

Universidade Estadual Paulista Julio de Mesquita Filho - UNESP
Instituto de Biociências
Departamento de Bioestatística, Biologia Vegetal, Parasitologia e
Zoologia
Programa de Pós-Graduação em Biometria
Métodos Numéricos e Computacionais

Trabalho Computacional 1 - Métodos Diretos para Sistemas Lineares

Aluno(a): Richard Castro Júnior

Professor: Daniela Renata Cantane

Botucatu-SP

2022

1 Introdução

1.1 Sistemas lineares

Um sistema linear de m equações e n incógnitas é um conjunto de m equações lineares onde cada uma possui n variáveis. Seja $A = [a_{ij}]$ uma matriz de ordem $m \times n$ definida sobre \mathbb{R} , isto é, seus elementos $a_{i,j} \in \mathbb{R}$ para $1 \leq i \leq m$ e $1 \leq j \leq n$. Consideremos o problema de encontrar escalares $x_1, \dots, x_n \in \mathbb{R}$ satisfazendo simultaneamente o seguinte sistema de equações lineares (PULINO, 2012):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = y_1 \\ \vdots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = y_i \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = y_m \end{cases}$$

conhecendo os escalares $y_1, \dots, y_n \in \mathbb{R}$.

O sistema linear pode ser representado em sua forma matricial $AX = Y$, onde

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & a_{in} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} \quad \text{e} \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_m \end{bmatrix}$$

sendo a matriz A de ordem $m \times n$ a matriz dos coeficientes do sistema linear; o vetor coluna X de ordem $n \times 1$, o vetor de incógnitas; e o vetor coluna Y de ordem $m \times 1$, o vetor do lado direito do sistema linear (PULINO, 2012).

O sistema $AX = Y$ pode ser representado também pela sua matriz aumentada:

$$[A \mid Y] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & y_1 \\ a_{21} & a_{22} & \dots & a_{2n} & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & y_n \end{array} \right]$$

1.2 Sistemas triangulares

Denomina-se sistema triangular superior a todo sistema $AX = Y$ em que $a_{ij} = 0 \quad \forall j < i$, ou seja, a sistemas da forma (SOUZA, 2022):

$$\left\{ \begin{array}{cccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + \dots + a_{1n}x_n & = & y_1 \\ & & a_{22}x_2 & + & a_{23}x_3 & + \dots + a_{2n}x_n & = & y_2 \\ & & & & a_{33}x_3 & + \dots + a_{3n}x_n & = & y_3 \\ & & & & & & \vdots & \vdots \\ & & & & & & a_{nn}x_n & = & y_n \end{array} \right.$$

Tais sistemas são resolvidos por substituições retroativas, por meio de equações da

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}} \quad \forall i = n, \dots, 1$$

Denomina-se sistema triangular inferior a todo sistema $Ax = b$ em que $a_{ij} = 0 \quad \forall j > i$ ou seja, a sistemas da forma (SOUZA, 2022):

$$\left\{ \begin{array}{cccccc} a_{11}x_1 & & & & & = & y_1 \\ a_{21}x_1 & + & a_{22}x_2 & & & = & y_2 \\ \vdots & \vdots & \vdots & & & \vdots & y_3 \\ a_{n1}x_1 & + & a_{n2}x_2 & + \dots + a_{nn}x_n & = & y_n \end{array} \right.$$

Tais sistemas são resolvidos por substituições progressivas, por meio de equações da

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}} \quad \forall i = n, \dots, 1$$

Para a solução de um sistema triangular superior ou inferior de ordem n , o esforço computacional (quantidade de operações elementares necessárias para calcular a solução do problema) é $E_C = n^2$, sendo:

- n operações de divisão;
- $n(n-1)/2$ operações de adição/subtração;
- $n(n-1)/2$ operações de multiplicação.

1.3 Métodos diretos

Os métodos diretos são métodos que, a menos de erros de arredondamento, fornecem a solução exata do sistema linear, caso ela exista, após um número finito de operações. A característica dos

Métodos Diretos é que eles transformam as equações lineares originais em equações equivalentes. Para isso são utilizadas 3 operações elementares, que não alteram a solução, mas elas alteram a determinante dos coeficientes da matriz: troca da posição de duas linhas, multiplicação de uma equação por um constante diferente de zero e multiplicação de uma equação por um constante diferente de zero e subtração por outro linha de equação (CENDRON, 2020).

1.3.1 Método de Eliminação de Gauss

Considera-se um sistema de equações lineares $Ax = b$, onde A é uma matriz (de números reais) quadrada de ordem n e não singular (possui matriz inversa e o determinante não é nulo).

O Método de Eliminação de Gauss consiste em transformar este sistema num sistema triangular superior equivalente aplicando a sequência das 3 operações elementares mencionadas anteriormente. Disso, obtém-se um novo sistema $A'x = b'$, que seja triangular superior, tal que os sistemas $Ax = b$ e $A'x = b'$ são equivalentes (não se altera a solução, mas a determinante dos coeficientes da matriz), uma vez que A é uma matriz não singular, garantindo que a solução obtida seja única.

1.3.2 Método de Decomposição LU

Considere um sistema linear $Ax = b$, onde a matriz A é não singular. A fim de resolver o sistema, podemos fatorar a matriz A como o produto de uma matriz L triangular inferior e uma matriz U triangular superior, ou seja, $A = LU$ (4.4... 2020).

Sendo assim, o sistema pode ser reescrito da seguinte forma (4.4... 2020):

$$Ax = b(LU)x = bL(Ux) = bLy = b \quad e \quad Ux = y$$

Isto significa que, ao invés de resolvermos o sistema original, podemos resolver o sistema triangular inferior $Ly = b$ e, então, o sistema triangular superior $Ux=y$, o qual nos fornece a solução de $Ax = b$ (4.4... 2020).

A matriz U da fatoração LU é a matriz obtida ao final do escalonamento da matriz A . A matriz L é construída a partir da matriz identidade I , ao longo do escalonamento de A . Os elementos da matriz L são os múltiplos do primeiro elemento da linha de A a ser zerado dividido pelo pivô acima na mesma coluna (4.4... 2020).

1.3.3 Método de Decomposição de Cholesky

A decomposição de Cholesky procura decompor uma matriz A na forma $A = GG^T$, onde G é uma matriz triangular inferior com elementos da diagonal principal estritamente positivos (ANDRADE, 2022).

Uma matriz A é dita definida positiva se A for simétrica e se $x^T Ax > 0$, para todo $x \neq 0$. Uma matriz simétrica A é definida positiva se, e somente se, o processo de eliminação de Gauss pode ser realizado sem permutação de linhas ou colunas e tem todos os elementos pivots positivos. Então, uma matriz simétrica A é definida positiva se, e somente se, pode ser fatorada como GG^T , onde G é uma matriz triangular inferior com elementos positivos na diagonal (ANDRADE, 2022).

Conhecidos $A = GG^T$, para resolver o sistema linear $Ax = b$, deve-se resolver dois sistemas lineares:

$$Ax = b \Rightarrow GG^T x = b \Rightarrow \begin{cases} Gy = b \\ G^T x = y \end{cases}$$

2 Objetivos

Implementar computacionalmente os métodos Eliminação de Gauss, Decomposição LU e Decomposição de Cholesky; fazer os três ordenamentos das matrizes vistos (Método do Mínimo Grau, Método Forma Bloco Triangular e Método de Triangularização de Björck) e comparar todos os casos em relação à densidade/esparsidade da matriz.

3 Procedimento

Os métodos diretos (Eliminação de Gauss, Decomposição LU e Decomposição de Cholesky) foram implementados utilizando o MATLAB ®. Trata-se de uma linguagem de programação que expressa diretamente matrizes e arranjos matemáticos, combinando um ambiente de desktop ajustado para análise iterativa e processos de design com uma linguagem de programação.

Em seguida, foram implementados os reordenamentos: Mínimo Grau, Método Forma Bloco Triangular e Método de Triangularização de Björck. A densidade/esparsidade da matriz com e sem reordenamento foram, então, analisados e comparados.

Por fim, um pré-condicionador foi implementado para o Método de Cholesky e os resultados do método com e sem o uso de pré-condicionador foram comparados.

3.1 Métodos diretos

3.1.1 Método de Eliminação de Gauss

Inicialmente, é preciso construir a seguinte matriz aumentada:

$$(A^{(1)} | b^{(1)}) = (A | b) = \left(\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & a_{1n+1}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & a_{2n+1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & a_{nn+1}^{(1)} \end{array} \right)$$

em que $a_{ij}^{(1)} = a_{ij}$, $i, j = 1, \dots, n$, $a_{11}^{(1)} \neq 0$, $a_{in+1}^{(1)} = b_i$.

No passo k , o elemento $a_{kk}^{(k)}$ é chamado pivô e os elementos $m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ são chamados de multiplicadores do passo k .

Passo 1: Eliminar a variável x_1 das equações $2, \dots, n$

$$(a_{11}^{(1)} \neq 0)$$

Para $i = 2$ até n faça

$$m_{i1} = a_{i1}^{(1)} / a_{11}^{(1)}$$

$$L_i^{(2)} \leftarrow L_i^{(1)} - m_{i1} * L_1^{(1)}$$

Nova matriz:

$$\left(\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & a_{1n+1}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & a_{2n+1}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & a_{nn+1}^{(2)} \end{array} \right), \quad \begin{array}{l} \text{em que } a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1} * a_{ij}^{(1)} \\ i = 2, \dots, n, j = 1, \dots, n + 1. \end{array}$$

Passo 2: Eliminar a variável x_2 das equações $3, \dots, n$

$$\left(a_{22}^{(2)} \neq 0\right)$$

Para $i = 3$ até n faça

$$m_{i2} = a_{i2}^{(2)} / a_{22}^{(2)}$$

$$L_i^{(3)} \leftarrow L_i^{(2)} - m_{i2} * L_2^{(2)}$$

Nova matriz:

$$\left(\begin{array}{ccccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & a_{1n}^{(1)} & a_{1n+1}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} & a_{2n+1}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \dots & a_{3n}^{(2)} & a_{3n+1}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & & \\ \vdots & & & & & \\ 0 & 0 & a_{n3}^{(3)} & \dots & a_{nn}^{(3)} & a_{nn+1}^{(3)} \end{array} \right) \quad \begin{array}{l} i = 3, \dots, n \\ j = 2, \dots, n+1 \end{array}$$

Passo n-1: Eliminar a variável x_{n-1} da equação n

$$\left(a_{n-1n-1}^{(n-1)} \neq 0\right)$$

$$m_{nn-1} = a_{nn-1}^{(n-1)} / a_{n-1n-1}^{(n-1)}$$

$$L_n^{(n)} \leftarrow L_n^{(n-1)} - m_{nn-1} * L_{n-1}^{(n-1)}$$

Matriz final:

$$\left(\begin{array}{ccccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & a_{1n+1}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & a_{2n+1}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{nn}^{(n)} & a_{nn+1}^{(n)} \end{array} \right), \quad \begin{array}{l} a_{nj}^{(n)} = a_{nj}^{(n-1)} - m_{nn-1} * a_{n-1j}^{(n-1)}, \\ j = n-1, n+1. \end{array}$$

Assim, obtemos um sistema triangular superior equivalente ao original que deve ser resolvido como visto anteriormente.

3.1.2 Método de Decomposição LU

As matrizes L e U são determinadas usando a ideia básica do Método de Eliminação de Gauss. Neste caso, os elementos $I_{ij}, i > j$ são os multiplicadores m_{ij} obtidos no processo de eliminação de Gauss e os elementos u_{ij} são os mesmos elementos obtidos no final da fase de triangularização do Método de Eliminação de Gauss.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix} = LU$$

Considere $A^{(1)} = A$. O cálculo do Passo 1 (que é eliminar os elementos abaixo de $a_{11}^{(1)}$) é equivalente a pré-multiplicar $A^{(1)}$ por uma matriz M_1 , em que

$$M_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -m_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ -m_{n1} & 0 & \cdots & 1 \end{pmatrix}.$$

Assim, ao final do Passo 1 teremos

$$A^{(2)} = M_1 A^{(1)}.$$

Do mesmo modo, no Passo 2, para eliminar os elementos abaixo de $a_{22}^{(2)}$ é só pré-multiplicarmos a matriz anterior por:

$$M_2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & -m_{32} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & -m_{n2} & \cdots & 1 \end{pmatrix}$$

e assim, ao final do Passo 2 nossa matriz é dada por:

$$A^{(3)} = M_2 M_1 A^{(1)}$$

No final dos $n - 1$ passos, nossa matriz escalonada será dada por:

$$A^{(n)} = \underbrace{M_{n-1} \cdots M_2 M_1}_M A^{(1)}.$$

Isso significa que nossa matriz escalonada será triangular superior, que será chamada de U :

$$A^{(n)} = M A^{(1)} = M A = U.$$

Como cada matriz M_k é não-singular ($\det(M_k) = 1 \neq 0$) e

$$M_k^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & m_{k,k+1} & \ddots & 0 & 0 \\ 0 & 0 & \vdots & 0 & \ddots & 0 \\ 0 & 0 & m_{nk} & 0 & 0 & 1 \end{pmatrix}$$

então $M = M_{n-1} \cdots M_2 M_1$ também é. Além disso,

$$M^{-1} = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1}.$$

É fácil verificar que

$$M^{-1} = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ m_{31} & m_{32} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ m_{n1} & m_{n2} & \cdots & 1 \end{pmatrix},$$

isto é, a matriz tem diagonal unitária e abaixo dela, tem-se todos os multiplicadores que foram usados nas $n - 1$ etapas para zerar cada elemento.

Assim,

$$MA = U \implies A = M^{-1}U,$$

então chamando

$$M^{-1} = L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ m_{31} & m_{32} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ m_{n1} & m_{n2} & \cdots & 1 \end{pmatrix},$$

temos $A = LU$ a fatoração desejada.

Para fatorar a matriz A no produto de uma matriz triangular inferior L de diagonal unitária por uma matriz triangular superior U , basta fazermos a Eliminação de Gauss. Guardamos os multiplicadores para zerar cada elemento de A e com eles montamos a matriz L . A matriz escalonada final é a matriz U .

3.1.3 Método de Decomposição de Cholesky

Considerando A simétrica e definida positiva, podemos escrever:

$$A = \begin{pmatrix} g_{11} & 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & 0 & \cdots & 0 \\ g_{31} & g_{32} & g_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_{n1} & g_{n2} & g_{n3} & \cdots & g_{nn} \end{pmatrix} \begin{pmatrix} g_{11} & g_{21} & g_{31} & \cdots & g_{n1} \\ 0 & g_{22} & g_{32} & \cdots & g_{n2} \\ 0 & 0 & g_{33} & \cdots & g_{n3} \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & g_{nn} \end{pmatrix}$$

Agora, determinamos os elementos g_{ij} :

1. Construção dos elementos da diagonal de G .

O produto da linha 1 e coluna 1 resulta em

$$g_{11}^2 = a_{11} \Rightarrow g_{11} = \sqrt{a_{11}}$$

O produto da linha 2 e coluna 2 resulta em

$$g_{21}^2 + g_{22}^2 = a_{22} \Rightarrow g_{22} = (a_{22} - g_{21}^2)^{1/2}$$

O produto da linha 3 e coluna 3 resulta em

$$g_{31}^2 + g_{32}^2 + g_{33}^2 = a_{33} \Rightarrow g_{33} = (a_{33} - g_{31}^2 - g_{32}^2)^{1/2}$$

O produto da linha j e coluna j resulta em

$$g_{j1}^2 + g_{j2}^2 + \cdots + g_{jj}^2 = a_{jj} \Rightarrow g_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2 \right)^{1/2}, \text{ com}$$

$$j = 1, 2, \dots, n.$$

2. Construção dos elementos não pertencentes à diagonal de G .

- Primeira coluna de G :

O produto da linha 2 e coluna 1 resulta em

$$g_{21}g_{11} = a_{21} \Rightarrow g_{21} = a_{21}/g_{11}$$

O produto da linha i e coluna 1 resulta em

$$g_{i1}g_{11} = a_{i1} \Rightarrow g_{i1} = a_{i1}/g_{11}, i = 2, 3, \dots, n$$

- Segunda coluna de G :

O produto da linha 3 e coluna 2 resulta em

$$g_{31}g_{21} + g_{32}g_{22} = a_{32} \Rightarrow g_{32} = (a_{32} - g_{31}g_{21})/g_{22}$$

O produto da linha i e coluna 2 resulta em

$$g_{i1}g_{21} + g_{i2}g_{22} = a_{i2} \Rightarrow g_{i2} = (a_{i2} - g_{i1}g_{21}) / g_{22},$$

$$i = 3, 4, \dots, n$$

- Terceira coluna de G :

O produto da linha 4 e coluna 3 resulta em

$$g_{41}g_{31} + g_{42}g_{32} + g_{43}g_{33} = a_{43} \Rightarrow g_{43} = (a_{43} - g_{41}g_{31} - g_{42}g_{32}) / g_{33}$$

O produto da linha i e coluna 3 resulta em

$$g_{i1}g_{31} + g_{i2}g_{32} + g_{i3}g_{33} = a_{i3} \Rightarrow g_{i3} = (a_{i3} - g_{i1}g_{31} - g_{i2}g_{32}) / g_{33},$$

$$i = 4, 5, \dots, n$$

- Coluna j de G :

O produto da linha $i > j$ e coluna j resulta em

$$g_{i1}g_{j1} + g_{i2}g_{j2} \cdots + g_{ij}g_{jj} = a_{ij} \Rightarrow g_{ij} = \frac{1}{g_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} g_{ik}g_{jk} \right)$$

com $i = j + 1, \dots, n$.

Primeiramente, é determinadi o primeiro elemento da coluna 1. Em seguida, todos os elementos abaixo dele. Determine o primeiro elemento da coluna 2. Sequencialmente, todos os elementos abaixo do mesmo. O processo continua até a última coluna.

Observação: O fato de A ser definida positiva garante que os termos dentro da raiz quadrada serão sempre positivos; caso não o seja, então A não terá fatoração de Cholesky.

3.2 Matriz de Hilbert

Uma função que receba um inteiro n e que retorne uma matriz de Hilbert de ordem n foi implementada e testados para todos os programas implementados utilizando a matriz de Hilbert

para resolver o sistema linear com

$$a_{ij} = \frac{1}{i+j-1} \quad b = \frac{1}{3} \sum_{j=1}^n a_{ij}$$

Foram considerados $n = 5$, $n = 10$, $n = 50$, $n = 100$ e $n = 500$, cujos resultados foram analisados e comparados.

A função *hilb*(n) do MATLAB ® foi utilizada para comparação com os resultados de matriz de Hilbert.

4 Resultados e discussão

4.1 Matriz de Hilbert

Implementados os cálculos para as matrizes de Hilbert, os resultados foram organizados de acordo com o método implementados. Para cada um deles, foram comparados os resultados para o código do *script* e a função *hilb*(n) do MATLAB, analisando o tempo computacional e o número de iterações. Os resultados estão dispostos conforme as tabelas 1, 2, 3, 4, 5 e 6 abaixo:

Tabela 1: Implementação do Método Gauss com código do script.

Ordem da matriz de Hilbert	Tempo computacional (s)	Número de iterações
5	3.032880e-02	353
10	4.300620e-02	2077
20	2.788370e-02	13648
28	4.383400e-02	35122

Tabela 2: Implementação do Método Gauss com função *hilb*(n).

Ordem da matriz de Hilbert	Tempo computacional (s)	Número de iterações
5	4.448710e-02	278
10	4.136300e-02	1777
20	2.722180e-02	12448
28	4.136300e-02	32770

Tabela 3: Implementação da Decomposição LU com código do script.

Ordem da matriz de Hilbert	Tempo computacional (s)	Número de iterações
5	2.549590e-02	384
10	2.818760e-02	978
20	3.608000e-04	6809
28	4.733300e-02	35015

Tabela 4: Implementação da Decomposição LU com função hilb(n).

Ordem da matriz de Hilbert	Tempo computacional (s)	Número de iterações
5	3.480470e-02	309
10	3.325100e-03	1823
20	3.946220e-02	12449
28	4.136300e-02	32663

Tabela 5: Implementação da Decomposição de Cholesky com código do script.

Ordem da matriz de Hilbert	Tempo computacional (s)	Número de iterações
5	4.102990e-02	87
10	4.346320e-02	1905
12	4.656350e-02	3006

Tabela 6: Implementação da Decomposição de Cholesky com função hilb(n).

Ordem da matriz de Hilbert	Tempo computacional (s)	Número de iterações
5	4.326280e-02	301
10	4.485340e-02	1605
12	4.700660e-02	458

A partir da ordem 29, a matriz de Hilbert tem determinante igual a zero. Deste modo, as matrizes de ordens maiores ou igual a 29 são matrizes singulares, isto é, não possuem matriz inversa. Então, a solução não é única para qualquer b , ou seja, o sistema tem infinitas soluções. Por isso, consideramos até a ordem 28 da matriz de Hilbert.

No caso das matrizes calculadas pela Decomposição de Cholesky, considerou-se até a ordem 12, pois, a partir da ordem 13, a matriz A não é definida positiva, dado que $\det(A_k) \geq 0$ para todo k .

$= 1, 2, \dots, n$, não admitindo a decomposição de Cholesky.

Vemos que não é vantajoso trabalhar com a Decomposição de Cholesky, ao menos para a matriz de Hilbert. Primeiramente, não é possível trabalhar com cálculos até a ordem 28, pois, a partir da ordem 13, a matriz de Hilbert não é definida-positiva. Apesar de não haver muitas iterações para calcular a solução, o tempo computacional é maior comparado aos outros métodos diretos. Podemos notar que o tempo computacional para $n=5$ na Decomposição de Cholesky é aproximadamente igual ao tempo computacional de $n=28$ para o Método de Gauss e para a Decomposição LU. Não obstante, não há uma variação significativa do tempo computacional na Decomposição de Cholesky de $n=5$ até $n=12$.

Uma comparação entre o Método de Gauss e a Decomposição LU mostra que a Decomposição é mais vantajosa para operar os cálculos. Analisando o número de iterações, notamos uma semelhança na variação conforme o aumento da ordem da matriz para ambos os métodos, mas o tempo computacional para a Decomposição é melhor, tendo em vista que, evidentemente para as menores ordens, ele é menor, o que reduz o custo computacional. Para maiores ordens de matriz, a diferença não é tão significativa quanto para as menores ordens. Vê-se uma variação considerável na ordem 20 no Método de Gauss, entretanto a Decomposição LU é apresentada mais relevância nos resultados apresentados, levando em conta, ainda, que pouco mais de iterações e tempo computacional similar a do Método de Gauss.

Além disso, a análise dos métodos diretos aplicados mostram que há uma falta de precisão conforme o aumento da ordem da matriz. Apesar das soluções serem exatas e únicas, ordens maiores, as matrizes de Hilbert apresentam um problema de mau condicionamento, isto é, o aumento da ordem da matriz faz com que os elementos da matrizes aproximem-se do problema de singularidade. Isso resulta em propagação de erros de forma mais relevante, tal que soluções podem ficar cada vez mais distintas, em outras palavras, acarretam em grandes variações na solução do sistema.

4.2 Aplicação do métodos de reordenamento

Os próximos cálculos também foram operados com os métodos anteriormente utilizados, mas, desta vez, foram acrescentados cálculos com reordenamentos aplicados também. Foi utilizada uma matriz extraída do site <https://math.nist.gov/MatrixMarket/>, que fornece acesso a um repositório de dados de teste para uso em estudos comparativos de algoritmos para álgebra linear numérica. A matriz escolhida foi `fs_541_1.mtx`, real e assimétrica, de ordem 514 e com 4285 entradas (dis-

ponível em: https://math.nist.gov/MatrixMarket/data/Harwell-Boeing/smtape/fs_541_1.html).

Nas tabelas 7, 8 e 9, temos os resultados obtidos a partir das cálculos feitos, constando o tempo computacional (em segundos), a esparsidade da matriz, o grau de esparsidade da mesma e número de iterações do cálculo:

Tabela 7: Implementação do Método Gauss.

Método de Reordenamento	Tempo computacional (s)	Esparsidade	Grau de esparsidade	Número de iterações
Sem reordenaento	7.940e+02	4.2161e+10	9.99999898e-01	212432177
Mínimo Grau	1.359e+03	4.2056e+10	9.99999898e-01	212432719
Bloco Triangular	1.489e+03	4.2056e+10	9.99999898e-01	63628356
Triangularização de Björck	1.753e+03	4.2295e+10	9.99999899e-01	212432720

Tabela 8: Implementação da Decomposição LU.

Método de Reordenamento	Tempo computacional (s)	Esparsidade	Grau de esparsidade	Número de iterações
Sem reordenaento	6.716e+02	4.2161e+10	9.99999898e-01	212290969
Mínimo Grau	5.609e+02	4.2056e+10	9.99999898e-01	212583111
Bloco Triangular	5.917e+02	4.2056e+10	9.99999898e-01	212583111
Triangularização de Björck	7.351e+02	4.2295e+10	9.99999899e-01	212291512

Tabela 9: Implementação da Decomposição de Cholesky.

Método de Reordenamento	Tempo computacional (s)	Esparsidade	Grau de esparsidade	Número de iterações
Sem reordenaento	2.556e+02	4.2161e+10	9.99999898e-01	159804775
Pré-condicionamento	2.437e+02	4.2586e+10	9.99999899e-01	159806860
Mínimo Grau	Matriz não definida positiva			
Bloco Triangular	Matriz não definida positiva			
Triangularização de Björck	Matriz não definida positiva			

Pelo Método de Gauss, observa-se uma ineficiência em implementar os reordenamentos, pois há um aumento no tempo computacional e não há uma mudança significativa no número de iterações e no grau de esparsidade. No entanto, ao implementar o reordenamento conforme o método de Formulação de Bloco Triangular, há uma grande diferença no número de iteração, cuja diferença é de, aproximadamente, 70%, o que não se reflete no tempo computacional, tendo em vista que, para o cálculo com reordenamento, este tempo basicamente dobra. Computacional, portanto, o gasto computacional é melhor, implementando o Método de Gauss, se for incorporado qualquer

um dos reordenamento, apesar de uma considerável mudança na esparsidade da matriz.

Em contrapartida ao primeiro método analisado, a Decomposição LU apresenta uma outra ocorrência, tendo em vista que a implementação do método junto aos reordenamentos mostra uma melhora e maior eficiência nos cálculos. Esta eficiência é visualizada a partir da comparação entre o grau de esparsidade e o número de iterações com o tempo computacional. Sem grandes diferenças das duas primeiras variáveis, o tempo computacional é reduzido pela incorporação dos métodos de reordenamento de Mínimo Grau e de Formulação de Bloco Triangular, cujas diferenças são, respectivamente, de 16,5% e de 11,9%. Deste modo, o reordenamento por Mínimo Grau é o mais eficiente na aplicação da Decomposição LU. O método de reordenamento de Triangularização de Björck mostrou-se ineficaz, pois, nas mesmas condições postas previamente, o tempo computacional aumentou, aproximadamente, 9,5%.

Na implementação da Decomposição de Cholesky, houve uma dificuldade na execução do cálculo, não pela implementação ou truncamento, mas pela dificuldade em encontrar matrizes cujos reordenamentos resultassem numa matriz definida positiva. Aplicando as matrizes (que fossem quadradas, reais e densas ou esparsas) do acervo utilizado, uma parte das matrizes encontradas não eram definidas positivas ou não era possível aplicar os métodos de reordenamentos, pois as matrizes tornavam-se não definidas positivas (ou eram definidas negativas ou com pontos críticos que são pontos de sela). Assim, utilizando a mesma matriz aplicadas para os outros métodos, só foi possível aplicar o pré-condicionamento, com base numa função do próprio MATLAB. O pré-condicionamento mostrou-se eficaz, uma vez que o tempo computacional reduziu em torno de 4,7% com ligeiras modificações na esparsidade, no grau de esparsidade e no número de iterações.

Os métodos diretos implementados forneceram uma solução exata do sistema linear, após um número finito de operações, entretanto são métodos cuja desvantagem está nos erros de arredondamentos acumulativos, além de serem lentos e imprecisos, dependendo da estrutura e tamanho da matriz. Ainda sim, não há envolvimento de erro de truncamento, isto é, limitações do número de dígitos à direita da vírgula decimal, ao menos pelos métodos aplicados (deve-se levar em conta o processamento das máquinas, cuja capacidade é limitada para armazenar dígitos significativos de valores numéricos).

Comparando ambos os métodos, nota-se que a Decomposição de Cholesky é o melhor método a ser aplicado na resolução de sistemas lineares, buscando uma solução única, ou seja, cujo sistema seja consistente, possível e determinado. Mesmo comparados aos cálculos operados com métodos de

reordenamentos eficientes pela Decomposição LU, a Decomposição de Cholesky tem menores tempos computacionais e menos iterações, consequentemente, gerando menores custos computacionais. Ressalta-se ainda que a incorporação do pré-condicionamento, inclusive, trazem mais eficiência nos cálculos, mas esta decomposição não mostrou eficaz na aplicação de métodos de reordenamentos, pois, no conjunto de possíveis matrizes, as matrizes definidas negativas ou matrizes cujos pontos críticos são pontos de sela.

5 Conclusão

Foram implementar computacionalmente os métodos diretos de resolução de sistemas lineares, onde foram feitos os três ordenamentos das matrizes. Comparando os casos em relação à esparsidade da matriz, números de iterações, grau de esparsidade e tempo computacional, a Decomposição de Cholesky apresentou os melhores resultados e mostrou-se o método mais eficiente para a resolução de sistema, apesar de ter uma limitação relacionada ao acervo de matrizes utilizados e pela exigência de que a matriz seja definida positiva. A análise dos resultados e das implementações provou que os métodos diretos apresentam uma solução exata, cujo número de operações é finito, e há também a vantagem de haver erros de truncamento, no entanto conta-se com erros de arredondamentos acumulativos e lentidão e imprecisão segundo a estrutura e tamanho da matriz, o que foi observado na análise dos cálculos das matrizes de Hilbert. Ainda sim, não há envolvimento de erro de truncamento, isto é, limitações do número de dígitos à direita da vírgula decimal, ao menos pelos métodos aplicados (deve-se levar em conta o processamento das máquinas, cuja capacidade é limitada para armazenar dígitos significativos de valores numéricos).

6 Referências

4.4 Fatoração LU. 2020. Disponível em: https://www.ufrgs.br/reamat/CalculoNumerico/livro-oct/sdsl-fatoracao_lu.html. Acesso em: 21 set. 2022.

ANDRADE, Doherty. Decomposição LU e Cholesky. Disponível em: <http://www.dma.uem.br/kit/topicos-especiais/cholesky.pdf>. Acesso em: 21 set. 2022.

CENDRON, Marcelo. Definição de sistemas lineares. 2020. Disponível em: <http://userdir.luzerna.ifc.edu.br/>

celocendron/tutoriais/calculo/sist _linear01.php. Acesso em: 21 set. 2022.

PULINO, Petronio. **Álgebra Linear e suas Aplicações**. Campinas: Pulinus, 2012.

SOUZA, Marcone Jamilson Freitas. Sistemas Lineares. Disponível em: <http://www.decom.ufop.br/prof/marc>
Acesso em: 21 set. 2022.