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 sem pré-condicionador é dada por:

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

   Note que os índices subscritos no código acima indicam o número da iteração, mas uma implementação não precisa ter os vetores duplicados.

   Implemente uma função que resolva um sistema linear pelo método Gradientes Conjugados, sem 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. A função deve sobrescrever a solução final em $\mathbf{x}$ e retornar o número de iterações efetuado. O protótipo da função deve ser:

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

2. Teste e analise a eficiência do método achando a solução do sistema abaixo, usando tolerância 10^{-7} e estimativa inicial igual ao vetor nulo. Seu programa teste deve exibir na tela o número de iterações e a solução encontrada.

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

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*) deve ser enviado 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 o final do dia.