INF1608 – Análise Numérica
Lab 12: Gradientes Conjugados

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

Para estes exercícios, considere a representação de matrizes simétricas como quadradas cheias $M_{n\times n}$, e use o código do Lab 0.

O método dos Gradiente Conjugado é um método direto/iterativo para solução de sistemas lineares $Ax = b$. Sua implementação segue os procedimentos a seguir, sem e com pré-condicionador M:

$x_0 =$ estimativa inicial
$d_0 = r_0 = b - Ax$
for $k = 0, 1, \cdots, n-1$ **do**
  if $||r_k||_2 < tol$ **then**
    stop
  end
  $\alpha_k = \dfrac{r_k^T r_k}{d_k^T A d_k}$
  $x_{k+1} = x_k + \alpha_k d_k$
  $r_{k+1} = r_k - \alpha_k A d_k$
  $\beta_k = \dfrac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
  $d_{k+1} = r_{k+1} + \beta_k d_k$
end

$x_0 =$ estimativa inicial
$r_0 = b - Ax$
$d_0 = z_0 = M^{-1} r_0$
for $k = 0, 1, \cdots, n-1$ **do**
  if $||r_k||_2 < tol$ **then**
    stop
  end
  $\alpha_k = \dfrac{r_k^T z_k}{d_k^T A d_k}$
  $x_{k+1} = x_k + \alpha_k d_k$
  $r_{k+1} = r_k - \alpha_k A d_k$
  $z_{k+1} = M^{-1} r_{k+1}$
  $\beta_k = \dfrac{r_{k+1}^T z_{k+1}}{r_k^T z_k}$
  $d_{k+1} = z_{k+1} + \beta_k d_k$
end

1. Implemente os procedimentos Gradientes Conjugados para resolver $Ax = b$, dada uma estimativa inicial da solução x. Quando a norma-2 do resíduo for menor que a tolerância especificada, a solução é considerada válida e as iterações devem ser interrompidas. Sua função deve atualizar o valor de `x` com a solução do sistema e retornar o número de iterações realizadas. Para a versão com pré-condicionador, deve-se particularizar a implementação para o pré-condicionador de Jacobi: $M = D$, onde D é a matriz diagonal de A.

   (a) Sem pré-condicionador:

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

   (b) Com pré-condicionador de Jacobi:

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

2. Teste suas implementações determinando e comparando a convergência e a solução dos sistemas abaixo, usando tolerância 10^{-7}. Seu programa deve exibir na tela os números de iterações e as soluções encontradas para cada sistema em cada método.

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \quad e \quad \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0.5 \\ -1 & 3 & -1 & 0 & 0.5 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0.5 & 0 & -1 & 3 & -1 \\ 0.5 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 1.5 \\ 1 \\ 1 \\ 1.5 \\ 2.5 \end{bmatrix}$$

   Sabe-se que as soluções destes sistemas são [1 2] e [1 1 1 1 1 1], respectivamente.

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”) devem ser enviados via página da disciplina no EAD. O prazo final para envio é **sexta-feira, dia 16 de novembro**.