

Lab 5: Sistemas Lineares e Interpolação

Prof. Waldemar Celes

Departamento de Informática, PUC-Rio

Para este exercício, considere a representação de matrizes implementada no primeiro laboratório do semestre. A matriz é representada por um vetor de ponteiros, onde cada elemento aponta para o vetor linha.

1. Para a solução de sistemas lineares na forma $Ax = b$, considere a fatoração $PA = LU$, onde P representa a matriz permutação, L a matriz triangular inferior e U a matriz triangular superior. A matriz U é obtida através do procedimento de eliminação de Gauss da matriz A . Para melhorar a estabilidade do método, deve-se empregar a estratégia de *pivotamento*, isto é, as linhas da matriz são trocadas para garantir que o elemento pivô da eliminação de cada coluna seja sempre o elemento de maior valor absoluto da coluna em questão. A matriz P é inicializada com o valor identidade e registra cada troca de linha ocorrida na eliminação de Gauss: a cada troca de linhas da matriz A , troca-se as mesmas linhas na matriz P . A matriz L tem os elementos da diagonal iguais a 1 e os valores abaixo da diagonal iguais aos *fatores* usados na eliminação de Gauss da matriz A .

Com a matriz fatorada, pode-se resolver o sistema linear para um vetor independente b qualquer, em ordem $O(n^2)$, usando substituições *bottom-up* e *top-down*.

$$Ax = b$$

$$PAx = Pb$$

$$LUx = Pb$$

$$Ly = Pb, \text{ onde } y = Ux$$

Assim, dada a matriz fatorada LU , acha-se y com substituição *top-down*: $Ly = Pb$. Em seguida, acha-se x com substituição *bottom-up* (substituição regressiva ou retro-substituição): $Ux = y$.

Pede-se:

- (a) Implemente uma função que receba como parâmetros uma matriz quadrada A e faça a fatoração $PA = LU$. As matrizes L e U devem ser armazenadas no espaço de memória da matriz A , sobrescrevendo os valores dos elementos de A . A diagonal da matriz L (com valores iguais a 1) não é armazenada. A função deve retornar a matriz de permutação P alocada dinamicamente dentro da função. O protótipo da função deve ser:

```
double** fatoracao (int n, double** A);
```

- (b) Implemente uma função que receba como entrada a matriz A fatorada (isto é, a matriz LU), a matriz de permutação P e um vetor independente b , e retorne um vetor solução x alocado dinamicamente pela função. O protótipo da função é dado por:

```
double* substituicao (int n, double** A, double** P, double* b);
```

- (c) Implemente uma função para calcular os coeficientes c_i do polinômio que interpola um conjunto de n pontos na forma:

$$P_{n-1}(x) = c_0 + c_1x + c_2x^2 + \dots + c_{n-1}x^{n-1}$$

Para resolver o sistema de equações, utilize as funções `fatoracao` e `substituicao` implementadas. A função de interpolação deve receber como parâmetros o número de pontos n , as coordenadas px e py desses pontos, e retornar o vetor dos coeficientes alocado dinamicamente, seguindo o protótipo:

```
double* interpola (int n, double* px, double* py);
```

Implemente também uma função que avalia o polinômio resultante no ponto x , retornando o valor avaliado, seguindo o protótipo:

```
double* avalia (int n, double* c, double x);
```

2. O polinômio interpolante de Lagrange é dado por:

$$P_{n-1}(x) = y_0L_0(x) + \dots + y_{n-1}L_{n-1}(x)$$

onde:

$$L_k(x) = \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_{n-1})}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_{n-1})}$$

Escreva uma função para avaliar o polinômio interpolante de Lagrange em um ponto x dado. A função deve receber como parâmetros o conjunto de pontos e o valor de x , retornando o valor avaliado, seguindo o protótipo:

```
double lagrange (int n, double* px, double* py, double x);
```

3. Escreva uma função *main* para testar suas funções. O objetivo é determinar e avaliar o polinômio que interpola os pontos:

$$(-5, 5), (-4, 5), (-3, 5), (-2, 5), (-1, 5), (0, 5), (1, 5), (2, 5), (3, 5), (4, 5), (5, 42)$$

Use os dois métodos de interpolação acima e exiba na tela os valores avaliados para os polinômios interpolantes para os seguintes valores de x : $-1, -0.8, -0.6, -0.4, -0.2, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0$.

Organize seu código da seguinte forma. O arquivo “`interp.c`” deve conter a implementação das funções `fatoracao`, `substituicao`, `interpola`, `avalia` e `lagrange`, com seus respectivos

protótipos no arquivo “interp.h”. O arquivo “teste_interp.c” deve conter os testes realizados (função *main*).

Entrega: O código fonte deste trabalho (isto é, os arquivos “interp.c”, “interp.h” e “teste_interp.c”) deve ser enviado para inf1608@tecgraf.puc-rio.br (não envie os arquivos comprimidos). A implementação completa deve ser enviada até **sexta-feira, dia 7 de outubro (prazo final)**. O assunto da mensagem para envio da implementação completa deve ser: **Lab5: XXXXXXXX**, onde **XXXXXXX** representa o número de matrícula do aluno sem o dígito de controle.