Departamento de Informática - CT/UFES
$1^{\underline{o}}$ Trabalho de Cálculo Numérico - 08/1
Curso de Física

Sistemas Lineares

Data de entrega: 25/05/08

Objetivos: Implementar algoritmos iterativos de solução de sistemas esparsos e fazer um estudo de desempenho com relação ao tempo de processamento para os métodos diretos e iterativos usando o MatLab (ou Octave).

Definição das matrizes esparsas

Neste trabalho será considerado dois tipos de matrizes esparsas: **matrizes tridiagonais** (A_t) e **matrizes pentadiagonais** (A_p) assim definidas:

$$A_t = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \dots & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_{n-2,n-1} & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

$$A_p = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \dots & a_{1,q} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \dots & 0 & a_{2,q+1} & 0 & \dots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \dots & 0 & 0 & a_{3,q+2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{q,1} & 0 & 0 & 0 & \dots & a_{q,q} & a_{q,q+1} & 0 & \dots & a_{q,n} \\ 0 & a_{q+1,2} & 0 & 0 & \dots & a_{q+1,q} & a_{q+1,q+1} & a_{q+1,q+2} & \dots & 0 \\ 0 & 0 & a_{q+2,3} & 0 & \dots & 0 & a_{q+2,q+1} & a_{q+2,q+2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_{n,q} & 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

Na implementação de métodos iterativos de sistemas esparsos é comum utilizar um armazenmento otimizado da matriz dos coeficientes. Para esses dois tipos de sistema considere as seguintes matrizes para armazenar os coeficientes de A_t e A_p:

$$AA_t = \begin{bmatrix} 0 & a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} \\ a_{32} & a_{33} & a_{34} \\ \vdots & \vdots & \vdots \\ a_{n-2,n-1} & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{n,n} & 0 \end{bmatrix}$$

$$AA_p = \begin{bmatrix} 0 & 0 & a_{11} & a_{12} & a_{1,q} \\ 0 & a_{21} & a_{22} & a_{23} & a_{2,q+1} \\ 0 & a_{32} & a_{33} & a_{34} & a_{3,q+2} \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ a_{q,1} & a_{q,q-1} & a_{q,q} & a_{q,q+1} & a_{q,n} \\ a_{q+1,2} & a_{q+1,q} & a_{q+1,q+1} & a_{q+1,q+2} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n-2,q-2} & a_{n-2,n-3} & a_{n-2,n-2} & a_{n-2,n-1} & 0 \\ a_{n-1,q-1} & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} & 0 \\ a_{n,q} & a_{n,n-1} & a_{n,n} & 0 & 0 \end{bmatrix}$$

Dado um sistema $Ax = b$ com a matriz dos coeficientes armazenada usando uma das formas citadas acima, pede-se:

1. Considerar a implementação de solução de sistemas padrão do MatLab (ou Octave) para métodos diretos considerando o armazanamento global das matrizes A_t e A_p. Neste ítem não está previsto implementação, mas o uso da função "\".

2. Implementar o algoritmo baseado no Método Iterativo de Seidel considerando as matrizes A_t e A_p armazenadas de forma otimizada respectivamente por AA_t e AA_p.

Descrições importantes:

- Para testar sua implementação considere as matrizes definidas por:

  para todo coeficiente $a_{ij} \neq 0$ tem-se

$$\begin{aligned} a_{ij} &= 8.0 \quad \text{se} \quad i = j \\ a_{ij} &= -2.0 \quad \text{se} \quad i < j \\ a_{ij} &= -1.0 \quad \text{se} \quad i > j \end{aligned}$$

  Para matrizes do tipo A_p considere o parâmetro p para definir os coeficientes não nulos das diagonais secundárias que distam p linhas da diagonal principal. Para exemplificar considere uma matriz de ordem 7 e $p = 5$:

$$A = \begin{bmatrix} 8.0 & -2.0 & 0.0 & 0.0 & -2.0 & 0.0 & 0.0 \\ -1.0 & 8.0 & -2.0 & 0.0 & 0.0 & -1.0 & 0.0 \\ 0.0 & -1.0 & 8.0 & -2.0 & 0.0 & 0.0 & -2.0 \\ 0.0 & 0.0 & -1.0 & 8.0 & -2.0 & 0.0 & 0.0 \\ -1.0 & 0.0 & 0.0 & -1.0 & 8.0 & -2.0 & 0.0 \\ 0.0 & -1.0 & 0.0 & 0.0 & -1.0 & 8.0 & -2.0 \\ 0.0 & 0.0 & -1.0 & 0.0 & 0.0 & -1.0 & 8.0 \end{bmatrix}$$

  O vetor dos termos independentes pode ser definido por:

$$b_i = \sum_{\forall \;\; j \text{ tal que } a_{ij} \neq 0} a_{ij}$$

- O tempo de processamento para os métodos iterativos depende da tolerância adotada. Considere vários valores para a tolerância, por exemplo, 0.001, 0.0001, 0.00001 e 0.000001.

- Para testar o desempenho com relação ao tempo de processamento faça um estudo da variação da tolerância com relação a razão do tempo de processamento entre os dois tipos de métodos, para tal considere:

$$r = \frac{tempo_{\textit{método iterativo}}}{tempo_{\textit{método direto}}}$$

  Para os vários valores de tolerância considerados trace o gráfico $tol \times r$.

- Considere o tempo de processamento para sistemas de ordens variadas, por exemplo considere $n = 100$, $n = 500$,$n = 1000$, $n = 5000$, $n = 10000$, etc. Na sequência de ordens escolhidas é importante comparar o desempenho dos métodos diretos e iterativos com relação ao tempo de processamento. Em cada caso trace o gráfico $n \times r$ e discuta os comportamentos obtidos.

- O MatLab (ou Octave) dispõe de funções que podem auxiliar sua implementação, tais como: **tic**, **toc**, **plot**, etc.

- Serão aceitos grupos de até 2 alunos.

- Um relatório deve ser elaborado contendo as conclusões de cada item. O código, o relatório e um arquivo texto explicando como rodar o programa devem ser enviados por e-mail.

- **Lembrando as regras de envio:** o e-mail deve ter como assunto CalcNum:trab1:<nome> e um anexo contendo um arquivo compactado. Neste caso <nome> deve conter nome e último sobrenome dos 2 (dois) participantes do grupo.