Lab 5: Método dos Mínimos Quadrados

INF1608 – Análise Numérica

Leonardo Seperuelo Duarte
lduarte@tecgraf.puc-rio.br
Departamento de Informática, PUC-Rio

Para este exercício, considere a representação de matrizes por vetor de ponteiros do Lab 0 e o método de solução de sistemas lineares do Lab 3. *Se usar os códigos dos laboratórios anteriores, envie suas implementações junto com a solução deste laboratório para a correção.* Se preferir, você pode copiar as funções necessárias já existentes para o código deste exercício.

Podemos resolver um sistema inconsistente na forma $A_{m \times n} x_n = b_m$, com $m > n$, através do Método dos Mínimos Quadrados (MMQ). Na sua forma mais direta, a solução do MMQ é feita resolvendo o sistema linear $n \times n$ definido pela equação normal:

$$A^T A\, \bar{x} = A^T b$$

onde A^T representa a matriz transposta de A e $\bar{x}$ a solução aproximada do problema. O erro do método pode ser avaliado pelo vetor residual $r = b - A\bar{x}$. Como métrica de erro, podemos usar a norma-2 desse vetor:

$$e = \|r\|_2 = \sqrt{\sum_{i=1}^{m} r_i^2}$$

1. Pede-se:

   (a) Implemente uma função que resolva o sistema $A_{m \times n} x_n = b_m$ pelo método dos mínimos quadrados. A função também recebe como parâmetro o vetor $\bar{x}$, já alocado com dimensão n, que deve ser preenchido com a solução aproximada. A função deve retornar a norma-2 do vetor resíduo, que indica o quão ajustado está a solução.

   ```
   double mmq (int m, int n, double** A, double* b, double* x);
   ```

   (b) Usando a função do item anterior, implemente uma função que ajuste uma parábola, $y = a + bx + cx^2$, a um conjunto de n pontos (px_i, py_i) fornecido. A função deve determinar os coeficientes a, b, e c, preenchendo os endereços respectivos recebidos, e retornar o erro (expresso como norma-2 do resíduo) do ajuste.

   ```
   double ajuste_parabola (int n, double* px, double* py,
                           double* a, double* b, double* c);
   ```

   (c) Similar ao item anterior, implemente uma função que ajuste uma cúbica, $y = a + bx + cx^2 + dx^3$, a um conjunto de n pontos (px_i, py_i) fornecido.

   ```
   double ajuste_cubica (int n, double* px, double* py,
                         double* a, double* b, double* c, double* d);
   ```

2. Ideias para testar suas funções:

   (a) Escreva um código que resolva os sistemas inconsistentes abaixo usando o MMQ. Para cada sistema, exiba na tela o vetor que representa a solução aproximada e seu respectivo erro associado (norma-2).

$$\text{i.}\quad \begin{bmatrix} 3 & -1 & 2 \\ 4 & 1 & 0 \\ -3 & 2 & 1 \\ 1 & 1 & 5 \\ -2 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ -5 \\ 15 \\ 0 \end{bmatrix} \qquad \text{ii.}\quad \begin{bmatrix} 4 & 2 & 3 & 0 \\ -2 & 3 & -1 & 1 \\ 1 & 3 & -4 & 2 \\ 1 & 0 & 1 & -1 \\ 3 & 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 2 \\ 0 \\ 5 \end{bmatrix}$$

   (b) Encontre a melhor parábola e a melhor cúbica que ajusta cada conjunto de pontos abaixo. Exiba na tela os coeficientes encontrados e os erros dos ajustes. Qual teve melhor ajuste, parábola ou cúbica, para cada conjunto de pontos?

      i. $(0,0),\ (1,3),\ (2,3),\ (5,6)$
      ii. $(1,2),\ (3,2),\ (4,1),\ (6,3)$

Agrupe os protótipos das funções pedidas em um módulo "mmq.h" e as implementações em um módulo "mmq.c". Escreva um outro módulo "main.c" para o código de teste da sua implementação.

Entrega: O código fonte deste trabalho (isto é, os arquivos "mmq.h", "mmq.c" e "main.c", *e eventuais códigos de laboratórios passados usados na solução*, **não** zipados) devem ser enviados via página da disciplina no EAD até o final da aula. O sistema receberá trabalhos com atraso (com perda de 1 ponto na avaliação) até 2 horas após o fim da aula.