INF1608 – Análise Numérica
Lab 11: Métodos Iterativos para Sistemas Lineares

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

Para estes exercícios, considere a representação de matrizes quadradas $M_{n\times n}$ como um vetor de vetores do Lab 0.

1. Considere o método Gradientes Conjugados para solução de sistemas lineares $A\mathbf{x} = \mathbf{b}$. A versão do método Gradientes Conjugados com pré-condicionador M é dada por:

$$
\begin{array}{l}
\mathbf{x}_0 = \text{estimativa inicial} \\
\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 \\
\mathbf{d}_0 = \mathbf{z}_0 = M^{-1}\mathbf{r}_0 \\
\textbf{for}\ \ k = 0, 1, \cdots, n-1\ \ \textbf{do} \\
\quad \textbf{if}\ \ ||\mathbf{r}_k||_2 < tol\ \ \textbf{then} \\
\quad\quad \textbf{stop} \\
\quad \textbf{end} \\
\quad \alpha_k = \dfrac{\mathbf{r}_k^T \mathbf{z}_k}{\mathbf{d}_k^T A \mathbf{d}_k} \\
\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \\
\quad \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A \mathbf{d}_k \\
\quad \mathbf{z}_{k+1} = M^{-1}\mathbf{r}_{k+1} \\
\quad \beta_k = \dfrac{\mathbf{r}_{k+1}^T \mathbf{z}_{k+1}}{\mathbf{r}_k^T \mathbf{z}_k} \\
\quad \mathbf{d}_{k+1} = \mathbf{z}_{k+1} + \beta_k \mathbf{d}_k \\
\textbf{end}
\end{array}
$$

   Se fizermos $M = I$, matriz identidade, usamos o método sem pré-condicionador, e as operações com M^{-1} não precisam ser realizadas. Se fizermos $M = D$, matriz diagonal de A, temos o método com o pré-condicionador de Jacobi.

   Implemente duas funções que resolvam um sistema linear pelo método Gradientes Conjugados, sem e com pré-condicionador, dada uma estimativa inicial da solução $\mathbf{x}$. Quando a norma-2 do resíduo for menor que a tolerância especificada, a solução é considerada válida e as iterações devem ser interrompidas. As funções devem sobrescrever a solução final em $\mathbf{x}$ e retornar o número de iterações efetuado. Os protótipos das funções devem ser:

```
int gradconj (int n, double** A, double* b, double* x, double tol);
int gradconj_jacobi (int n, double** A, double* b, double* x, double tol);
```

2. Teste, analise e compare a eficiência dos métodos, sem ou com pré-condicionador, achando a solução do sistema abaixo, usando tolerância 10^{-7} e estimativa inicial igual ao vetor nulo. Seu programa deve exibir na tela o número de iterações e a solução encontrada para cada um dos métodos.

$$\begin{bmatrix} 1.0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 2.0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 3.0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 4.0 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 5.0 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 6.0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 7.0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 8.0 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 9.0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 10.0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} 1.4 \\ 2.8 \\ 3.8 \\ 4.8 \\ 5.8 \\ 6.8 \\ 7.8 \\ 8.8 \\ 9.8 \\ 10.4 \end{bmatrix}$$

Sabe-se que a solução desse sistema é $[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$.

Agrupe os protótipos das funções pedidas em um módulo "gradconj.h" e as implementações em um módulo "gradconj.c". Escreva o teste em outro módulo "main.c".

Entrega: O código fonte deste trabalho (isto é, os arquivos "gradconj.c", "gradconj.h" e "main.c", *e eventuais códigos de laboratórios passados usados na solução*) devem ser enviados via página da disciplina no EAD. O prazo final para envio é **domingo, dia 5 de junho**.