Cálculo Numérico - MS211F - 1S/2024

Sarah Pereira Teixeira Silva
RA: 258968

Pedro Henrique Martins Belo
RA: 267809

5 de junho de 2024

Seja A uma matriz real, quadrada de ordem n e b um vetor do $\mathbb{R}^n$. Escreva em metalinguagem o algoritmo para decomposição LU sem pivoteamento. Conte o número de operações desse algoritmo.

1. Valores de entrada:

   Matriz U como uma matriz A de ordem n.

   Matriz L como uma matriz identidade de ordem n.

   Vetor b pertencente a $\mathbb{R}^n$.

2. Para cada coluna j de U, exceto a última:

   (a) Para cada linha i abaixo da diagonal principal em U:

      i. Calcule o fator m_{ij}, que é o elemento ij de U dividido pelo elemento jj de U.

      ii. Atualize o elemento correspondente de L para m_{ij}.

      iii. Para cada elemento k da linha i em U, abaixo do elemento ij:

         A. Subtraia m_{ij} multiplicado pelo elemento correspondente da linha j em U de k.

3. Repetir o processo do passo 2 para cada coluna, exceto a última, mas sem a atribuição de L para m_{ij}.

4. Construção da matriz U:

   (a) Para cada elemento i, j da matriz U:

      i. Se j for menor ou igual a i, copie o elemento correspondente de A para U.

      ii. Caso contrário, atribua 0 ao elemento.

5. Construir a matriz L a partir dos elementos de A resultantes da etapa anterior:

   (a) Para cada elemento i, j da matriz L:

      i. Se j for maior que i, copie o elemento correspondente de A para L.

      ii. Caso contrário, atribua zero ao elemento e um à diagonal principal.

Saída: Retornar L, U.

Contagem de operações:

O algoritmo de decomposição LU sem pivoteamento exige o mesmo número de flops de multiplicação e divisão que a eliminação de Gauss. A diferença se mostra no menor esforço gasto na fase de decomposição. Assim, o número de flops de multiplicação e divisão envolvidos na fase de decomposição pode ser calculado por aproximadamente $\frac{2}{3}n^3$, em que n é a ordem da matriz A.

Para cada valor j no laço 2.iii são realizadas duas operações: uma multiplicação e uma adição. Assim, são necessárias

$$\sum_{j=k+1}^{n} 2 = 2(n-k+1).$$

Já no segundo laço do algoritmo (em 2.i), além das operações acima, para cada i será realizada uma divisão; assim, o número de operações correspondente a i será

$$\sum_{j=k+1}^{n}[1+2(n-k+1)] = [1+2(n-k+1)](n-k).$$

Para obter o número total de operações fazemos a soma em k

$$\sum_{k=1}^{n-1}(n-k)+2(n-k+1)(n-k) = \sum_{k=1}^{n-1}(n-k)+2\sum_{k=1}^{n-1}(n-k+1)(n-k) = \frac{n(n-1)}{2}+2\frac{n^3-n}{3} = \frac{2}{3}n^3+\frac{n^2}{2}-\frac{7}{6}n.$$

Os cálculos acima foram obtidos atráves não apenas de uma progressão aritmética, mas também do seguinte resultado:

$$\sum_{k=1}^{n-1} k^2 = \frac{(n-1)n(2n-1)}{6}.$$

Para contabilizar o número total de operações no Método de Eliminação Gaussiana, somamos o número de operações necessárias para resolver um sistema triangular, como deduzido. Dessa, forma, na resolução de um sistema com n equações e n incógnitas, o número total de operações é

$$\frac{2}{3}n^3+\frac{n^2}{2}-\frac{7}{6}n+n^2 = \frac{2}{3}n^3+\frac{3n^2}{2}-\frac{7}{6}n,$$

que é um número próximo de $\frac{2}{3}n^3$ para n grande.

Seja A uma matriz quadrada de ordem n. A é dita uma *matriz de banda* L se a_{ij}, sempre que $|i-j| > L$. Por exemplo, a matriz abaixo é de ordem 4 e banda 1.

$$A = \begin{bmatrix} 2 & 9 & 0 & 0 \\ 1 & 2 & 3 & 0 \\ 0 & 1 & 7 & 1 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

a) É possível adaptar seu algoritmo para o caso de matrizes de banda, tirando proveito dessa informação? Qual seria o custo computacional, em termos do número de operações desse algoritmo?

É possível. Para isso, devemos iterar apenas sobre a banda, já que nos outros termos fora da banda já temos garantido os termos zeros.

Nesse caso como estamos trabalhando com uma matriz de banda, o algoritmo dependerá da banda L que for atribuída a matriz. De qualquer forma, o custo operacional será relativamente menor comparado a uma matriz tradicional, já que dependerá de menos operações.

Como o número de operações de uma matriz está diretamente ligada a Eliminação Gaussiana (que faz com que transformemos em zero os elementos abaixo da diagonal principal), e em uma matriz de banda teremos um número de elementos, não nulos, reduzidos, isso resultará em um número reduzido de operações, dado por um valor aproximado de $2L^2n + O(n^3)$.

$$\sum_{k=1}^{n-1} 2 \cdot min(L, n-j) \cdot min(L+1, n-j+1) \approx 2L^2n$$

b) Considere uma matriz de banda 5 e ordem 100. Quantos elementos são seguramente nulos nessa matriz? Quais os custos computacionais para o cálculo da decomposição LU com o algoritmo convencional e com o algoritmo adaptado?

Para sabermos os elementos seguramente nulos, vamos primeiro calcular os seguramente não nulos e então subtrair do total de elementos.

Como a matriz é de ordem 100, ela possui $100 \cdot 100 = 10.000$ elementos.

Abaixo, segue uma fórmula deduzida para calcular o número de posições não nulas de uma matriz de banda L e ordem n. A seguir, é apresentado o cálculo feito para a matriz em questão de ordem $100 \cdot 100$ e banda 5:

Ilustrando a dedução da fórmula através de um exemplo:

Seja a matriz A, de ordem 5 e $L = 2$:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} \\ 0 & 0 & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

Nas primeiras L linhas, onde $L = banda$, teremos $L + i$ entradas não nulas, com i variando de 1 a L. Nas últimas entradas, o padrão se repete para L linhas.

Para $n - 2L$ linhas, onde n é a ordem da matriz, temos $2L + 1$ termos não nulos. Daí, deduzimos a seguinte fórmula:

Números de entradas não nulas da matriz A, de ordem n e ordem L:

$$2\sum_{i=1}^{L}(L+i) + [(n-2L)\cdot(2L+1)]$$

Para o caso acima, teremos:

$$2\sum_{i=1}^{2}(2+i) = 2\left[(2+1)+(2+2)\right] = 14;\ (5-2\cdot 2)\cdot(2\cdot 2+1) = 5$$
$$2\sum_{i=1}^{L}(L+i) + [(n-2L)\cdot(2L+1)] = 14 + 5 = 19 \text{ entradas não nulas;}$$

Com isso, o número de elementos certamente nulos é:

$$n^2 - \left\{2\sum_{i=1}^{L}(L+i) + [(n-2L)\cdot(2L+1)]\right\}$$

No caso do exemplo acima, o número de zeros é:

$$25 - 19 = 6$$

Agora, para o cálculo para a matriz de ordem 100 e banda $L = 5$:

Pela fórmula deduzida acima, o número de termos nulos é:

$$n^2 - \left\{2\sum_{k=1}^{L}(L+i) + [(n-2L)\cdot(2L+1)]\right\} =$$
$$10.000 - \{2\left[6+7+8+9+10\right] + [100-2\cdot 5]\left[2\cdot 5+1\right]\} = 10.000 - [80+990] = 8.930 \text{ termos nulos}$$

Os custos operacionais para o método convencional em relação ao número de operações é proporcional a $\frac{2}{3}\cdot n^3$, enquanto o método para as matrizes de banda relacionado ao número de operações é proporcional a nL^2, o que mostra que o algoritmo feito para a matriz de banda terá um custo bem menor se comparado ao algoritmo convencional.

Assim, temos que os custos operacionais sem a implementação do método seriam:

$$\frac{2}{3}\cdot(100^3) = 666.666,66...$$

Enquanto que com a implementação do código seria:

$$100\cdot(5^2) = 2.500$$

A decomposição LU com pivoteamento surgiu como uma ferramenta para resolver sistemas lineares, porém se mostra útil também para outros fatores. Pesquise ao menos três situações em que a decomposição LU é útil, justificando. Mostre, para cada situação, um exemplo não trivial (cite sua fonte).

1. Decomposição LU no cálculo de autovalores de matrizes[1]:

   A decomposição LU é usada indiretamente para determinar autovetores e autovalores, sendo incorporada em algoritmos que realizam esse cálculo e assim tornando o processo mais eficiente.

   Um exemplo de sua aplicação em algoritmos é o algoritmo QR, que é um método iterativo para encontrar autovalores. A decomposição LU é aplicada como parte do processo iterativo para decompor a matriz na forma LU, tornando mais fácil realizar as manipulações durante o cálculo dos autovalores.

2. Decomposição LU na análise de estabilidade e convergência[2]:

   A decomposição LU também pode ser usada na análise de estabilidade e convergência de métodos numéricos para EDP's. Uma das maneiras é ela ser empregada na análise da estabilidade de métodos de diferenças finitas ou de elementos finitos, ajudando a determinar os limites de estabilidade desses métodos.

   Um exemplo prático de seu uso pode ser encontrado na simulação de fluxo de calor em uma peça de metal. Para que haja uma garantia que a temperatura na peça de metal permaneça dentro de limites seguros durante o funcionamento da máquina, é modelado o processo de resfriamento da peça de metal usando equações diferenciais parciais (EDPs) e métodos numéricos, em que é usada a decomposição LU para uma maior precisão e estabilidade dos métodos.

3. Decomposição LU para achar a inversa de uma matriz quadrada[3]:

   Sua aplicação se torna mais eficiente que métodos mais tradicionais, principalmente para matrizes maiores. A partir do momento que uma matriz A é decomposta em LU, a inversa de A pode ser computada através da resolução do sistema linear com matrizes identidade. Especificamente, se A foi decomposta em LU, de modo com que $A = LU$, então $A^{-1} = U^{-1}L^{-1}$. Como L e U são matrizes triangulares, torna-se fácil calcular suas inversas.

   As implementações padrão que usam o método simplex (de Dantzig's) para programação linear são baseadas na formação de matrizes inversas e em suas atualizações a cada etapa do método. Como essas implementações tem propriedades ruins de erro de arredondamento, a decomposição LU passa a ser usada. Apesar de lenta, é mais eficaz e resulta um melhor comportamento ao erro de arredondamento.

Referências

[1]Strang, G. (2006). Introduction to Linear Algebra. Wellesley-Cambridge Press.

[2]Karniadakis, George & Sherwin, Spencer. (2005). Spectral/HP Element Methods for Computational Fluid Dynamics. 10.1093/acprof:oso/9780198528692.001.0001.

[3]Golub, G. H.: The simplex method of linear programming using LU decomposition. Comm. ACM. 12, 266-268 (1969).