EP1 - Decomposição LU para Matrizes Tridiagonais

MAP3121 - Métodos Numéricos e Aplicações

Prof° Renato Vicente

Igor Costa D’Oliveira - 11391446

Paulo Gomes Ivonica - 11804532

1 de Maio de 2022

---

1. Introdução

Este relatório descreve as atividades ao estudar problemas de decomposição LU de matrizes. Neste exercício computacional, analisou-se e implementou-se um algoritmo que decompõe matrizes tridiagonais cíclicas e posteriormente às soluciona.

2. Matrizes Tridiagonais Acíclicas

2.1 Definição

Matrizes tridiagonais são aquelas que possuem elementos não nulos na diagonal principal e nas diagonais secundárias acima e abaixo desta. Portanto todos os outros elementos são iguais a zero. Segue um exemplo abaixo:

Figura 1 - Matriz tridiagonal acíclica de ordem N.

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & . & . & . & & 0 \\ a_{21} & a_{22} & a_{23} & 0 & & & & . \\ 0 & a_{32} & a_{33} & a_{34} & 0 & & & . \\ . & 0 & a_{43} & a_{44} & . & . & & . \\ . & & 0 & . & . & . & . & \\ . & & & . & . & . & . & . \\ & & & & . & . & . & . \\ 0 & . & . & . & & . & . & a_{nn} \end{bmatrix}$$

2.2 Algoritmo

A decomposição LU e a posterior resolução do sistema linear correspondente à uma matriz tridiagonal são mais otimizadas do que uma matriz não tridiagonal.

Fato este que ocorre porque as operações e o armazenamento envolvendo os valores sabidamente nulos são facilmente descartáveis. Portanto, para implementar o algoritmo da função LU basta armazenar três vetores, um para cada diagonal não nula. Tais vetores são:

$$A = (a_1, a_2, \ldots, a_n)$$
$$B = (b_1, b_2, \ldots, b_n)$$
$$C = (c_1, c_2, \ldots, c_n)$$

Esses três vetores são os elementos não nulos da figura 1. E para obtermos L e U de forma eficiente serão utilizadas as deduções matemáticas apresentadas na literatura da tarefa deste exercício, no qual a figura 2 ilustra a função "LU_TriDiagonal" responsável pela resolução da matriz apresentada pelos vetores a, b, c e d. Na figura também mostra o código comentado sobre as equações e funções usadas.

Figura 2 - Função para a decomposição LU.

```python
# Essa função recebe uma matriz tridiagonal definida por quatro
# vetores [a, b, c, d] e retorna as soluções x do problema.
# Para isso utiliza a decomposição de dois parâmetros L e U.

def LU_TriDiagonal(a, b, c, d, ciclica):
    n = len(a)
    u = [0] * n
    l = [0] * n
    x = [0] * n
    y = [0] * n

    u[0] = b[0]
    for i in list(range(1, n, 1)):
        # Multiplicadores
        l[i] = a[i]/u[i - 1]
        u[i] = b[i] - l[i]*c[i - 1]

    # Solução de Ly = d
    y[0] = d[0]
    for i in list(range(1, n, 1)):
        y[i] = d[i] - l[i]*y[i - 1]

    # Solução de Ux = y
    x[n - 1] = y[n - 1]/u[n - 1]
    i = n - 2
    while i >= 0:
        x[i] = (y[i] - c[i] * x[i + 1]) / u[i]
        i -= 1

    if ciclica == 0:
        apresentar_Parametros(u, l, y, 0, 0, 0, 0)

    return x
```

2.3 Testes

Após a implementação do algoritmo em python para a resolução de sistemas lineares tridiagonais usando a decomposição LU, foi feita uma função geradora de matrizes tridiagonais para testar o algoritmo. Para isso foram utilizados como coeficientes da matriz A os seguintes valores:

$$
\begin{aligned}
&a_i = \frac{2i-1}{4i},\ 1 \le i \le n-1,\ a_n = \frac{2n-1}{2n},\\
&c_i = 1 - a_i,\ 1 \le i \le n,\\
&b_i = 2,\ 1 \le i \le n,
\end{aligned}
$$

E o lado direito do sistema linear é dado pela seguinte equação.

$$d_i = \cos\left(\frac{2\pi i^2}{n^2}\right),\ 1 \le i \le n.$$

Logo, foi feita a função “Gerador_TriDiagonal” que apresenta justamente o algoritmo para realizar os testes das demais funções apresentadas até agora. Além disso, foram desenvolvidas algumas funções para que os resultados aparecerem para o usuário de uma forma fácil e intuitiva e que permitissem a interação dele com os testes. A figura 3 ilustra a interface desenvolvida para a interação entre o usuário e o programa, logo é possível ver que o usuário pode tanto realizar os testes usando o gerador das matrizes tridiagonais apresentado anteriormente, quanto digitar o seu próprio problema linear usando matrizes tridiagonais.

Figura 3 - Interface do programa.

```
-------- Métodos Numéricos EP1 --------

 1 - Gerador de Matrizes Tridiagonais.
 2 - Digitar uma Matriz Tridiagonal.
 3 - Sair.

 Digite uma opção:

-------- Gerador de Matrizes Tridiagonais --------

 1 - Gerador de Matrizes não Cíclicas.
 2 - Gerador de Matrizes Cíclicas.
 3 - Voltar.

 Digite uma opção:
```

Após selecionar a opção 1 duas vezes (gerador de matrizes não cíclicas) é possível testar o algoritmo usando um tamanho n para a matriz tridiagonal gerada (An,n). Na imagem 4 está ilustrado o teste de uma matriz com n igual a

5. Nele é possível perceber todos os elementos da matriz A gerada e os valores de D. Também é apresentado os parâmetros calculados para a resolução do problema e as suas soluções.

Figura 4 - Teste de uma matriz tridiagonal acíclica com n igual a 5.

```
-------- Gerador de Matrizes não Cíclicas --------

Digite um número para n [n > 0]: 5

Problema:

| 2.00 0.75 0.00 0.00 0.00 | * | X1 |  =  | 0.97 |
| 0.38 2.00 0.62 0.00 0.00 | * | X2 |  =  | 0.54 |
| 0.00 0.42 2.00 0.58 0.00 | * | X3 |  =  | -0.64 |
| 0.00 0.00 0.44 2.00 0.56 | * | X4 |  =  | -0.64 |
| 0.00 0.00 0.00 0.90 2.00 | * | X5 |  =  | 1.00 |


Parâmetros Calculados:


U = | 2.00|1.86|1.86|1.86|1.73|
L = | 0.00|0.19|0.22|0.24|0.48|
Y = | 0.97|0.35|-0.72|-0.47|1.23|


Soluções do Problema:

X1 =  0.382695981561
X2 =  0.270921597341
X3 =  -0.239243828463
X4 =  -0.465977711511
X5 =  0.709689970180
```

3. Matrizes Tridiagonais Cíclicas

3.1 Definição

Matrizes tridiagonais são aquelas que possuem elementos não nulos na diagonal principal e nas diagonais secundárias acima e abaixo desta. Já as matrizes tridiagonais cíclicas possuem todos os outros elementos iguais a zero, como as tridiagonais, porém possui o termo $A1n$ e Ann não nulos . Segue um exemplo na figura 5.

Figura 5 - Matriz tridiagonal cíclica de ordem N.

$$A = \begin{bmatrix} b_1 & c_1 & & & a_1 \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & & & a_n & b_n \end{bmatrix}$$

Com isso pode-se observar que é possível obter a decomposição LU de de maneira eficiente dividindo a matriz cíclica em uma matriz superior tridiagonal não cíclica e dois vetores v e w.

3.2 Algoritmo

Novamente será utilizada a dedução matemática das referências da tarefa, no qual para a solução do sistema linear será reaproveitado o algoritmo para matrizes tridiagonais feita no tópico 2 deste trabalho. Logo, na figura 6 está ilustrado o algoritmo em python usado no programa para a resolução de matrizes tridiagonais cíclicas, e é possível ver que ele utiliza a função “LU_TriDiagonal” para calcular o y_til e o z_til.

Figura 6 - Função para a resolução de matrizes tridiagonais cíclicas.

```python
# Essa função recebe uma matriz tridiagonal cíclica definida
# por quatro vetores [a, b, c, d] e retorna as soluções x do problema.
# Para isso divide a matriz cíclica em duas partes, uma matriz tridiagonal
# não cíclica e dois vetores v e w.

def TriDigCiclica(a, b, c, d):
    n = len(a)
    v = [0] * (n-1)
    w = [0] * (n-1)

    # Determinação de v
    v[0] = a[0]
    v[n - 2] = c[n - 2]
    # Determinação de w
    w[0] = c[n - 1]
    w[n - 2] = a[n - 1]

    a_til = [0] * (n - 1)
    b_til = [0] * (n - 1)
    c_til = [0] * (n - 1)
    d_til = [0] * (n - 1)
    y_til = [0] * (n - 1)
    z_til = [0] * (n - 1)
    x = [0] * n

    # Determinação de T
    for i in list(range(0, n - 1, 1)):
        a_til[i] = a[i]
    for i in list(range(0, n - 1, 1)):
        b_til[i] = b[i]
    for i in list(range(0, n - 1, 1)):
        c_til[i] = c[i]
    for i in list(range(0, n - 1, 1)):
        d_til[i] = d[i]

    y_til = LU_TriDiagonal(a_til, b_til, c_til, d_til, 1)
    z_til = LU_TriDiagonal(a_til, b_til, c_til, v, 1)

    x[n - 1] = (d[n - 1] - c[n - 1] * y_til[0] - a[n - 1] * y_til[n-2])/(b[n - 1] - c[n - 1] * z_til[0] - a[n - 1] * z_til[n - 2]

    for i in list(range(0, n - 1, 1)):
        # Determinação das soluções
        x[i] = y_til[i] - (x[n - 1] * z_til[i])

    apresentar_Parametros(0, 0, y_til, v, w, z_til, 1)

    return x
```

3.3 Testes

Agora será necessário testar a função “TriDigCiclica”, tanto quanto as soluções encontradas, quanto a velocidade para a resolução do problema. Logo, para realizar os testes foi utilizado o mesmo gerador de matrizes tridiagonais apresentado no tópico 2.3 deste trabalho, porém também é possível testar uma matriz qualquer inserindo os seus valores no programa.

Na figura 7 está o teste de uma matriz com ordem igual a cinco gerada pela função “Gerador_TriDiagonal”. Nela é possível visualizar as soluções do sistema linear encontrados pelo algoritmo, e para confirmar que estes valores são realmente os resultados esperados, foram multiplicados a matriz A pelos valores encontrados para X e foi visto que os valores do lado direito do sistema linear foram todos iguais aos esperados, logo o algoritmo está funcionando.

Figura 7 - Matriz tridiagonal cíclica de ordem n igual a 5.

```
-------- Gerador de Matrizes Cíclicas --------

Digite um número para n [n > 1]: 5

Problema:

| 2.00 0.75 0.00 0.00 0.25 | * | X1 |  =  | 0.97 |
| 0.38 2.00 0.62 0.00 0.00 | * | X2 |  =  | 0.54 |
| 0.00 0.42 2.00 0.58 0.00 | * | X3 |  =  | -0.64 |
| 0.00 0.00 0.44 2.00 0.56 | * | X4 |  =  | -0.64 |
| 0.10 0.00 0.00 0.90 2.00 | * | X5 |  =  | 1.00 |


Parâmetros Calculados:


v = | 0.25|0.00|0.00|0.56|
w = | 0.10|0.00|0.00|0.90|
y = | 0.37|0.29|-0.31|-0.25|
z = | 0.12|0.00|-0.09|0.30|


Soluções do Problema:

X1 =  0.288852503747
X2 =  0.290338254597
X3 =  -0.245071044993
X4 =  -0.459867724306
X5 =  0.692497850750
```

Já para testar a eficiência computacional do algoritmo, contando a velocidade para a resolução dos problemas, foi inserido um valor de n igual a 20 no gerador de matrizes. A figura 8 apresenta o teste e é possível visualizar as soluções que começam em X1 e terminam no X20. O algoritmo apresentou o resultado em menos de um segundo, portanto é possível concluir que ele apresenta um algoritmo rápido e eficiente.

Figura 8 - Matriz tridiagonal cíclica de ordem n igual a 20.

```
-------- Gerador de Matrizes Cíclicas --------

Digite um número para n [n > 1]: 20

Problema:

| 2.00 0.75 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.25 | * | X1 |  =  | 1.00 |
| 0.38 2.00 0.62 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X2 |  =  | 1.00 |
| 0.00 0.42 2.00 0.58 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X3 |  =  | 0.99 |
| 0.00 0.00 0.44 2.00 0.56 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X4 |  =  | 0.97 |
| 0.00 0.00 0.00 0.45 2.00 0.55 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X5 |  =  | 0.92 |
| 0.00 0.00 0.00 0.00 0.46 2.00 0.54 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X6 |  =  | 0.84 |
| 0.00 0.00 0.00 0.00 0.00 0.46 2.00 0.54 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X7 |  =  | 0.72 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.47 2.00 0.53 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X8 |  =  | 0.54 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.47 2.00 0.53 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X9 |  =  | 0.29 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.47 2.00 0.53 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X10 |  =  | 0.00 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.48 2.00 0.52 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X11 |  =  | -0.32 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.48 2.00 0.52 0.00 0.00 0.00 0.00 0.00 0.00 0.00 | * | X12 |  =  | -0.64 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.48 2.00 0.52 0.00 0.00 0.00 0.00 0.00 0.00 | * | X13 |  =  | -0.88 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.48 2.00 0.52 0.00 0.00 0.00 0.00 0.00 | * | X14 |  =  | -1.00 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.48 2.00 0.52 0.00 0.00 0.00 0.00 | * | X15 |  =  | -0.92 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.48 2.00 0.52 0.00 0.00 0.00 | * | X16 |  =  | -0.64 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.49 2.00 0.51 0.00 0.00 | * | X17 |  =  | -0.17 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.49 2.00 0.51 0.00 | * | X18 |  =  | 0.37 |
| 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.49 2.00 0.51 | * | X19 |  =  | 0.82 |
| 0.03 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.97 2.00 | * | X20 |  =  | 1.00 |


Parâmetros Calculados:


v = | 0.25|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.51|
w = | 0.03|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.00|0.97|
y = | 0.38|0.32|0.33|0.32|0.31|0.28|0.24|0.18|0.10|0.00|-0.11|-0.21|-0.30|-0.34|-0.32|-0.23|-0.06|0.10|0.39|
z = | 0.14|-0.03|0.01|-0.00|0.00|-0.00|0.00|-0.00|0.00|-0.00|0.00|-0.00|0.00|-0.00|0.00|-0.01|0.02|-0.08|0.28|


Soluções do Problema:

X1 =  0.330315120446
X2 =  0.333697839550
X3 =  0.330820606659
X4 =  0.324585733578
X5 =  0.310538095217
X6 =  0.284981385394
X7 =  0.243757281980
X8 =  0.183491369092
X9 =  0.102744152222
X10 =  0.003606288097
X11 =  -0.106697235236
X12 =  -0.214727902174
X13 =  -0.301137459552
X14 =  -0.343308126549
X15 =  -0.320975007262
X16 =  -0.224510819657
X17 =  -0.063864400083
X18 =  0.125806761171
X19 =  0.287136437501
X20 =  0.355892047713
```

4. Conclusão

De acordo com os resultados dos testes, foi possível concluir que o algoritmo desenvolvido em python além de apresentar todas as soluções corretamente ele o fez de uma forma rápida e otimizada, visto que o algoritmo está resolvendo rapidamente matrizes com um n muito elevado (n > 50). Portanto, a decomposição em LU para resolver sistemas lineares com matrizes tridiagonais é uma ótima ferramenta computacionalmente para este propósito.

É importante salientar que os algoritmos foram desenvolvidos de forma que possam ser usados como partes de outros programas posteriormente de forma simples e fácil, sem precisar alterar parâmetros de entrada ou saída do código.

Por fim, foram revistos conceitos, estudados durante o oferecimento de matérias como Álgebra Linear e Cálculo. É importante ter um bom entendimento de algoritmos de cálculo, e a sua implementação, porque é necessário buscar técnicas para programar buscando o melhor desempenho possível para o algoritmo.

5. Referências

[1] **Enunciado Exercício-Proposto 1.**
<https://edisciplinas.usp.br/pluginfile.php/6990213/mod_resource/content/3/tarefa1_2022.pdf>.
Acessado em 23 de Abril de 2022.

[2] **Figura 1**
Disponível em:
https://www.cenapad.unicamp.br/parque/manuais/Essl/html/esygr30.gif
Acessado em 27 de Abril de 2022.