ESTRATÉGIAS DE PIVOTEAMENTO PARCIAL E TOTAL NA FATORAÇÃO LU PARA AUMENTAR ESPARSIDADE

Fillipe Rafael Bianek Pierin
UFPR - Análise Numérica I - 2019/01
bianekpierin@gmail.com

1. Introdução

Matrizes esparsas têm aplicações em problemas de computação, para armazenamento de dados; engenharia, física, no método das malhas para resolução de circuitos elétricos. E também essas matrizes são usadas na resolução de sistemas lineares. Essas matrizes, por definição, são matrizes que contém grande quantidade de elementos nulos.

A resolução de sistemas lineares $Ax = b$, com $A \in \mathbf{R}^{n\times n}$ e $b \in \mathbf{R}^n$, requer que a matriz A seja invertível, porém em problemas práticos a invertibilidade de uma matriz é computacionalmente custoso. Para facilitar os cálculos computacionais, sugerimos o uso de algum tipo de fatoração na matriz A. Empregaremos neste projeto a fatoração LU. Nesta fatoração, buscamos que nos fatores L e U os elementos nulos não sejam preenchidos ou tenham pouco preenchimento, para isso utilizaremos a estratégia de LU esparso.

O objetivo deste projeto será maximizar a estabilidade em conjunto com a esparsidade usando na fatoração LU a estratégia do pivoteamento de Markowitz com o pivoteamento completo, conforme [1], que será chamada de LU esparso. Desta forma, queremos diminuir o preenchimento da matriz A, ou seja, se temos k elementos nulos na matriz A, ou nos fatores L e U, desejamos que após a fatoração LU os fatores tenham número igual ou menor de elementos nulos.

O projeto está organizado da seguinte maneira: Na Seção 2 apresenta-se a fundamentação teórica, explicando a fatoração LU com pivoteamento parcial ou completo, LU com estratégia de Markowitz, a resolução de sistemas $Ax = b$ usando a fatoração LU, apresentando os respectivos algoritmos. Na Seção 3 mostramos os resultados das implementações, apresentados exemplos e gráficos ilustrando a estratégia usada para maximizar a estabilidade em conjunto com a esparsidade da fatoração LU. Por fim, na Seção 4 temos uma conclusão com os principais aspectos concluídos da análise, implementação e exemplos realizados nas seções anteriores.

2. Fundamentação teórica

2.1. Fatoração LU

Pela técnica de decomposição LU, a matriz A, não singular, é decomposta no produto de duas matrizes L e U, sendo L é uma matriz triangular inferior e U é uma matriz triangular superior. Ou seja,

$$A = LU.$$

A decomposição LU baseia-se no método de eliminação de Gauss ou pivoteamento da matriz A. Esse método consiste em fazer operações elementares de um sistema $Ax = b$, na matriz ampliada, até que em $n-1$ obtemos um sistema triangular superior $Ux = b$. Sendo que a matriz ampliada é

$$\left[\ A \mid b\ \right] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right]. \tag{1}$$

Definição 1 (Operações Elementares) *Considere $A \in \mathbf{R}^{n\times n}$, L_i a linha i da matriz A e $\alpha \in \mathbf{R}$. As operações elementares que operamos em $Ax = b$ no método de eliminação de Gauss são* *(i) Permutar duas linhas da matriz:* $L_i \longleftrightarrow L_j$*;*

(ii) Multiplicar uma linha por um escalar $\alpha \neq 0$*:* $L_i \longleftarrow \alpha L_i$*;*

(iii) Somar à uma linha da matriz outra multiplicada por um escalar $\alpha \neq 0$*:* $L_i \longleftarrow L_i + \alpha L_j$*.*

Logo, em termos gerais fazemos substituições no sistema dado por operações elementares para facilitar a resolução do sistema $Ax = b$.

$$\underbrace{\left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_2 & b_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right]}_{Ax=b} \xrightarrow{\text{transformação elementar}} \underbrace{\left[\begin{array}{cccc|c} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1n} & \tilde{b}_1 \\ 0 & \tilde{a}_{22} & \dots & \tilde{a}_2 & \tilde{b}_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & \tilde{a}_{nn} & \tilde{b}_n \end{array}\right]}_{Ux=c}$$

Em cada passo k da eliminação de Gauss, os multiplicadores da eliminação são calculados usando a expressão

$$m_{ij} = \frac{a_{ij}}{a_{jj}},$$

onde durante a fatoração LU guardamos os multiplicadores m_{ij} abaixo do pivô de cada passo, nos elementos que serão zerados. Diferentemente da eliminação Gaussiana, em que não salvamos os multiplicadores. Logo, obtemos

$$\tilde{A} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \\ m_{12} & \tilde{a}_{22} & \dots & \tilde{a}_2 \\ \vdots & \vdots & \dots & \vdots \\ m_{n1} & m_{12} & \dots & \tilde{a}_{nn} \end{bmatrix}$$

essa matriz $\tilde{A}$ chamaremos de matriz A no formato LU.

Após obtido a matriz $\tilde{A}$, extraímos dela as matrizes L e U, de maneira que

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ m_{12} & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ m_{n1} & m_{12} & \dots & 1 \end{bmatrix} \text{ e } U = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \\ 0 & \tilde{a}_{22} & \dots & \tilde{a}_2 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \tilde{a}_{nn} \end{bmatrix}.$$

Portanto, conseguimos a fatoração LU da matriz A, tal que:

(i) U é a matriz triangular superior obtida no final da eliminação Gaussiana da matriz A;

(ii) L é a matriz triangular inferior, com a diagonal principal unitária e abaixo da diagonal se encontrar os multiplicadores de cada passo k.

2.2. Pivoteamento Parcial

Na estratégia de pivoteamento parcial, antes do k-ésimo passo da eliminação Gaussiana, escolhemos como pivô, o maior em elemento em módulo, entre os elementos coluna k da matriz A. Tal que

$$\left|a_{jj}^{(k-1)}\right| \geq \left|a_{ij}^{(k-1)}\right|, \forall i = j, \dots, n,$$

em que $a_{jj}^{(k-1)}$ é o pivô do passo k.

2.3. Pivoteamento Completo

Usando eliminação Gaussiana com pivoteamento completo, no início de cada passo k escolhemos como pivô o maior elemento em valor absoluto dos que ainda atuam no processo de eliminação, ou seja,

$$\text{pivô } = \text{pv } = \max_{\forall i,j \geq k} \left| a_{ij}^{(k-1)} \right|,$$

em que $a_{jj}^{(k-1)}$ é o pivô do passo k.

Tanto no pivoteamento parcial quanto no completo, após a escolha do pivô, passamos ele para a posição $a_{kk}, k \in \{1, \ldots, n\}$, fazendo troca de linhas no parcial, e linhas e/ou colunas no completo. Para a implementação estamos considerando r a linha do pivô e t a coluna do pivô, que usaremos para trocar as linhas e colunas necessárias. Para tal cálculo salvamos o número das linhas trocadas no vetor p e de colunas no vetor q, que são os vetores de permutação de linhas e colunas, respectivamente.

Durante a fatoração LU podemos realizar tanto o pivoteamento completo quanto o pivoteamento parcial. Normalmente usamos o pivoteamento parcial na fatoração LU. Porém, para fazer a estratégia LU esparso que é o objetivo deste projeto, usaremos o pivoteamento completo. Pois, desta forma diminuímos o preenchimento da matriz A no formato LU.

2.4. Algoritmo - Fatoração LU com pivoteamento parcial e completo

Agora apresentamos o algoritmo LU usando pivoteamento parcial e completo. Neste algoritmo fazemos:

- nas linhas 1 a 4 criamos os vetores p e q com a numeração das linhas para salvar quais as linhas e colunas que serão trocadas;

- nas linhas 9 a 18 encontramos qual o maior valor $max\left\{|a_{ij}|\right\}$, que será escolhido como pivô pv no passo $k \in \{1, \ldots, n-1\}$, na coluna se for pivoteamento parcial ou na matriz inteira se for pivoteameto completo, e salvamos o valor da linha i e coluna j em r e t, e o pivô em pv;

- com os valores das linhas e colunas salvas, trocamos as linhas e colunas na matriz A, conforme as linhas 22 a 41 do algoritmo 1;

- por fim, nas linhas 42 a 48, fazemos o pivoteamento com o pivô pv do passo k, salvando na própria matriz A os valores m_{ij} usados em linha de forma a diminuir o gasto de memória computacional. Ou seja, dado que fazemos em cada linha pivoteada $L_i \leftarrow Li - m_{ij}Lj$, salva-se m_{ij} nas linhas embaixo do pivô.

Para fazer o pivoteamento parcial não usamos as linhas 3, 14 a 17 e 32 a 49 do algoritmo 1. Porque são as linhas referentes a escolha do maior valor de a_{ij} por coluna, após feito por linhas e a mudança das respectivas colunas da posição de $max\left\{|a_{ij}|\right\}$, cujas posições são salvas no vetor q criado na linha 3.

Algoritmo 1: FATORAÇÃO LU COM PIVOTEAMENTO PARCIAL E COMPLETO

```
Entrada: A
Saída: A no formato LU
para i ∈ 1, . . . , n faça
    p[i] = i
    q[i] = i
fim
para k ∈ 1, . . . , (n − 1) faça
    pv = |A[k, k]|
    r = k
    t = k
    para i ∈ (k + 1), . . . , n faça
        se |A[i, k]| > pv então
            pv = |A[i, k]|
            r = i
        fim
        se |A[i, j]| > pv então
            pv = |A[i, j]|
            t = j
        fim
    fim
    se pv = 0 então
        parar; a matriz A é singular
    fim
    se r ≠ k então
        aux = p[k]
        p[k] = p[r]
        p[r] = aux
        para j ∈ 1, . . . , n faça
            aux = A[k, j]
            A[k, j] = A[r, j]
            A[r, j] = aux
        fim
    fim
    se t ≠ k então
        aux = q[k]
        q[k] = q[t]
        q[r] = aux
        para j ∈ 1, . . . , n faça
            aux = A[j, k]
            A[j, k] = A[j, t]
            A[j, t] = aux
        fim
    fim
    para i ∈ (k + 1), . . . , n faça
        m = A[i, k]/A[k, k]
        A[i, k] = m
        para j ∈ (k + 1), . . . , n faça
            A[i, j] = A[i, j] − m ∗ A[k, j]
        fim
    fim
    retorna A no formato LU
fim
```

Também como apresentado, podemos utilizar a fatoração LU para resolver um sistema da forma $Ax = b$ com A quadrada e não singular, conforme o algoritmo 2.

2.5. Algoritmo - Resolução de sistema $Ax = b$ usando Fatoração LU

Para a resolução de um sistema $Ax = b$ usando LU, primeiramente resolvemos $c = Pb$, depois $Ly = c$ e por último $Ux = y$. Porque,

$$
\begin{aligned}
Ax &= b,\\
PAx &= Pb,\\
LUx &= c,\ c = Pb,\\
Ly &= c,\ y = Ux,\\
Ly &= c.
\end{aligned} \tag{2}
$$

Logo, $x = U^{-1}L^{-1}Pb$. Desta forma, usando a fatoração LU diminuímos o uso de memória computacional, não precisando fazer inversa de matrizes, que tem um alto custo computacional.

Observe que quando estamos utilizando o pivoteamento completo na fatoração LU, $PA \neq LU$, pois estamos neste caso trocando colunas também, logo $PAQ = LU$ no pivoteamento completo.

A seguir temos o algoritmo da resolução de sistema usando o procedimento mostrado acima usando LU, com matriz A quadrada.

Algoritmo 2: RESOLUÇÃO DE SISTEMA $Ax = b$ USANDO FATORAÇÃO LU

Entrada: A no formato LU, b
Saída: x
início
    para $i \in 1, \ldots, n$ **faça**
        $r = p[i]$
        $c[i] = b[r]$
    fim
    para $i \in 1, \ldots, n$ **faça**
        $soma = 0$
        para $j \in 1, \ldots, (i-1)$ **faça**
            $soma = soma + A[i,j] * y[j]$
        fim
        $y[i] = c[i] - soma$
    fim
    para $i \in n, (n-1), \ldots, 1$ **faça**
        $soma = 0$
        para $j \in (i+1), \ldots, n$ **faça**
            $soma = soma + A[i,j] * x[j]$
        fim
        $x[i] = (y[i] - soma)/A[i,i]$
    fim
fim
retorna x

2.6. LU Esparso

Definição 2 (Matriz Esparsa) *Seja A uma matriz de ordem n. Então, a esparsidade da matriz A é dada por*

$$esp(A) = \frac{nz(A)}{n \times m},$$

sendo $nz(A)$ a quantidade de elementos nulos na matriz A, n o número de linhas e m colunas da matriz A.

Na estratégia para matrizes esparsas procuramos diminuir o preenchimento das matrizes L e U, durante a realização da fatoração LU da matriz A, conforme [1]. Para isso precisamos interpolar as seguintes estratégias:

- Maximixar a <u>estabilidade</u>: escolhendo P e Q tal que $|a_{kk}| = max\{|a_{ij}|\}$
- Maximizar a <u>esparsidade</u>: escolhendo P e Q tal que a quantidade de entradas a_{ij} não zeradas é minimizada .

Consequentemente, minimizamos o números de elementos não nulos na matriz A, $nnz(A)$, para tal usamos a condição $nzlin \times nzcol$, em que $nzlin$ é o número de elementos nulos na linha i e $nzcol$ é o número de elementos nulos na coluna j. Considerando a seguinte restrição

$$|a_{ij}| \geq \tau |a_{jj}|, \text{ para } i < j. \tag{3}$$

em que $\tau \in [0, 1]$. Essa restrição é conhecida como o pivoteamento de Markowitz, e o parâmetro τ de parâmetro Threshold ou parâmetro limite. Para esta estratégia consideramos pivoteamento completo apresentado anteriormente.

2.6.1. Algoritmo LU Esparso

Nesta subseção apresentamos o algoritmo para LU esparso usando a estratégia de minimizar os elementos não nulos da matriz A, sujeito à restrição dada em 3.

Algoritmo 3: FATORAÇÃO LU ESPARSO

```
   Entrada: A
   Saída: A no formato LU
1  para i ∈ 1, ..., n faça
2  |  p[i] = i
3  |  q[i] = i
4  fim
5  para k ∈ 1, ..., n faça
6  |  cont = n^2
7  |  pv = |A[k, k]|
8  |  r = k
9  |  t = k
10 |  para i ∈ k, ..., n faça
11 |  |  para j ∈ k, ..., n faça
12 |  |  |  nzlin # elemen. não nulos linha i
13 |  |  |  nzcol # elemen. não nulos coluna j
14 |  |  |  se cont ≥ (nzlin ∗ nzcol) então
15 |  |  |  |  se |A[i, j]| ≥ τ|A[j, j]| então
16 |  |  |  |  |  pv = |A[i, j]|
17 |  |  |  |  |  cont = nzlin ∗ nzcol
18 |  |  |  |  |  r = i
19 |  |  |  |  |  t = j
20 |  |  |  |  fim
21 |  |  |  fim
22 |  |  fim
23 |  fim
24 |  Fazer linhas 19 a 48 do Algoritmo 1.
25 |  retorna A no formato LU
26 fim
```

Como default na implementação estamos usando $\tau = 10^{-4}$.

Nessa algoritmo fazemos:

- na linha 6 para cada passo k definimos $cont$, que é a contagem de elementos não nulos na linha i e coluna j;
- nas lihas 10 a 23 minimizamos a quantidade de elementos nulos, diminuindo o valor $cont$ tal que a desigualdade da linha 14 seja satisfeita;
- na linha 15 a 21 recalculamos $cont$, se satisfazer a o pivoteamento de Markowitz na linha 15, salvando o pivô, a linha e a coluna desse pivô.

No algoritmo 3 em comparação com o algoritmo 1, a diferença ocorre no processo de escolha do pivô, por isso que na linha 24 do algoritmo 3 colocamos para usar parte do algoritmo 1. Essa parte que se repete é a parte que muda as linhas e colunas na matriz A e calcula os valores de m_{ij} salvando-os.

3. Metodologia e Análise

Para a implementação utilizamos o software Julia 1.0.3, as implementações e exemplos que usaremos a seguir pode ser encontrado no GitHub.

Na análise numérica mostraremos alguns exemplos para verificar a propriedade ($PAQ = LU$), eficiência da implementação, fazer comparação entre LU com pivoteamento completo e LU esparso. Também faremos uma análise mais geral comparando o resultado de várias matrizes e uma análise para valores diferentes do parâmetros τ no LU esparso.

3.1. Exemplo de fatoração LU com pivoteamento parcial e completo

Primeiramente mostraremos, através de um exemplo, que $PA = LU$ e $PAQ = LU$, com pivoteamento parcial e completo, respectivamente.

Considere a seguinte sistema $Ax = b$, com $A = \begin{bmatrix} 2.00 & 1.00 & 2.00 \\ 1.00 & 4.00 & -3.00 \\ 5.00 & 7.00 & 1.00 \end{bmatrix}$ e $b = \begin{bmatrix} 9.00 \\ 3.00 \\ -2.00 \end{bmatrix}$.

1) Faremos a fatoração LU com pivoteamento parcial.

$$\begin{vmatrix} 2.00 & 1.00 & 2.00 \\ 1.00 & 4.00 & -3.00 \\ 5.00 & 7.00 & 1.00 \end{vmatrix} \sim \begin{vmatrix} 5.00 & 7.00 & 1.00 \\ 1.00 & 4.00 & -3.00 \\ 2.00 & 1.00 & 2.00 \end{vmatrix} \sim \begin{vmatrix} 5.00 & 7.00 & 1.00 \\ 0.20 & 2.60 & -3.20 \\ 0.40 & -1.80 & 1.60 \end{vmatrix} \sim \begin{vmatrix} 5.00 & 7.00 & 1.00 \\ 0.20 & 2.60 & -3.24 \\ 0.40 & -0.69 & -0.62 \end{vmatrix} \Rightarrow$$

$$\Rightarrow L = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.20 & 1.00 & 0.00 \\ 0.40 & -0.69 & 1.00 \end{bmatrix} \text{ e } U = \begin{bmatrix} 5.00 & 7.00 & 1.00 \\ 0.00 & 2.60 & -3.20 \\ 0.00 & 0.00 & -0.62 \end{bmatrix} \Rightarrow p = \begin{bmatrix} 3.00 \\ 2.00 \\ 1.00 \end{bmatrix} \text{ e } x = \begin{bmatrix} 35.75 \\ -23.00 \\ -19.75 \end{bmatrix}$$

.

Também temos que, $PA = LU$, ou seja,

$$PA = \begin{bmatrix} 0.00 & 0.00 & 1.00 \\ 0.00 & 1.00 & 0.00 \\ 1.00 & 0.00 & 0.00 \end{bmatrix} \begin{bmatrix} 5.00 & 7.00 & 1.00 \\ 0.20 & 2.60 & -3.20 \\ 0.40 & -0.69 & -0.62 \end{bmatrix} = \begin{bmatrix} 5.00 & 7.00 & 1.00 \\ 1.00 & 4.00 & -3.00 \\ 2.00 & 1.00 & 2.00 \end{bmatrix} =$$

$$= LU = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.20 & 1.00 & 0.00 \\ 0.40 & -0.69 & 1.00 \end{bmatrix} \begin{bmatrix} 5.00 & 7.00 & 1.00 \\ 0.00 & 2.60 & -3.20 \\ 0.00 & 0.00 & -0.62 \end{bmatrix}$$

2) Agora faremos a fatoração LU com pivoteamento completo.

$$\begin{vmatrix} 2.00 & 1.00 & 2.00 \\ 1.00 & 4.00 & -3.00 \\ 5.00 & 7.00 & 1.00 \end{vmatrix} \sim \begin{vmatrix} 5.00 & 7.00 & 1.00 \\ 1.00 & 4.00 & -3.00 \\ 2.00 & 1.00 & 2.00 \end{vmatrix} \sim \begin{vmatrix} 7.00 & 5.00 & 1.00 \\ 4.00 & 1.00 & -3.00 \\ 1.00 & 2.00 & 2.00 \end{vmatrix} \sim \begin{vmatrix} 7.00 & 5.00 & 1.00 \\ 0.57 & -1.86 & -3.57 \\ 0.14 & 1.29 & 1.86 \end{vmatrix} \sim$$

$$\sim \begin{vmatrix} 7.00 & 1.00 & 5.00 \\ 0.57 & -3.57 & -1.86 \\ 0.14 & 1.86 & 1.29 \end{vmatrix} \sim \begin{vmatrix} 7.00 & 1.00 & 5.00 \\ 0.57 & -3.57 & -1.86 \\ 0.14 & -0.52 & 0.32 \end{vmatrix} \Rightarrow L = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.57 & 1.00 & 0.00 \\ 0.14 & -0.52 & 1.00 \end{bmatrix}$$

$$\text{e } U = \begin{bmatrix} 7.00 & 1.00 & 5.00 \\ 0.00 & -3.57 & -1.86 \\ 0.00 & 0.00 & 0.32 \end{bmatrix} \Rightarrow p = \begin{bmatrix} 3.00 \\ 2.00 \\ 1.00 \end{bmatrix}, q = \begin{bmatrix} 2.00 \\ 3.00 \\ 1.00 \end{bmatrix} \text{ e } x = \begin{bmatrix} -23.00 \\ -19.75 \\ 35.75 \end{bmatrix}$$

.

Vejamos que, $PAQ = LU$, ou seja,

$$PAQ = \begin{bmatrix} 0.00 & 0.00 & 1.00 \\ 0.00 & 1.00 & 0.00 \\ 1.00 & 0.00 & 0.00 \end{bmatrix} \begin{bmatrix} 5.00 & 7.00 & 1.00 \\ 0.20 & 2.60 & -3.20 \\ 0.40 & -0.69 & -0.62 \end{bmatrix} \begin{bmatrix} 0.00 & 0.00 & 1.00 \\ 1.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 \end{bmatrix} =$$

$$= \begin{bmatrix} 7.00 & 1.00 & 5.00 \\ 4.00 & -3.00 & 1.00 \\ 1.00 & 2.0 & 2.00 \end{bmatrix} = LU = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.57 & 1.00 & 0.00 \\ 0.14 & -0.52 & 1.00 \end{bmatrix} \begin{bmatrix} 7.00 & 1.00 & 5.00 \\ 0.00 & -3.57 & -1.86 \\ 0.00 & 0.00 & 0.32 \end{bmatrix}$$

.

Essa propriedade vale também para a estratégia de LU esparso.

Os valores do vetor x, nos dois tipos de pivoteamento, estão em ordem diferente, pois quando resolvemos $Ax = b$, fazemos $PAx = Pb$ e não multiplicamos por Q no lado direito da expressão.

3.2. Exemplo de comparação entre LU com pivoteamento completo e LU esparso

Exibiremos como funciona a estratégia de LU esparso, apresentado na seção anterior. Seja $A = \begin{bmatrix} 2.00 & 0.00 & 2.00 & 0.00 & 0.00 \\ 3.00 & 1.00 & 4.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -3.00 & 2.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 4.00 & 2.00 & 2.00 & 1.00 \end{bmatrix}$, cuja esparsidade ($esp(A)$) é de 52%, ou seja, 52% da matriz contem elementos nulos. Então,

1) aplicando LU com pivoteamento completo obtemos

$$L = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 1.00 & 0.00 & 0.00 & 0.00 \\ -0.75 & -1.50 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.50 & -0.33 & 0.11 & 0.11 & 1.00 \end{bmatrix} \text{ e } U = \begin{bmatrix} 4.00 & 3.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & -1.50 & 3.50 & 2.00 & 1.00 \\ 0.00 & 0.00 & 6.00 & 5.00 & 1.50 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.17 \end{bmatrix},$$

$$p = \begin{bmatrix} 2.00 & 5.00 & 3.00 & 4.00 & 1.00 \end{bmatrix}^T \text{ e } q = \begin{bmatrix} 3.00 & 1.00 & 2.00 & 4.00 & 5.00 \end{bmatrix}^T,$$

com $esp(A) = 52\%$ da matriz L e $esp(A) = 52\%$ da matriz U.

2) agora usando a estratégia de LU esparso encontramos

$$L = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 2.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.50 & 1.00 & 0.00 \\ 2.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \text{ e } U = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 4.00 & 2.00 \\ 0.00 & 0.00 & 2.00 & 0.00 & 2.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -3.00 \end{bmatrix},$$

$$p = \begin{bmatrix} 4.00 & 5.00 & 1.00 & 2.00 & 3.00 \end{bmatrix}^T \text{ e } q = \begin{bmatrix} 4.00 & 5.00 & 1.00 & 2.00 & 3.00 \end{bmatrix}^T,$$

com $esp(A) = 68\%$ da matriz L e $esp(A) = 64\%$ da matriz U.

Logo, os fatores L e U na estratégia LU esparso diminui o preenchimento de cada fator, como podemos ver nesse exemplo o fator L tem 30.77% mais elementos nulos na estratégia LU esparso, e o fator U tem 23.08% mais elementos nulos. Assim, o objetivo de diminuir o preenchimento dos elementos nulos dos fatores L e U é verificado, como esperávamos.

3.3. Análise do parâmetro Threshold τ

Realizamos agora a comparação entre vários valores do parâmetro τ no pivoteamento de Markowitz, verificando o que muda em cada valor de τ. Para tal usamos o algoritmo de LU esparso. E utilizaremos a matriz A do exemplo anterior, que é uma matriz quadrada de ordem 5 com $esp(A) = 52\%$. O resultado da esparsidade desse matriz para cada valor de τ é apresentado na tabela 1.

τ	$esp(L)$	$esp(U)$
0.00	0.68	0.64
0.20	0.68	0.64
0.40	0.68	0.64
0.50	0.68	0.64
0.60	0.68	0.64
0.80	0.68	0.64
1.00	0.68	0.64

Tabela 1: Esparsidade dos fatores L e U mudando o valor do parâmetro τ.

Consequentemente, na estratégia LU esparso, a esparsidade dos fatores L e U independem do valor do parâmetro τ.

3.4. Análise de eficiência

Na análise de eficiência iremos aplicar a estratégia implementada usando várias matrizes de tamanho $n = 500$ e $\tau = 0.1$. Testamos os métodos com as seguintes matrizes

- triangular superior: $\begin{bmatrix} 1 & 2 & \dots & n \\ 0 & 2 & \dots & n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n \end{bmatrix}$,

- triangular inferior: $\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \dots & n \end{bmatrix}$,

- diagonal: $\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & n \end{bmatrix}$,

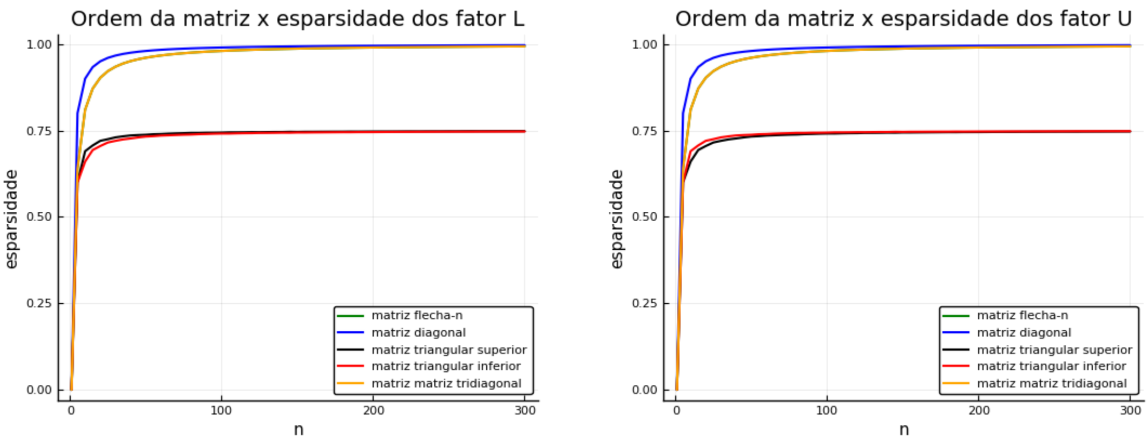
- tridiagonal: $\begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 1 & 2 & -1 & \dots & 0 \\ 0 & 1 & 3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & \dots & n \end{bmatrix}$,

- flecha-n: $\begin{bmatrix} n & 1 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix}$.

O resultado da análise mostrando a espasidade dos fatores L e U é apresentado na tabela 2. Onde podemos perceber pela tabela 2 que com LU esparso a esparsidade dos fatores L e U aumentam, em comparação ao pivoteamento completo. Também concluímos que para cada uma dessas matrizes quanto maior a ordem da matriz (n), mais esparsa ela será, veja a figura 1.

#	Tipo	pivô completo		LU esparso	
		$esp(L)$	$esp(U)$	$esp(L)$	$esp(U)$
1	triangular superior	0.4990	0.5010	0.7488	0.7482
2	triangular inferior	0.6647	0.5396	0.7482	0.7488
3	tridiagonal	0.4990	0.5010	0.9960	0.9960
4	diagonal	0.4990	0.5010	0.9980	0.9980
5	flecha-n	0.4990	0.4990	0.9966	0.9960

Tabela 2: Aplicação da fatoração LU com as estratégias de pivoteamento completo e LU esparso em matrizes especiais.



(a) eparsidade do fator L

(b) eparsidade do fator U

Figura 1: Ordem da matriz versus esparsidade dos fatores L

Observe que nos gráficos da 1 tanto a matriz diagonal quanto a tridiagonal tiveram os mesmos resultados de esparsidade.

4. Considerações finais

Concluímos, que para a estratégia LU esparso, a implementação feita em Julia se apresentou eficiente sendo melhor que o LU com pivoteamento completo, como era de esperar teoricamente. Porque, na estratégia de LU esparso, minimizamos o número de linhas e colunas com elementos preenchido os fatores L e U na fatoração LU usando a restrição 3.

Também, verificamos que não há uma relação entre o valor do parâmetro τ e a esparsidade da matriz. Além disso, averiguamos que quanto maior a ordem da matriz quadrada maior sua esparsidade, conforme figura 1.

Para trabalhos futuros podemos estender a estratégia LU esparso para matrizes não quadradas de ordem $n \times m$, e analisar se as mesmas conclusões obtidas para matrizes quadradas se mantêm.

Referências

[1] Golub, G. H. e Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU press.

[2] Ruggiero, M. A. G. e Lopes, V. L. d. R. (1997). *Cálculo numérico: aspectos teóricos e computacionais*. Makron Books do Brasil.

[3] Souza, M. J. F. (2018). Sistemas lineares. URL `http://www.decom.ufop.br/marcone/Disciplinas/CalculoNumerico/Sistemas.pdf`.