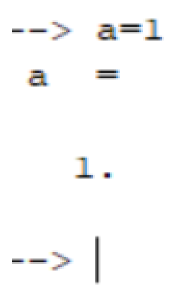


Figura 3 – Criação de variável a = 1
Fonte: Autores (2024)

Além disso, existem variáveis pré-definidas pelo *software* que não podem ser modificadas, como as descritas no Quadro 3.

Quadro 3 – Variáveis pré-definidas do *SCILAB*

SCI, WSCI	Contém o caminho raiz do SCILAB
SCIHOME	Caminho para preferência e histórico de arquivos de sua sessão
TMPDIR	Caminho do diretório temporário
Home	Caminho do diretório do usuário
%e	Número de Euler
%eps	Epsilon (relativo à precisão do ponto flutuante)
%f ou %F	Valor boleano para falso
%t ou %T	Valor boleano para verdadeiro
%i	Unidade imaginária
%inf	Infinito
%nan	Not-a-number – não é um número
%pi	Número PI
%s ou %z	Define polinômios

Fonte: Instituto Federal de Educação, Ciência e Tecnologia Catarinense, *campus* de Luzerna (2017).

O *software SCILAB* possui diversos comandos, por exemplo, as expressões lógicas que utilizadas para a comparação entre valores do mesmo tipo, o que pode ser visto no Quadro 4.

Quadro 4 – Operadores relacionais

Relação	**Operador**	**Uso**
Igual a	=	a = b
Diferente de	≠	a ≠ b
Maior que	>	a > b
Menor que	<	a < b
Maior ou igual	≥	a ≥ b
Menor ou igual	≤	a ≤ b

Fonte: Campos (2010, p. 19)

Além dos comandos de expressões lógicas, existem comandos básicos do *SCILAB* que podem ser utilizados, como o comando de limpar a tela do console (clc) e o comando para limpar as variáveis gravadas na memória (clear), entre outros, como disposto no Quadro 5.

Quadro 5 – Operadores básicos do *SCILAB*

Operador	**Definição**
Clc	Limpa a tela do console
Clear	Apaga os comandos gravados na memória do *software*
;	Faz com que não execute o comando criado na linha do código
:	Atribui intervalo à variável
//	Inclui comentários nas linhas dos comandos

Fonte: Autores (2024).

Neste sentido, o conhecimento dos comandos básicos pode facilitar na manipulação do *SCILAB*, como na execução de alguns comandos, a saber: entrar com a=2 e b=3, em seguida limpar o valor de b com o comando *clear* b (Figura 4). Na referida figura, o valor de x = 0:10 é dado quando x deve variar de 1 em 1, começando em 0 e terminando em 10. Outra possibilidade mostrada foi x = [0:5:20], em que o valor de x varia de 0 a 20, indo de 5 em 5 (Figura 4), além do comando clc para limpar as informações disponíveis na janela de comando (Figura 4).

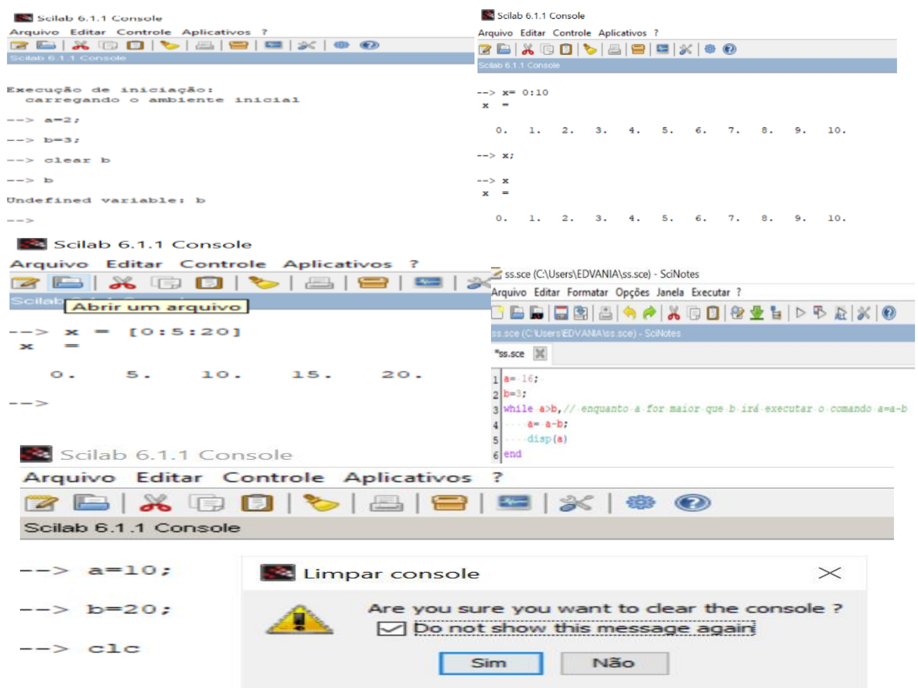


Figura 4 – Alguns comandos básicos do *SCILAB*
Fonte: Autores (2024).

O *software* permite, ainda, que sejam implementados comentários nas linhas dos comandos, indicados por // seguidas do comentário desejado (Figura 4). As duas barras indicam ao *SCILAB* que aquela informação não será executada, mas apenas um lembrete do significado contemplado naquela linha de comando.

Além dos comandos apresentados na Figura 4, existem muitos outros disponíveis no aplicativo. Um deles é utilizado para a construção de gráficos para representar uma função ou determinado problema, a exemplo de construir o gráfico de y = 2x + 2, com x variando de 1 em 1, iniciando em -4 e terminando em 15 (Figura 5). O comando para construção de gráficos bidimensionais é dado por *plot(x,y)*, em que deve ser informado o valor da variação no eixo x, a função e o comando para a construção do gráfico de duas dimensões.

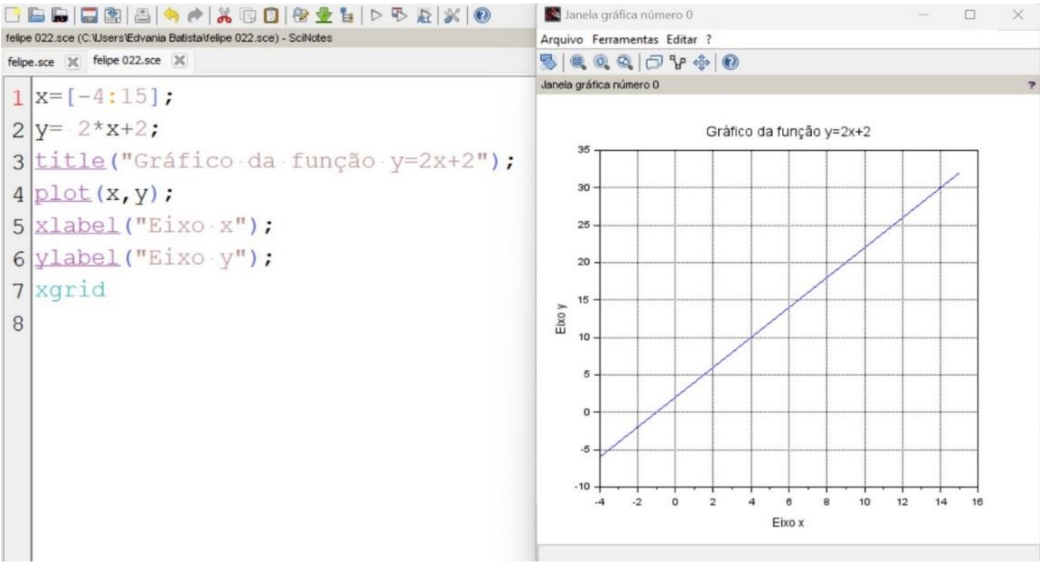


Figura 5 – Comandos para construção do gráfico da função f(x) = 2x+2
Fonte: Autores (2024).

Em que x= [-4:15] indica a variação dos valores de x de 1 em 1, de -4 até 15, e *title* ("Gráfico da função f(x)= 2x+2'') imprime a mensagem desejada como cabeçalho do gráfico. Para variações diferentes de 1 para os valores de x, basta indicar a variação desejada: assim, x= [-4:0.5:15] indica que x varia de -4 a 15 com passo 0,5. Finalmente, o comando *xgrid* quadricula a figura plotada no *SCILAB*.

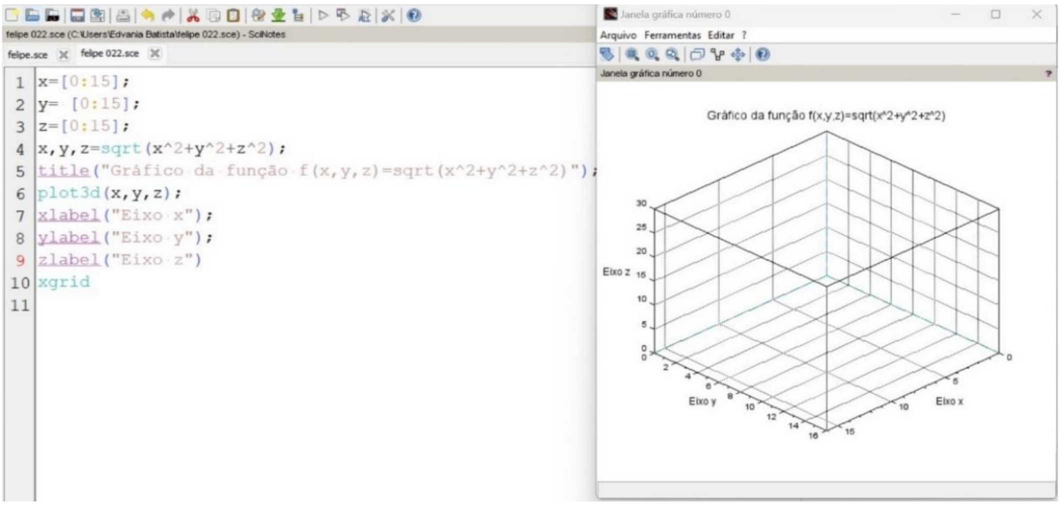


Figura 6 – Comandos para plotar o gráfico da função f(x,y,z) = $\sqrt{x^2 + y^2 + z^2}$
Fonte: Autores (2024).

A função f(x,y,z) = $\sqrt{x^2 + y^2 + z^2}$ é representada pelo comando *sqrt*, enquanto que *plot3d* (x,y,z) é o comando para apresentar o gráfico tridimensional (Figura 6).
Para construir um gráfico com dimensões diferentes, é necessário definir os vetores x e y com dimensões diferentes (ver Figura 7), em que x está no intervalo de 0 a 2* π, enquanto y varia de -2 a 2, com passo 0.1. Por sua vez, os comandos *x.label*, *y.label* e *z.label* dão nomes aos eixos x, y e z. Finalmente, o comando *scf ()* é utilizado para ajustar a janela da figura.

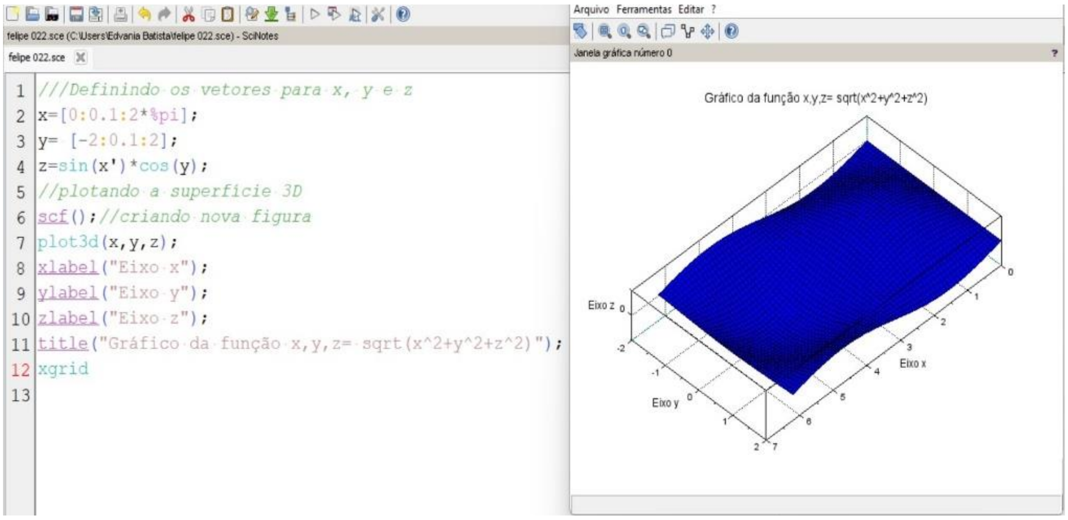


Figura 7 – Gráficos com dimensões diferentes
Fonte: Autores (2024).

2.5 *Comandos if, while e for*

Para implementar rotinas no *Scilab* no intuito de obter zeros de funções, são utilizados procedimentos iterativos. Dentre os principais comandos para estruturas lógicas estão os comandos condicionais *if*, *while* e *for* (Fontana, 2018).
"O comando *for* (para) é utilizado para criar laços ('*loops*') de uma dada operação, por um número definido de vezes. Este é um comando básico para trabalhar com vetores e matrizes, ele permite fazer com que uma dada variável varie entre um valor inicial e um valor final" (Fontana, 2018, p. 9). Um exemplo é dado na Figura 8, em que, na linha 1, é definida a variável i que varia de 1 a 5, ou seja, o "comando utilizado para executar uma dada ação se determinada condição for satisfeita" (Fontana, 2018, p. 10). Por sua vez, o comando *if* impõe uma condição a ser cumprida, no caso em questão, se i > 2, a resposta deverá ser o próprio valor de i, caso não seja satisfeita a condição, será executado o comando *else* (senão) de multiplicar o valor de i por dois.

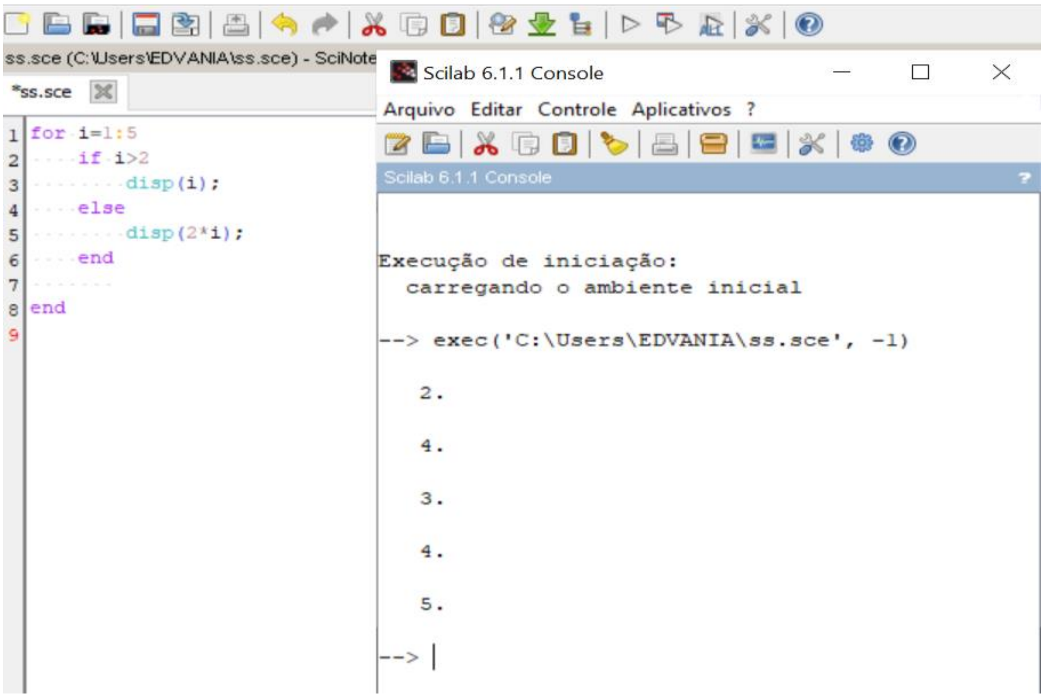


Figura 8 – Comandos *for*, *if* e *else* no *SCILAB*
Fonte: Autores (2024).

Por sua vez, o comando *while* (enquanto) indica "um condicional que repete uma dada operação enquanto um critério de parada não seja satisfeito" (Fontana, 2018, p. 10). Um exemplo pode ser observado na Figura 9, onde foram indicados os valores iniciais "a=16" e "b=3" e o comando *while* a > b e a = a - b.

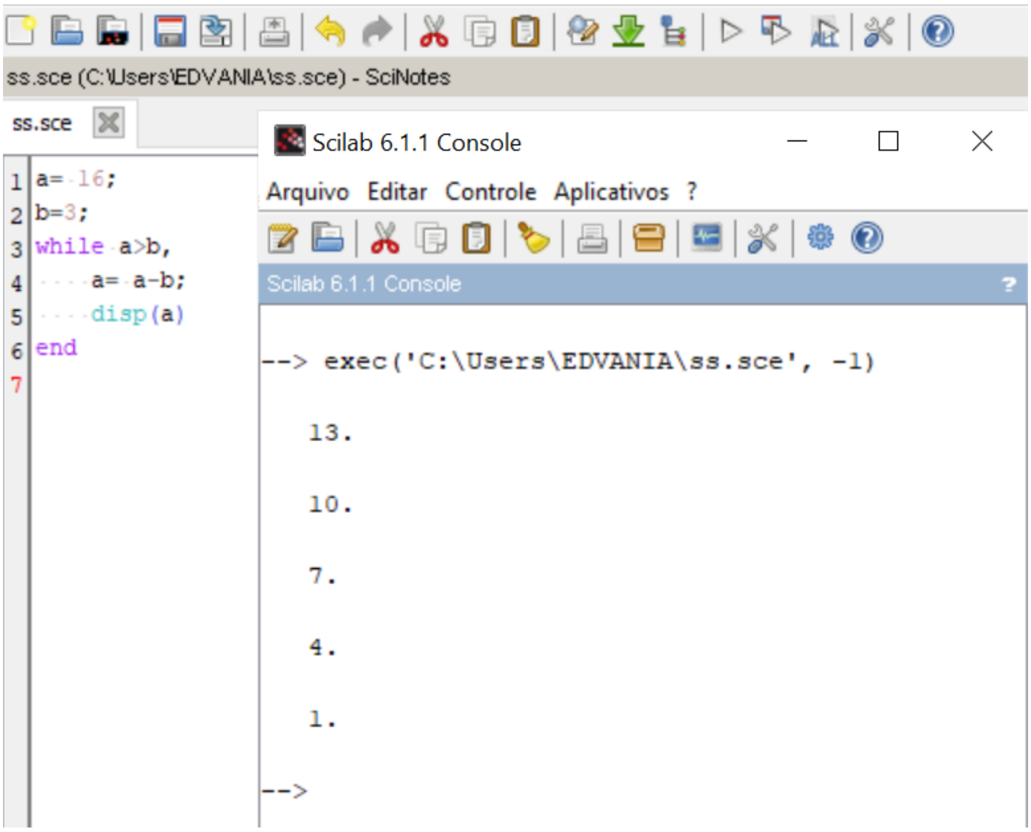


Figura 9 – Comando *while* do *SCILAB*
Fonte: Autores (2024).

O referido comando indica que, enquanto o valor de a for maior que o de b, o *Scilab* fará com que a receba a diferença entre os valores de a e b, ou seja, o valor de a irá variar de três em três até que seja menor que o valor inicial d e b. No entanto, quando o valor de a for menor do que o valor de b, o *SCILAB* irá interromper a execução dos comandos.

2.6 *Novas funções no software SCILAB*

Na biblioteca do *SCILAB* já existem algumas funções, como as funções modular, logarítmica, exponencial e trigonométrica. Contudo, ele permite a entrada de funções diferentes das constantes de sua biblioteca, por exemplo, a função y = a*tg(x) + 2. Para implementar a referida função no *SCILAB*, Figura 10, é necessário representar a função da seguinte forma: *function* [y] = f(x); em seguida, digitar valores para a e b além de y = (a*tan(x) + b).

A depender da situação, o *SCILAB* permite a opção de não por [] para o argumento de apenas uma saída. Mas, para dois ou mais argumentos de saída, deve-se utilizar [] como sintaxe.

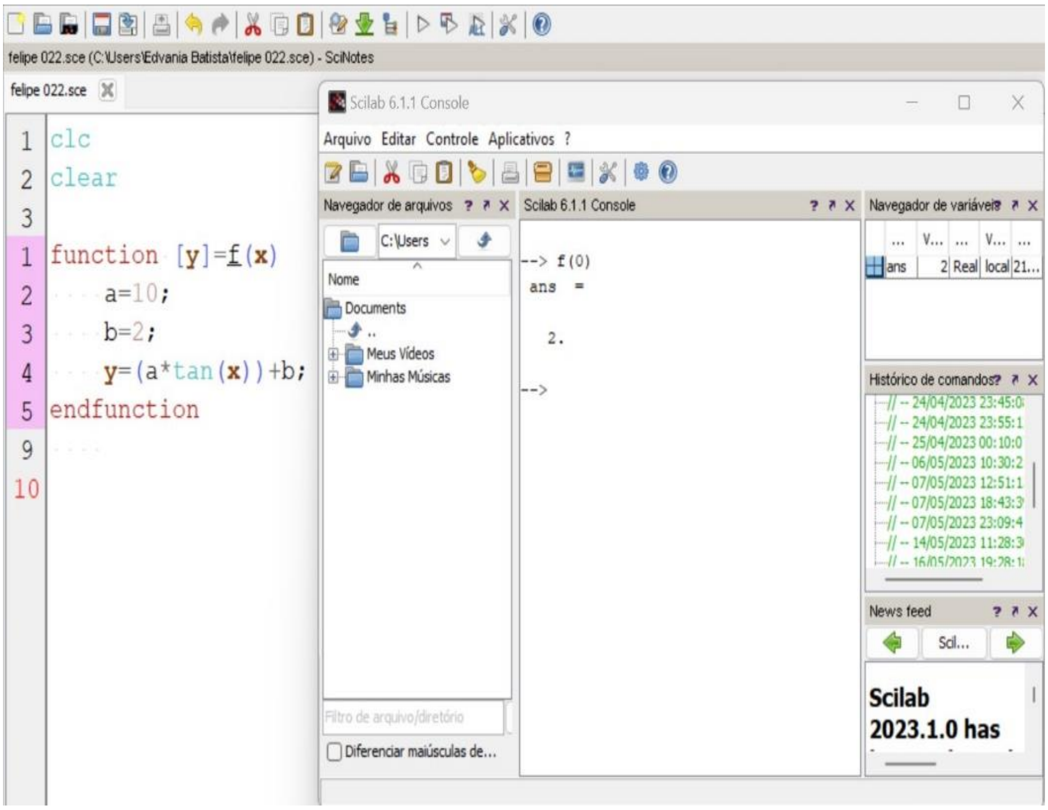


Figura 10 – Função no *SCILAB*
Fonte: Autores (2024).

2.7 *Zero de uma função*

Em matemática, função é uma relação de elementos de um conjunto X, denominado domínio, com elementos do subconjunto Y, denominado contradomínio. Essa relação é denotada por $f\colon X \to Y$, em que f é expressa por $y = f(x)$. Segundo Flemming e Gonçalves (2006), dados os conjuntos X com elementos x e Y com elementos y, diz-se que f de X e Y é uma relação que relaciona cada elemento x (denominado termo independente) pertencente a X, a um único elemento y (denominado termo dependente), em que $y = f(x)$ pertence a Y. As referidas funções, de acordo com suas especificidades, podem ser classificadas como: função afim (ou função polinomial do 1° grau), função quadrática (ou função polinomial do 2° grau), função trigonométrica, função modular, função exponencial, função logarítmica, entre outras.

Os valores do termo independente x de uma função que tem por imagem o termo dependente igual a zero, ou seja, $y = f(x) = 0$, é denominado zero ou raiz da função. O zero ou raiz da função é o valor que representa o local onde o gráfico da função intercepta o eixo x do plano cartesiano. A Figura 11 ilustra os zeros da função $f(x) = x^2 - 5x + 6$, ou seja, quando $x = 2$ ou $x = 3$ tem-se $f(x) = 0$, locais onde a função corta o eixo x.

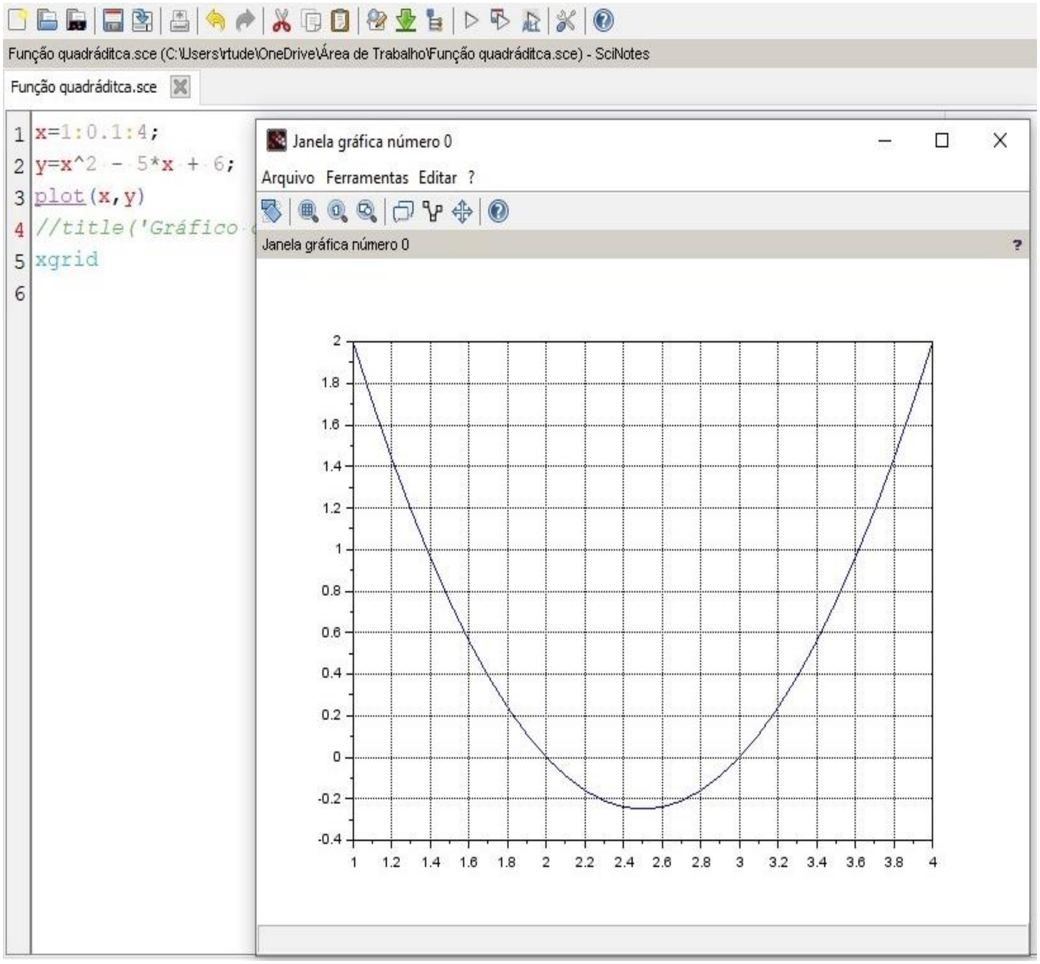


Figura 11 – Zeros da função $f(x) = x^2 - 5x + 6$ são $x = 2$ e $x = 3$
Fonte: Autores (2024).

No trabalho com Cálculo Numérico, além de outras ações, o que se busca é a solução (raiz) aproximada de uma função ou equação. Inicialmente, para calcular uma raiz aproximada, duas etapas devem ser seguidas, a saber:

1º) Isolar a raiz da equação $f(x) = 0$ em um intervalo $[a, b]$, o menor possível. Porém, para garantir a existência de pelo menos uma raiz no intervalo $[a, b]$, deve-se observar o seguinte resultado que diz: "se f for contínua no intervalo $[a, b]$ e $f(a)$ e $f(b)$ tiverem sinais opostos, então existe pelo menos um número c entre a e b tal que $f(c) = 0$" (Flemming; Gonçalves, 2006, p. 111), um caso particular do Teorema do Valor Intermediário.

Apesar de o Teorema do Valor Intermediário garantir a existência de pelo menos uma raiz no intervalo $[a, b]$, é necessário observar se no intervalo $[a, b]$ existe uma única raiz, o que pode ser visto na afirmação de Justo *et al.* (2020, p. 50) que indica que: "se f: $[a, b] \rightarrow \mathbb{R}$ é um função diferençável, $f(a) \cdot f(b) < 0$ e $f'(x) > 0$ (ou $f'(x) < 0$) para todo x $\in (a, b)$, então existe um único $x^* \in (a, b)$ tal que $f(x^*) = 0$".

2º) Refinar o intervalo para encontrar o valor da raiz aproximada, isto é, até o grau de exatidão desejada.

O processo para a determinação do intervalo que contenha uma raiz da função é iniciado com a atribuição de valores arbitrários a x, que, substituídos na função, obtêm os valores de y correspondentes até que ocorra a troca de sinal de y. Os dois valores de x que levam a troca do sinal de y pertencem aos extremos do intervalo $[a, b]$ de existência da raiz. Veja o exemplo para $f(x) = x^3 - 3x - 1$ que foram atribuídos os valores de x igual a 0, 1 e 2.

Exemplo 1: isolamento da raiz positiva de $f(x) = x^3 - 3x - 1$.

X	Y
0	-1
1	-3
2	1

Observe que quando $x = 1$ o valor de $y = -3$ e que para $x = 2$ o valor de $y = 1$, ou seja, $f(1) \cdot f(2) = -3 \cdot 1 = -3 < 0$, indica que a função corta o eixo das abscissas entre 1 e 2. Logo, existe uma raiz de $f(x)$ no intervalo (1, 2).

Um esboço do gráfico pode ser visto na Figura 12, que mostra as três raízes de $f(x)$.

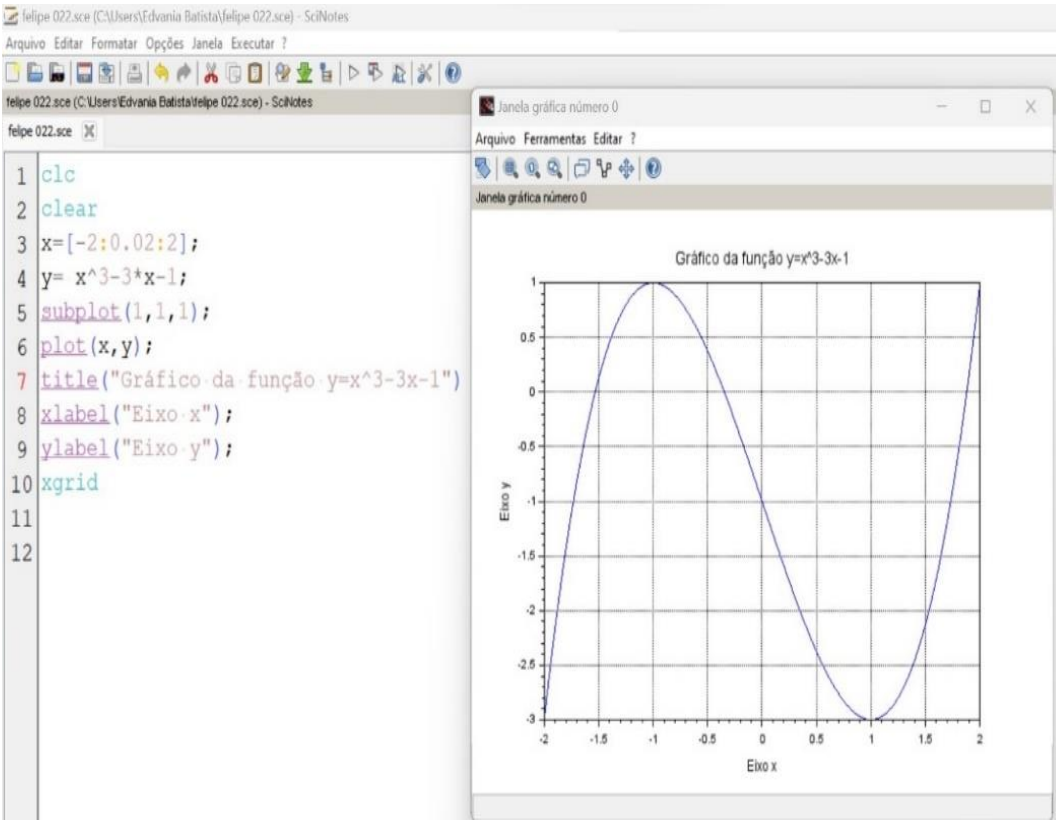


Figura 12 – Raiz da função $f(x) = x^3 - 3x - 1$.
Fonte: Autores (2024).

A presente pesquisa foi realizada pelo método qualitativo bibliográfico que, segundo Minayo (2001), responde e se preocupa com um nível de problemas que não podem ser quantificados. O referido método trabalha com o universo abrangente das nossas relações e com o universo dos fenômenos que não se deve reduzir a operações de variáveis.

Nessa perspectiva e para obter uma solução (raiz aproximada) do problema a ser analisado, com a precisão desejada, foram implementados códigos no *software* livre *SCILAB,* do método bissecção, além da função $f(x) = e^{-x} + x^2 + ln(x) - sin\,(x)$, da seguinte forma:

a) Determinar um intervalo $[a, b]$ em que exista uma raiz ε da função do problema;

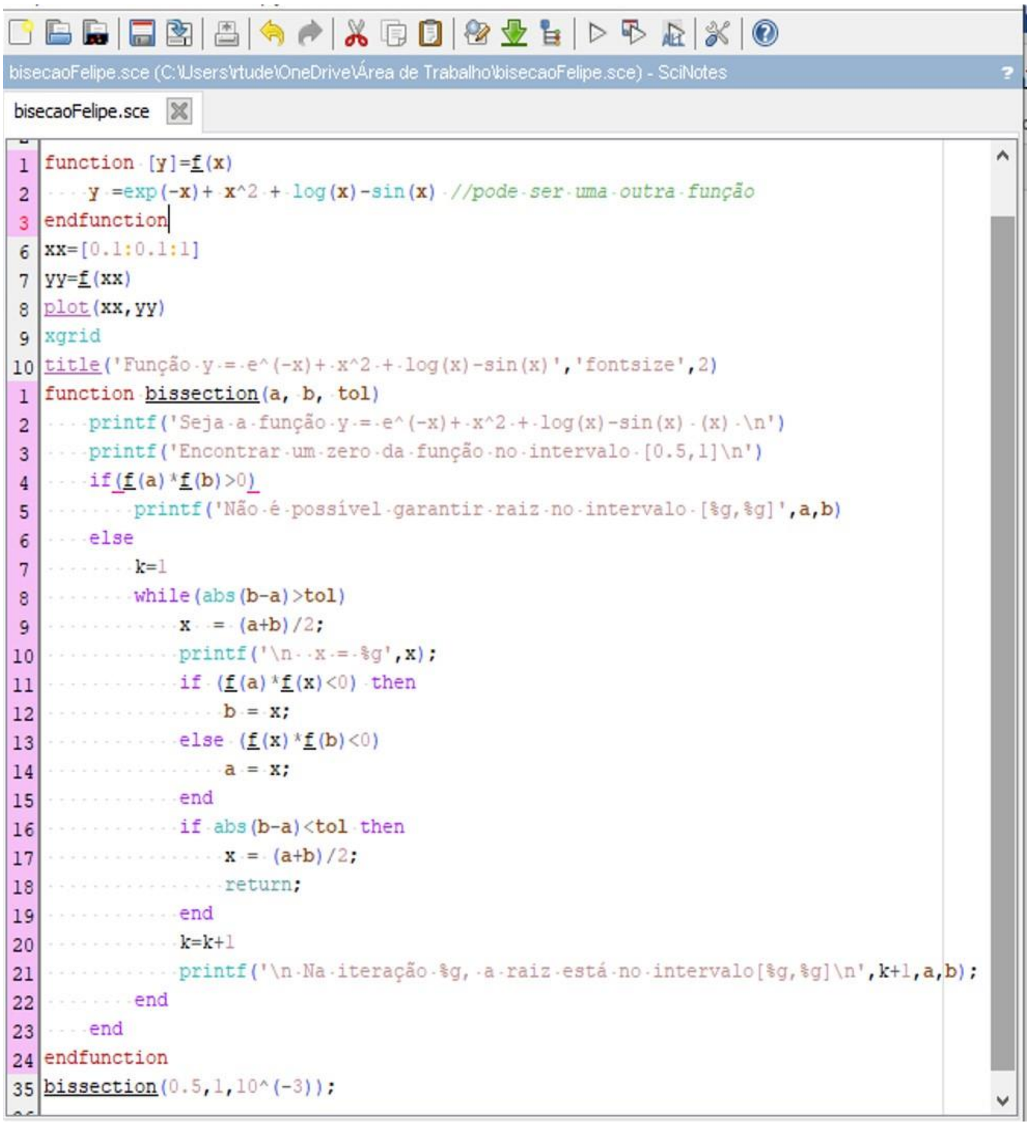
b) Reduzir o intervalo $[a, b]$ da raiz, com o seguinte procedimento: $x = \frac{a+b}{2}$, média dos extremos do intervalo;
c) Verificar se $f(x) = 0$, ou seja, se x é a solução da função dada, fazer $\varepsilon \simeq x$ e encerrar as operações, uma vez que se encontrou a solução;
d) Se $f(x) \neq 0$, determinar o novo intervalo que contenha a raiz, sendo $[a, x]$ ou $[x, b]$;
e) O intervalo novo, que contenha a raiz, é obtido pelo procedimento: se $f(a)*f(x) < 0$ então $\varepsilon \in [a, x]$, caso contrário, $\varepsilon \in [x, b]$; e,
f) Caso o módulo de $(b - a)$ seja menor que a precisão do problema, o procedimento será encerrado e o resultado será x. Caso contrário, repete-se a bipartição do intervalo e os procedimentos de "a" a "f", denominados iteração, até que se obtenha a raiz procurada.

3. Resultados e Discussão

O cálculo numérico possui vários métodos iterativos utilizados para obter o zero de uma função, como, por exemplo, Bissecção, Newton, Secante, entre outros. No entanto, aqui foram ilustradas as fases para a solução de um problema com o uso do método bissecção.

Bissecção é o nome dado ao método em que, a partir da definição de um intervalo que contenha uma raiz da função, é aplicada a técnica de divisão do intervalo ao meio até que se obtenha a raiz com a precisão estabelecida. A depender da complexidade da função, ela pode ser resolvida de forma manual, se o problema for de baixa complexidade, ou computacional, se o problema for de uma complexidade mais elevada.

Aqui foram ilustrados os procedimentos para encontrar o zero de uma função que não possui solução analítica, a saber, $f(x) = e^{-x} + x^2 + \ln(x) - \sin(x)$, com precisão de 10-3, com o método bissecção, resolvida com o uso do software SCILAB. Foi escolhido o intervalo $[1/2, 1]$, que contém uma raiz positiva, mas foi necessário garantir um único zero de f no intervalo $[\frac{1}{2}, 1]$. Em seguida, foram implementados os códigos do método bisseção no SCILAB, (Figura 13) para a solução completa da função; e, então, apresentada a tela com as iterações (resultados) da solução da referida função (Figura 14).




```
function [y]=f(x)
    y =exp(-x)+ x^2 + log(x)-sin(x) //pode ser uma outra função
endfunction
xx=[0.1:0.1:1]
yy=f(xx)
plot(xx,yy)
xgrid
title('Função y = e^(-x)+ x^2 + log(x)-sin(x)','fontsize',2)
function bissection(a, b, tol)
    printf('Seja a função y = e^(-x)+ x^2 + log(x)-sin(x) (x) \n')
    printf('Encontrar um zero da função no intervalo [0.5,1]\n')
    if(f(a)*f(b)>0)
        printf('Não é possível garantir raiz no intervalo [%g,%g]',a,b)
    else
        k=1
        while(abs(b-a)>tol)
            x  = (a+b)/2;
            printf('\n  x = %g',x);
            if (f(a)*f(x)<0) then
                b = x;
            else (f(x)*f(b)<0)
                a = x;
            end
            if abs(b-a)<tol then
                x = (a+b)/2;
                return;
            end
            k=k+1
            printf('\n Na iteração %g, a raiz está no intervalo[%g,%g]\n',k+1,a,b);
        end
    end
endfunction
bissection(0.5,1,10^(-3));
```

Figura 13. *Script* com os códigos do *Scilab* de $f(x) = e^{-x} + x^2 + ln(x) - sin(x)$, no intervalo [0.5, 1], com precisão de 10^{-3}
Fonte: Autores (2024).

Function [y] = f(x) indica que y recebe f(x), o comando $y = \%e\hat{}(-x) + x^2 + \log(x) - \sin(x)$ representa a função dada no problema, enquanto o comando endfunction encerra o comando function [y] = f(x). Observa-se que após encerrar a função [y] = f(x), inicia-se a escrita dos códigos da bissecção, function bissection (a, b, tol), onde: bissection indica o método iterativo escolhido, a e b são os extremos do intervalo que contêm uma raiz e tol é a precisão desejada para a aproximação da raiz. Já o comando if que significa se, indica uma condição, exemplo: se f(a) * f(b) > 0, nada pode ser feito, uma vez que em [a, b] não é possível afirmar se existe uma raiz da função $y = e^{-x} + x^2 + \ln(x) - \sin(x)$. No entanto, se a condição de f(a) * f(b) > 0 não for obedecida, é dado o comando else, que significa "se não", ou seja, f(a)* f(b) < 0, que indica a existência de, pelo menos, uma raiz no intervalo [a, b].

O comando *printf* é utilizado para imprimir a informação que o usuário deseja que apareça no console do *Scilab* e *while* (*abs*(b-a)>*tol*) significa que enquanto o valor absoluto da diferença entre os extremos do intervalo for maior que a tolerância, terá que continuar com os procedimentos iterativos. Já o contador de iteração é iniciado com k = 1, onde k = 1, 2, 3..., ou seja, k = k+1, enquanto o comando x = (a+b)/2 faz a média do intervalo. A escolha do novo intervalo é feita com a verificação de que: se f(a) * f(x) < 0, então b recebe o valor de x, b = x, ou seja, o novo intervalo será [a, x], se não f(x) * f(b) <0, então a recebe o valor de x, a = x, e o novo intervalo será [x, b].

Após a escolha do novo intervalo, se o problema tiver alcançado a raiz com a precisão desejada, o programa será encerrado e apresentará o resultado da raiz aproximada; se não, o programa retornará para o comando *while* e iniciará a próxima iteração. O comando *end* é dado para encerrar os comandos *else*, *for*, *if* e *while*, sendo necessário um *end* para cada um. Por fim, a função é executada com o comando *bissection* (0.5, 1, 10^(-3)), em que 0.5 e 1 são os extremos do intervalo [a, b] e 10^(-3) é a tolerância (precisão), definida pelo usuário.

Após a execução dos comandos anteriores, o *Scilab* será executado e os resultados apresentados no console (Figura 14), que mostra o resultado de cada iteração realizada do problema de encontrar uma raiz de $f(x) = e^{-x} + x^2 + ln(x) - sin\,(x)$, com precisão de 0,001, no intervalo (0.5, 1). Sendo assim, ele mostra que foram necessárias 12 iterações para chegar à raiz aproximada no intervalo [0.5, 1], que é 0.708984.

A solução de $f(x)$ também poderia ser realizada de forma manual, em que as diferenças entre o resultado computacional e o manual estão na rapidez e na melhor precisão da solução obtida pelo *software*. Além do mais, a execução manual exige que sejam repetidos todos os procedimentos iterativos, um a um, para cada iteração, até encontrar a solução do problema, enquanto no *SCILAB* basta trocar a função $f(x)$, o intervalo e a tolerância, pois o restante dos procedimentos serão os mesmos já dispostos no *script* apresentado no problema de encontrar uma raiz de $f(x) = e^{-x} + x^2 + ln(x) - sin\,(x)$.

```
Scilab 6.1.1 Console
Seja a função y = e^(-x)+ x^2 + log(x)-sin(x) (x)
Encontrar um zero da função no intervalo [0.5,1]

 x = 0.75
 Na iteração 2, a raiz está no intervalo[0.5,0.75]

 x = 0.625
 Na iteração 3, a raiz está no intervalo[0.625,0.75]

 x = 0.6875
 Na iteração 4, a raiz está no intervalo[0.6875,0.75]

 x = 0.71875
 Na iteração 5, a raiz está no intervalo[0.6875,0.71875]

 x = 0.703125
 Na iteração 6, a raiz está no intervalo[0.703125,0.71875]

 x = 0.710938
 Na iteração 7, a raiz está no intervalo[0.703125,0.710938]

 x = 0.707031
 Na iteração 8, a raiz está no intervalo[0.707031,0.710938]

 x = 0.708984
 Na iteração 9, a raiz está no intervalo[0.708984,0.710938]

 x = 0.709961
```

Figura 14 – Resultado da execução de $f(x) = e^{-x} + x^2 + ln(x) - sin\,(x)$ no *SCILAB*
Fonte: Autores (2024).

O resultado da raiz aproximada encontrado ao final das iterações (Figura 14) é condizente com o ilustrado no gráfico da Figura 15, que mostra a raiz da função entre 0.6 e 0.8.

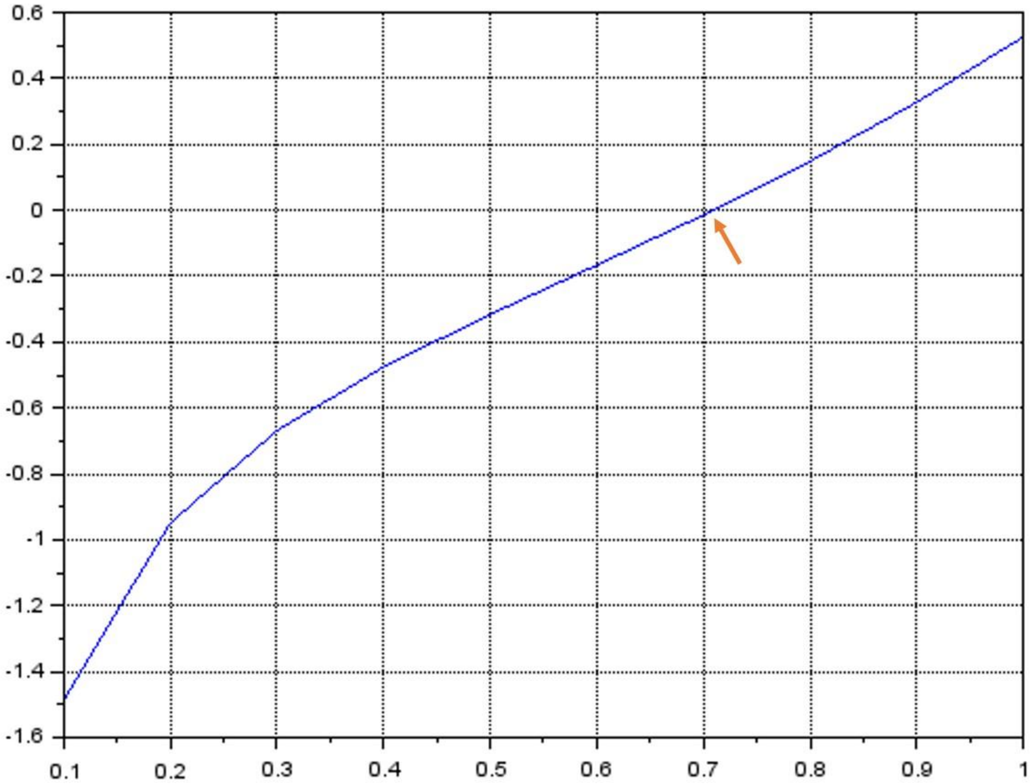


Figura 15 – Raiz (≅0.708984) da função $f(x) = e^{-x} + x^2 + ln(x) - sin\,(x)$
Fonte: Autores (2024).

O resultado das iterações para a solução de $f(x)$ mostrou boa precisão do *software SCILAB* para a resolução do problema em questão e poderá ser utilizado tanto para problemas simples quanto para problemas mais complexos, em especial para os que não possuem solução analítica. O uso da resolução de problemas de forma computacional pode resultar em economia de tempo, uma vez que não é necessário escrever um novo *script* para cada problema, mas apenas trocar a função desejada, o intervalo e a precisão para resolver o novo problema.

4. Conclusão

O Cálculo Numérico é uma área da matemática que utiliza métodos iterativos para encontrar o zero de funções cuja solução não é analítica. Dentre os métodos iterativos, estão a Bissecção, Newton e Secante.

No presente trabalho foi utilizado o método bissecção com o objetivo de obter a raiz aproximada de um problema de solução não analítica, utilizando o *software SCILAB*. Bissecção é o nome dado ao método em que, a partir da definição de um intervalo que contenha uma raiz da função, é aplicada a técnica de dividir o intervalo ao meio até que se obtenha a raiz com a precisão estabelecida.

Os resultados mostraram que o *software* foi eficaz para obter a solução do problema proposto e poderá ser utilizado tanto para problemas simples quanto para problemas mais complexos, em especial para os que não possuem solução analítica. Outro detalhe a se destacar em relação à solução via programa computacional, foi a eficiência e a rapidez para encontrar a solução do problema, bem como na possibilidade de apenas substituir a função desejada, o intervalo e a precisão no *script* do programa para resolver um novo problema.

A pesquisa também evidenciou a importância de se conhecer tanto o *software* quanto os métodos numéricos a serem aplicados no *SCILAB*.

Ademais, é importante ressaltar que a utilização do *SCILAB* não se limita à aplicação do método bissecção. Ele pode ser utilizado em diversos outros métodos iterativos para a resolução de problemas matemáticos mais complexos. Portanto, sugere-se a continuidade das pesquisas nessa área, explorando as diversas possibilidades que a ferramenta oferece.

Por fim, o presente estudo poderá contribuir para uma reflexão sobre a importância das tecnologias na educação, assim como para o aprimoramento do ensino de Cálculo Numérico como uma alternativa no processo de ensino-aprendizagem.

Agradecimentos: Os autores agradecem ao Departamento de Matemática e Estatística da Universidade Federal de Rondônia - UNIR por possibilitar o desenvolvimento desta pesquisa.

Conflitos de interesse: Os autores declaram não haver conflito de interesses.

Referência bibliográfica

ALMEIDA, H. R. F. L. (2016). *Das tecnologias às tecnologias digitais e seu uso na educação matemática. Nuances: Estudos sobre Educação*, Presidente Prudente, v. 26, n. 2, p. 224-240.

AMARAL, T. R.; LEITE, G. N., & SILVA, O. A. O ensino do cálculo numérico utilizando o SCILAB. In: Congresso Internacional do Ensino da Matemática, VI, 16, 17 e 18, outubro, Canoas, 2013. Anais... Canoas, 2013.

BONA, B. O. (2009). Análise de softwares educativos para o ensino de matemática nos anos iniciais do ensino fundamental. Experiências em Ensino de Ciências, Cuiabá, v. 4, n. 1, p. 35-55. Disponível em: https://fisica.ufmt.br/eenciojs/index.php/eenci/article/view/300. Acesso em: 23 fev. 2023.

CAMPOS, F. F. Fundamentos de SCILAB. Belo Horizonte: Universidade Federal de Minas Gerais/Departamento de Ciência da Computação do ICEx, 2010. Disponível em: https://www.ime.unicamp.br/~encpos/VIII_EnCPos/Apostila_Scilab.pdf. Acesso em: 07 abr. 2023.

FERREIRA, J. A. T. O uso do Scilab no Cálculo Numérico. Ouro Preto: Instituto de Ciências Exatas e Biológicas/Departamento de Computação, 2009. Disponível em: https://www.cin.ufpe.br/~rrbs/Favip/Materiais%20SciLab/scilab_notas%20de%20aulas.pdf. Acesso em: 22 mar. 2023.

FLEMMING, D. M., & GONÇALVES, M. B. (2006). *Cálculo A*: funções, limites, derivação e integração. 6. ed. Florianópolis: Pearson, p. 111.

FONTANA, E. Breve Introdução à Programação em Scilab 6.0. Curitiba: Universidade Federal do Paraná/Departamento de Engenharia Química, 2018. Disponível em: https://fontana.paginas.ufsc.br/files/2017/02/intro_scilab.pdf. Acesso em: 07 abr. 2023.

FURTADO, D. A. Introdução ao SCILAB e à Programação de Computadores Cursos de Engenharia. Uberlândia: UFU, 2014. Disponível em: http://www.furtado.prof.ufu.br/site/teaching/FC/FC_Introdu%C3%83%C2%A7%C3%83%C2%A3oScilab_FURTADO.pdf. Acesso em: 07 abr. 2023.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA CATARINENSE – campus luzerna. Apostila de lógica aplicado a Scilab. 2017. Disponível em: https://professor.luzerna.ifc.edu.br/marcelo-cendron/wp-content/uploads/sites/40/2017/02/Apostila-Conceitos-de-Programa%C3%A7%C3%A3o-Scilab.pdf. Acesso em: 13 abr. 2023.

JUSTO, D. A. R.; SAUTER, S.; AZEVEDO, F. S.; GUIDI, L. F., & KONZEN, p. H. A. (2020). *Cálculo Numérico*: um livro colaborativo. [S.I.: s.n.], p. 50.

LIMA, J. A.; SANTOS, R. T.; REIS, E. S., & CARDOSO, F. L. Algumas contribuições da planilha eletrônica do Software Libre Office no ensino do Cálculo Numérico. RECIMA21 – Revista Científica Multidisciplinar, v. 4, n. 4, p. e443002, 2023. https://doi.org/10.47820/recima21.v4i4.3002

MINAYO, M. C. S. (2001). *Pesquisa social*. Teoria, método e criatividade. 18. ed. Petrópolis: Vozes, p. 79-82.

SOBRAL, J. B. M. Dos Primórdios da Matemática aos Sistemas Formais da Computação. 1. ed. Florianópolis: Edição dAutores (2024), 2015. Disponível em https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/157316/teste-livro-1vFINAL.pdf?sequence=1&isAllowed=y. Acesso em: 15 de abr. 2023.