Instituto Tecnológico de Aeronáutica
Divisão de Ciência da Computação (IEC)
CCI-22 Matemática Computacional
Prof. Dr. Johnny Cardoso Marques (johnny@ita.br)
Página do Professor: www.comp.ita.br/~johnny

Roteiro de Laboratório 3 – Sistemas Lineares

Entrega: 13/05/2019

Produzir um Relatório Descritivo e enviar para **johnny@ita.br**

Incluir no campo Assunto: **CCI-22 Lab 3 – Nome do Aluno**

Requisitos para o Relatório Descritivo:

1. Conter nome completo;
2. Conter os códigos-fonte gerados para cada questão de laboratório;
3. Conter as impressões (“prints”) dos resultados obtidos; e
4. Discutir os resultados obtidos.

Foram apresentados em sala de aula, os seguintes métodos iterativos para solução de Sistemas Lineares:

1. Método da Gauss-Jacobi
2. Método da Gauss-Siedel

Cada um dos métodos pode ser colocado em uma solução computacional usando os seguintes pseudocódigos apresentados no Anexo I e II deste roteiro.

Questão de Laboratório 1

Construa um programa em MATLAB ou em outro ambiente computacional que implemente o pseudocódigo do Método de Gauss-Jacobi.

Questão de Laboratório 2

Construa um programa em MATLAB ou em outro ambiente computacional que implemente o pseudocódigo do Método de Gauss-Siedel.

Questão de Laboratório 3

Teste os dois programas construídos usando o Exercício 1 e 2 da Listex 3.

Anexo I: Pseudocódigo do Método da Gauss-Jacobi

```
INÍCIO
Passo 1: Recebe a ordem do sistema ord
Passo 2: Recebe a matriz dos coeficientes A
Passo 3: Recebe o vetor X inicial
Passo 4: Recebe o vetor dos termos independentes b
Passo 5: Recebe a precisão Ɛ
Passo 6: Iniciar o contador de iterações k com 1
Passo 7: Para i = 1:ord
     Passo 8: Para j = 1:ord
          Passo 9a: Se i = j
                         C(i, j) = 0;
                         g (i, 1) = b(i, 1) / A(i, i);
          Passo 9b: Senão
                         C(i, j) = -A(i, j) / A(i, i);
Passo 10: Enquanto 1
          Passo 11: X1=C*X+g;
               Passo 12: Para i=1:ord
                              deltaX(i,1)=módulo(X1(i,1)-X(i,1));
          Passo 13: Encontre o maior valor em módulo dentro de X1 (dr1)
          Passo 14: Encontre o maior valor em módulo dentro de deltax (dr2)
          Passo 15: Faça dr=dr2/dr1;
          Passo 16: Se dr<Ɛ    %CRITÉRIO DE PARADA
                         Sair do loop enquanto
          Passo 17: X = X1;
          Passo 18: k = k+1;
Passo 19: Imprimir o número de iterações k
Passo 20: Imprimir X1 (solução do sistema)
Passo 21: Imprimir a precisão calculada dr que atendeu ao critério de parada
(dr<Ɛ)
FIM
```

ANEXO II: Pseudocódigo do Método da Gauss-Siedel

```
INÍCIO
Passo 1: Recebe a ordem do sistema ord
Passo 2: Recebe a matriz dos coeficientes A
Passo 3: Recebe o vetor X inicial
Passo 4: Recebe o vetor dos termos independentes b
Passo 5: Recebe a precisão Ɛ
Passo 6: Iniciar o contador de iterações k com 1
Passo 7: Para i = 1:ord
     Passo 8: Para j = 1:ord
          Passo 9a: Se i = j
                         C(i, j) = 0;
                         g (i, 1) = b(i, 1) / A(i, i);
          Passo 9b: Senão
                         C(i, j) = -A(i, j) / A(i, i);
Passo 10: Enquanto 1
          Passo 11: X1=X;
          Passo 12: Para i=1:ord
                    X1(i,1) = C(i,:)*X1 + g(i,1);
                    % (OBS. A EXPRESSÃO C(I,:)*X1 REALIZA UMA MULTIPLICAÇÃO
                    % ENTRE DOIS VETORES (UMA LINHA DA MATRIZ E O VETOR X))
          Passo 13: Para i=1:ord
                         deltaX(i,1)=módulo(X1(i,1)-X(i,1));
                         % CALCULAR A MATRIZ DE DIFERENCAS ENTRE O RESULTADO
                         % DA ITERAÇÃO ATUAL E A ANTERIOR
          Passo 14: Encontre o maior valor em módulo dentro de X1 (dr1)
          Passo 15: Encontre o maior valor em módulo dentro de deltax (dr2)
          Passo 16: Faça dr=dr2/dr1;
          Passo 17: Se dr<Ɛ    %CRITÉRIO DE PARADA
                         Sair do loop enquanto
          Passo 18: X = X1;
          Passo 19: k = k+1;
Passo 20: Imprimir o número de iterações k
Passo 21: Imprimir X1 (solução do sistema)
Passo 22: Imprimir a precisão calculada dr que atendeu ao critério de parada
dr<Ɛ)
FIM
```

DICA: No MATLAB para sair do loop quando uma condição for satisfeita, use *break*.