Álgebra Linear Algorítmica - ICP115 (2021-2)
João Vitor de Oliveira Silva

PS

TODAS AS RESPOSTAS **devem ser justificadas**, POR MEIO DE **cálculos, explicação textual e/ou argumentos geométricos**. É POSSÍVEL (E RECOMENDADO) USAR A FUNÇÃO `GAUSSIAN_ELIMINATION`, DISPONIBILIZADA NO CLASSROOM, NAS QUESTÕES. COLOQUE UM PRINT DA SAÍDA DO ALGORITMO NA QUESTÃO (MATRIZ AUMENTADA INICIAL E FINAL, AS INTERMEDIÁRIAS TAMBÉM SE JULGAR NECESSÁRIO). **OBS: a substituição reversa dos sistemas deve ser feita toda por você, com os passos apresentados em sua prova.** NÃO SERÁ ACEITO APENAS O PRINT DE UMA EXECUÇÃO DA FUNÇÃO NATIVA `SYM.LINSOLVE`. PODE USAR FUNÇÕES NATIVAS PARA CÁLCULO DE DETERMINANTES, PRODUTO MATRICIAL, TRANSPOSIÇÃO E POTÊNCIA DE MATRIZES. RESPOSTAS QUE CARECEM DE JUSTIFICATIVA **não serão consideradas.**

1. (15 pontos) Seja $b > 10^{900}$ um número real. Determine **todos os valores** de a para os quais os vetores

$$(1, 1, 1), \quad (1, a, a^2), \quad (1, b, b^3)$$

são coplanares.

2. (30 pontos) Uma maneira de calcular o valor aproximado de uma função não polinomial é por meio de interpolação polinomial. Isso é feito da seguinte forma:

   - São escolhidos valores $x_1, x_2, \ldots, x_n$ em que $f(x_1), f(x_2), \ldots, f(x_n)$ são conhecidos;
   - Calcula-se os coeficientes de um polinômio $p(x)$ em que $p(x_k) = f(x_k)$;
   - Para um novo x_q não usado na interpolação, calcula-se $p(x_q)$.

   (a) Considere a função raiz quadrada $f(x) = \sqrt{x}$. Uma vez que para $x_1 = 1$, $x_2 = 4$, $x_3 = 9$ e $x_4 = 16$ o valor da função é conhecido, encontre um polinômio interpolador que passe pelos pontos

$$\{(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)), (x_4, f(x_4))\}$$

   Dica: pense primeiro qual o grau do polinômio que é necessário para passar por todos estes pontos, só depois monte o sistema e encontre seus coeficientes.

(b) Usando $x_q = 2$, qual o valor de $p(x_q)$? Este valor é próximo de $\sqrt{2}$? Para responder a pergunta anterior, compare com o valor retornado por `np.sqrt(2)`$\rightarrow$ 1.414213562373.

(c) Repita o processo anterior, mas usando $x_q = 50$. Neste caso, `np.sqrt(50)`$\rightarrow$ 7.0710678118654755.

(d) Encontre a melhor parábola na forma $q(x) = ax^2 + bx + c$ que aproxima os pontos do item (a).

(e) Faça um gráfico das funções encontradas no item (a), (d) e da função $f(x) = \sqrt{x}$. Comente brevemente suas impressões.

3. (25 pontos) Rotações no espaço tridimensional podem ser representadas por ângulos de Euler. As matrizes usadas neste processo são:

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \quad R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

Os ângulos α, β e γ são conhecidos como *yaw*, *pitch* e *roll*, respectivamente. A composição das transformações acima

$$R = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma)$$

é construída e depois aplicada sobre um vetor $v \in \mathbb{R}^3$.

(a) Construa a matriz R usando como ângulos (em radianos) $\alpha = \dfrac{\pi}{6}$, $\beta = \dfrac{\pi}{2}$ e $\gamma = \dfrac{\pi}{2}$.

(b) Mostre que R é uma matriz de rotação.

(c) **Sem usar eliminação gaussiana**, resolva o sistema linear $Rx = b$, em que $b = [1, 1, -1]^t$. *Dica: use propriedades da matriz R.*

(d) Calcule (se existir) o autoespaço associado ao autovalor 1 da matriz R. Caso o mesmo não exista, justifique.

Curiosidade: rotações realizadas desta forma estão sujeitas ao famoso problema de Gimbal Lock.

4. (30 pontos) Matrizes tridiagonais costumam ser muito comuns de aparecer em problemas físicos. Uma matriz M é dita tridiagonal se for uma matriz quadrada tal que apenas os elementos M_{ij} em que $|i-j| < 2$ podem ser diferentes de 0. Um exemplo de matriz tridiagonal 4×4 é

$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix}$$

Em muitas destes problemas físicos, é comum ser necessário resolver sistemas lineares com uma mesma matriz tridiagonal M, variando apenas o vetor b do sistema $Mx = b$.

Um certo cientista precisa resolver problemas que envolvem matriz tridiagonais. O mesmo está usando o seguinte algoritmo para se obter a solução dos sistemas lineares.

(a) Qual o algoritmo apresentado?

(b) Mostre o funcionamento do algoritmo para as entradas

$$M = \begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 10 \\ 10 \\ 0 \\ -10 \end{bmatrix}.$$

(c) O desempenho do algoritmo não pareceu tão satisfatório para matrizes tridiagonais muito grandes. Seria possível alterar o código anterior, de modo que a solução dos sistemas lineares ocorresse de forma mais eficiente? Explique detalhadamente sua resposta.

(d) Seria possível usar parte do algoritmo para calcular o determinante de matrizes tridiagonais? Justifique cuidadosamente.

Algoritmo 1: resolve_sistema(M, b)

Entrada: Matriz $(n \times n)$ M, Vetor $(n \times 1)$ b
Variável: Matriz $(n \times n)$ U, Vetor $(n \times 1)$ y, Vetor $(n \times 1)$ x,
NumReal m, NumReal soma, NumInteiro i, j, k, n

início
- $n \leftarrow$ número de elementos de b
- $U \leftarrow M$
- $y \leftarrow b$
- $x \leftarrow b * 0$
- **para** $i = 0, \ldots, n-1$ **faça**
  - **para** $j = i+1, \ldots, n-1$ **faça**
    - $m \leftarrow -\dfrac{U[j,i]}{U[i,i]}$
    - **para** $k = i, \ldots, n-1$ **faça**
      - $U[j,k] \leftarrow U[j,k] + m * U[i,k]$
    - **fim**
    - $y[j] \leftarrow y[j] + m * y[i]$
  - **fim**
- **fim**
- **para** $i = 0, \ldots, n-1$ **faça**
  - soma $\leftarrow 0$
  - **para** $j = 0, \ldots, i-1$ **faça**
    - soma $\leftarrow$ soma $+ U[n-1-i, n-1-j] * x[n-1-j]$
  - **fim**
  - $x[n-1-i] \leftarrow \dfrac{y[n-1-i] - \text{soma}}{U[n-1-i, n-1-i]}$
- **fim**
- **retorna** x

fim

OBS: o loop `for` inclui o último número apresentado no comando, só para após seu sucessor.