

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MARY ESTELLA DALA COSTA PAGNO

**MÉTODO DAS DIFERENÇAS FINITAS NA SOLUÇÃO DE EQUAÇÕES
DIFERENCIAIS ORDINÁRIAS COM O AUXÍLIO DO SOFTWARE OCTAVE**

PATO BRANCO

2024

MARY ESTELLA DALA COSTA PAGNO

**MÉTODO DAS DIFERENÇAS FINITAS NA SOLUÇÃO DE EQUAÇÕES
DIFERENCIAIS ORDINÁRIAS COM O AUXÍLIO DO SOFTWARE OCTAVE**

**Finite difference method in solving ordinary differential equations with the
aid of Octave software**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Licenciado em Matemática do Curso
de Licenciatura em Matemática da Universidade
Tecnológica Federal do Paraná.

Orientador: Professor Me. Vanderlei Martins

PATO BRANCO

2024



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

MARY ESTELLA DALA COSTA PAGNO

**MÉTODO DAS DIFERENÇAS FINITAS NA SOLUÇÃO DE EQUAÇÕES
DIFERENCIAIS ORDINÁRIAS COM O AUXÍLIO DO SOFTWARE OCTAVE**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Licenciado em Matemática do Curso
de Licenciatura em Matemática da Universidade
Tecnológica Federal do Paraná.

Data de aprovação: 24/junho/2024

Vanderlei Martins
Mestre em Matemática e Computação Científica
Universidade Tecnológica Federal do Paraná

Gilberto Souto
Mestre em Matemática e Computação Científica
Universidade Tecnológica Federal do Paraná

Aureo Quintas Garcia
Mestre em Engenharia Mecânica
Universidade Tecnológica Federal do Paraná

PATO BRANCO
2024

AGRADECIMENTOS

Primeiramente quero agradecer a Deus por ter me proporcionado mais essa conquista que encaminha o encerramento de mais um ciclo da minha vida.

Agradeço aos meus pais Diane Dala Costa Pago e Gilvan Fabio Pagno, as minhas irmãs Monike Dala Costa Pagno e Nayla Dala Costa Pagno e a William Cittadin pelo incentivo e todo o apoio que me deram ao longo de toda essa jornada.

Agradeço a todos os professores que fizeram parte dessa jornada de aprendizado. Em especial quero agradecer ao Professor Me. Vanderlei Martins, que está comigo desde o início desse projeto. Sou grata por toda a ajuda que recebi ao longo do desenvolvimento da pesquisa, pela paciência e pelos aprendizados que adquiri.

Quero agradecer também aos professores que se disponibilizaram a participar da banca desse projeto, o Professor Me. Gilberto Souto e o Professor Me. Aureo Quintas Garcia.

Por fim quero agradecer a UTFPR por me dar essa oportunidade e ajudar na minha formação profissional.

RESUMO

Este trabalho de conclusão de curso traz como tema central os métodos numéricos. O principal objetivo foi estudar o método das diferenças finitas, implementado no software Octave para resolver equações diferenciais ordinárias. Inicialmente são abordados como pré-requisitos os sistemas de equações lineares e a série de Taylor. Na sequência, o método das diferenças finitas é abordado. Por fim, são resolvidos dois exemplos com o auxílio do software Octave usando dois algoritmos diferentes na resolução e observando o tempo gasto com cada um.

Palavras-chave: método das diferenças finitas; métodos numéricos; octave.

ABSTRACT

This course conclusion work has numerical methods as its central theme. The main objective was to study the finite difference method, implemented in the Octave software to solve ordinary differential equations. Initially, systems of linear equations and the Taylor series are discussed as prerequisites. Next, the finite difference method is discussed. Finally, two examples are solved with the help of Octave software using two different algorithms in the resolution and observing the time spent with each one.

Keywords: finite difference method; numerical methods; octave.

LISTA DE FIGURAS

Figura 1 – Gráfico exemplo 1	29
Figura 2 – Gráfico exemplo 2 para $n = 4$	33
Figura 3 – Gráfico exemplo 2 para $n = 200$	33

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Inserindo os dados do sistema no Octave	11
Listagem 2 – Triangularizando o sistema pelo método de eliminação de Gauss. . .	15
Listagem 3 – Resolução do sistema triangularizado	17
Listagem 4 – Resolução de sistema tridiagonal	19
Listagem 5 – Inserindo o exemplo 1 no Octave	27
Listagem 6 – Código completo para resolver o exemplo 1 pelo método de Gauss .	28
Listagem 7 – Inserindo o exemplo 2 no Octave	32
Listagem 8 – Resolução do exemplo 2 pelo método de Gauss	34
Listagem 9 – Inserindo o exemplo 2 no Octave para resolver pelo código de siste- mas tridiagonais	35
Listagem 10 – Resolução do exemplo 2 pelo código para sistemas tridiagonais . .	36

SUMÁRIO

1	INTRODUÇÃO	8
2	PRÉ-REQUISITOS	10
2.1	Sistemas de equações lineares	10
2.1.1	Método de Eliminação Gaussiana	11
2.1.2	Sistemas triangulares	15
2.1.3	Sistemas tridiagonais	17
2.2	Série de Taylor	19
2.3	Equações diferenciais ordinárias	21
3	MÉTODO	23
3.1	Método das diferenças finitas	23
4	EXEMPLOS E RESULTADOS	25
4.1	Exemplo 1	25
4.2	Exemplo 2	29
5	CONCLUSÃO	37
	REFERÊNCIAS	38

1 INTRODUÇÃO

Os modelos matemáticos estão presentes e são fundamentais nas mais diversas áreas da ciência como na física, na química e principalmente nas engenharias. Eles permitem retratar e entender muitos fenômenos complexos de forma mais precisa, como as equações de Newton que retratam o movimento dos corpos, ou as equações que descrevem as taxas químicas, são exemplos disso.

Contudo, quando se trata da solução desses problemas nem sempre essa pode ser obtida de maneira exata, principalmente quando se trabalha com sistemas mais complexos ou problemas não lineares, ou quando as soluções analíticas não são possíveis ou difíceis de se obter. Nesses cenários que entram os métodos numéricos, pois eles trazem soluções aproximadas por meio de algoritmos e cálculos computacionais. Eles são utilizados para solucionar problemas envolvendo equações diferenciais, integração numérica, sistemas de equações lineares entre outros.

Os métodos numéricos vão trabalhar com a discretização dos problemas contínuos. Em seguida algoritmos são aplicados a partir de softwares para se obter soluções numéricas aproximadas. Apesar de serem soluções aproximadas, são muito significativas devido à sua capacidade de lidar com problemas complexos e, com a disponibilidade cada vez maior de computadores de alta performance, permite a aplicação de algoritmos mais avançados e obter resultado mais precisos.

Dentre os métodos numéricos existe o método das diferenças finitas, que é amplamente utilizado em várias áreas e, assim como outros métodos, oferece uma maneira eficaz de obter soluções aproximadas para problemas complexos.

Este trabalho irá se concentrar na aplicação do método das diferenças finitas para buscar soluções de equações diferenciais ordinárias em diferentes cenários, implementando-o com o uso do software Octave, explorando algoritmos diferentes para resolver essas equações, analisando a eficiência de cada abordagem. Para essa análise será levado em conta principalmente o tempo de execução do algoritmo para resolver o mesmo problema mas com abordagens diferentes, para entender melhor as vantagens de se usar determinado algoritmo.

Assim esse trabalho se justifica pelo interesse em entender como o método das diferenças finitas pode ser aplicado para resolver equações diferenciais ordinárias para auxiliar e facilitar a resolução de problemas, e quais as vantagens que essa aplicação gera.

O objetivo principal desse projeto é a implementação do método das diferenças finitas para resolver numericamente algumas equações diferenciais ordinárias.

Como objetivos específicos temos:

- Estudar o método das diferenças finitas.
- Implementar o método da eliminação gaussiana para a resolução de sistemas de equações lineares.

- Implementar o método das diferenças finitas no software GNU Octave para resolver numericamente algumas equações diferenciais ordinárias.

2 PRÉ-REQUISITOS

2.1 Sistemas de equações lineares

Esta seção está baseada em Cunha (2000). Informações mais detalhadas sobre o assunto podem ser obtidas em Kühlkamp (2011) ou em livros de Álgebra Linear, como por exemplo Anton e Rorres (2012).

Os sistemas de equações lineares são de extrema importância na matemática pois uma grande quantidade de problemas convertem-se em sistemas de equações lineares em sua resolução.

Um sistema de n equações lineares com n incógnitas é representado por:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}.$$

Os coeficientes das incógnitas x_i formam uma matriz quadrada \mathbf{A} com n linhas e n colunas:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}.$$

Definindo os vetores coluna $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$ e $\mathbf{b} = [b_1 \ b_2 \ \cdots \ b_n]^T$, podemos representar o sistema na forma matricial $\mathbf{Ax} = \mathbf{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

em que n é o número de equações do sistema, \mathbf{A} é a matriz dos coeficientes, \mathbf{x} é a matriz (vetor) das incógnitas e \mathbf{b} é a matriz (vetor) dos termos independentes.

Exemplo. Considere o sistema $\mathbf{Ax} = \mathbf{b}$ sendo

$$a_{ij} = \frac{1}{i+j-1} \quad \text{e} \quad b_i = \frac{1}{i}.$$

Com vistas à familiarização com o Octave, para $n = 5$, os dados do sistema (\mathbf{A} e \mathbf{b}) podem ser inseridos da seguinte forma (Listagem 1):

Listagem 1 – Inserindo os dados do sistema no Octave

```

1 n = 5;
2 for i=1:n
3     for j=1:n
4         A(i,j) = 1/(i+j-1);
5     end
6     b(i)=1/i;
7 end
8 A
9 b = b'
```

Fonte: Autoria própria (2024).

Quando este código é executado, obtém-se o seguinte resultado:

$$\mathbf{A} = \begin{bmatrix} 1.0000 & 0.5000 & 0.3333 & 0.2500 & 0.2000 \\ 0.5000 & 0.3333 & 0.2500 & 0.2000 & 0.1667 \\ 0.3333 & 0.2500 & 0.2000 & 0.1667 & 0.1429 \\ 0.2500 & 0.2000 & 0.1667 & 0.1429 & 0.1250 \\ 0.2000 & 0.1667 & 0.1429 & 0.1250 & 0.1111 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 1.0000 & 0.5000 & 0.3333 & 0.2500 & 0.2000 \end{bmatrix}^T.$$

Para resolver os sistemas de equações lineares, neste trabalho será utilizado o método de Eliminação de Gauss.

2.1.1 Método de Eliminação Gaussiana

A eliminação gaussiana é um método que tem como base as operações elementares sobre as equações (linhas) de um sistema de equações lineares. A solução do sistema $\mathbf{Ax} = \mathbf{b}$ não se altera se o submetemos a uma sequência de operações elementares do tipo:

- (i) multiplicação de uma equação por uma constante não nula;

- (ii) soma de um múltiplo de uma equação à outra;
- (iii) troca da ordem das equações.

Considerando o sistema linear

$$\begin{cases} x_1 & + x_3 = 0 & l_1 : 1^\circ \text{ equação} \\ x_1 + x_2 & = 1 & l_2 : 2^\circ \text{ equação} \\ 2x_1 + 3x_2 + x_3 & = 1 & l_3 : 3^\circ \text{ equação} \end{cases}$$

será realizada, de forma sistemática, uma sequência conveniente de operações dos tipos (i), (ii) e (iii) com o intuito de transformar este sistema linear em um sistema triangular equivalente (sistema com o mesmo conjunto solução). É importante que a eliminação seja feita de maneira sistemática para elaborar algoritmos para os programas computacionais.

Primeiramente serão zerados os coeficientes da primeira coluna que estão abaixo da diagonal. Isso corresponde a eliminar x_1 das duas últimas equações, o mantendo apenas na primeira. Para isso vamos subtrair, da segunda e da terceira equações, um múltiplo adequado da primeira equação. Assim fica: $l'_2 = l_2 - l_1$ e $l'_3 = l_3 - 2l_1$. Quando reescrevermos o sistema ele ficará da forma:

$$\begin{cases} x_1 & + x_3 = 0 & l'_1 = l_1 \\ & x_2 - x_3 = 1 & l'_2 = l_2 - \frac{a_{21}}{a_{11}}l_1 \\ & 3x_2 - x_3 = 1 & l'_3 = l_3 - \frac{a_{31}}{a_{11}}l_1 \end{cases}.$$

Para chegar à forma triangular, basta eliminar o elemento da segunda coluna que está abaixo da diagonal, o que corresponde à eliminação de x_2 na última equação. Para isso tomamos $l''_3 = l'_3 - 3l'_2$. Agora estamos utilizando a segunda equação como base para a eliminação. Então temos um sistema triangular:

$$\begin{cases} x_1 & + x_3 = 0 \\ & x_2 - x_3 = 1 \\ & & x_3 = -1 & l''_3 = l'_3 - \frac{a_{32}}{a_{22}}l'_2 \end{cases}.$$

Generalizando o procedimento descrito acima, será estabelecido o processo sistemático de triangularização de um sistema com n equações e n incógnitas.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix},$$

na qual cada linha l_i , $i = 1 : n$ representa uma equação do sistema linear que queremos triangularizar.

• Eliminação da primeira coluna

Supondo $a_{11} \neq 0$, para eliminar a incógnita x_1 nas $n - 1$ últimas equações vamos subtrair a primeira linha multiplicada pelo fator $m_{i1} = a_{i1}/a_{11}$ de todas as outras linhas l_i , para $i = 2 : n$.

Dessa maneira, estamos modificando os elementos das $n - 1$ últimas linhas da seguinte forma:

$$\text{Para } i = 2 : n, \quad a_{ij}^2 = a_{ij} - m_{i1}a_{1j}, \quad j = 2 : n$$

$$b_i^2 = b_i - m_{i1}b_1.$$

O índice superior 2 em a_{ij}^2 e b_i^2 indica que vamos usar um segundo valor para a_{ij} e b_i .

No fim deste estágio, as matrizes \mathbf{A} e \mathbf{b} foram modificadas de modo que assumem a seguinte configuração:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^2 & a_{23}^2 & \cdots & a_{2n}^2 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{n2}^2 & a_{n3}^2 & \cdots & a_{nn}^2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2^2 \\ \vdots \\ b_n^2 \end{bmatrix}.$$

• Eliminação da segunda coluna

Para eliminar a incógnita x_2 das $n - 2$ últimas equações repetimos o procedimento anterior tomando, agora, a segunda linha como auxiliar no processo de eliminação:

$$l_i^3 = l_i^2 - m_{i2}l_2^2, \quad i = 3 : n, \quad \text{em que } m_{i2} = \frac{a_{i2}}{a_{22}}, \quad i = 3 : n$$

Nesta etapa, estamos supondo que $a_{22} \neq 0$. Os coeficientes serão modificados segundo as relações

$$\text{Para } i = 3 : n \quad a_{ij}^3 = a_{ij}^2 - m_{i2}a_{2j}^2, \quad j = 3 : n$$

$$b_i^3 = b_i^2 - m_{i2}b_2^2.$$

Continuamos eliminando os elementos que estão abaixo da diagonal em cada coluna, repetindo a ideia descrita acima.

No processo de eliminação, os elementos $a_{11}, a_{22}^2, a_{33}^3, \dots, a_{jj}^j$ que aparecem na diagonal de \mathbf{A} são chamados de pivôs. Se eles não se anulam, após este procedimento chegamos à forma triangularizada do sistema:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}x_2 + \dots + a_{2n}x_n = b_2^2 \\ \vdots \\ a_{nn}x_n = b_n^n \end{array} \right.$$

Se no processo de eliminação um dos pivôs se anular, devemos trocar as linhas (sempre escolhendo aquelas abaixo da diagonal para não perder a eliminação anterior), de modo que sejam escolhidos elementos não nulos para pivôs.

Tendo a forma generalizada para a triangularização de sistemas de equações lineares, vamos testar o método. Continuando com o primeiro exemplo, vamos inserir o código que vai triangularizar o sistema que já foi inserido no programa (Listagem 2). Continuaremos denotando as matrizes A e b atualizadas na triangularização por A e b para não guardar várias cópias dessas matrizes, economizando espaço de armazenamento, principalmente no caso de sistemas grandes.

Depois de executar este programa o resultado obtido é o seguinte:

$$\mathbf{A} = \begin{bmatrix} 1.00000 & 0.50000 & 0.33333 & 0.25000 & 0.20000 \\ 0.00000 & 0.08333 & 0.08333 & 0.07500 & 0.06667 \\ 0.00000 & 0.00000 & 0.00556 & 0.00833 & 0.00952 \\ 0.00000 & 0.00000 & 0.00000 & 0.00036 & 0.00071 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00002 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

Listagem 2 – Triangularizando o sistema pelo método de eliminação de Gauss.

```

1  n = 5;
2  for i=1:n
3      for j=1:n
4          A(i,j) = 1/(i+j-1);
5      end
6      b(i)=1/i;
7  end
8  A
9  b
10 for k=1:n-1
11     for i=k+1:n
12         m=A(i,k)/A(k,k);
13         b(i)=b(i)-m*b(k);
14         for j=k:n
15             A(i,j)=A(i,j)-m*A(k,j);
16         end
17     end
18 end
19 A
20 b=b'
```

Fonte: Autoria própria (2024).

2.1.2 Sistemas triangulares

Quando a matriz dos coeficientes de um sistema tem todos os elementos abaixo ou acima da diagonal principal nulos, o sistema é dito triangular.

Observando o sistema triangularizado

$$\begin{cases} x_1 & + x_3 = 0 \\ & x_2 - x_3 = 1 \\ & & x_3 = -1 \end{cases},$$

verifica-se que ele pode ser resolvido facilmente. Começando pela terceira equação temos que $x_3 = -1$. Na segunda usando o valor de x_3 e isolando x_2 , temos que $x_2 = 2$. Por fim na primeira equação, usando os valores de x_2 e x_3 , isolamos x_1 e assim temos que $x_1 = 1$. Nesse caso, a matriz dos coeficientes \mathbf{A} é uma matriz triangular superior pois todos os elementos abaixo da diagonal principal são nulos, isto é, $a_{ij} = 0$ se $i > j$.

Agora para generalizar, considere o sistema triangular:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n-1}x_{n-1} + a_{1n}x_n = b_1 \\ a_{22}x_2 + \dots + a_{2n-1}x_{n-1} + a_{2n}x_n = b_2 \\ \vdots \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1} \\ a_{n,n}x_n = b_n \end{array} \right. .$$

Admitindo que $a_{ii} \neq 0$, $i = 1 : n$, podemos determinar o valor das incógnitas x_1, x_2, \dots, x_n fazendo as seguintes operações:

$$\left\{ \begin{array}{l} x_n = b_n/a_{n,n} \\ x_{n-1} = (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1} \\ \vdots \\ x_1 = (b_1 - a_{1n}x_n - \dots - a_{13}x_3 - a_{12}x_2)/a_{11} \end{array} \right. .$$

Assim resta solucionar o exemplo inicial agora resolvendo o sistema triangular no Octave (Listagem 3).

Listagem 3 – Resolução do sistema triangularizado

```

1  n = 5;
2  for i=1:n
3      for j=1:n
4          A(i,j) = 1/(i+j-1);
5      end
6      b(i)=1/i;
7  end
8  A
9  b
10 for k=1:n-1
11     for i=k+1:n
12         m=A(i,k)/A(k,k);
13         b(i)=b(i)-m*b(k);
14         for j=k:n
15             A(i,j)=A(i,j)-m*A(k,j);
16         end
17     end
18 end
19 A
20 b
21 x(n)=b(n)/A(n,n);
22 for k=n-1:-1:1
23     soma=b(k);
24     for j=k+1:n
25         soma=soma-A(k,j)*x(j);
26     end
27     x(k)=soma/A(k,k);
28 end
29 x=x'
```

Fonte: Autoria própria (2024).

O resultado obtido com o programa é

$$\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

2.1.3 Sistemas tridiagonais

Quando temos um sistema tridiagonal, os elementos não nulos da matriz dos coeficientes estão nas três diagonais centrais. Assim, a eliminação em cada coluna será de apenas um elemento. Para esse tipo de sistema linear pode-se utilizar quatro vetores, um para a diagonal principal, dois para as diagonais secundárias e um para o termo independente. Segue a

ilustração do sistema:

$$\begin{bmatrix} d_1 & c_1 & & & \\ a_2 & d_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & d_{n-1} & c_{n-1} \\ & & & & a_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

A representação de matrizes tridiagonais em quatro vetores é bastante importante pois economiza espaço de armazenamento já que não é necessário guardar os elementos nulos. O algoritmo da triangularização pode ser adaptado para sistemas tridiagonais visto que é necessário zerar apenas um elemento abaixo da diagonal principal. Isto resulta em um algoritmo específico para resolver sistemas tridiagonais (Listagem 4).

Como exemplo, considere o sistema

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{bmatrix}.$$

Listagem 4 – Resolução de sistema tridiagonal

```

1 n=4;
2 for i=2:n
3     a(i)=-1;
4 end
5 for i=1:n
6     d(i)=2.04;
7 end
8 for i=1:n-1
9     c(i)=-1;
10 end
11 b(1)=40.8;
12 b(n)=200.8;
13 for i=2:n-1
14     b(i)=0.8;
15 end
16 a
17 d
18 c
19 b
20 for i=1:n-1
21     d(i+1)=d(i+1)-(a(i+1)/d(i))*c(i);
22     b(i+1)=b(i+1)-(a(i+1)/d(i))*b(i);
23 end
24 x(n)=b(n)/d(n);
25 for i=n-1:-1:1
26     x(i)=(b(i)-c(i)*x(i+1))/d(i);
27 end
28 x=x'
```

Fonte: Autoria própria (2024).

2.2 Série de Taylor

Esta seção, baseada em Stewart, Clegg e Watson (2022), abordará de forma resumida a série de Taylor e sua construção.

Considerando uma função f que pode ser representada por uma série de potências para $|x - a| < R$ com $R > 0$,

$$f(x) = c_0 + c_1(x - a) + c_2(x - a)^2 + c_3(x - a)^3 + \dots,$$

vamos determinar os coeficientes c_n em termos de f . Fazendo $x = a$ nessa equação, todos os termos a partir do primeiro se anulam, resultando em

$$f(a) = c_0.$$

Derivando a série de potências (derivação termo a termo), tem-se

$$f'(x) = c_1 + 2c_2(x - a) + 3c_3(x - a)^2 + 4c_4(x - a)^3 + \dots$$

Fazendo $x = a$ novamente, agora nesta última equação:

$$f'(a) = c_1.$$

Derivando novamente a série termo a termo, tem-se

$$f''(x) = 2c_2 + 2 \cdot 3c_3(x - a) + 3 \cdot 4c_4(x - a)^2 + \dots$$

Novamente fazendo $x = a$:

$$f''(a) = 2c_2 \Rightarrow c_2 = \frac{f''(a)}{2}.$$

Fazendo uma nova derivação termo a termo:

$$f'''(x) = 2 \cdot 3c_3 + 2 \cdot 3 \cdot 4c_4(x - a) + 3 \cdot 4 \cdot 5c_5(x - a)^2 + \dots$$

E, novamente fazendo $x = a$:

$$f'''(a) = 2 \cdot 3c_3 = 3!c_3 \Rightarrow c_3 = \frac{f'''(a)}{3!}.$$

Percebe-se que existe um padrão. Se a série continuar a ser derivada termo a termo mais vezes e substituindo x por a obtém-se

$$f^{(n)}(a) = 2 \cdot 3 \cdot 4 \cdots n c_n = n! c_n,$$

ou seja,

$$c_n = \frac{f^{(n)}(a)}{n!}.$$

Substituindo os coeficientes na série de potências tem-se

$$\begin{aligned}
 f(x) &= f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots \\
 &= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n,
 \end{aligned}$$

que é dita série de Taylor da função f centrada em a (ou em torno de a).

Exemplo. Encontrar a série de Taylor da função $f(x) = e^x$ centrada em $a = 0$.

Solução: como $f(x) = e^x$, então $f^{(n)}(x) = e^x$. Assim, $f^{(n)}(0) = e^0 = 1$ para todo n . Portanto, a série de Taylor de f centrada em 0 é

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Assim, se e^x tiver uma expansão em série de potências centrada em $a = 0$ então

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

2.3 Equações diferenciais ordinárias

Tópico baseado em Zill (2016). Uma equação diferencial ordinária (EDO) é uma equação que contém derivadas de uma função desconhecida que depende de apenas uma variável. Essas equações são classificadas de acordo com a ordem e a linearidade.

A ordem da equação é a ordem da maior derivada que aparece na equação. Uma equação diferencial ordinária de ordem n pode ser escrita na forma geral

$$F(x, y, y', \dots, y^{(n)}) = 0.$$

Aqui F é uma função, que assume valores reais, de $n + 2$ variáveis: $x, y, y', \dots, y^{(n)}$, sendo $y^{(n)} = \frac{d^n y}{dx^n}$. Quanto à linearidade, uma equação diferencial ordinária de ordem n é dita linear se F for linear em $y, y', \dots, y^{(n)}$, ou seja, quando é possível escrever a equação na forma

$$a_n(x) \frac{d^n y}{dx^n} + a_{n-1}(x) \frac{d^{n-1} y}{dx^{n-1}} + \dots + a_1(x) \frac{dy}{dx} + a_0(x) y = g(x).$$

Exemplos de equações diferenciais ordinárias lineares:

$$(1) \frac{dy}{dx} + 5y = e^x.$$

$$(2) \frac{d^2y}{dx^2} - 4y = 0.$$

$$(3) y'' - y' + xy = e^x(x^2 + 1).$$

Nos exemplos, o primeiro é uma equação de primeira ordem enquanto o segundo e o terceiro são equações de segunda ordem.

Uma função φ definida em um intervalo I que tem pelo menos n derivadas contínuas em I , as quais quando substituídas na equação diferencial ordinária de ordem n reduzem a equação a uma identidade, é uma solução da equação no intervalo. Assim, uma solução de uma equação diferencial ordinária de ordem n no intervalo I é uma função φ com pelo menos n derivadas contínuas que satisfaz $F(x, \varphi(x), \varphi'(x), \dots, \varphi^{(n)}(x)) = 0$ para todo x em I .

A função $\varphi(x) = \frac{1}{6}e^x - 5ce^{-5x}$ é uma solução para a equação do exemplo 1, a função $\varphi(x) = c_1e^{2x} + c_2e^{-2x}$ é uma solução para a equação do exemplo 2 e a função $y(x) = xe^x$ é uma solução para a equação do exemplo 3 (em qualquer intervalo I de números reais). Para ilustração, vamos verificar que $y(x) = xe^x$ satisfaz a equação $y'' - y' + xy = e^x(x^2 + 1)$. Como $y'(x) = e^x + xe^x$ e $y''(x) = 2e^x + xe^x$ temos

$$\begin{aligned} y'' - y' + xy &= 2e^x + xe^x - (e^x + xe^x) + xxe^x \\ &= e^x + x^2e^x \\ &= e^x(x^2 + 1). \end{aligned}$$

3 MÉTODO

3.1 Método das diferenças finitas

Este capítulo foi baseado em Cunha (2000).

A incógnita de uma equação diferencial ordinária é uma função $y(x)$ que está definida em todos os pontos do intervalo em que a equação está sendo resolvida. O primeiro passo para um método que tem como finalidade a solução numérica de equações diferenciais é discretizar a região onde se procura a solução. Assim é definida uma malha, que é um conjunto finito de pontos, que são chamados de nós da malha. Seja x_0 um ponto de referência e h um número positivo. A malha de passo h associada a x_0 é o conjunto de pontos

$$x_i = x_0 + ih, \quad i = 1, 2, \dots, n.$$

As aproximações para $y(x)$ serão calculadas nos pontos da malha.

No método das diferenças finitas, o próximo passo é a discretização das derivadas que estão na equação diferencial. Essas derivadas são aproximadas por diferenças entre valores da solução discretizada. Para realizar essas aproximações das derivadas utiliza-se a série de Taylor. Lembrando que a série de Taylor da função f centrada em a é

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \frac{f^{(4)}(a)}{4!}(x-a)^4 + \dots,$$

trabalhando com uma função y e com $x - a = h$ tem-se

$$y(a+h) = y(a) + \frac{y'(a)}{1!}h + \frac{y''(a)}{2!}h^2 + \frac{y'''(a)}{3!}h^3 + \frac{y^{(4)}(a)}{4!}h^4 + \dots$$

Trocando a por x

$$y(x+h) = y(x) + y'(x)h + \frac{y''(x)}{2!}h^2 + \frac{y'''(x)}{3!}h^3 + \frac{y^{(4)}(x)}{4!}h^4 + \dots \quad (1)$$

Isolando o termo $y'(x)$:

$$y'(x) = \frac{y(x+h) - y(x)}{h} - \frac{y''(x)}{2!}h - \frac{y'''(x)}{3!}h^2 + \dots$$

Trocando h por $-h$ na equação (1):

$$y(x-h) = y(x) + y'(x)(-h) + \frac{y''(x)}{2!}(-h)^2 + \frac{y'''(x)}{3!}(-h)^3 + \frac{y^{(4)}(x)}{4!}(-h)^4 + \dots \quad (2)$$

Novamente isolando o termo $y'(x)$, tem-se a seguinte equação:

$$y'(x) = \frac{y(x) - y(x-h)}{h} + \frac{y''(x)}{2!}h - \dots$$

Subtraindo (2) de (1):

$$y(x+h) - y(x-h) = 2y'(x)h + \frac{2y'''(x)}{3!}h^3 + \dots$$

Isolando $y'(x)$

$$y'(x) = \frac{y(x+h) - y(x-h)}{2h} - y'''(x)\frac{h^2}{3!} \dots$$

Para cada ponto da malha no interior do intervalo em que a equação diferencial está definida, denota-se por y_i a aproximação de $y(x_i)$. Assim usa-se as seguintes equações:

$$y'(x_i) \approx \frac{y_{i+1} - y_i}{h}, \quad y'(x_i) \approx \frac{y_i - y_{i-1}}{h} \quad \text{ou} \quad y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}$$

ditas fórmulas avançada, atrasada e centrada respectivamente.

Agora seguindo as mesmas ideias, pode-se estabelecer uma expressão para o cálculo da aproximação da segunda derivada. Somando as equações (1) e (2) e depois isolando $y''(x)$:

$$y''(x) = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} - \frac{h^2}{12}y^{(4)}(\xi). \quad (3)$$

Desprezando o termo $\frac{h^2}{12}y^{(4)}(\xi)$ (que representa o erro) obtém-se a aproximação para a derivada de segunda ordem:

$$y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

4 EXEMPLOS E RESULTADOS

Neste capítulo constam dois problemas de valores de contorno (PVC) que foram resolvidos com o uso do método das diferenças finitas.

4.1 Exemplo 1

Exemplo apresentado em Cunha (2000). Aplicação do método das diferenças finitas para encontrar uma aproximação para a solução da equação diferencial ordinária

$$y'' - y' + xy = e^x(x^2 + 1), \quad x \in (0,1),$$

com as condições de contorno

$$y(0) = 0, \quad y(1) = e.$$

Discretizando o intervalo, tomando a malha associada a $h = 0.1$, usando a fórmula centrada para aproximar a primeira derivada e a aproximação para a segunda derivada tem-se:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \frac{y_{i+1} - y_{i-1}}{2h} + x_i y_i = e^{x_i}(x_i^2 + 1).$$

A equação acima é uma equação linear nas incógnitas y_{i-1} , y_i e y_{i+1} , que são as aproximações para $y(x)$ nos pontos x_{i-1} , x_i e x_{i+1} . Como $x_i = ih$, esta equação pode ser escrita na forma

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \frac{y_{i+1} - y_{i-1}}{2h} + i h y_i = e^{ih}(i^2 h^2 + 1).$$

Multiplicando por $2h^2$ e agrupando os termos obtém-se

$$(2 + h)y_{i-1} + (2ih^3 - 4)y_i + (2 - h)y_{i+1} = 2h^2 e^{ih}(i^2 h^2 + 1).$$

Nesta forma discretizada a equação pode ser escrita para todos os pontos internos da malha $x = x_i$, $i = 1 : 9$, obtendo-se um sistema tridiagonal com nove equações nas nove incógnitas y_i , $i = 1 : 9$. Escrevendo a equação para cada valor de i tem-se:

$$i = 1 : 2(h^3 - 2)y_1 + (2 - h)y_2 = 2h^2e^h(h^2 + 1) - (2 + h)y_0.$$

$$i = 2 : (2 + h)y_1 + 2(2h^3 - 4)y_2 + (2 - h)y_3 = 2h^2e^{2h}(4h^2 + 1)$$

$$i = 3 : (2 + h)y_2 + 2(3h^3 - 4)y_3 + (2 - h)y_4 = 2h^2e^{3h}(9h^2 + 1)$$

$$i = 4 : (2 + h)y_3 + 2(4h^3 - 4)y_4 + (2 - h)y_5 = 2h^2e^{4h}(16h^2 + 1)$$

$$i = 5 : (2 + h)y_4 + 2(5h^3 - 4)y_5 + (2 - h)y_6 = 2h^2e^{5h}(25h^2 + 1)$$

$$i = 6 : (2 + h)y_5 + 2(6h^3 - 4)y_6 + (2 - h)y_7 = 2h^2e^{6h}(36h^2 + 1)$$

$$i = 7 : (2 + h)y_6 + 2(7h^3 - 4)y_7 + (2 - h)y_8 = 2h^2e^{7h}(49h^2 + 1)$$

$$i = 8 : (2 + h)y_7 + 2(8h^3 - 4)y_8 + (2 - h)y_9 = 2h^2e^{8h}(64h^2 + 1)$$

$$i = 9 : (2 + h)y_8 + 2(9h^3 - 2)y_9 = 2h^2e^{9h}(81h^2 + 1) - (2 - h)y_{10}$$

que resulta no sistema linear

$$\left\{ \begin{array}{l} a_{11} y_1 + a_{12} y_2 \\ a_{21} y_1 + a_{22} y_2 + a_{23} y_3 \\ a_{32} y_2 + a_{33} y_3 + a_{34} y_4 \\ a_{43} y_3 + a_{44} y_4 + a_{45} y_5 \\ a_{54} y_4 + a_{55} y_5 + a_{56} y_6 \\ a_{65} y_5 + a_{66} y_6 + a_{67} y_7 \\ a_{76} y_6 + a_{77} y_7 + a_{78} y_8 \\ a_{87} y_7 + a_{88} y_8 + a_{89} y_9 \\ a_{98} y_8 + a_{99} y_9 \end{array} \right. \begin{array}{l} = b_1 \\ = b_2 \\ = b_3 \\ = b_4 \\ = b_5 \\ = b_6 \\ = b_7 \\ = b_8 \\ = b_9 \end{array} ,$$

sendo $a_{i,i-1} = 2 + h$, $a_{i,i} = 2ih^3 - 4$, $a_{i,i+1} = 2 - h$ e $b_i = 2h^2e^{ih}(i^2h^2 + 1)$.

Escrevendo no formato $\mathbf{Ax} = \mathbf{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} & a_{67} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{76} & a_{77} & a_{78} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{87} & a_{88} & a_{89} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{89} & a_{99} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \end{bmatrix}.$$

Depois do sistema estar implementado, segue-se a resolução deste pelo método de eliminação de Gauss.(Listagem 5).

Listagem 5 – Inserindo o exemplo 1 no Octave

```

1 a=0;
2 b=1;
3 ya=0;
4 yb=e;
5 n=9;
6 h=(b-a)/(n+1);
7 A(1,1) = 2*h^3-4;
8 A(1,2) = 2-h;
9 A(n,n-1) = 2+h;
10 A(n,n) = 2*n*h^3-4;
11 for i= 2:n-1
12     A(i,i-1) = 2+h;
13     A(i,i) = 2*i*h^3-4;
14     A(i,i+1) = 2-h;
15     b(i)=2*h^2*e^(i*h)*(i^2*h^2+1);
16 end
17 b(1) = 2*h^2*e^h*(h^2+1) - (2+h)*ya;
18 b(n) = 2*h^2*e^(n*h)*(n^2*h^2+1) - (2-h)*yb;
19 A
20 b = b'
```

Fonte: Autoria própria (2024).

Depois do sistema estar implementado, segue-se a resolução do sistema pelo método de eliminação de Gauss.(Listagem 6)

A solução obtida com o programa é:

$$y = [0.1104 \ 0.2441 \ 0.4046 \ 0.5963 \ 0.8239 \ 1.0928 \ 1.4092 \ 1.7801 \ 2.2134]^T.$$

Listagem 6 – Código completo para resolver o exemplo 1 pelo método de Gauss

```

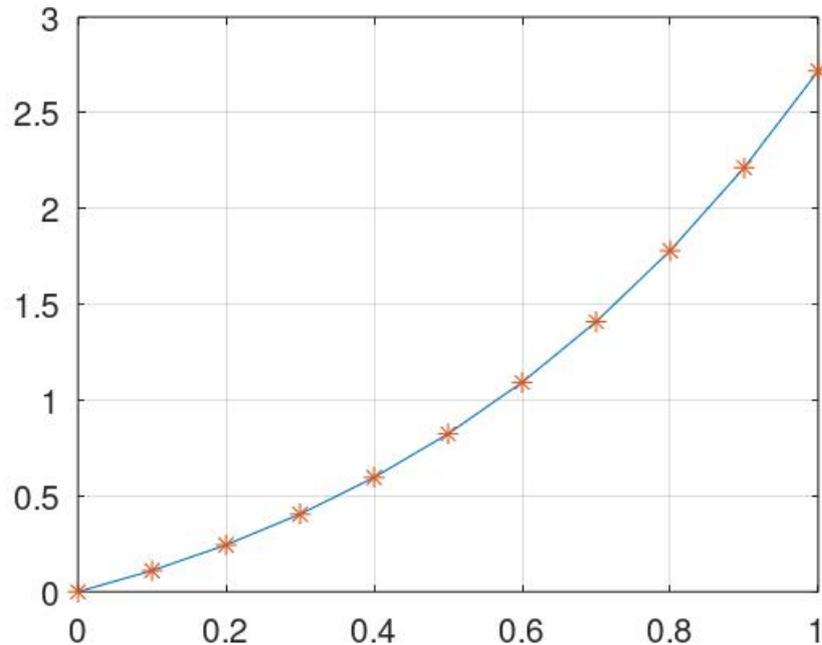
1 clear
2 xa=0
3 xb=1
4 ya=0
5 yb=e
6 n=9
7 h=(xb-xa)/(n+1)
8 for i=1:n;
9     x(i)=xa+i*h;
10    end
11 A(1,1) = 2*h^3-4
12 A(1,2) = 2-h
13 A(n,n-1) = 2+h
14 A(n,n) = 2*n*h^3-4
15 for i= 2:n-1
16     A(i,i-1) = 2+h
17     A(i,i) = 2*i*h^3-4
18     A(i,i+1) = 2-h
19     b(i)=2*h^2*e^(i*h)*(i^2*h^2+1)
20 end
21 b(1) = 2*h^2*e^h*(h^2+1) - (2+h)*ya
22 b(n) = 2*h^2*e^(n*h)*(n^2*h^2+1) - (2-h)*yb
23 A
24 b=b'
25 for k=1:n-1
26     for i=k+1:n
27         m=A(i,k)/A(k,k)
28         b(i) = b(i)-m*b(k)
29         for j=k:n
30             A(i,j)=A(i,j)-m*A(k,j)
31         end
32     end
33 end
34
35 y(n)=b(n)/A(n,n)
36 for k=n-1:-1:1
37     soma=b(k)
38     for j=k+1:n
39         soma=soma-A(k,j)*y(j)
40     end
41     y(k)=soma/A(k,k)
42 end
43 y=y'
44 x=[xa x xb]
45 y=[ya y yb]
46 z= x.*e.^x
47 plot(x,z,x,y,'*')

```

Fonte: Autoria própria (2024).

O gráfico gerado pelo código com as soluções numérica e exata:

Figura 1 – Gráfico exemplo 1



Fonte: Gerada com o programa Gnu Octave.

4.2 Exemplo 2

Exemplo apresentado em Chapra e Canale (2016). A conservação do calor pode ser usada para desenvolver uma equação que descreve a temperatura em uma haste longa e fina. Se a haste não estiver isolada ao longo de seu comprimento e se o sistema estiver em estado estacionário, a equação resultante será

$$\frac{d^2T}{dx^2} + c(T_a - T) = 0,$$

sendo T a temperatura da barra ($^{\circ}C$), c o coeficiente de transferência de calor $\left(\frac{W}{m^2K}\right)$ e T_a a temperatura do ar em torno da haste. Para existir uma solução dessa equação, devem existir condições de contorno adequadas. Um caso simples é quando as temperaturas nas extremidades da haste são mantidas em valores fixos. Estas podem ser expressas matematicamente por

$$\begin{aligned} T(0) &= T_1 \\ T(L) &= T_2. \end{aligned}$$

O problema será solucionado considerando uma haste de 10 m com $c = 0.01 \frac{W}{m^2K}$, $T_a = 20^\circ C$ e as seguintes condições de contorno

$$\begin{aligned} T(0) &= 40^\circ C \\ T(10) &= 200^\circ C. \end{aligned}$$

Fazendo a aproximação por diferenças finitas para a segunda derivada:

$$\frac{d^2T}{dx^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2}.$$

Substituindo na equação diferencial:

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} - c(T_i - T_a) = 0.$$

Agora agrupando os termos:

$$-T_{i-1} + (2 + ch^2)T_i - T_{i+1} = ch^2T_a.$$

Considerando $n = 4$, para cada valor de i esta equação fica

$$i = 1 : (2 + ch^2)T_1 - T_2 = ch^220 + 40$$

$$i = 2 : -T_1 + (2 + ch^2)T_2 - T_3 = ch^220$$

$$i = 3 : -T_2 + (2 + ch^2)T_3 - T_4 = ch^220$$

$$i = 4 : -T_3 + (2 + ch^2)T_4 = ch^220 + 200$$

resultando no sistema de equações lineares:

$$\left\{ \begin{array}{l} a_{11} T_1 + a_{12} T_2 = b_1 \\ a_{21} T_1 + a_{22} T_2 + a_{23} T_3 = b_2 \\ a_{32} T_2 + a_{33} T_3 + a_{34} T_4 = b_3 \\ a_{43} T_3 + a_{44} T_4 = b_4 \end{array} \right. ,$$

sendo $a_{i,i-1} = a_{i,i+1} = -1$, $a_{ii} = 2 + ch^2$ e $b_i = ch^2 20$ para $i = 2,3$, $b_1 = ch^2 20 + 40$ e $b_4 = ch^2 20 + 200$.

Agora escrevendo na forma $\mathbf{Ax} = \mathbf{b}$:

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{bmatrix}.$$

Este exemplo será resolvido utilizando dois algoritmos diferentes, que foram abordados no capítulo dos pré-requisitos. Primeiramente será utilizado o algoritmo do método de eliminação de Gauss. Inicia-se inserindo os dados no software. (Listagem 7)

Depois do sistema inserido no programa, segue-se a resolução pelo método de eliminação de Gauss. (Listagem 8)

No código deste exemplo foram inseridas duas ferramentas a mais. Um comando para calcular o tempo utilizado para a resolução do problema e uma ferramenta para gerar o gráfico da solução. A intenção dessas duas ferramentas é para fins de comparação, pois como serão utilizadas duas formas diferentes, os tempos de resolução poderão ser comparados para verificar qual é mais vantajoso.

Agora, o mesmo exemplo usando o algoritmo para sistemas tridiagonais. Primeiramente os dados devem ser inseridos no programa. (Listagem 9)

Depois, resolvendo com o código para sistemas tridiagonais (Listagem 10):

Para ambos os algoritmos o programa gerou o seguinte resultado:

$$y = [40.00 \quad 65.970 \quad 93.778 \quad 124.538 \quad 159.480 \quad 200.000].$$

E o gráfico da solução (aproximada) obtida nos dois métodos foi (Figura 1):

O tempo de resolução do algoritmo pelo método de eliminação de Gauss e o tempo de resolução para o método de sistemas tridiagonais não apresentou diferença significativa.

O tempo de resolução dos dois códigos não diferenciou muito. Quando se trabalha com o n pequeno usando qualquer um dos códigos, a solução será encontrada de maneira rápida. Agora, vamos ver o que acontece quando o mesmo problema é resolvido com um valor mais alto para n . Faremos com $n = 200$.

Com o algoritmo para sistemas tridiagonais o tempo de resolução com $n = 200$ foi aproximadamente 100 vezes menor que com o algoritmo para o método de eliminação de Gauss.

Listagem 7 – Inserindo o exemplo 2 no Octave

```

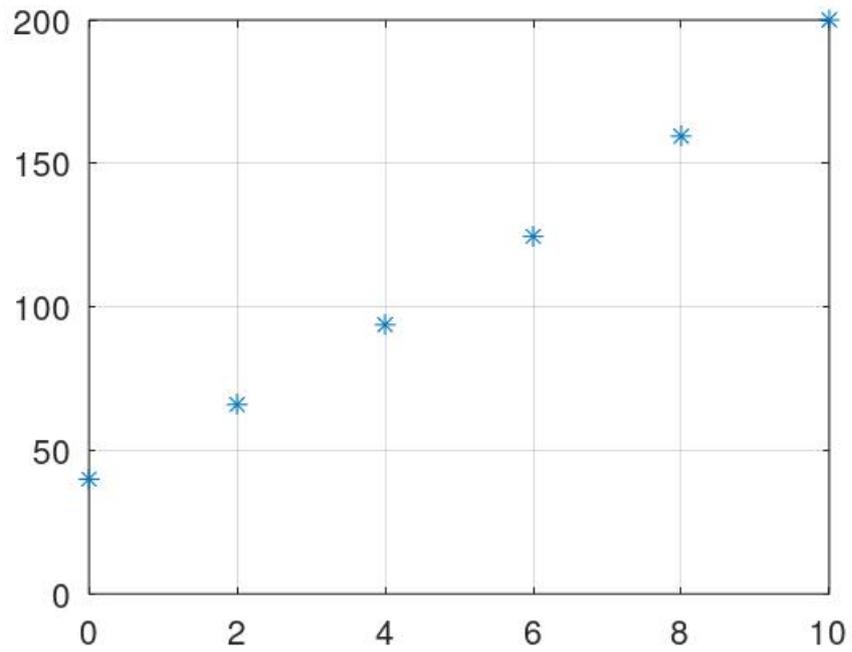
1 clear
2 xa=0;
3 xb=10;
4 ya=40;
5 yb=200;
6 c=0.01;
7 ta=20;
8 n=4;
9 h=(xb-xa)/(n+1);
10 for i=1:n;
11     x(i)=xa+i*h;
12 end
13 A(1,1)=2+c*h^2;
14 A(1,2)=-1;
15 A(n,n-1)=-1;
16 A(n,n)=2+c*h^2;
17 for i=2:n-1;
18     A(i,i-1)=-1;
19     A(i,i)=2+c*h^2;
20     A(i,i+1)=-1;
21     b(i)=c*h^2*ta;
22 end
23 b(1)=c*h^2*ta+ya;
24 b(n)=c*h^2*ta+yb;
25 A
26 b=b'
```

Fonte: Autoria própria (2024).

Em ambos os métodos a solução obtida foi a mesma, gerando o seguinte gráfico para a solução aproximada (Figura 2).

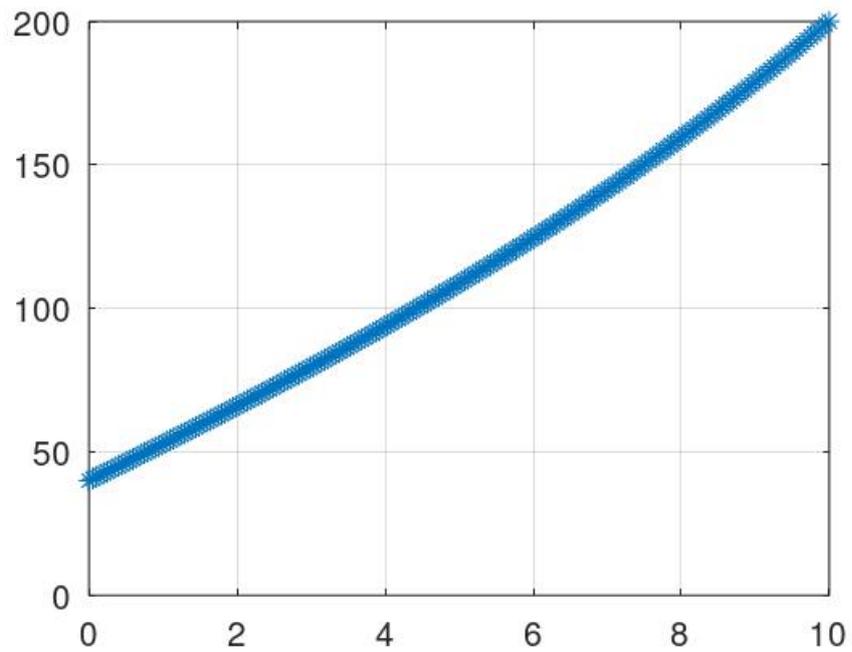
Analisando as duas resoluções percebe-se que a segunda se mostrou mais rápida para determinar a solução do problema com n grande.

Figura 2 – Gráfico exemplo 2 para $n = 4$



Fonte: Gerada com o programa Gnu Octave.

Figura 3 – Gráfico exemplo 2 para $n = 200$



Fonte: Gerada com o programa Gnu Octave.

Listagem 8 – Resolução do exemplo 2 pelo método de Gauss

```

1 clear
2 tic
3 xa=0;
4 xb=10;
5 ya=40;
6 yb=200;
7 c=0.01;
8 ta=20;
9 n=4;
10 h=(xb-xa)/(n+1);
11 for i=1:n;
12     x(i)=xa+i*h;
13 end
14 A(1,1)=2+c*h^2;
15 A(1,2)=-1;
16 A(n,n-1)=-1;
17 A(n,n)=2+c*h^2;
18 for i=2:n-1;
19     A(i,i-1)=-1;
20     A(i,i)=2+c*h^2;
21     A(i,i+1)=-1;
22     b(i)=c*h^2*ta;
23 end
24 b(1)=c*h^2*ta+ya;
25 b(n)=c*h^2*ta+yb;
26 A
27 b
28 for k=1:n-1;
29     for i=k+1:n;
30         m=A(i,k)/A(k,k);
31         b(i) = b(i)-m*b(k);
32         for j=k:n;
33             A(i,j)=A(i,j)-m*A(k,j);
34         end
35     end
36 end
37 y(n)=b(n)/A(n,n);
38 for k=n-1:-1:1;
39     soma=b(k);
40     for j=k+1:n;
41         soma=soma-A(k,j)*y(j);
42     end
43     y(k)=soma/A(k,k);
44 end
45 x=[xa x xb]
46 y=[ya y yb]
47 toc
48 plot(x,y,'*')

```

Fonte: Autoria própria (2024).

Listagem 9 – Inserindo o exemplo 2 no Octave para resolver pelo código de sistemas tridiagonais

```
1 clear
2 xa=0;
3 xb=10;
4 ya=40;
5 yb=200;
6 cc=0.01;
7 ta=20;
8 n=4;
9 h=(xb-xa)/(n+1);
10 for i=1:n;
11     x(i)=xa+i*h;
12 end
13 for i=2:n;
14     a(i)=-1;
15 end
16 for i=1:n
17     d(i)=2+cc*h^2;
18 end
19 for i=1:n-1
20     c(i)=-1;
21 end
22 b(1)=cc*h^2*ta+ya;
23 b(n)=cc*h^2*ta+yb;
24 for i=2:n-1
25     b(i)=cc*h^2*ta;
26 end
27 a
28 d
29 c
30 b
```

Fonte: Autoria própria (2024).

Listagem 10 – Resolução do exemplo 2 pelo código para sistemas tridiagonais

```

1 clear
2 tic
3 xa=0;
4 xb=10;
5 ya=40;
6 yb=200;
7 cc=0.01;
8 ta=20;
9 n=4;
10 h=(xb-xa)/(n+1);
11 for i=1:n;
12     x(i)=xa+i*h;
13 end
14 for i=2:n;
15     a(i)=-1;
16 end
17 for i=1:n
18     d(i)=2+cc*h^2;
19 end
20 for i=1:n-1
21     c(i)=-1;
22 end
23 b(1)=cc*h^2*ta+ya;
24 b(n)=cc*h^2*ta+yb;
25 for i=2:n-1
26     b(i)=cc*h^2*ta;
27 end
28 a
29 d
30 c
31 b
32 for i=1:n-1
33     d(i+1)=d(i+1)-(a(i+1)/d(i))*c(i);
34     b(i+1)=b(i+1)-(a(i+1)/d(i))*b(i);
35 end
36 y(n)=b(n)/d(n);
37 for i=n-1:-1:1
38     y(i)=(b(i)-c(i)*y(i+1))/d(i);
39 end
40 y
41 x=[xa x xb]
42 y=[ya y yb]
43 toc
44 plot(x,y,'*')

```

Fonte: Autoria própria (2024).

5 CONCLUSÃO

Este trabalho investigou o método das diferenças finitas, um recurso amplamente usado em uma variedade de problemas, mas que aqui foi focado na sua aplicação em equações diferenciais ordinárias. Utilizando o software Octave como uma ferramenta para implementação, o estudo mostrou como resolver tais equações por meio desse método.

Assim, através deste estudo podemos observar como a programação pode ser uma grande aliada na resolução de problemas matemáticos, fornecendo soluções precisas. A comparação dos tempos de execução de dois algoritmos de eliminação permitiu identificar qual o algoritmo mais eficaz para o problema estudado.

Por fim, este trabalho demonstrou a utilidade do método das diferenças finitas e também a força da programação como ferramenta para resolver problemas matemáticos de maneira eficiente.

REFERÊNCIAS

- ANTON, H.; RORRES, C. **Matrizes e sistemas de equações lineares**. 10. ed. Porto Alegre: Bookman, 2012. Recurso online.
- CHAPRA, S. C.; CANALE, R. P. **Métodos numéricos para engenharia**. 7. ed. Porto Alegre: AMGH, 2016. Recurso online.
- CUNHA, M. C. C. **Métodos numéricos**. 2. ed. Campinas: Editora Unicamp, 2000.
- KÜHLKAMP, N. **Matrizes e sistemas de equações lineares**. 3. ed. Florianópolis: UFSC, 2011.
- STEWART, J.; CLEGG, D.; WATSON, S. **Cálculo**. 6. ed. São Paulo: Cengage Learning, 2022. v. 2. Recurso online.
- ZILL, D. G. **Equações diferenciais com aplicações em modelagem**. 3. ed. São Paulo: Cengage Learning, 2016. Recurso online.