30/Nov/2020

5o. PROGRAMA - EQUAÇÕES DIFERENCIAIS PARCIAIS PARABÓLICAS
EQUAÇÃO DO CALOR OU EQUAÇÃO DE DIFUSÃO
MÉTODO DE CRANK-NICOLSON

A equação do calor é tipicamente uma equação diferencial parcial parabólica e pode ser expressa por

$$u_t - \sigma u_{xx} = 0. \tag{1}$$

Resolva numericamente a eq. do calor com as condições

$$\begin{aligned} u(x,0) =& \sin(\pi x),\ 0 \leq x \leq 1 \text{ (condição inicial)}, \\ u(0,t) =& u(1,t) = 0,\ t \geq 0 \quad \text{(condições de contorno)}. \end{aligned} \tag{2}$$

A solução analítica exata é dada por [1]

$$u(x,t) = \exp(-\pi^2 t) \sin(\pi x). \tag{3}$$

1) Use o método de discretização explícito

$$U_m^{n+1} = U_m^n + \frac{k\sigma}{h^2}\left[U_{m+1}^n - 2U_m^n + U_{m-1}^n\right] \tag{4}$$

para resolver a equação (1) com as condições (2) para $t = 0.02$, 0.04, 0.06, 0.08,
usando $h = 0.1$ e $k = 0.005$ ($r = k\sigma/h^2 = 0.5$) (estável)
$h = 0.05$ e $k = 0.0025$ ($r = k\sigma/h^2 = 1$) (instável)
Este método também é chamado de "forward time centered space(FTCS)" [2]. É $\mathcal{O}(k + h^2)$.

2) a) Aplique a Regra do Trapézio e mostre que a expressão (1) pode ser discretizada como

$$U_m^{n+1} - \frac{1}{2}r\left[U_{m+1}^{n+1} - 2U_m^{n+1} + U_{m-1}^{n+1}\right] = U_m^n + \frac{1}{2}r\left[U_{m+1}^n - 2U_m^n + U_{m-1}^n\right] \tag{5}$$

onde $r = \frac{k\sigma}{h^2}$, $U_m^0 = u(mh,0) = \sin(\pi hm)$ ($m = 1,...,M-1$), $U_0^n = U_M^n = 0$ (condições de contorno). Esta discretização corresponde ao método de Crank-Nicolson[3]. É $\mathcal{O}(k^2 + h^2)$.

b) A equação (5) pode ser escrita na forma matricial

$$\begin{bmatrix} 1+r & -r/2 & 0 & & & 0 \\ -r/2 & 1+r & -r/2 & & & 0 \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & -r/2 & 1+r & -r/2 \\ & & & & -r/2 & 1+r \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ \cdot \\ \cdot \\ \cdot \\ U_{M-1}^{n+1} \end{bmatrix} = \begin{bmatrix} 1-r & r/2 & 0 & \cdot & \cdot & 0 \\ r/2 & 1-r & r/2 & & & 0 \\ & \cdot & \cdot & \cdot & & \cdot \\ & & \cdot & \cdot & \cdot & \cdot \\ & & & r/2 & 1-r & r/2 \\ & & & & r/2 & 1-r \end{bmatrix} \begin{bmatrix} U_1^n \\ U_2^n \\ \cdot \\ \cdot \\ \cdot \\ U_{M-1}^n \end{bmatrix} \tag{6}$$

Para avançar no tempo basta resolver este sistema de equações. Como a matriz da esquerda é tridiagonal sua solução pode ser feita por decomposição **LU** (veja apêndice A). Já que a matriz da esquerda é constante, a decomposição **LU** só é necessária uma vez.

Calcule U_m^{n+1}, $m = 1,...,M-1$ para os tempos, condições e passos do item 1). Sugestão de algoritmo no Apêndice B.

Referências

[1]D.Quinney, *An Introduction to Numerical Solution of Differential Equations*, revised ed., John Wiley & Sons, 1987.
[2]W.H.Press, S.A.Teukolsky, W.T.Vetterling, B.P.Flannery, *Numerical Recipes in Fortran(C),* 2nd ed., Cambridge Univ. Press, 1992.
[3]J.Crank and P.Nicolson, Proc. Camb. Phil. Soc. **32**, 50 (1947). op. cit. in [4].
[4]W.F.Ames, *Numerical Methods for Partial Differential Equations*, 3rd ed.,1992.

APÊNDICE A

Seja um sistema de equações do tipo **Ax=y** onde y é um vetor conhecido e **A** é uma matriz tridiagonal

$$\mathbf{A}=\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{bmatrix}$$

A matriz **A** pode ser decomposta como **A=LU** onde

$$\mathbf{L}=\begin{bmatrix} 1 & & & & & \\ l_2 & 1 & & & & \\ & \cdot & \cdot & & & \\ & & \cdot & \cdot & & \\ & & & l_{n-1} & 1 & \\ & & & & l_n & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{U}=\begin{bmatrix} w_1 & v_1 & & & \\ & w_2 & v_2 & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & \cdot & v_{n-1} \\ & & & & w_n \end{bmatrix}.$$

Os elementos de **L** e **U** podem ser encontrados com as relações $v_i = c_i$, $w_1 = b_1$, $l_j = a_j/w_{j-1}$, $w_j = b_j - l_j c_{j-1}$, $j = 2, ...n$. Temos então o sistema **LUx=y**. Chamando **z=Ux** resolve-se **Lz=y** por substituição para frente (forward substitution) encontrando **z**. Em seguida resolve-se **Ux=z** por substituição para trás (backward substitution), determinando finalmente o vetor **x**, solução do sistema **Ax=y.**

APÊNDICE B

Algoritmo para solução da equação do calor pelo método de Crank-Nicolson:

$\pi =$acos(-1.0)
$\sigma = 1$
$imp = 10$
$mult = 1$
$M = mult \times imp$! no. de espaços em x $(M \times hx = 1.)$
$hx = 0.1$! discretização em espaco (h)
$ht = 0.005$! discretização no tempo (k)
$r = \sigma ht/(hx)^2$
************decomposição tridiagonal
Fazer de $j = 1$ até $M - 1$! carrega matriz **A**
$a_j = -0.5r$
$b_j = 1 + 2r$
$c_j = -0.5r$
fim

$w_1 = b_1$
Fazer de $j = 2$ até M ! determina as matrizes **L** e **U**
$l_j = a_j/w_{j-1}$
$w_j = b_j - l_j c_{j-1}$
fim
escreva 'fim da decomposição LU'
****
Fazer de $m = 1$ ate $M - 1$! função inicial $u(x, t = 0)$ a ser evoluida no tempo
$x = m \times hx$
$u_m = \sin(\pi \times x)$
fim
\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#avanco no tempo
ler n_f ! no. de passos a ser evoluido no tempo

Fazer de $n = 1$ até n_f
$u_0 = 0$
$u_M = 0$
Fazer de $m = 1$ até $M - 1$! carrega vetor y
$y_m = 0.5r \times u_{m-1} + (1 - r)u_m + 0.5r \times u_{m+1}$
fim
$z_1 = y_1$
Fazer de $j = 2$ até $M - 1$! substituição para frente
$z_j = y_j - z_{j-1}l_j$
fim
$u_{M-1} = z_{M-1}/w_{M-1}$! substituição para trás
Fazer de $i = M - 2$ até 1
$u_i = (z_i - c_iu_{i+1})/w_i$
fim
######´Fim do avanço no tempo'
***********impressão dos resultados
$tempo = n_f \times ht$
escreva n_f, $tempo$
Fazer $i = 0$ até imp
$m = mult \times i$
$x = m \times hx$
$exato = \exp(-\pi^2 \times tempo) \times \sin(\pi \times x)$
escreva x, $exato$, 'exato'
escreva x, u_m, 'CN'
fim