Decomposição LU para Matrizes Tridiagonais

Tarefa 1 - MAP3121 - **Data de entrega: 01/05/2022**

Regras do Jogo

- Você deve implementar o exercício programa em C/C++ (alunos da elétrica) ou Python3.7 (demais alunos).
- Python:
  - Pode usar: Matplotlib, NumPy (apenas para trabalhar com aritmética de vetores, matrizes, leitura/escrita de dados), bibliotecas básicas auxiliares: sys, time, datetime, os, math.
  - **Não** pode usar: SciPy ou outras bibliotecas de algebra linear computacional
- C, C++:
  - **Não** pode usar recursos de versões além de C/C++14.
  - Pode usar qualquer biblioteca nativa do gcc/g++ (que não exiga instalção adicional).
- Incluir, obrigatoriamente, um arquivo LEIAME.txt com instruções de compilação e execução, indicando versão de interpretador/compilador necessário.
- O exercício pode ser feito em duplas. A dupla **permanecerá a mesma** nas tarefas subsequentes.
- Apenas um aluno deve entregar o exercício, destacando no relatório e no código o nome de ambos os alunos.
- A entrega deve conter o relatório (em .pdf), contendo a análise do problema estudado, e o código usado para as simulações computacionais (arquivos fonte). A entrega deve ser feita em um arquivo compactado único.
- O relatório deve apresentar resultados e análises de todas as tarefas descritas neste enunciado.
- O seu código deve estar bem documentado, de forma a facilitar a correção. Rodar os testes também deve ser fácil para o usuário do seu programa, sem que este tenha que editar seu código. Ou seja, você deve pedir como entrada qual teste o usuário quer rodar, qual método e os parâmetros para o teste.

Decomposição LU

Matrizes pertencentes a certas classes oriundas de aplicações podem ser triangularizadas pelo Método de Eliminação de Gauss sem trocas de linhas e sem a necessidade de condensação pivotal para a estabilidade numérica. Por exemplo, matrizes diagonais dominantes, matrizes simétricas definidas positivas e algumas matrizes em problemas de aproximação por splines estão nesta situação.

Suponha que A é uma matriz triangularizável pelo Método de Eliminação de Gauss sem trocas de linhas. Se denotarmos por U a matriz triangular superior obtida da triangularização, e por L a matriz triangular inferior formada pelos multiplicadores L_{ij}, $i > j$, abaixo da diagonal e com elementos diagonais iguais a 1, então pode-se mostrar que

$$A = LU,$$

chamada de decomposição LU de A.

Sabendo-se da existência da decomposição LU, podemos obter os coeficientes de L e de U usando-se somente as propriedades dessas matrizes, sem termos de necessariamente fazer as contas na ordem da eliminação de Gauss. Note que, se $A = LU$, com L triangular inferior e U triangular superior, então

$$A_{ij} = \sum_{k=1}^{\min\{i,j\}} L_{ik} U_{kj}. \qquad \text{(exercício)}$$

Usando-se também o fato de que $L_{ii} = 1$, os coeficientes podem ser calculados em uma ordem diferente do Método de Eliminação de Gauss da seguinte maneira (exercício):

para i =1, $\cdots$, n **faça**

$$U_{ij} = A_{ij} - \sum_{k=1}^{i-1} L_{ik} U_{kj}, \quad j = i, \cdots, n \tag{1}$$

$$L_{ji} = \left(A_{ji} - \sum_{k=1}^{i-1} L_{jk} U_{ki} \right) / U_{ii}, \quad j = i+1, \cdots, n \tag{2}$$

fim

Observações:

- Quando $i = 1$, as somatórias em (1) e (2) não são calculadas.
- Quando $i = n$, a expressão (2) não é calculada.
- Para a implementação em Python, *somente o laço em i* é necessário. As expressões (1) e (2) podem ser **vetorizadas** usando multiplicação de matrizes do NumPy, evitando-se o uso ineficiente de mais laços. Note que essas expressões têm a forma

$$U[i, i:n] = A[i, i:n] - L[i, 1:(i-1)] \cdot U[1:(i-1), i:n],$$
$$L[(i+1):n, i] = \frac{1}{U_{ii}} \cdot \Big(A[(i+1):n, i] - L[(i+1):n, 1:(i-1)] \cdot U[1:(i-1), i] \Big).$$

Podemos então resolver qualquer sistema linear $Ax = b$ resolvendo-se um sistema triangular inferior e outro triangular superior da seguinte forma (por que?):

$$\begin{cases} Ly = & b, \\ Ux = & y. \end{cases}$$

Novamente, somente um laço é necessário em cada resolução de sistema triangular, se você vetorizar usando produtos internos. Esta metodologia pode ser usada para a resolução de vários sistemas lineares com a mesma matriz A e lados direitos diferentes. Para matrizes tridiagonais, as vetorizaçãoes mencionadas acim não são necessárias, como veremos a seguir.

Matrizes tridiagonais

Matrizes tridiagonais possuem elementos diferentes de zero somente na diagonal principal e nas diagonais secundárias acima e abaixo da diagonal principal, ou seja, $a_{ij} = 0$ se $|i - j| > 1$. Elas aparecem frequentemente em aplicações e por isso merecem um tratamento especial. A sua estrutura pode ser explorada para obtermos L e U de forma eficiente.

Se A é uma matriz tridiagonal triangulrizável pelo Método de Eliminação de Gauss sem trocas de linhas, os únicos elementos de U que podem ser não nulos são U_{ii} e $U_{i,i+1}$, e os únicos multiplicadores que podem ser não nulos são $L_{i+1,i}$. Além disso, $U_{i,i+1} = a_{i,i+1}$ (tente demonstrar estas afirmações). Consequentemente, o número de operações aritméticas é reduzido consideravelmente ao descartarmos contas cujos resultados sabemos que são nulos.

O armazenamento também pode ser feito de forma eficiente, sendo desnecessário guardar os valores que sabemos que são nulos. A matriz tridiagonal

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

pode ser armazenada em três vetores

$$a = (0, a_2, \ldots, a_{n-1}, a_n), \quad b = (b_1, b_2, \ldots, b_{n-1}, b_n) \quad c = (c_1, c_2, \ldots, c_{n-1}, 0)$$

e os coeficientes $u_i = U_{ii}$ e $l_{i+1} = L_{i+1,i}$ da decomposição LU podem ser calculados pelo algoritmo (exercício)

$u_1 = b_1$

para $i = 2, \cdots, n$ **faça**

$l_i = a_i / u_{i-1}$ (multiplicador)

$u_i = b_i - l_i c_{i-1}$

fim

sendo possível armazená-los também em vetores. Lembre-se que $U_{i,i+1} = c_i$.

Tendo L e U (armazenados como descrito acima), a solução de um sistema $Ax = d$ é então obtida de

$Ly = d$:

$y_1 = d_1$

para $i = 2, \cdots, n$ **faça**

$y_i = d_i - l_i y_{i-1}$

fim

$Ux = y$:

$x_n = y_n / u_n$

para $i = n-1, \cdots, 1$ **faça**

$x_i = (y_i - c_i\, x_{i+1}) / u_i$

fim

Sistemas tridiagonais cíclicos

No tratamento numérico de certos problemas envolvendo periodicidade, aparecem sistemas tridiagonais cíclicos $Ax = d$ onde a matriz tem a seguinte estrutura:

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

É possível obter a decomposição LU de A de maneira eficiente, mas aqui estamos interessados na resolução do sistema linear aproveitando o algoritmo para matrizes tridiagonais. O sistema $Ax = d$ pode ser escrito como

$$T\tilde{x} + x_n v = \tilde{d}$$
$$w^t \tilde{x} + x_n b_n = d_n$$

onde T é a submatriz principal $(n-1) \times (n-1)$, *que é tridiagonal*, v e w são os vetores de $\mathbb{R}^{n-1}$ definidos por $v = (a_1, 0, \ldots, 0, c_{n-1})^t$ e $w = (c_n, 0, \ldots, 0, a_n)^t$, respectivamente, $\tilde{x} = (x_1, \ldots, x_{n-1})^t$ e $\tilde{d} = (d_1, \ldots, d_{n-1})^t$. A solução é dada por

$$x_n = \frac{d_n - c_n \tilde{y}_1 - a_n \tilde{y}_{n-1}}{b_n - c_n \tilde{z}_1 - a_n \tilde{z}_{n-1}} \quad e \quad \tilde{x} = \tilde{y} - x_n \tilde{z}$$

onde $\tilde{y}$ é a solução do sistema linear $T\tilde{y} = \tilde{d}$ e $\tilde{z}$ é a solução do sistema linear $T\tilde{z} = v$, ambos com a mesma matriz tridiagonal T de ordem $n-1$.

Tarefa

Implemente o algoritmo descrito acima para a decomposição LU de uma matriz tridiagonal A $n \times n$. As matrizes A, L e U devem ser armazenadas em vetores conforme descrito no texto. Implemente também o algoritmo para a resolução de um sistema linear tridiagonal usando a decomposição LU da matriz. *Faça as implementações de forma que elas possam ser usadas como partes de outros programas.*

Teste os algoritmos na resolução do sistema linear tridiagonal cíclico $Ax = d$, onde os coeficientes da matriz A são

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

e o lado direito do sistema linear é dado por

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

Use $n = 20$. Pense em uma forma adequada para o programa apresentar os dados e as respostas.