MAC 300 – Métodos Numéricos de Álgebra Linear

Exercício Programa 1

Resolução de sistemas de equações lineares

Este exercício programa tem dois objetivos: (i) estudar e desenvolver algoritmos para resolver sistemas de equações lineares; (ii) constatar que os algoritmos devem ser desenvolvidos levando em consideração as estruturas de dados disponíveis na plataforma onde serão implementados.

Para que os programas possam ser facilmente testados, adotaremos um formato de entrada padronizado para os sistemas de equações. Os arquivos de entrada deverão ter o seguinte formato: na primeira linha do arquivo deverá haver um único número inteiro n indicando a dimensão do sistema (número de variáveis e equações); depois n^2 linhas com três números (dois inteiros e um real) indicando linha, coluna e valor do elemento da matriz de coeficientes, respectivamente. Finalmente n linhas com um inteiro e um real indicando a posição e o valor dos elementos do lado direito do sistema.

O exercício programa deverá ser implementado em C. As tarefas estão divididas em duas partes. A primeira está relacionada a sistemas definidos positivos e a segunda a sistemas gerais. Um detalhe que vale para todas as rotinas que deverão ser implementadas: nenhuma delas precisará da declaração de vetores ou matrizes locais.

Primeira parte: sistemas definidos positivos.

Nesta etapa devem ser implementadas as seguintes funções:

- `int cholcol(int n, double A[][nmax]);`

  Recebe um inteiro n e uma matriz $A \in I\!R^{n\times n}$ e devolve, na parte triangular inferior da própria A, o fator de Cholesky G. Suponha que só a parte triangular inferior da matriz A foi armazenada (i.e., sua implementação não deve utilizar para nada os elementos acima da diagonal da matriz A. Nem usá-los nem alterá-los.) A função também deve devolver 0 se G foi calculada com sucesso e -1 se a matriz A não for definida positiva.

- `int forwcol(int n, double A[][nmax], double b[]);`

  Recebe um inteiro n, uma matriz triangular inferior $A \in I\!R^{n\times n}$ e um vetor $b \in I\!R^n$ e devolve, no próprio b, a solução do sistema triangular inferior $Ax = b$. A função deve devolver 0 se o sistema foi resolvido com sucesso e -1 se a matriz A for singular.

- `int backcol(int n, double A[][nmax], double b[], int trans);`

  Recebe um inteiro n, uma matriz triangular $A \in I\!R^{n\times n}$, um vetor $b \in I\!R^n$ e um inteiro *trans* que vale 0 se o sistema triangular superior que deve ser resolvido está dado por $Ax = b$ e 1 se esse sistema está dado por $A^T x = b$. A função devolve, no próprio b, a solução do sistema desejado, 0 se o sistema foi resolvido com sucesso e -1 se a matriz A for singular.

- As 3 rotinas acima mencionadas devem ser orientadas a coluna. As 3 de baixo devem fazer o mesmo, mas devem ser implementadas orientadas a linha.

  `int cholrow(int n, double A[][nmax]);`

```
int forwrow(int n, double A[][nmax], double b[]);
int backrow(int n, double A[][nmax], double b[], int trans);
```

Para testar suas implementações da decomposição de Cholesky utilize os sistemas (com matrizes de coeficientes simétricas) que estão nos arquivos a1.dat, a2.dat, ..., a9.dat (você deve gerar estes arquivos com o gerador genmatsim.c). Os sistemas dos arquivos a8.dat e a9.dat admitem infinitas soluções pois suas matrizes de coeficientes são singulares. Seu algoritmo deve detectar isso! Já os sistemas que estão nos arquivos a1.dat, a2.dat, ..., a7.dat têm como solução única $x = (x_0, x_1, \ldots, x_{n-1})^T$ tal que $x_i = 1 + i\%(n/100)$, onde n é a dimensão do problema.

Depois dos testes, verifique se as versões orientadas a linha e a coluna estão implementadas corretamente (i.e., percorrem, nos seus laços mais internos, a matriz de coeficientes por linha e coluna, respectivamente). Para isso, meça o tempo de execução das duas versões para cada um dos 7 primeiros problemas. Num Pentium III 550 Mhz com 256Mb de RAM os tempos (medidos em segundos) são, aproximadamente, os que aparecem na Tabela 1. Seus programas devem mostrar o mesmo tipo de comportamento. Entregue, junto com o programa (código-fonte) e a tabela relacionada aos seus próprios testes, um breve comentário sobre as diferenças de tempo encontradas para cada versão (orientada a linha e a coluna) do programa.

PROBLEMA	DECOMPOSIÇÃO DE CHOLESKY					
	ORIENTADA A LINHA			ORIENTADA A COLUNA		
	$A = GG^T$	$Gy = b$	$G^T x = y$	$A = GG^T$	$Gy = b$	$G^T x = y$
1	0.00	0.00	0.01	0.02	0.00	0.00
2	0.07	0.00	0.00	0.21	0.01	0.00
3	0.21	0.00	0.00	1.10	0.00	0.00
4	0.52	0.00	0.01	3.22	0.01	0.01
5	1.25	0.01	0.00	6.81	0.01	0.01
6	2.41	0.01	0.01	12.14	0.03	0.01
7	4.12	0.03	0.01	19.55	0.03	0.02

Tabela 1: Tempos da Decomposição de Cholesky

Segunda parte: sistemas gerais.

Nesta etapa devem ser implementadas as seguintes funções:

- `int lucol(int n, double A[][nmax], int p[]);`

  Recebe um inteiro n e uma matriz $A \in \mathbb{R}^{n \times n}$ e devolve, na própria A, as matrizes L e U da decomposição LU da matriz PA, onde P é a matriz de permutação obtida ao utilizar pivoteamento parcial. A matriz P deve ser devolvida no vetor $p \in \mathbb{N}^n$. A função também devolve 0 se a decomposição LU foi calculada com sucesso e -1 se a matriz A for singular.

- `int sscol(int n, double A[][nmax], int p[], double b[]);`

  Recebe um inteiro n, uma matriz $A \in \mathbb{R}^{n \times n}$ e um vetor $p \in \mathbb{N}^n$ (ambos saída da função anterior) e um vetor $b \in \mathbb{R}^n$ e devolve, no próprio b, a solução do sistema $LUx = Pb$. Para isso, calcula-se primeiro Pb (e guarda-se em b). Em seguida, resolve-se o sistema triangular

inferior $Ly = Pb$ (guardando y em b) e, finalmente, resolve-se o sistema triangular superior $Ux = y$ (guardando x em b). A função deve devolver 0 se o sistema foi resolvido com sucesso e -1 se a matriz U for singular.

- As 2 rotinas acima mencionadas devem ser orientadas a coluna. As 2 de baixo devem fazer o mesmo, só que devem ser implementadas orientadas a linha.

```
int lurow(int n, double A[][nmax], int p[]);
```

```
int ssrow(int n, double A[][nmax], int p[], double b[]);
```

Teste suas rotinas com os sistemas que estão nos arquivos m1.dat, m2.dat, ..., m9.dat (você deve gerar estes arquivos com o gerador genmat.c). Da mesma forma que os outros arquivos de teste, os dois últimos têm matrizes de coeficientes singulares e os 7 primeiros têm as soluções descritas anteriormente.

PROBLEMA	DECOMPOSIÇÃO LU			
	ORIENTADA A LINHA		ORIENTADA A COLUNA	
	$PA = LU$	$LUx = Pb$	$PA = LU$	$LUx = Pb$
1	0.02	0.00	0.02	0.00
2	0.13	0.00	0.21	0.00
3	0.53	0.01	1.06	0.01
4	1.58	0.01	3.23	0.01
5	3.23	0.03	6.74	0.02
6	5.63	0.02	11.79	0.04
7	8.86	0.04	19.99	0.05

Tabela 2: Tempos da Decomposição LU

As mesmas coisas apontadas para a decomposição de Cholesky (com respeito as implementações orientadas a linha e a coluna) devem ser verificadas aqui. A Tabela 2 mostra os tempos de ambas as versões. Note que nos problemas grandes onde as operações que são executadas em $O(n^3)$ dominam a medição do tempo, a decomposição LU demora aproximadamente o dobro da decomposição de Cholesky. Comente brevemente esse fato. As tabelas e os comentários devem ser entregues em formato texto, ps ou pdf.