Universidade Federal do Rio de Janeiro
Escola Politécnica
Departamento de Recursos Hídricos e Meio Ambiente

EEH591 - MODELAGEM HIDRÁULICA & AMBIENTAL

TP5 – Quinto Trabalho Prático

Entrega: 15/05/2017 (pode ser feito em grupos de até 4 alunos) Devolução: 31/05/2017

Questão 1:

a) Escreva uma subrotina para resolver sistemas de equações tri-diagonais, usando o algoritmo de dupla varredura (*double sweep*) como apresentado na seção 4.6.3 do livro *Computational Fluid Dynamics* (Abbott & Basco). [Dica: use precisão dupla.]

Obs: Mando em anexo um exemplo escrito em FORTRAN. Atenção: no exemplo, os vetores A e C estão trocados em relação ao que consta no capítulo 4 do livro.

b) Escreva um programa, usando a subrotina do item a), para resolver um sistema tridiagonal de **100 equações lineares**, cuja matriz de coeficientes tem em cada posição o índice da linha como inteiro e o da coluna dividido por 1000 como decimal, e o lado direito de cada equação é igual ao índice da linha. Tal esquema está exemplificado abaixo:

$$\begin{bmatrix} 1.001 & 1.002 & 0 & 0 & . & . & . & . & . \\ 2.001 & 2.002 & 2.003 & 0 & 0 & . & . & . & . \\ 0 & 3.003 & 3.003 & 3.004 & 0 & 0 & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & 0 & 0 & 98.097 & 98.098 & 98.099 & 0 \\ . & . & . & . & 0 & 0 & 99.098 & 99.099 & 99.10 \\ . & . & . & . & . & 0 & 0 & 100.099 & 100.100 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ x_{98} \\ x_{99} \\ x_{100} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ . \\ 98 \\ 99 \\ 100 \end{bmatrix}$$

Como visto acima, para uma matriz de 100 equações a primeira linha seria 1.001; 1.002 e a última 100.099; 100.100. Tal matriz será mal condicionada, por isso é importante usar precisão dupla. O programa deve criar os vetores de coeficientes A, B, C e D, e chamar a subrotina para resolver o sistema. Dica: se quiser checar o programa, este caso dá para resolver no Excel ou similar.

c) Faça um gráfico de barra indicando no eixo horizontal as variáveis e no vertical a solução encontrada.

Questão 2:

Resolva os problemas 7 e 10 do capítulo 4 do livro *Computational Fluid Dynamics*.

Questão 3:

Repita o problema 7 indicado na questão 2, com as seguintes modificações:

a) Use o esquema implícito de Crank-Nicholson (θ = ½).
b) Idem. Use a subrotina da questão 1 para computar.
c) Idem.
d) Compare e comente os resultados das duas versões do problema 7.

____ Ω ____

```
*********************************************************************
*
      SUBROUTINE VARRE2 (A,B,C,D,N,NORDEM)
*
*********************************************************************
*
*  Subrotina para cálculo de sistemas tridiagonais Algoritmo  "doublé
*  sweep". Secao 4.6.3. do livro Computational Fluid Dynamics (Abbot &
*  Basco).
*     N = numero de equações
*     Equações do tipo
*         A[j] Z(j-1) + B[j] Z(j) + C[j] Z(j+1) = Dj              (1)
* ####  Note que A e C estão trocados em relação ao livro   ####
*
*     Condições de contorno:
* ####  ATENÇÃO: caso haja condição do tipo mista ou tipo Neumman, ou
* ####           seja, BET1 ou BETN são diferentes de zero, o ideal ,
* ####           armar o sistema de modo a colocar tal condição
* ####           no ponto 1.
*
*      ponto 1  (ALF1 e BET1 são constantes)
*       ALF1*Z(1)+BET1*(dZ(1)/dx)=CC1                             (2)
*     para um esquema de diferenças finitas progressivas de segunda
*     ordem, usando Z(3)=E2{E1*Z(1)+F1}+F2, e Z(2)=E1*Z(1)+F1,
*     ou seja, a recorrência do double sweep, tem-se:
*       Z(1)={CC1-BET1/(2DX)*[F1(4-E2)-F2]}/
*            {ALF1+BET1/(2DX)*[E1(4-E2)-3]}
*     ponto N  (ALFN e BETN são constante)
*       ALFN*Z(N)+BETN*(DZ(N)/DX)=CCN                             (3)
*     que pode ser escrito como
*             -E[N-1]*Z(N-1) + Z(N) = F[N-1]
*     onde E[N-1] e F[N-1] são funções da esquematização de (3).
*     por exemplo:
*         - esquema de diferenças regressivas de primeira ordem;
*               E[N-1] = BETN/DX/{ALFN+BETN/DX}
*               F[N-1] = CCN/{ALFN+BETN/DX}
*         - esquema de diferenças regressivas de segunda ordem;
*               E[N-1] = 4BETN/(2DX)/{ALFN+3*BETN/(2DX)}
*               F[N-1] = {CCN-BETN/(2DX)*Z'(N-2)}/{ALFN+3*BETN/(2DX)}
* onde Z'(N-2) seria o valor de Z(N-2) extrapolado ou interpolado
* para o instante de tempo adequado, dependendo do esquema de
* discretização temporal adotado. Note que Z'(N-2) só e necessário no
* caso de condição de contorno com derivada, i.e., BETN=/=0.
* De qualquer modo, os valores de E[N-1] e F[N-1] são passados para a
* subrotina pelo programa principal.
* Deste modo define-se os valores dos vetores A, B, C , D nas posições 1 e * e N-1, ou seja:
*        ponto 1:
*           A[1]=0.0
*           B[1]=ALF1
*           C[1]=BET1/(DX)  ...esquema de ordem DX
*           C[1]=BET1/(2DX) ...esquema de ordem DX^2
*           D[1]=CC1
*        ponto N:
*           A[N]=-E(N-1)
*           B[N]=1.0
*           C[N]=0.0
*           D[N]=F(N-1)
*
*     Resposta sai no vetor D.
*********************************************************************
*
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (NM=200)
c
      DIMENSION A(NM),B(NM),C(NM),D(NM),E(NM),F(NM)
c
C     le E(N-1) & F(N-1)
      E(N-1)=-A(N)
      F(N-1)= D(N)
C
C     Varre de N-1 ate 2 para calcular E & F de N-2 ate 1.
      DO J=N-1,2,-1
         DEN=C(J)*E(J)+B(J)
         E(J-1)=-A(J)/DEN
         F(J-1)=(D(J)-C(J)*F(J))/DEN
      ENDDO
C
```

```fortran
C      Calcula Z(1) e coloca em D[1]
       IF (C(1).EQ.0.0) THEN
          D(1)=D(1)/B(1)
       ELSEIF (NORDEM.EQ.2) THEN
          D(1)=(D(1)-C(1)*(F(1)*(4.0-E(2))-F(2)))/
      &        (B(1)+C(1)*(E(1)*(4.0-E(2))-3.0))
       ELSE
          D(1)=(D(1)-C(1)*F(1))/(B(1)+C(1)*(E(1)-1.0))
       ENDIF
C
C     Varre de 1 ate N-1 para calcular resposta de 2 ate N, pondo em D.
       DO J=1,N-1
          D(J+1)=D(J)*E(J)+F(J)
       ENDDO
C
       RETURN
       END
```