Preparação do TRABALHO 7

Condução de Calor numa barra de cobre (1D)

(Nota : A equação é uma equação às derivadas parciais (PARABÓLICA). Os métodos de integração numérica, propostos neste trabalho, baseiam-se em métodos já estudados para ODE, nomedamente, no método de EULER, no método de CRANK-NICOLSON e no método das DIFERENÇAS FINITAS)

$$\boxed{\frac{\partial T(x,t)}{\partial t} = \frac{k}{c\rho}\frac{\partial^2 T(x,t)}{\partial x^2}}$$

T – temperatura, x – posição , t- tempo, k- conditividade térmica do material

c- calor específico, ρ- massa volúmica , L – comprimento da barra

x e t são duas variáveis independentes.

7.1) Resolução pelo método de EULER

▶ Discretização da equação no ponto (x,t)→(i,n)

$\boxed{\frac{\partial^2 T(x,t)}{\partial x^2}}$ → a 2ª derivada, em x, é aproximada por diferenças centradas

$$\frac{\partial^2 T(x,t)}{\partial x^2} \approx \frac{T(i-1,t) - 2T(i,t) + T(i+1,t)}{(\Delta x)^2}$$

$\boxed{\frac{\partial T(x,t)}{\partial t}}$ → a 1ª derivada, em t, é aproximada por diferenças avançadas

$$\frac{\partial T(x,t)}{\partial t} \approx \frac{T(i,n+1) - T(i,n)}{\Delta t}$$

Obtém-se:

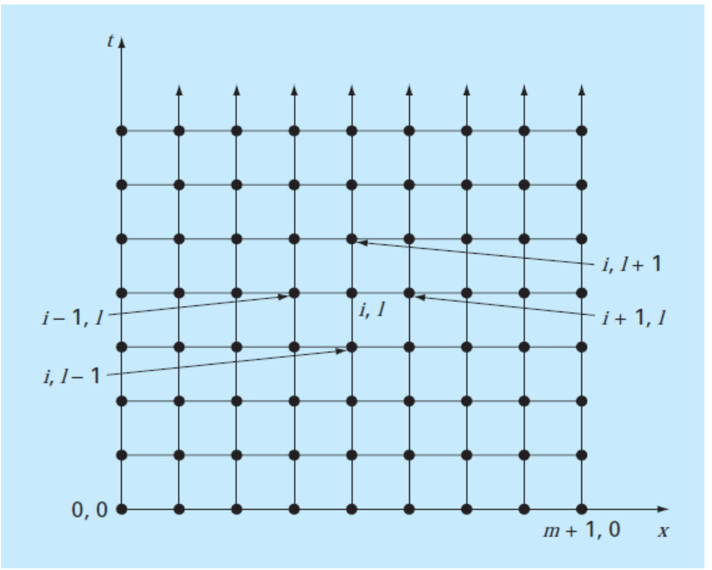
$$\boxed{T(i,n+1) \ = T(i,n) + \frac{k\,\Delta t}{c\rho(\Delta x)^2}\left[T(i-1,n) - 2T(i,n) + T(i+1,n)\right]}$$

O método de Euler é estável se e só se,

$$\frac{k}{c\rho(\Delta x)^2} \leq \frac{1}{2}$$

▶Sugestões para a construção do código

A equação vai ser resolvida sobre uma grelha discreta, **x=0: dx: L**, em que no eixo dos **xx** está representada a barra, com **Nx** pontos, (**Nx-2** pontos no seu interior), e no eixo dos tt, representam-se os diferentes instantes, **t= 0: dt: tf**. Repare que a solução em cada instante **n**, é dada por **T(1:Nx, n)**. Em todos os instantes a solução será dada por **T(1:Nx, 1:Nt)**.



Deve construir:

-vetor t, dt=0.1 s

-vetor x, dx =0.5 cm

- vetor de temperaturas T= zeros(Nx,Nt)

- Deve Aplicar as CI+ CF

T(2:Nx-1,1) = 100;
T(1,:) = 0; T(Nx,:) = 0;

Terá que fazer dois ciclos um para calcular a distribuição da temperatura ao longo da barra em cada instante, e outro para os vários instantes.

Para representar os resultados, use as rotinas `mesh` e `contourf` (faça help **`mesh e contourf`**)

Questão: Pode usar-se o valor das grandezas em unidades do sistema CGS ? A equação é válida ???

7.2) Resolução pelo método de Crank-Nicolson (+ estável)

► Discretização da equação no ponto (x,t)→(i,n)

Recorde o algoritmo de CN, (aula 2 , pág12) para a solução numérica da equação $\frac{dy}{dt} = f(t, y)$:

$$y_{k+1} = y_k + \left[f(t_{k,} y_k) + f(t_{k+1,} y_{k+1}) \right] * \frac{h}{2}$$

Se se considerar a função f(t,x) como $\frac{k}{c\rho}\frac{\partial^2 T(x,t)}{\partial x^2}$, a 2ª derivada pode escrever-se como:

$$\frac{\partial^2 T(x,t)}{\partial x^2} \approx \frac{1}{2}\left[\frac{T(i-1,n) - 2T(i,n) + T(i+1,n)}{(\Delta x)^2} + \frac{T(i-1,n+1) - 2T(i,n+1) + T(i+1,n+1)}{(\Delta x)^2}\right]$$

E considerando-se diferenças avançadas para a derivada temporal, obtém-se:

$$\boxed{\begin{aligned} T(i,n+1) &= T(i,n) \\ &+ \frac{k\,\Delta t}{2c\rho(\Delta x)^2}\left[T(i-1,n) - 2T(i,n) + T(i+1,n) + T(i-1,n+1) - 2T(i,n+1) + T(i+1,n+1)\right] \end{aligned}}$$

Fazendo um rearranjo de termos, de forma a que os termos em T(-, *n+1*) fiquem agrupados no 1º membro e os termos em T(-, *n*) fiquem agrupados no 2º membro, obtém-se,

$$\boxed{\begin{aligned} -T(i-1,n+1) + \left(\frac{2}{\eta}+2\right)T(i,n+1) - T(i+1,n+1) \\ = T(i-1,n) + \left(\frac{2}{\eta}-2\right)T(i,n) + T(i+1,n) \end{aligned}}$$

Com,

$$\eta = \frac{k}{c\rho(\Delta x)^2}$$

Como T(1,n+1) e T(Nx,n+1), são conhecidas (Condições Fronteira), a temperatura em cada extremidade da barra, a equação anterior pode escrever-se na forma matricial. Para tal escreva a equação anterior para i=2, i=3,…i=Nx-1. (i=1 e i=Nx as temperaturas são conhecidas.

Vai obter o seguinte sistema, na forma matricial

$(A\boldsymbol{Z} = \boldsymbol{b})$

$$A\boldsymbol{T}_{\boldsymbol{n+1}} = \boldsymbol{T}_{\boldsymbol{n}}$$

Tn -temperaturas ao longo da barra no instante n, conhecidas da iteração anterior

Tn+1- temperaturas ao longo da barra no instante n+1, a determinar na presente iteração

$$
\begin{bmatrix}
\left(\frac{2}{\eta}+2\right) & -1 & & & & \\
-1 & \left(\frac{2}{\eta}+2\right) & -1 & & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & -1 & \left(\frac{2}{\eta}+2\right) & -1 \\
 & & & & -1 & \left(\frac{2}{\eta}+2\right)
\end{bmatrix}
\begin{bmatrix}
T(2,n+1) \\
T(3,n+1) \\
\vdots \\
T(Nx-2,n+1) \\
T(Nx-1,n+1)
\end{bmatrix}
$$

$$
= \begin{bmatrix}
\color{red}{\boldsymbol{T(1,n+1)}} + T(1,n) + \left(\frac{2}{\eta}-2\right)T(2,n) + T(3,n) \\
T(2,n) + \left(\frac{2}{\eta}-2\right)T(3,n) + T(4,n) \\
\vdots \\
T(Nx-3,n) + \left(\frac{2}{\eta}-2\right)T(Nx-2,n) + T(Nx-1,n) \\
T(Nx-2,n) + \left(\frac{2}{\eta}-2\right)T(Nx-1,n) + T(Nx,n) + \color{red}{\boldsymbol{T(Nx,n+1)}}
\end{bmatrix}
$$

Os valores a vermelho são conhecidos! Por isso figuram no vetor independente e não nas incógnitas a determinar.

►Sugestões para a construção do código

►Pode verificar-se que a matriz é diagonal.

►O método consiste em resolver um sistema de equações algébricas cuja matriz é tridiagonal. Este processo é repetido para cada iteração no tempo.

a) Como escrever a matriz?

Como A é tridiagonal, pode escrever-se como a soma de três matrizes, uma diagonal, outra diagonal inferior e outra diagonal superior.

%Escrita da Matriz

NM=Nx-2 ; nº de pontos da Matriz

$\eta = \cdots$

D1=$\left(\frac{2}{\eta} + 2\right)$;

A= eye (NM);

A=D1*A; % matriz diagonal, diagonal principal

A(1,2)=-1; % 2º elemento da 1ª linha

```
    for i=2:NM-1
        A(i,i-1)=-1;  % diagonal superior
        A(i,i+1)=-1;  % diagonal inferior
     End
```

A (NM,NM-1)=-1; % penúltimo elemento da última linha

%---

%Escrita do vetor b   (b= Tn)

b=zeros(Nx-2,1);

D2=$\left(\frac{2}{\eta} - 2\right)$;

%-------a)---

%ciclo   para se obter a temperatura ao longo da barra em cada instante, e para nt instantes,

```
for n=1:Nt-1
    for i=1:Nx-2
        b = T(i,n)+D2*T(i+1,n)+T(i+2,n);
     end

      b(1)=b(1)+T(1,n+1); % é preciso adicionar a CF
      b(NM)=b(NM)+T(Nx,n+1); % CF

          T(2:Nx-1,n+1)=linsolve(A,b);

end
```