MS211 - Cálculo Numérico

Guilherme Nunes Trofino

1 de abril de 2021

Conteúdo

1 Sistemas Lineares

Definição Seja $A = (a_{ij})$ uma matriz $n \times n$ e $y_1, \cdots, y_n$ escalares então um *Sistema Linear* com *n-equações* e *n-incógnitas* é dada pela família:

$$\begin{cases} a_{11}x_1 & a_{12}x_2 & \cdots & a_{1n}x_n = b_1 \\ a_{21}x_1 & a_{22}x_2 & \cdots & a_{2n}x_n = b_2 \\ \cdots \\ a_{n1}x_1 & a_{n2}x_2 & \cdots & a_{nn}x_n = b_n \end{cases}$$

Representação Simplificadamente estes sistemas podem ser apresentados como matrizes e vetores multiplicados, $A \times X = B$, onde A representa a *Matriz dos Coeficientes*, X representa o *Vetor de Variáveis* e B representa o *Vetor de Resultados.*

$$Ax = b$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Matriz Inversa A matriz $A \in R^{n \times n}$ é não-singular, isto é, inversível, se, e somente se, exista uma matriz A^{-1}, nomeada como inversa de A, tal que:

$$A \times A^{-1} = A^{-1} \times A = I$$

em que $I \in R^{n \times n}$ representa a matriz identidade. Algebricamente temos que A será não-singular se, e somente se, $det(A) \neq 0$.

2 Méodo da Eliminação de Gauus

Definição Este algoritmo permite solucionar sistemas lineares quadrados de maneira sistématica independente de seu tamanho ou coeficientes.

1. Primeiramente será necessário concatenar a matriz dos coeficientes com o vetor de resultados.

$$[A|B]^0 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & | & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & | & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & | & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & | & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & | & b_n \end{bmatrix}$$

2. Em seguida aplicam-se sucessivas operações elementares pocurando transformar a concatenada em uma *Matriz Triangular Superior*, para isso, modificando cada entrada de uma coluna, abaixo de seu pivô, para zero.

$$[A|B]^1 = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & | & b_1^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & | & b_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & | & b_3^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & | & \vdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} & | & b_n^{(1)} \end{bmatrix}$$

$$[A|B]^2 = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} & | & b_1^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & | & b_2^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & | & b_3^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & | & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} & | & b_n^{(2)} \end{bmatrix}$$

$$\boxed{[A|B]^n = \begin{bmatrix} a_{11}^{(n)} & a_{12}^{(n)} & a_{13}^{(n)} & \cdots & a_{1n}^{(n)} & | & b_1^{(n)} \\ 0 & a_{22}^{(n)} & a_{23}^{(n)} & \cdots & a_{2n}^{(n)} & | & b_2^{(n)} \\ 0 & 0 & a_{33}^{(n)} & \cdots & a_{3n}^{(n)} & | & b_3^{(n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & | & \vdots \\ 0 & 0 & 0^{(n)} & \cdots & a_{nn}^{(n)} & | & b_n^{(n)} \end{bmatrix}}$$

3. Posteriormente, se for necessário, deve-se realizar a substituição reversa para obter as soluções, no caso de um sistema de equações.

Este processo possui complexidade $O(n^3)$ e a substituição reversa possui complexidade O^2

Algoritmo

2.1 Pivoteamento Parcial

Motivação No algoritmo implementado anteriormente a eliminação irá falhar quando ao menos um dos pivôs na matriz considerada for igual a zero, pois este elemento será o denominador de uma das operações.

Definição Afim de evitar este problema antes de iniciar o j-ésimo estágio do algoritmo, permutam-se as linhas da matriz A de modo a obter a seguinte organização:

$$|a_{jj}^{(j-1)}| \geq |a_{ij}^{(j-1)}|; \forall i = j, \ldots, n$$

Algoritmo

2.2 Substituição Reversa

Definição

3 Fatoração LU

Definição Este algoritmo permite solucionar sistemas lineares quadrados de maneira sistématica independente de seu tamanho ou coeficientes. Diferentemente da *Eliminação de Gauss*, este método permite resolver diferentes sistemas lineares sem refazer todos os cálculos.

São geradas duas matrizes: L, *Triangular Inferior* originada da Identidade, e U, *Triangular Superior*. Tais que $LU = A$, representadas genericamente como:

1. Primeiramente incializamos as matrizes bases para este método. L será a matriz identidade enquanto U será uma cópia da matriz inicial A.

$$U^0 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \quad L^0 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

2. Em seguida realizamos operações elementares para transformar U em uma matriz diagonal superior e os coeficientes do método de gauss são salvos na respectiva entrada de L.

$$U^1 = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & u_{32} & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & u_{n2} & u_{n3} & \cdots & u_{nn} \end{bmatrix} \quad L^1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ i_{21} & 1 & 0 & \cdots & 0 \\ i_{31} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ i_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$U^2 = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & u_{n3} & \cdots & u_{nn} \end{bmatrix} \quad L^2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ i_{21} & 1 & 0 & \cdots & 0 \\ i_{31} & i_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ i_{n1} & i_{n2} & 0 & \cdots & 1 \end{bmatrix}$$

$$\boxed{U^n = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{bmatrix}} \quad \boxed{L^n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ i_{21} & 1 & 0 & \cdots & 0 \\ i_{31} & i_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ i_{n1} & i_{n2} & i_{n3} & \cdots & 1 \end{bmatrix}}$$

3.

3.1 Pivoteamento Parcial

Motivação A fatoração, como implementada anteriormente, irá falhar quando qualquer um dos pivôs da matriz for igual a zero, pois este elemento será posteriormente o denominador de uma das operações.

Definição Afim de evitar este problema antes de iniciar o j-ésimo estágio do algoritmo, permutam-se as linhas da matriz A de modo a satisfazer:

$$PA = LU$$

Onde P é a *Matriz de Permutação*, obtida permutando as linhas da matriz identidade.

Teorema Qualquer matriz $A \in \mathbb{R}^{n \times n}$ não-singular pode ser fatorada como

$$PA = LU$$

em que U é traingular superior, L é triangular inferior com diagonal unitária e P é a matriz de permutação.

3.2 Solução de Sistemas

Definição

4 Ponto Flutuante

Definição Computadores não operam com números reais, estes utilizam da representação por *Ponto Flutuante Normalizado* que torna suas operações mais eficientes. Entretanto essa conversão apresenta limitações e consequentemente haverá erros pela maneira como os números são representados.

$$x = \pm 0.d_1 d_2 \cdots d_t \times \beta^e$$

Notação Denotamos por $F(\beta, t, m, M)$ o conjunto de todos os pontos flutuantes representados nessas condições.

1. β é a base em que o número será representado;
2. t é o número de dígitos representados do número de acordo com as regras que seguem:
   (a) $d_1 \neq 0$, do contrário o ponto poderia ser deslocado e expoente incrementado;
3. e é o expoente da base do número representado, variando entre $-m \geq e \geq M$

4.1 Erro Absoluto

Definição Essa representação numérica não compreende todos os números reais e portanto quando a função $F(\beta, t, m, M)$ é aplicada para um número real X será retornado $\overline{X}$, o arredondamento de X em ponto flutuante tal que o *Erro Relativo*, denotado abaixo, seja mínimo:

$$\boxed{|X - \overline{X}|}$$

4.2 Erro Relativo

Definição Há situações em que a representação absoluta não será útil pois possirá uma esquema distinta dos números. Isso é solucionado pela normalização do erro com relação ao número utilizado. O *Erro Absoluto* pode ser obtido a partir da seguinte equação:

$$\boxed{\frac{|X - \overline{X}|}{|X|}}$$

4.3 Erro de Márquina

Definição Afim de estimar a precissão dos computadores denota-se o *Épsilon da Márquina* como metade da distância entre 1 e o menor ponto flutuante estritamente maior que 1.

$$\boxed{\epsilon_{mach} = \frac{1}{2}\beta^{1-t}}$$

5 Atividades

5.1 Laboratórios 1

Questão 1 Quando $n = 10^0$ temos erro negativo, visto que $2 - 2.71828 = -0.71828$. Conforme n cresce esta diferença se reduz até que, em $n = 10^9$, obtemos um valor calculado superior ao esperado, pois quando n se torna demasiado grande as aproximações computacionais se tornam relevantes e influenciam no resultado.

Tal comportamento se mantém até $n = 10^{15}$ quando obtemos um "erro"aproximado de 3.035 e, em $n = 10^{18}$, obtemos um "erro"$e = 1$. Neste ponto o valor computado para 1/n é considerado como 0 e portanto se torna desprezível. Nesta simulação está claro que a capacidade computacional é limitada e deve ser considerando em cálculos de grande precisão.

Questão 2 Considerando que este algoritmo é baseado no método de Eliminação de Gauss são efetuadas $O(n^3)$ operações. A Eliminação de Gauss realiza $O(n^3)$ para solucionar um sistema linear $Ax = b$ assim como a Fatoração LU, $O(n^3)$, entretanto a solução dos sistemas triangulares $Ux = y$ e $Ly = b$ demandam apenas $O(n^2)$.

Questão 3 A matrix não singular a seguir apresenta falha ao ser solucionada:

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Isso se deve ao fato de que tanto a Fatoração LU quanto a Eliminação de Gauss, implementadas sem pivoteamento parcial, utilizam os pivôs da matriz no denominador dos cálculos em certa altura dos algoritmos. Essa divisão é matematicamente impossível e assim retorna erro de execução do código.

Questão 4 Primeiramente a equação pode ser reescrita como:

$$Bx = (2A + I)(C^{-1} + A)b$$

Simplificando a notação adotando: $G = (2A + I)$ e $h = (C^{-1} + A)b$, "G"não representa problemas computacionais e pode ser calculada diretamente. Por outro lado, "h"representa problemas computacionais por apresentar o cálculo de uma inversa, o que não é desejado. Pode-se evitar esse desgaste resolvendo o seguinte sistema:

$$C(h - Ab) = b$$

Aplicando Eliminação de Gauss ou Fatoração LU, mais eficientes do que o cálculo da inversa e, finalmente, é necessário solucionar o sistema novamente implementando um dos métodos citados anteiormente:

$$Bx = (Gh)$$

Questão 5 Solucionando algebricamente obtemos:

$$Hx = Hu \to H^{-1}Hx = H^{-1}Hu \to x = u$$

Portanto esperava-se que $x = [1, 1, ..., 1]^T$, independente do valor escolhido de n, entretanto ao realizar as operações descobrimos que os resultados obtidos flutuam e dificilmente são como esperado. Isso ocorre pois quanto maior a matriz de Hilbert menor os coeficientes em suas entradas, até que em certo momento essas sofreram com arredondamentos da máquina. Isso gera uma reação em cadeia que traz resultados equivocados.