



Uso do Matlab no estudo de matrizes e sistemas lineares

Using Matlab to study matrices and linear systems

Camila Fredegoto Santos¹, Wellington José Corrêa²

RESUMO

A resolução de sistemas lineares é uma tarefa fundamental em várias áreas da matemática e engenharia, desempenhando um papel crucial na solução de problemas do mundo real. O MATLAB, uma poderosa plataforma de computação numérica, oferece uma ampla gama de ferramentas e recursos para abordar eficientemente esse tipo de problema. No que concerne este artigo, exploraremos como o MATLAB pode ser utilizado para resolver sistemas lineares. Além disso, o MATLAB oferece a capacidade de trabalhar com matrizes, tornando-o particularmente adequado para lidar com sistemas de equações lineares representados nessa forma. Através de exemplos práticos, ilustraremos como utilizar as funcionalidades do MATLAB para resolver sistemas lineares de diferentes tamanhos e complexidades. Por fim, destacaremos as vantagens de utilizar o MATLAB para resolução de sistemas lineares, incluindo sua eficiência computacional, capacidade de lidar com sistemas grandes e complexos, e a facilidade de visualização de resultados sendo uma grande ferramenta para ajudar o aluno em seus estudos e pesquisas.

PALAVRAS-CHAVE: álgebra linear; MATLAB; sistemas lineares.

ABSTRACT

Solving linear systems is a fundamental task in various fields of mathematics and engineering, playing a crucial role in addressing real-world problems. MATLAB, a powerful numerical computing platform, offers a wide range of tools and resources to efficiently tackle this type of problem. In this article, we will explore how MATLAB can be used to solve linear systems. Furthermore, MATLAB provides the capability to work with matrices, making it particularly suitable for handling systems of linear equations represented in that form. Through practical examples, we will illustrate how to use MATLAB's features to solve linear systems of different sizes and complexities. Finally, we will highlight the advantages of using MATLAB for solving linear systems, including its computational efficiency, ability to handle large and complex systems, and ease of result visualization, making it a valuable tool for assisting students in their studies and research.

KEYWORDS: linear algebra; MATLAB; linear systems.

INTRODUÇÃO

Sistemas lineares têm uma ampla gama de aplicações em matemática, física, engenharia, economia, ciência da computação e muitas outras disciplinas. Eles são usados para modelar e resolver problemas do mundo real, como circuitos elétricos, análise de estruturas, planejamento de produção e muito mais.

¹ Bolsista do CNPq. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: camilafredegoto@gmail.com. ID Lattes: 5269307111893849.

² Docente no Departamento Acadêmico de Matemática. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: wcorrea@utfpr.edu.br. ID Lattes: 1045931096324971.



No que concerne os métodos iterativos (item (ii)), estudamos o método de Gauss-Jacobi e o Método de Gauss-Seidel. No método de Gauss-Jacobi, todas as variáveis são atualizadas simultaneamente após cada iteração, com base nos valores das variáveis na iteração anterior. No método de Gauss-Seidel, as variáveis são atualizadas uma a uma à medida que são calculadas, com base nos valores atualizados já disponíveis.

É oportuno mencionar que, também estudamos vários tipos de matrizes, incluindo matrizes tridiagonais, estas que desempenham um papel essencial nos cálculos matemáticos. As matrizes tridiagonais têm uma estrutura especial com elementos diferentes de zero apenas nas três diagonais: principal, imediatamente acima e imediatamente abaixo. Tal conceito é muito útil em interpolação e discretização de equações diferenciais parciais. Como este trabalho se concentra no uso do MATLAB empregando estes métodos, para o estudo da teoria e aplicação dos métodos citados acima, recomendamos o leitor consultar as seguintes obras: (BURDEN, R. L.; FAIRES; BURDEN, A. M., 2016), (DORNELLES FILHO, 2016), (FRANCO, 2006) dentre outras.

MATERIAIS E MÉTODOS

Como mencionado anteriormente, o escopo deste trabalho é utilizar o MATLAB para resolver sistemas lineares empregando métodos numéricos descritos em (i) e (ii). Segundo (DORNELLES FILHO, 2016), O MATLAB (acrônimo de *Matrix Laboratory*) é um software que permite ao usuário efetuar cálculos via digitação direta de comandos e também a construção de programas que automatizam procedimentos de cálculo mais complexos. O MATLAB é uma ferramenta de larga utilização tanto no mundo acadêmico (ensino, pesquisa, etc.) quanto no profissional (desenvolvimento de produtos, análise de problemas, etc.). Tem um interface simples e intuitiva, e constitui ferramenta indispensável para o estudante de ciências exatas e engenharia.

Para cada um dos métodos expostos acima, foi gerado um arquivo na extensão (.m). Tal arquivo contém o código MATLAB, que pode incluir comandos para realizar cálculos, processar dados, criar gráficos e realizar uma variedade de tarefas numéricas. Eles são essencialmente *scripts* que podem ser executados no ambiente MATLAB para realizar tarefas específicas, contendo a sequência de comandos necessária para compilação. Especificamente em nosso trabalho, foram elaborados códigos dos métodos numéricos citados na introdução recorrendo aos pseudo-códigos dados em (BURDEN, R. L.; FAIRES; BURDEN, A. M., 2016). Em alguns códigos, basta o usuário inserir as matrizes A e B para obter a solução X do sistema Eq. (2). Em outras situações, ao executar o código, o usuário pode optar em inserir um arquivo no formato (.DTA) contendo por exemplo, a matriz aumentada do sistema Eq. (2) no qual o código devolve a solução do referido sistema linear.

As Figura 1 e 2 se referem ao código implementado da fatoração de Cholesky e eliminação de Gauss com pivotamento parcial, respectivamente:



Figura 1 – Método de Cholesky

```
clear;
[ret, name] = system('hostname');
if ret ~= 0,
    if ispc
        name = getenv('COMPUTERNAME');
    else
        name = getenv('HOSTNAME');
    end
end
disp('data:');
disp(datetime('now'));
disp('Nome do computador:');
disp(name);
A = input('Insira as entradas da matriz A: por exemplo, [4 2 -4;2 10 4; -4 4
9]\n ');
B=input('Insira a matriz B: por exemplo, [0; 6; 5]\n');
G=chol(A);
disp('A Matriz Triangular inferior G é');
disp(G);
disp('Solução do sistema linear G*Y=B:');
Y = G\B;
disp(Y);
disp('Solução do sistema linear G^t*X=Y:');
X=G\Y;
disp(X);
```

Fonte: Autoria própria (2023)

Figura 2 – Eliminação de Gauss com pivotamento parcial

```
clear;
[ret, name] = system('hostname');
if ret ~= 0,
    if ispc
        name = getenv('COMPUTERNAME');
    else
        name = getenv('HOSTNAME');
    end
end
disp('data:');
disp(datetime('now'));
disp('Nome do computador:');
disp(name);
A = input('Insira as entradas da matriz A: por exemplo, [1 2 3; 3 1 0;0 3
4]\n');
B=input('Insira a matriz B: por exemplo, [3; 4; 3]\n');
N = length(B);
X= zeros(N,1);
Aug = [A B];
disp('Matriz ampliada');
disp(Aug);
for j=1:N-1
    %partial pivoting
    [M,P] = max(abs(Aug(j:N,j)));
    C=Aug(j,:);
    Aug(j,:) = Aug(P+j-1,:);
    Aug(P+j-1,:) = C;
    %echlon form
    for i=j+1:N
        m= -Aug(i,j)/Aug(j,j);
        Aug(i,:) = Aug(i,.)+ m*Aug(j,:);
    end
    fprintf('%3s\n','Forma escalonada do sistema usando eliminação de Gauss com
pivotamento parcial');
    fprintf('%3s\n','Matriz ampliada:');
    Aug;
    disp(Aug);
end

x=zeros(N,1);
for i= N:-1:1
    x(i) = (Aug(i,N+1)-Aug(i,1:N)*x)/Aug(i,i);
end
disp('A solução do sistema linear A*X=B é');
disp(x);
```

Fonte: Autoria própria (2023)



RESULTADOS E DISCUSSÕES

Os algoritmos utilizados em nosso estudo forneceram como resultados apresentados, as saídas das implementações em MATLAB dos métodos estudados. As Figuras 3 e 4 nos dão as saídas dos códigos implementados apresentados nas Figuras 1 e 2, respectivamente:

Figura 3 – Método de Cholesky

```
Command Window
>> Cholesky
data:
  19-Sep-2023 10:33:00

Nome do computador:
LAPTOP-OR9KOMMI

Insira as entradas da matriz A: por exemplo, [4 2 -4; 2 10 4; -4 4 9]
[4 2 -4; 2 10 4; -4 4 9]
Insira a matriz B: por exemplo, [0; 6; 5]
[0; 6; 5]
A Matriz Triangular inferior G é
  2     1    -2
  0     3     2
  0     0     1

Solução do sistema linear G*Y=B:
  0
  2
  1

Solução do sistema linear G^t*X=Y:
  1
  0
  1

fx >>
```

Fonte: Autoria própria (2023)

Figura 4 – Eliminação de Gauss com pivotamento parcial

```
Command Window
Matriz ampliada
  1     2     3     3
  3     1     0     4
  0     3     4     3

Forma escalonada do sistema usando eliminação de Gauss com pivotamento parcial
Matriz ampliada:
  3.0000  1.0000     0  4.0000
  0  1.6667  3.0000  1.6667
  0  3.0000  4.0000  3.0000

Forma escalonada do sistema usando eliminação de Gauss com pivotamento parcial
Matriz ampliada:
  3.0000  1.0000     0  4.0000
  0  3.0000  4.0000  3.0000
  0     0  0.7778     0

A solução do sistema linear A*X=B é
  1
  1
  0

fx
```

Fonte: Autoria própria (2023)

Note que na Figura 3, o código apresenta além da solução do sistema linear Eq. (2), a matriz triangular inferior G e as soluções dos sistemas triangulares $GY = B$ e $G^T X = Y$. Por outro lado, a Figura 4, o código fornece além da solução, todas as matrizes ampliadas com pivotamento parcial. Em tais situações, além de fornecer a resposta do sistema linear proposto usando determinado método, tais códigos oferecem de forma didática a resolução do exercício, no qual o usuário e graduando em



alguma disciplina que exija a solução de um sistema linear, pode conferir uma etapa intermediária à solução do sistema.

CONCLUSÃO

O MATLAB se mostrou uma importante ferramenta na obtenção da solução de sistemas lineares dados em Eq. (2). Além disso, o MATLAB permite que os educadores ensinem conceitos matemáticos e científicos complexos de uma maneira mais acessível e visual. Ele pode ser usado para ilustrar conceitos de álgebra, cálculo, estatísticas, equações diferenciais, dentre outros, tornando o aprendizado desses tópicos mais concreto e prático.

Agradecimentos

Agradeço ao meu orientador professor Dr. Wellington José Corrêa, pelo apoio e contribuições durante a atuação do projeto de pesquisa. Agradeço, também, ao CNPq pelo fomento financeiro.

Disponibilidade de Código

A implementação computacional desenvolvida está disponível no *Google Drive*, por meio do link: <https://drive.google.com/drive/folders/1BhyffsosOb6wW9ilPXXfXNUk-giuxa8y?usp=sharing>.

Conflito de interesse

Não há conflito de interesse.

REFERÊNCIAS

- BURDEN, Richard L; FAIRES, J Douglas; BURDEN, Annette M. **Análise numérica**. [S.l.]: Cengage Learning, 2016.
- DORNELLES FILHO, Adalberto Ayjara. **Fundamentos de cálculo numérico**. [S.l.]: Bookman Editora, 2016.
- FRANCO, Neide Bertoldi. **Cálculo numérico**. [S.l.]: Pearson, 2006.