

### TEMA 3

## EQUAÇÕES DIFERENCIAIS ORDINÁRIAS

Os próximos temas irão abordar a resolução de **equações diferenciais (ED's)**. Como o nome indica, uma equação diferencial é uma equação que envolva uma (ou mais) derivadas de uma função desconhecida. Uma grande parte dos problemas em física envolve este tipo de equações: o movimento de fluídos, circulação da corrente eléctrica, propagação de ondas, fluxos de calor. No entanto as equações diferenciais são fundamentais para compreender a dinâmica de muitos outros sistemas, desde a biologia à economia.

### Equações Diferenciais Ordinárias (EDO's)

Uma equação diferencial designa-se *ordinária* sempre que inclua apenas derivadas em ordem a uma *única variável independente*, como é o caso dos exemplos apresentados a seguir. A ordem de uma EDO é dada pela maior ordem das derivadas presentes.

As EDO's são usadas numa grande variedade de problemas físicos. Alguns exemplos incluem:

#### Termodinâmica

Lei de Newton do Arrefecimento (arrefecimento de um corpo à temperatura  $T$ , num ambiente a  $T_a$ , com um coeficiente de transferência  $\alpha$ ) – EDO 1ª ordem:

$$\frac{dT}{dt} = -\alpha(T - T_a)$$

#### Mecânica

Deslocamento vertical de um corpo de massa  $m$ , com velocidade inicial  $v_0$ , num campo gravítico com aceleração  $g$  – EDO 2ª ordem:

$$\frac{d^2z}{dt^2} = -g$$

Esta EDO de 2ª ordem pode ser decomposta num sistema de EDO's de 1ª ordem:

$$\frac{dv}{dt} = -g$$

$$\frac{dz}{dt} = v$$

Dinâmica de populações

Variação do número de elementos de uma determinada população ( $N$ ), com taxa de crescimento  $r$ , inseridos num ecossistema com capacidade limite  $B$  – EDO 1ª ordem:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{B}\right)$$

**Resolução de ED's**

A resolução analítica de equações diferenciais, abordada em cursos de Cálculo Diferencial, permite determinar uma função a partir da sua equação diferencial, como por exemplo no caso da equação geral do movimento de um corpo que relaciona a velocidade ( $v$ ) com a aceleração ( $a$ ):

$$\frac{dv}{dt} = -g$$

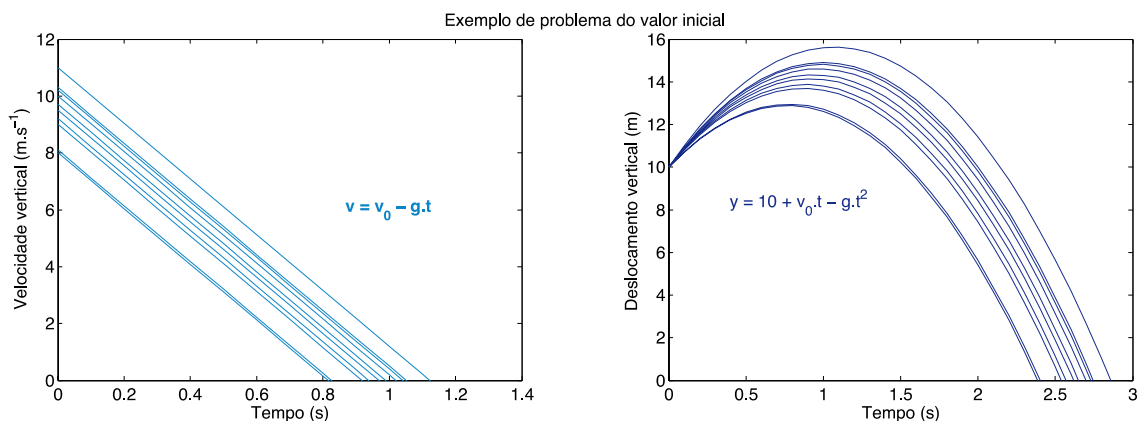
Integrando:

$$\int_{v_0}^{v'} v = \int_{t_0}^{t'} -g \cdot dt$$

Então:

$$v = v_0 - g(t - t_0)$$

Tal como o exemplo anterior indica, a solução da equação diferencial anterior admite uma infinidade de soluções, i.e.  $v_0$  pode tomar qualquer valor. A figura abaixo exemplifica várias soluções possíveis para a equação anterior, bem como as suas implicações na trajetória de um corpo, lançado à altura de 10m:



Apesar de todas as rectas no gráfico da esquerda corresponderem a soluções possíveis da equação, é fundamental definir uma condição adicional (que pode ser o valor inicial ou outro) que permita identificar a solução para um problema em particular – **Problema de Cauchy**. Este problema pode ser enunciado do seguinte modo:

Pretende-se determinar  $y(t)$ , com  $t$  pertencente ao intervalo de integração, que satisfaça as seguintes condições:

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y(t=0) &= y_0 \end{aligned}$$

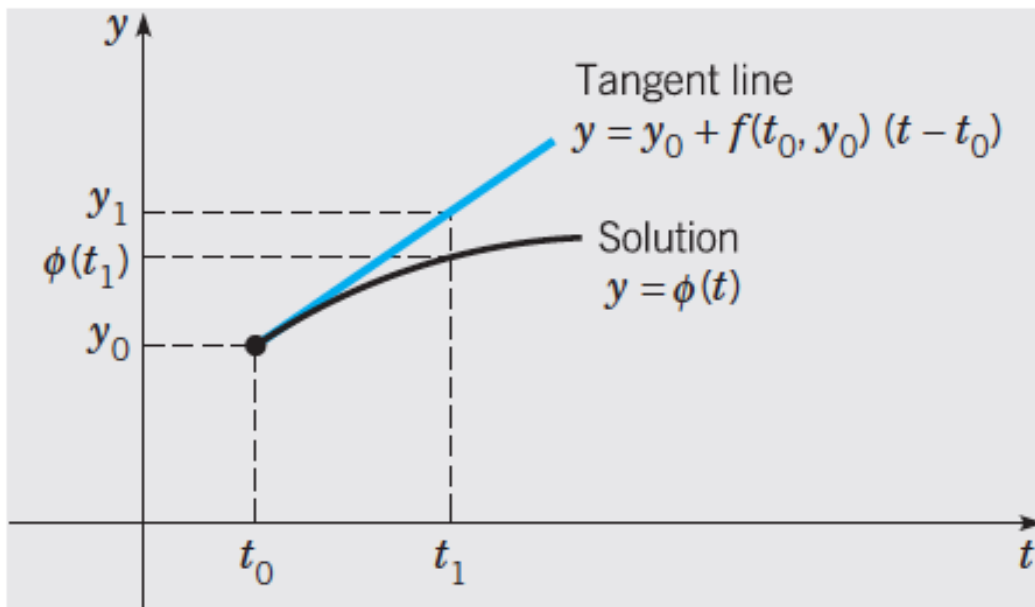
Este problema é abordado em mais detalhe no Tema 4.

### Métodos numéricos – diferenças finitas

Em muitos problemas físicos, em particular problemas não lineares, os métodos analíticos nem sempre permitem resolver ED's. É, por isso, particularmente relevante o estudo de métodos numéricos que permitam resolver problemas diferenciais.

A resolução numérica de equações diferenciais do tipo  $y'(t) = f(t, y(t))$ , com  $y(t_0) = y_0$ , pode ser feita através do cálculo de valores aproximados de  $y$  num determinado conjunto (discreto) de valores de  $t$ .

O valor de  $y'(t_0)$  corresponde ao declive da recta tangente à função no ponto  $t_0$ . Pode-se então assumir que, para um  $t_1$  suficientemente próximo de  $t_0$ , o valor da solução  $y(t)$  se encontra dentro de um intervalo de erro definido:



Retirado de Boyce & Di-Prima, *Elementary Differential Equations and Boundary Value Problems*, 10<sup>th</sup> ed, 2012, Wiley.

A transformação de uma ED de um domínio contínuo para um domínio discreto, e a sua transformação numa equação algébrica implica a sua *discretização*.

É necessário definir inicialmente o intervalo de discretização  $\Delta t$  adequado bem como a malha de  $N$  pontos que define o domínio discreto de integração:

$$t = t_0 + k\Delta t, k = \{1, 2, \dots, N\}$$

Definido o domínio, pode-se substituir as derivadas pela sua aproximação algébrica. O método mais simples consiste na aproximação por *diferenças finitas*:

$$\left(\frac{dy}{dt}\right)_{t=a} = \frac{y(a + \Delta t) - y(a)}{\Delta t}$$

Neste caso representam-se as diferenças finitas *avançadas*. Notar que esta expressão corresponde ao declive da recta tangente ao ponto  $y(a)$ , tal como acima descrito.

A escolha de um determinado método numérico exige algumas considerações prévias:

- Precisão – relacionada com o erro de aproximação
- Estabilidade – o erro é estável?
- Convergência – a solução numérica converge para a solução analítica quando  $\Delta t \rightarrow 0$ ?

De forma genérica, podem ainda definir-se três tipos de diferenças finitas:

#### Avançadas

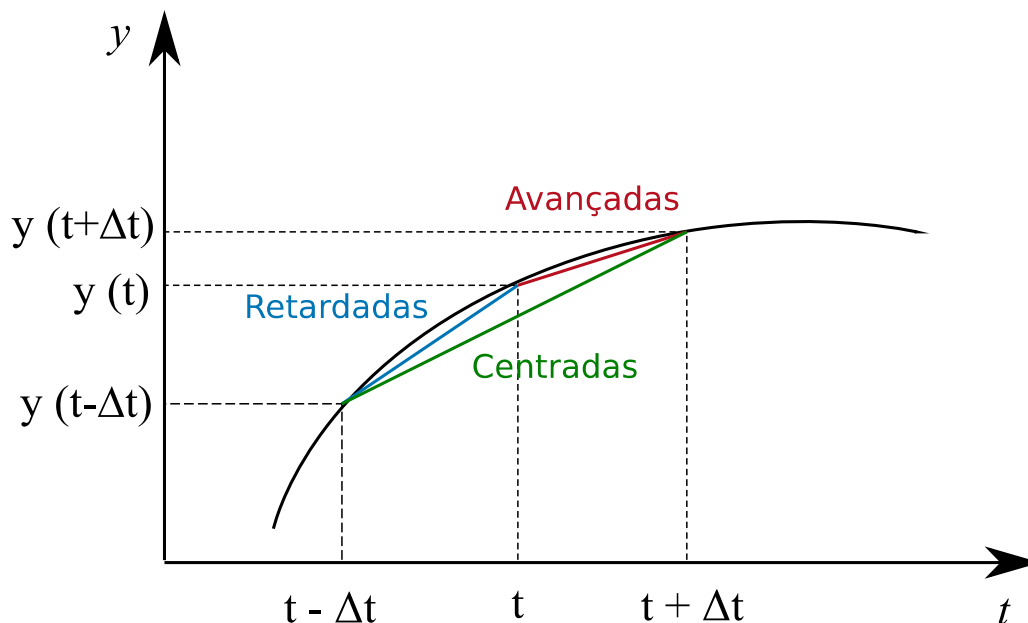
$$\frac{dy}{dt} \approx \frac{y(t + \Delta t) - y(t)}{\Delta t}$$

#### Retardadas

$$\frac{dy}{dt} \approx \frac{y(t) - y(t - \Delta t)}{\Delta t}$$

#### Centradas

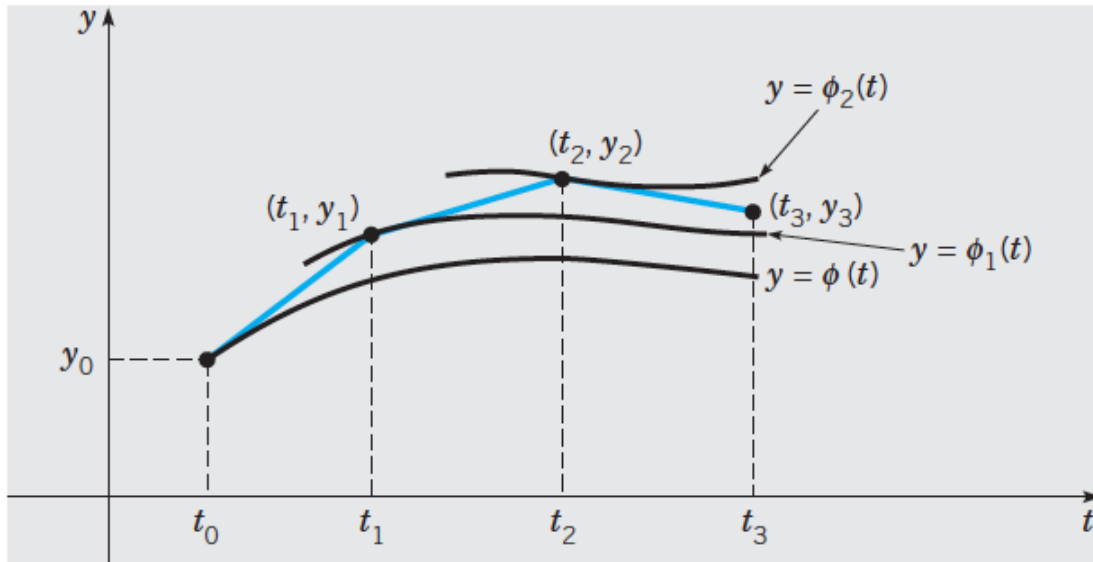
$$\frac{dy}{dt} \approx \frac{y(t + \Delta t) - y(t - \Delta t)}{2\Delta t}$$



## Método de Euler

O Método de Euler é o método mais antigo e simples para resolver EDO's, tendo sido desenvolvido por Euler por volta de 1768. É também muitas vezes chamado de método da recta tangente.

Como o nome indica, consiste em aproximar em cada ponto, partindo de  $t_0$ , o valor da função  $y(t)$  através do ajuste sucessivo de rectas tangentes, cujo declive é dado por  $y'(t)$ , tal como exemplificado na figura:



Retirado de Boyce & Di-Prima, *Elementary Differential Equations and Boundary Value Problems*, 10<sup>th</sup> ed, 2012, Wiley.

Partindo do ponto inicial conhecido  $(t_0, y_0)$ , pode-se calcular  $(t_1, y_1)$  como o ponto localizado sobre a recta tangente ao ponto inicial, para o instante  $t_1$ . Conhecido o valor de  $y_1$ , pode-se traçar novamente uma recta tangente ao ponto que permitirá determinar  $(t_2, y_2)$ , e assim sucessivamente, i.e.:

$$y(t + \Delta t) = y(t) + y'(t, y(t)) \cdot \Delta t$$

ou, na forma discreta:

$$y_{k+1} = y_k + y'(t_k, y_k) \cdot h$$

Em que  $h$  corresponde ao intervalo entre os pontos da malha (neste caso,  $\Delta t$ ).

O Método de Euler pode ser implementado de forma genérica segundo os seguintes passos:

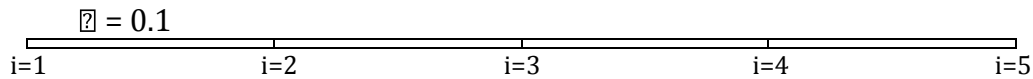
- 1 – Definir  $f(t, y)$
- 2 – Atribuir valores iniciais a  $t_0$  e  $y_0$
- 3 – Definir  $dt$  ( $\Delta t$ ) e número de pontos do domínio de integração  $N$
- 4 – Iterações  $i$  de 1 até  $N-1$ 

$k_1 = f(t_i, y_i)$	<i>derivada no ponto <math>i</math></i>
$y_{i+1} = y_i + dt \cdot k_1$	<i>cálculo de <math>y</math> no ponto seguinte</i>
$t_{i+1} = t_i + dt$	<i>cálculo de <math>t</math> no ponto seguinte</i>
- 5 – Output de  $t$  e  $y$

**Exercício 3.1**

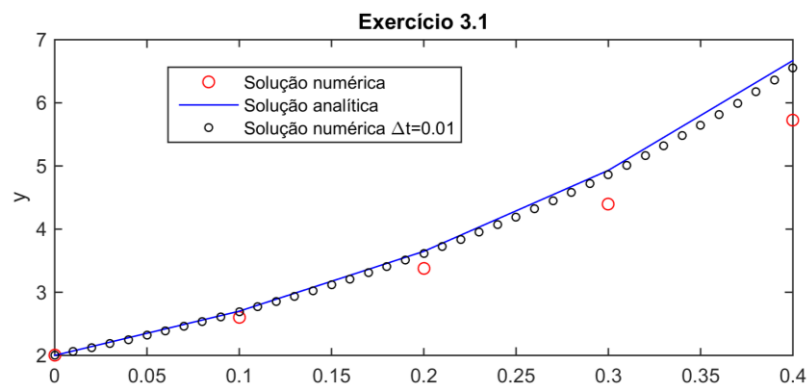
Considere a seguinte EDO:  $y'(t, y) = 3y + t^2$

Resolva numericamente a equação para uma malha de 5 pontos, com intervalo de 0.1 unidades, sabendo que  $t_0=0$  e  $y_0=2$ .



Resolução:

Iteração	t	$y_i$	$y'$	$y_{i+1}$
i = 1	0	2	6 (3x2+0)	2.6 (2+0.1x6)
i= 2	0.1	2.60	7.81	3.38
i= 3	0.2	3.38	10.18	4.47
i= 4	0.3	4.47	15.5	6.02
i= 5	0.4	6.02	23.22	9.34



Como é possível observar, para  $\Delta t=0.1$  a solução numérica diverge rapidamente em relação à solução real. Para  $\Delta t=0.01$  a solução numérica aproxima-se da solução real, indicando a existência de *convergência*. No entanto, pode-se ver que o erro aumenta consideravelmente ao fim de poucas iterações.

Exemplo de código em MATLAB para a resolução do Exercício 3.1:

```

dt=.1; % intervalo de tempo
N=5; % número de pontos da malha

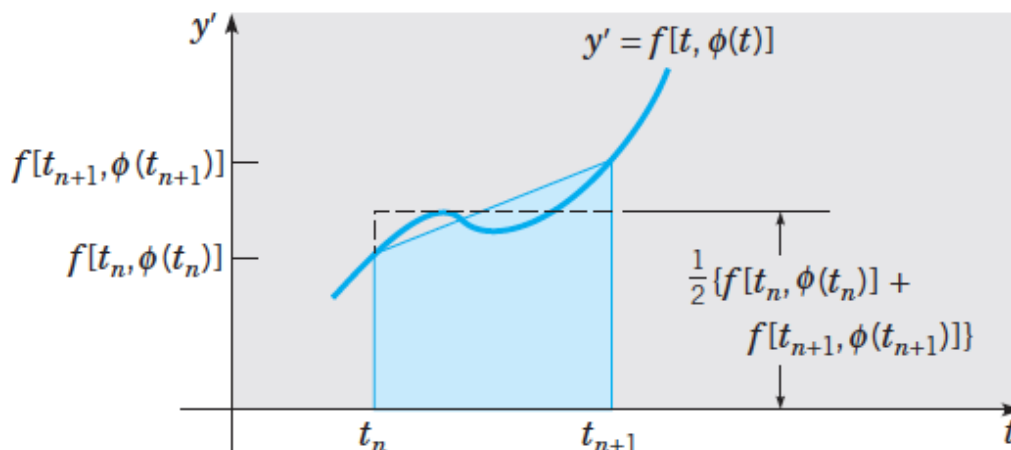
% inicialização dos vectores t e y
t=zeros(1,N);
y=zeros(1,N);

% condições iniciais t0 e y0
t(1)=0; y(1)=2;

% Método de Euler
for i=1:N-1
y(i+1)=y(i)+dt*(3*y(i)+t(i).^2); % para a EDO y'=3y+t^2
t(i+1)=t(i)+dt;
end
    
```

## Método de Crank-Nicolson

O método de Crank-Nicolson assemelha-se ao Método de Euler, no entanto utiliza-se um valor médio de  $y'(t, y)$ , calculado nos dois pontos de cada intervalo  $\Delta t$ , como representado na figura:



Retirado de Boyce & Di-Prima, *Elementary Differential Equations and Boundary Value Problems*, 10<sup>th</sup> ed, 2012, Wiley.

Genericamente:

$$y_{k+1} = y_k + \frac{h}{2} (y'(t_k, y_k) + y'(t_{k+1}, y_{k+1})) \quad , \quad k = 0, 1, \dots, N-1$$

ou

$$y_k = y_{k-1} + \frac{h}{2} (y'(t_{k-1}, y_{k-1}) + y'(t_k, y_k)) \quad , \quad k = 1, 2, \dots, N$$

No entanto,  $y_{k+1}$  (ou  $y_k$  no segundo exemplo) não é conhecido – é precisamente o que se pretende calcular.

Para o caso da EDO do Exercício 3.1 ( $y'(t, y) = 3y + t^2$ ) tem-se:

$$y_{k+1} = y_k + \frac{h}{2} (3y_k + t_k^2 + 3y_{k+1} + t_{k+1}^2)$$

Então:

$$y_{k+1} = \left[ y_k + \frac{h}{2} (3y_k + t_k^2 + t_{k+1}^2) \right] / \left( 1 - \frac{3h}{2} \right)$$

### Exercício 3.2

Resolva, para a mesma malha do Exercício 3.1, a EDO  $y'(t, y) = 3y + t^2$  usando o Método de Crank-Nicolson (C-N).

Exemplo de código em MATLAB para a resolução do Exercício 3.2:

```
% Continuação
% Método de Crank-Nicolson para a EDO y'=3y+t^2
y(1)=2; t(1)=0;
for i=1:N-1
    t(i+1)=t(i)+dt;
    y(i+1)=(y(i)+0.5*dt*(3*y(i)+t(i).^2 + t(i+1).^2))/(1-3*dt/2);
end
```

## Método de Runge-Kutta

Os vários métodos de Runge-Kutta permitem maior precisão, mas envolvem mais cálculos para cada iteração. O método evolui uma média ponderada de valores de  $y'(t, y)$  em diferentes pontos do intervalo  $t_n \leq t \leq t_{n+1}$  e pode ser escrito genericamente como:

$$y_{k+1} = y_k + h \sum_i^M b_i K_i, \quad k \geq 0$$

$$K_i = f(t_k + c_i \cdot h, y_k + h \sum_j^M a_{ij} K_j)$$

$i=1, 2, \dots, N; h=\Delta t$

Em que M corresponde ao número de etapas do método, ou a sua *ordem*.

### Método Runge-Kutta de ordem 4 (RK-4)

A expressão geral para o RK-4 é:

$$y_{k+1} = y_k + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

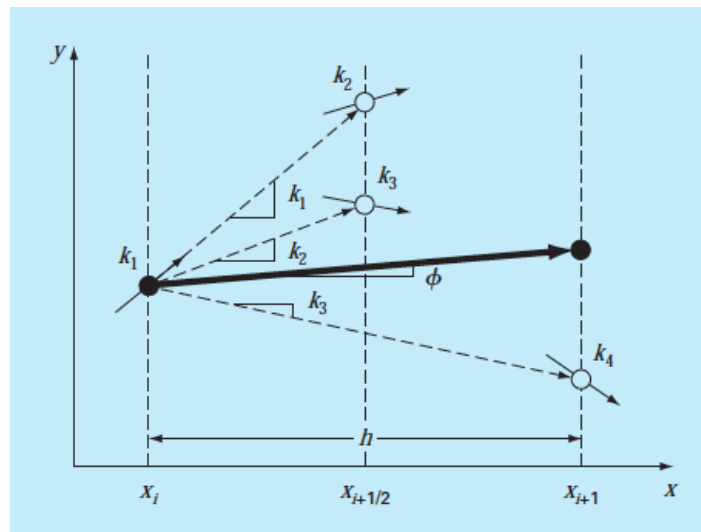
Em que

$$K_1 = f(t_k, y_k)$$

$$K_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} K_1\right)$$

$$K_3 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} K_2\right)$$

$$K_4 = f(t_{k+1}, y_k + hK_3)$$



Retirado de Chapra & Canale, *Numerical Methods for Engineers*, 6th ed, 2010, McGraw Hill

Como se pode ver na figura, o RK-4 requer o cálculo de valores de  $y'(t, y)$  em três pontos distintos,  $K_1$  corresponde à derivada no primeiro ponto do intervalo ( $t$ ), que é usada para determinar um ponto a meio do intervalo ( $t+1/2$ ) onde se avalia novamente o valor de  $y'$ , ou  $K_2$ . O valor de  $K_2$  é então usado para determinar um novo ponto a meio do intervalo, onde se calcula um outro valor possível para a derivada,  $K_3$ . Por fim,  $K_3$  permite determinar um quarto ponto em  $t+1$ , onde se avalia novamente o valor de  $y'$ .

Tem-se então 4 valores “possíveis” para  $y'$  no intervalo  $[t, t+1]$ :  $K_1, K_2, K_3$  e  $K_4$ . O valor da derivada a usar para aproximar a função no ponto  $t+1$ , é uma média ponderada dos quatro valores encontrados, sendo que se atribui mais peso aos dois valores intermédios, i.e.  $K_2$  e  $K_3$ .



A implementação genérica deste método é, assim, bastante semelhante ao método de Euler, aparte o cálculo intermédio dos termos  $K$ .

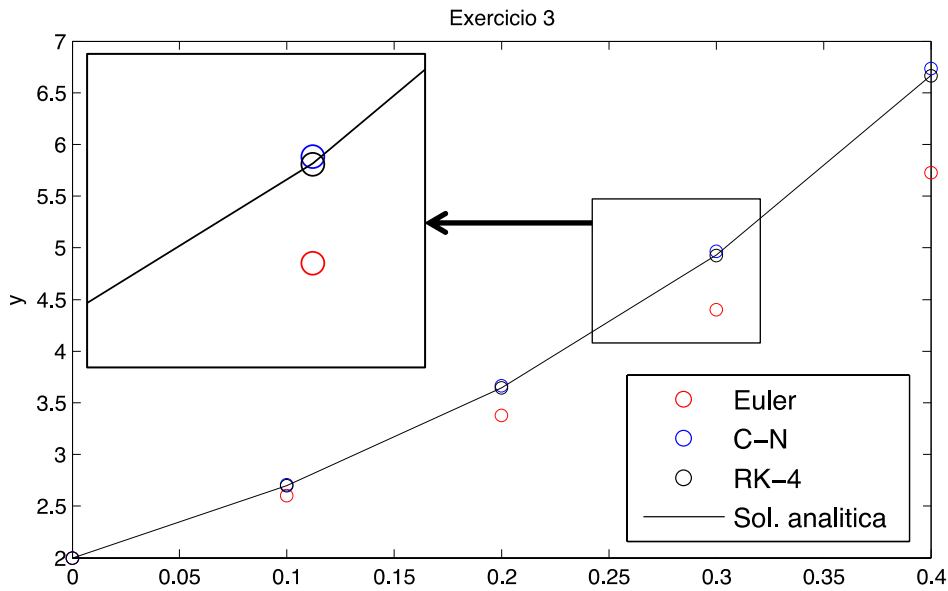
**Exercício 3.3**

Resolva, para a mesma malha do Exercício 3.1, a EDO  $y'(t, y) = 3y + t^2$  usando o Método de Runge-Kutta de ordem 4 (RK-4).

Exemplo de código em MATLAB para a resolução do Exercício 3.3:

```
%Continuação
% Método Runge-Kutta 4
y(1)=2; t(1)=0;
for i=1:N-1
    t1=t(i); % 1º ponto
    y1=y(i); % y no 1º ponto
    K1=3*y(i)+t(i)^2; % y' no 1º ponto
    t2=t(i)+dt/2; % ponto intermédio no intervalo t t+1
    y2=y(i)+0.5*dt*K1; % 1º valor de y no ponto intermédio
    K2=3*y2+t2^2; % y' avaliada em t2,y2
    y3=y2+0.5*dt*K2; % 2º valor de y no ponto intermédio
    K3=3*y3+t2^2; % y' avaliada em t2,y3
    t(i+1)=t(i)+dt; % último ponto do intervalo (t+1)
    y4=y(i)+dt*K3; % y em t+1
    K4=3*y4+t(i+1)^2; % y' avaliada em t+1
    y(i+1)=y(i)+(dt/6)*(K1+2*K2+2*K3+K4);
end
```

Comparação dos três métodos para o exemplo do Exercício 3:

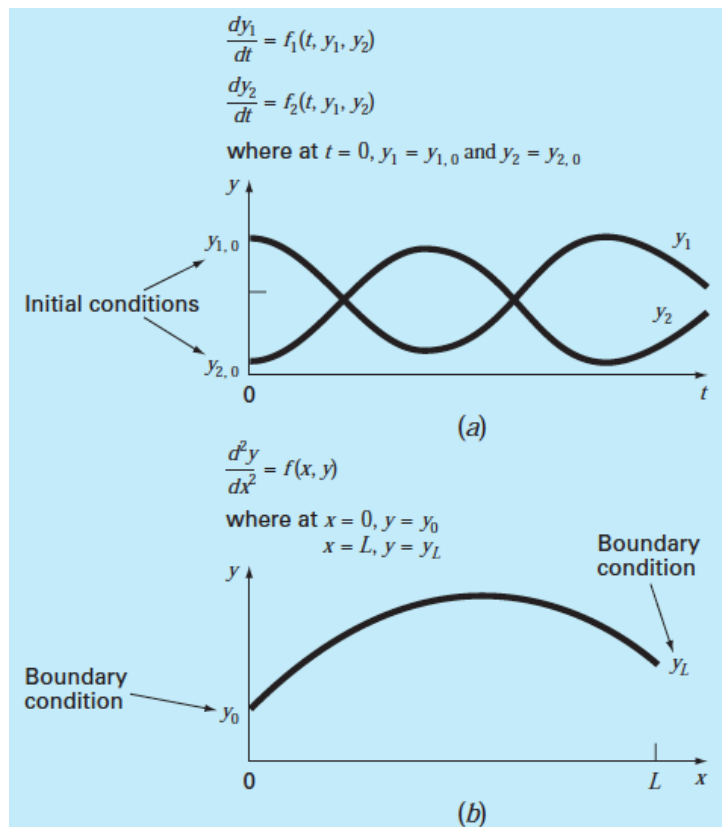


É possível observar a maior precisão dos Métodos C-N e RK-4.

## TEMA 4

### EQUAÇÕES DIFERENCIAIS ÀS DERIVADAS PARCIAIS: PROBLEMAS DE VALORES DE FRONTEIRA

No Tema anterior foi exemplificada a importância de definir corretamente uma condição adicional que permita resolver a EDO, no caso de uma EDO de 1ª ordem. Para EDO's de ordem  $n$ , torna-se necessário definir  $n$  condições adicionais. Nos temas anteriores foram apresentados problemas em que era conhecido o 1º valor do domínio, estes chamam-se *problemas de valor inicial*. No entanto, existem aplicações em que as condições não são conhecidas num único ponto, mas numa série de pontos do domínio. Usualmente (mas não obrigatoriamente) estas condições correspondem aos extremos (ou fronteiras) do domínio, pelo que estes problemas são apelidados de *problemas de valores de fronteira*. A diferença entre estes dois tipos de problema é exemplificada na figura seguinte.



Retirado de Chapra & Canale, *Numerical Methods for Engineers*, 6th ed, 2010, McGraw Hill

Podem ser definidos dois tipos de condição fronteira:

**Dirichlet:** fixam-se os valores de  $y(t_1)$ ,  $y(t_N)$  (ou de outros pontos no domínio)

**Neumann:** fixam-se os valores de  $y'(t_1)$ ,  $y'(t_N)$  (ou de outros pontos no domínio)

Isto é, no caso das condições fronteira de *Dirichlet*, os valores da função nos extremos são conhecidos, no caso das condições fronteira de *Neumann*, os valores das derivadas nos extremos são conhecidos.

No tema anterior foram apresentados diferentes métodos que permitem resolver problemas de valor inicial. Neste tema serão apresentados métodos para a resolução de problemas de valores de fronteira, utilizando as diferentes condições apresentadas anteriormente. Assim como nos métodos anteriores, serão utilizadas *diferenças finitas*, em que para cada ponto do domínio se aproxima a derivada por diferenças entre dois pontos. Na generalidade, a resolução destes problemas pode ser resumida nas seguintes etapas:

**1) Discretização do domínio de integração**

i.e. definir os  $N$  pontos da malha (1D, 2D, 3D, ...) onde se pretende resolver o problema;

**2) Discretização das equação diferencial**

i.e. substituir as derivadas pelo seu correspondente num domínio discreto e escrever a equação diferencial para cada ponto ( $y_i$ ) da malha;

**3) Construção do sistema de equações discretas**

o conjunto das equações para cada um dos  $N$  pontos pode ser organizado num sistema de  $N$  equações, e resumido em forma matricial, tal que:

$$A \cdot y = b$$

em que  $y$  é a solução.

**4) Definição das condições fronteira**

modificar o sistema de forma a incorporar as condições de fronteira (Dirichelet, se conhecido o valor de  $y$ , ou Neumann, se conhecido  $y'$ );

**5) Solução**

resolver o sistema de equações em 4).

**Discretização de derivadas**

Para uma dada Equação Diferencial, podem substituir-se os valores das derivadas pelo seu equivalente discreto.

1ª ordem (ou 1ª derivada): as aproximações discretas para as derivadas de primeira ordem podem ser avançadas, retardadas, ou centradas, como referido anteriormente.

$$\text{Avançadas: } \frac{dy}{dx} \approx \frac{y(x+\Delta x)-y(x)}{\Delta x} = \frac{y_{i+1}-y_i}{\Delta x}$$

$$\text{Retardadas: } \frac{dy}{dt} \approx \frac{y(t)-y(t-\Delta t)}{\Delta x} = \frac{y_i-y_{i-1}}{\Delta x}$$

$$\text{Centradas: } \frac{dy}{dx} \approx \frac{y(x+\Delta x)-y(x-\Delta x)}{2\Delta x} = \frac{y_{i+1}-y_{i-1}}{2\Delta x}$$

A precisão destas aproximações pode ser obtida utilizando uma série de Taylor. Para as diferenças avançadas:

$$y(x+\Delta x) = y(x) + \Delta x \left. \frac{\partial y}{\partial x} \right|_x + \frac{\Delta x^2}{2} \left. \frac{\partial^2 y}{\partial x^2} \right|_x + \frac{\Delta x^3}{3!} \left. \frac{\partial^3 y}{\partial x^3} \right|_x + \mathcal{O}((\Delta x)^4),$$

então,

$$\left. \frac{dy}{dx} \right|_x = \frac{y(x + \Delta x) - y(x)}{\Delta x} - \frac{\Delta x}{2} \left. \frac{\partial^2 y}{\partial x^2} \right|_x - \frac{\Delta x^2}{6} \left. \frac{\partial^3 y}{\partial x^3} \right|_x + \mathcal{O}((\Delta x)^3)$$

o que significa que, para  $\Delta x$  pequeno, as diferenças avançadas têm um erro de truncatura de primeira ordem  $\mathcal{O}(\Delta x)$ . O mesmo se passa para diferenças atrasadas:

$$y(x - \Delta x) = y(x) - \Delta x \left. \frac{\partial y}{\partial x} \right|_x + \frac{\Delta x^2}{2} \left. \frac{\partial^2 y}{\partial x^2} \right|_x - \frac{\Delta x^3}{3!} \left. \frac{\partial^3 y}{\partial x^3} \right|_x + \mathcal{O}((\Delta x)^3),$$

então,

$$\left. \frac{dy}{dx} \right|_x = \frac{y(x) - y(x - \Delta x)}{\Delta x} + \frac{\Delta x}{2} \left. \frac{\partial^2 y}{\partial x^2} \right|_x - \frac{\Delta x^2}{6} \left. \frac{\partial^3 y}{\partial x^3} \right|_x + \mathcal{O}((\Delta x)^3)$$

i.e. o erro de truncatura é também de primeira ordem  $\mathcal{O}(\Delta x)$ .

Comparando os erros de truncatura obtidos para as equações avançadas e retardadas, verifica-se que eles têm a mesma ordem, mas sinais opostos, pelo que fazendo uma média das duas aproximações obtém-se um erro de truncatura de segunda ordem:

$$\frac{\left. \frac{dy}{dx} \right|_x \text{ avançadas} + \left. \frac{dy}{dx} \right|_x \text{ atrasadas}}{2} = \frac{y(x + \Delta x) - y(x - \Delta x)}{2\Delta x} + \mathcal{O}((\Delta x)^2)$$

O erro de truncatura da aproximação por **diferenças centradas** é de segunda ordem, portanto esta aproximação deve ser utilizada, sempre que possível.

2ª ordem (ou 2ª derivada):

A aproximação para a segunda derivada é dada por:

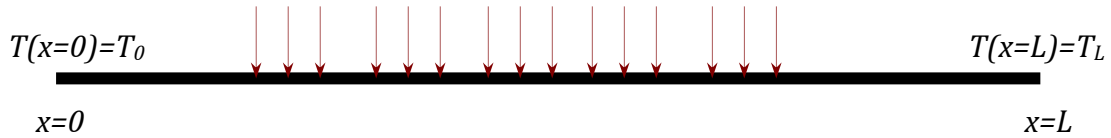
$$\left( \frac{d^2 y}{dx^2} \right)_i \approx \frac{y(x + \Delta x) - 2y(x) + y(x - \Delta x)}{(\Delta x)^2} = \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2}$$

Notar que esta expressão vem simplesmente da diferença (discreta) entre os valores das 1ª derivadas avançada e retardada para o elemento  $i$ .

Os problemas que envolvem *Equações Diferenciais às Derivadas Parciais*, muito comuns em problemas de física, são geralmente resolvidos de acordo com este método, pelo que serão analisadas em mais detalhe para dois casos (1D e 2D).

**PROBLEMA DE VALORES DE FRONTEIRA - 1D**

Consideremos um cabo circular de comprimento  $L$ , sujeito a determinadas condições fronteira nos extremos e a uma fonte de calor estacionária:



A distribuição de temperatura no cabo pode ser escrita como uma **equação de Poisson**:

$$\frac{\partial^2 T}{\partial x^2} = -f(x)$$

Em que  $f(x)$  corresponde à fonte de calor e as temperaturas nas extremidades são fixas (c.f. Dirichelet).

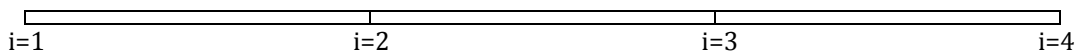
Exercício 4.1

Resolva o problema para uma malha com  $N=4$  elementos e com condições fronteira tais que:

$$T(x = 0) = 10^\circ\text{C}; \quad T(x = L) = 12^\circ\text{C}; \quad f(x) = 2K \cdot m^{-2}$$

Resolução do Exercício 4.1:

Para resolver o problema, discretiza-se o domínio em  $N$  elementos, neste caso, 4:



De seguida, discretiza-se a derivada presente na equação e escreve-se a sua expressão genérica:

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} = -f(x_i)$$

ou:

$$T_{i+1} - 2T_i + T_{i-1} = -f(x_i) \cdot \Delta x^2$$

ou ainda:

$$a_1 \cdot T_{i+1} + a_2 \cdot T_i + a_3 \cdot T_{i-1} = -f(x_i) \cdot \Delta x^2$$

Em que  $a_1=1$ ,  $a_2=-2$  e  $a_3=1$  são os **coeficientes da discretização**.

Pode-se então escrever o sistema de equações para o conjunto dos  $N=4$  pontos:

$$\begin{aligned} T_2 - 2T_1 &= -f(x_1) \cdot \Delta x^2 \\ T_3 - 2T_2 + T_1 &= -f(x_2) \cdot \Delta x^2 \\ T_4 - 2T_3 + T_2 &= -f(x_3) \cdot \Delta x^2 \\ -2T_4 + T_3 &= -f(x_4) \cdot \Delta x^2 \end{aligned}$$

Este sistema pode ser escrito na forma matricial como:

$$\begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = -D\mathbf{x}^2 \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix}$$

Finalmente, podem definir-se as condições fronteira, escrevendo as equações para os pontos das extremidades:

$$\begin{aligned} T_1 &= T_0 = 10 \\ T_4 &= T_L = 12 \end{aligned}$$

E modificando o sistema de equações de acordo com as mesmas:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 10 \\ -D\mathbf{x}^2 \cdot f(x_2) \\ -D\mathbf{x}^2 \cdot f(x_3) \\ 12 \end{bmatrix}$$

Pode-se então determinar  $T$  resolvendo o sistema matricial.

Exemplo de código em MATLAB para resolver o Exercício 4.1:

```

N=4; % n° pontos da malha
dx=1; % intervalo dx
x=0:dx:(N-1)/dx; % domínio xx

A=zeros(N,N); % inicialização da matriz A
b=zeros(N,1); % inicialização do vector b

M(1,1)=1; b(1)=10; % condições fronteira x=0
M(N,N)=1; b(N)=12; % condições fronteira x=L
fx=2; % valor de f K/m2

for i=2:N-1
    % coeficientes da discretização
    a1=1; a2=-2; a3=1;

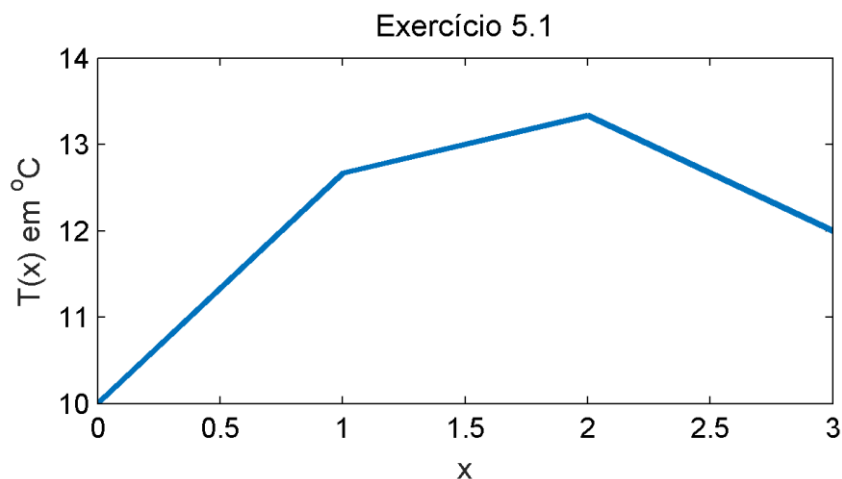
    % definição do interior da matriz A
    M(i,i-1)=a1; % elemento à esquerda da diagonal
    M(i,i)=a2; % diagonal
    M(i,i+1)=a3; % elemento à direita da diagonal

    % definição do interior do vector b
    b(i)=-fx*dx^2;
end

T=M\b; % resolução do sistema de equações

% apresentação do resultado
figure
plot(x,T)
title('Exercício 5.1')
xlabel('x'); ylabel('T(x)')
  
```

A distribuição de temperaturas no cabo circular pode finalmente ser apresentada:



**PROBLEMA DE VALORES DE FRONTEIRA – 2D**

A equação de Poisson também é válida para problemas a duas dimensões espaciais, contendo derivadas parciais de segunda ordem, em ordem a  $x'$  e a  $y'$ :

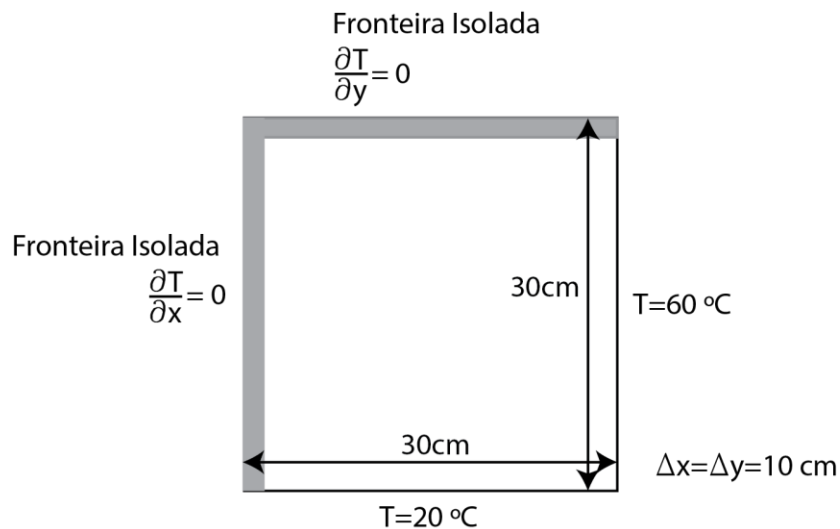
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f(x, y)$$

onde  $f(x,y)$  é uma função que representa fontes ou sumidouros de calor.

De seguida será apresentado um problema para ilustrar a resolução deste tipo de equações.

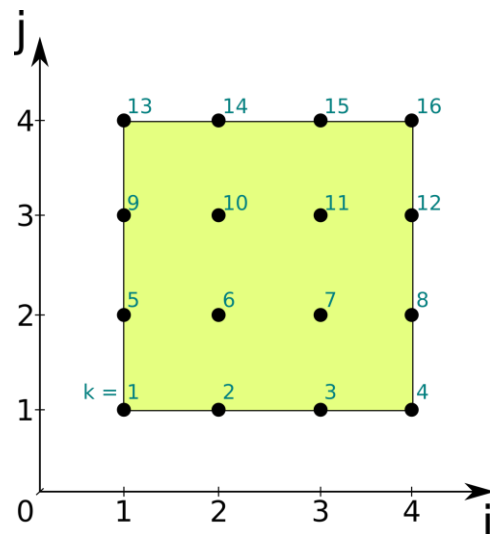
Exercício 4.2

Calcule a distribuição de temperatura numa placa, considerando a informação da seguinte figura:



Resolução do Exercício 4.2:

Para resolver o problema, discretiza-se o domínio em  $N$  pontos da malha, neste caso, 16. Numeram-se os pontos da malha sequencialmente, como representado na figura seguinte, de forma a encontrar uma equação para cada ponto, tal como se procedeu para as equações a uma dimensão.





Neste caso  $f(x_i, y_j) = 0$  em todo o domínio, com a exceção das fronteiras inferior e lateral direita, onde a temperatura tem um valor fixo.

De seguida, discretizam-se as duas derivadas para os pontos da malha  $x(i)$  e  $y(j)$ ,

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial^2 T}{\partial y^2} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2}$$

Como  $T$  tem duas dimensões, consideram-se os índices  $i$  e  $j$ , respectivamente para as direções  $xx'$  e  $yy'$ . Substituindo as aproximações das derivadas na equação de Poisson obtém-se:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} = f(i, j)$$

como neste caso  $\Delta x = \Delta y$ , obtém-se:

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = f(x_{i,j})\Delta x^2$$

não é obrigatório que  $\Delta x$  seja igual a  $\Delta y$  e nesse caso não se podia fazer esta simplificação. Pode-se reescrever a equação geral para este problema:

$$RT_{i+1,j} + LT_{i-1,j} + UT_{i,j+1} + DT_{i,j-1} + CT_{i,j} = f(x_{i,j})\Delta x^2$$

em que  $R(right)=1$ ,  $L(left)=1$ ,  $U(up)=1$ ,  $D(down)=1$  e  $C(center)=-4$  são os coeficientes da discretização.

Temos portanto dois referenciais para identificar um determinado ponto da malha, a sua posição  $(i, j)$  ou o índice absoluto  $(k)$ .

Neste caso é necessário considerar separadamente os pontos do interior da placa e dos diferentes lados, uma vez que têm condições de fronteira diferentes. Os pontos correspondentes são:

Interior:  $k=6, 7, 10, 11$ ;  
 Fronteira esquerda:  $k= 5, 9$ ;  
 Fronteira direita:  $k=8, 12$ ;  
 Fronteira de baixo:  $k=2, 3$ ;  
 Fronteira de cima:  $k=14, 15$ ;  
 Cantos da placa:  $k=1, 4, 13, 16$ .

No **interior**, neste exercício, não existe nenhuma fonte nem sumidouro, pelo que a equação para os pontos do interior se resume a:

$$RT_{i+1,j} + LT_{i-1,j} + UT_{i,j+1} + DT_{i,j-1} + CT_{i,j} = 0$$

Na **fronteira esquerda** temos uma condição de fronteira de Neumann, onde  $\frac{\partial T}{\partial x} = 0$ , que representa uma fronteira isolada. Para aplicar esta condição de fronteira considera-se um ponto imaginário, situado fora do domínio, em  $(i-1)$ , e admite-se que este ponto tem o mesmo valor que  $T_{i+1,j}$ , dado que:

$$\frac{\partial T}{\partial x} = \frac{T_{i+1,j} - T_{i-1,j}}{\Delta x} = 0$$

então,

$$T_{i-1,j} = T_{i+1,j}$$

substituindo na equação geral, obtém-se a equação para os pontos da fronteira esquerda:

$$(R + L)T_{i+1,j} + UT_{i,j+1} + DT_{i,j-1} + CT_{i,j} = 0$$

Na **fronteira superior** temos algo semelhante, mas aqui com  $\frac{\partial T}{\partial y} = 0$ . Para aplicar esta condição de fronteira considera-se um ponto imaginário, situado fora do domínio, em  $(j+1)$ , e admite-se que este ponto tem o mesmo valor que  $T_{i,j-1}$ .

$$\frac{\partial T}{\partial y} = \frac{T_{i,j+1} - T_{i,j-1}}{\Delta y} = 0$$

então,

$$T_{i,j+1} = T_{i,j-1}$$

substituindo na equação geral, obtém-se a equação para os pontos da fronteira esquerda:

$$RT_{i+1,j} + LT_{i-1,j} + (U + D)T_{i,j-1} + CT_{i,j} = 0$$

Na **fronteira inferior** temos uma condição de fronteira de Dirichlet, onde  $T_{i,j}=20$ , portanto a equação para os pontos da fronteira inferior resume-se a:

$$T_{i,j} = 20$$

Da mesma forma, na **fronteira direita**, temos:

$$T_{i,j} = 60$$

Restam as equações para os 4 cantos da placa. Os cantos são pontos de descontinuidade e devem ser tratados como isolados nas duas direcções, ou seja, com condições de fronteira de Neumann, em  $i$  e  $j$ .

**Canto inferior esquerdo:** consideramos que é isolado nos dois sentidos, ou seja,

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial y} = 0$$

De onde se obtém que  $T_{i-1,j} = T_{i+1,j}$  e  $T_{i,j-1} = T_{i,j+1}$ . Substituindo na equação geral obtém-se a equação para este canto:

$$(R + L)T_{i+1,j} + (U + D)T_{i,j+1} + CT_{i,j} = 0$$

**Canto superior esquerdo:** consideramos que é isolado nos dois sentidos, ou seja,

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial y} = 0$$

De onde se obtém que  $T_{i-1,j} = T_{i+1,j}$  e  $T_{i,j+1} = T_{i,j-1}$ .

Substituindo na equação geral obtém-se a equação para este canto:

$$(R + L)T_{i+1,j} + (U + D)T_{i,j-1} + CT_{i,j} = 0$$

**Canto superior direito:** consideramos que é isolado nos dois sentidos, ou seja,

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial y} = 0$$

De onde se obtém que  $T_{i+1,j} = T_{i-1,j}$  e  $T_{i,j+1} = T_{i,j-1}$ .

Substituindo na equação geral obtém-se a equação para este canto:

$$(R + L)T_{i-1,j} + (U + D)T_{i,j-1} + CT_{i,j} = 0$$

**Canto inferior direito:** consideramos que é isolado nos dois sentidos, ou seja,

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial y} = 0$$

De onde se obtém que  $T_{i+1,j} = T_{i-1,j}$  e  $T_{i,j-1} = T_{i,j+1}$ .

Substituindo na equação geral obtém-se a equação para este canto:

$$(R + L)T_{i-1,j} + (U + D)T_{i,j+1} + CT_{i,j} = 0$$

É preciso então considerar as interações entre os vários pontos da malha. Por exemplo o ponto do interior  $k=6$ , interage com os pontos à sua esquerda  $(i-1, j)$  ou  $(k=5)$ , direita  $(i+1, j)$  ou  $k=7$ , acima  $(i, j+1)$  ou  $k=10$ , e abaixo  $(i, j-1)$  ou  $k=2$ .

O sistema de equações, em que cada linha corresponde a um dos pontos da malha, fica então:

$$\left\{ \begin{array}{l} CT_1 + (R + L)T_2 + (U + D)T_5 = 0 \\ T_2 = 20 \\ T_3 = 20 \\ (R + L)T_3 + CT_4 + (U + D)T_8 = 0 \\ DT_1 + CT_5 + (R + L)T_6 + UT_9 = 0 \\ DT_2 + LT_5 + CT_6 + RT_7 + UT_{10} = 0 \\ DT_3 + LT_6 + CT_7 + RT_8 + UT_{11} = 0 \\ T_8 = 60 \\ DT_5 + CT_9 + (R + L)T_{10} + UT_{13} = 0 \\ DT_6 + LT_9 + CT_{10} + RT_{11} + UT_{14} = 0 \\ DT_7 + LT_{10} + CT_{11} + RT_{12} + UT_{15} = 0 \\ T_{12} = 60 \\ (U + D)T_9 + CT_{13} + (R + L)T_{14} = 0 \\ (U + D)T_{10} + LT_{13} + CT_{14} + RT_{15} = 0 \\ (U + D)T_{11} + LT_{14} + CT_{15} + RT_{16} = 0 \\ (U + D)T_{12} + (R + L)T_{15} + CT_{16} = 0 \end{array} \right.$$

Ou, na forma matricial:

$$\begin{bmatrix} C & (R+L) & 0 & 0 & (U+D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (R+L) & C & 0 & 0 & 0 & (U+D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & C & (R+L) & 0 & 0 & U & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D & 0 & 0 & L & C & R & 0 & 0 & U & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D & 0 & 0 & L & C & R & 0 & 0 & U & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D & 0 & 0 & 0 & C & (R+L) & 0 & 0 & U & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D & 0 & 0 & L & C & R & 0 & 0 & U & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D & 0 & 0 & L & C & R & 0 & 0 & U & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (U+D) & 0 & 0 & 0 & C & (R+L) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (U+D) & 0 & 0 & L & C & R & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (U+D) & 0 & 0 & L & C & R \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (U+D) & 0 & 0 & (R+L) & C \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \\ T_{10} \\ T_{11} \\ T_{12} \\ T_{13} \\ T_{14} \\ T_{15} \\ T_{16} \end{bmatrix} = \begin{bmatrix} 0 \\ 20 \\ 20 \\ 0 \\ 0 \\ 0 \\ 0 \\ 60 \\ 0 \\ 0 \\ 0 \\ 60 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Pode-se então determinar  $T$  para os vários pontos da placa resolvendo o sistema matricial.

Código em MATLAB para resolver o Exercício 4.2:

```

%Dados do exercicio
Tr=60;Td=20;
Lx=30; Ly=30; dx=10; dy=dx;
nx=Lx/dx+1; ny=Ly/dy+1; N=nx*ny;

%Coeficientes da discretizacao
R=1; L=1; U=1; D=1; C=-4;

%Inicializacao da matriz A e do vector b
A=zeros(N,N); b=zeros(N,1);

%Preenchimento da matriz nos pontos do interior, linha a linha
for j=2:ny-1
    for i=2:nx-1
        k=(j-1)*nx+i; %numeracao geral, k
        %pontos que interagem com o ponto k: U(up), D(down), L(left), R(right)
        kU=k+nx; kD=k-nx; kL=k-1; kR=k+1;
        A(k,k)=-4;
        A(k,kU)=U; A(k,kD)=D; A(k,kL)=L;A(k,kR)=R;
    end
end

%Fronteira Esquerda (isolada)
for j=2:nx-1
    k=(j-1)*nx+1;
    A(k,k)=C;
    A(k,k-nx)=D; A(k,k+nx)=U; A(k,k+1)=R+L;
end

%Fronteira Superior (isolada)
for i=2:nx-1
    k=(ny-1)*nx+i;
    A(k,k)=C;
    A(k,k-1)=L; A(k,k+1)=R; A(k,k-nx)=U+D;
end

%Fronteira Inferior (T=Td)
for i=2:nx-1
    k=i;
    A(k,k)=1; b(k)=Td;
end

%Fronteira Direita (T=Tr)
for j=2:1:ny-1
    k=(j-1)*nx+nx;
    A(k,k)=1; b(k)=Tr;
end

%Canto Inferior Esquerdo
A(1,1)=C; A(1,2)=L+R; A(1,nx+1)=U+D;

%Canto Superior Esquerdo
k=(ny-1)*nx+1; A(k,k)=C; A(k,k+1)=L+R; A(k,k-nx)=U+D;

%Canto Superior Direito
A(N,N)=C; A(N,N-1)=L+R; A(N,N-nx)=U+D;

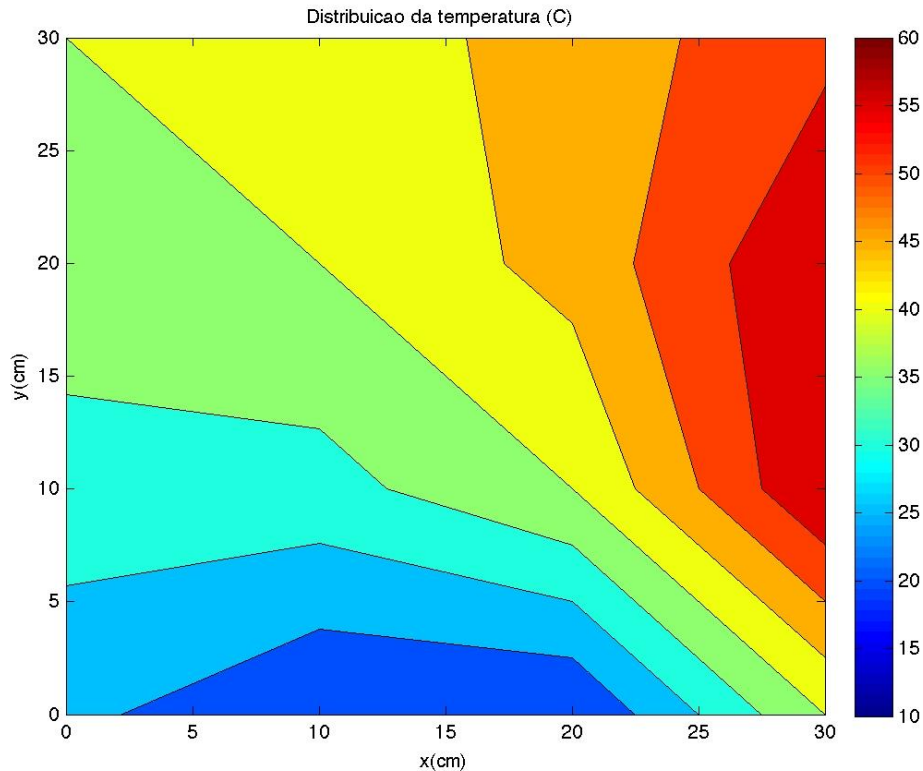
%Canto Inferior Direito
A(nx,nx)=C; A(nx,nx-1)=L+R; A(nx,nx+nx)=U+D;

%Resolucao do Sistema
T=A\b;

%Reorganizacao do vector T, numa matrix nx por ny
x=[0:dx:Lx]; y=[0:dy:Ly];
for j=1:ny
    for i=1:nx
        k=(j-1)*nx+i;
        Z(j,i)=T(k);
    end
end
[X,Y]=meshgrid(x,y);

%Representacao da Solucao
figure
[C,h]=contourf(X,Y,Z); colorbar; caxis([10 60])
title('Distribuicao da temperatura (C)'); xlabel('x(cm)'); ylabel('y(cm)')
    
```

De onde se obtém a distribuição da temperatura na placa:



## TEMA 5

### EQUAÇÕES DIFERENCIAIS ÀS DERIVADAS PARCIAIS: PROBLEMAS TRANSIENTES

No capítulo anterior foram descritos métodos para resolver problemas de valores de fronteira. Estas equações eram estacionárias no tempo. Neste capítulo vamos estudar equações transientes, ou seja, que evoluem ao longo do tempo. Estas equações são utilizadas, por exemplo, em problemas de difusão e advecção. São exemplos deste tipo de equações: problemas de transferência de calor, de difusão de massa num meio, de difusão molecular, entre outros.

#### Equação da Transferência de Calor

$$k \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t}$$

Estas equações também podem ser resolvidas numericamente aproximando as derivadas por diferenças finitas. No entanto, neste caso não se tem apenas derivadas parciais no espaço, mas também derivadas no tempo. Outra diferença é que, enquanto na resolução das equações elípticas se tem condições de fronteira bem definidas, neste caso, tem-se condições de fronteira no espaço e um problema de valor inicial, como no caso das equações diferenciais ordinárias. Existem duas formas diferentes de resolver estas equações, que se denominam de **método explícito**, e **método implícito**.

No método explícito calcula-se, explicitamente, o valor da variável, que no caso da equação da transferência de calor é a temperatura, para cada ponto da malha e cada instante do tempo.

No método implícito escreve-se, para cada passo de tempo, um sistema de equações, com uma equação para cada ponto da malha e resolve-se o sistema, num procedimento semelhante ao apresentado para os problemas de valor de fronteira.

Os diferentes métodos serão apresentados em seguida, juntamente com um exemplo da sua utilização.

## MÉTODOS EXPLÍCITOS

O primeiro passo para resolver numericamente uma equação diferencial é sempre discretizá-la. No método explícito as derivadas no **tempo** são aproximadas por diferenças avançadas e as derivadas no **espaço** por diferenças centradas, pelo que o método tem uma precisão de primeira ordem no tempo, e de segunda ordem no espaço. Discretizando os vários termos da equação:

$$\frac{\partial T}{\partial t} \approx \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

onde  $\Delta t$  é o passo de tempo,  $i$  é o índice espacial do ponto a considerar, e a letra  $l$  é utilizada para definir o índice no tempo. Esta aproximação é semelhante à utilizada no método de Euler e de Runge-Kutta.

Para a discretização no espaço utilizam-se diferenças centradas, no passo de tempo  $l$ . Neste caso, como temos uma segunda derivada, a aproximação é:

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i-1}^l - 2T_i^l + T_{i+1}^l}{\Delta x^2}$$

onde  $\Delta x$  é o espaçamento horizontal da malha.

Substituindo as duas aproximações, temos:

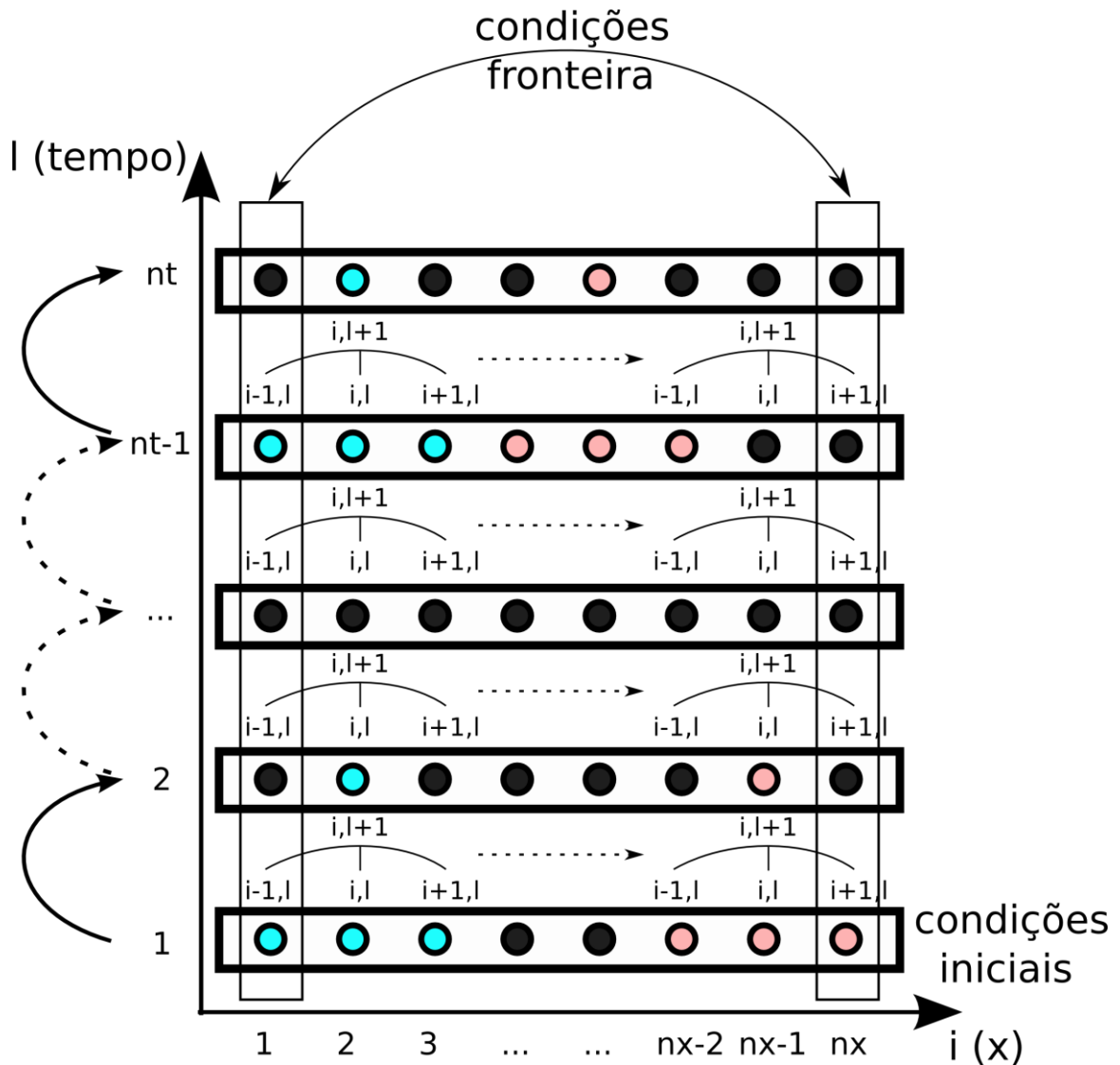
$$k \frac{T_{i-1}^l - 2T_i^l + T_{i+1}^l}{\Delta x^2} = \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

O termo  $T_i^l$  aparece dos dois lados da equação. Reorganizando os termos de forma a ficar o único termo em  $(l+1)$  do lado esquerdo da equação, temos:

$$T_i^{l+1} = C_1 T_{i-1}^l + (1 - 2C_1) T_i^l + C_1 T_{i+1}^l$$

em que  $C_1 = k \frac{\Delta t}{\Delta x^2}$

Esta equação pode ser aplicada a qualquer ponto da malha, onde a temperatura para o novo passo de tempo, num determinado ponto da malha ( $T_i^{l+1}$ ) é obtida a partir da temperatura do instante anterior, do próprio ponto ( $T_i^l$ ) e dos pontos vizinhos ( $T_{i-1}^l$  e  $T_{i+1}^l$ ). O procedimento está esquematizado na figura seguinte:



Tem de ser conhecida a temperatura inicial para todos os pontos da barra. A partir destes calcula-se iterativamente a temperatura para os vários pontos da barra no instante  $l=2$ . As temperaturas nas extremidades são conhecidas para todos os instantes (condições de fronteira).

### Exercício 5.1

Uma barra longa e fina, de  $L=20$  cm de comprimento, e com um coeficiente de transferência de calor  $k=0.835$  cm<sup>2</sup>/s, encontra-se inicialmente à temperatura de



20°C. As duas extremidades da barra são colocadas em gelo ( $T(0,L)=0^{\circ}\text{C}$ ). Calcule a evolução da temperatura da barra, por difusão, ao longo de 5 minutos, considerando um espaçamento horizontal  $\Delta x=5\text{cm}$ , e um passo de tempo  $\Delta t=2$  segundos. Utilize o método explícito.

### Resolução do Exercício 5.1:

Para resolver este problema temos de começar por discretizar a equação da difusão, o que já foi feito em cima. Portanto, partimos da equação discretizada:

$$T_i^{l+1} = C_1 T_{i-1}^l + (1 - 2C_1) T_i^l + C_1 T_{i+1}^l$$

em que  $C_1 = k \frac{\Delta t}{\Delta x^2}$ .

Podemos escrever um código no matlab para resolver esta equação. Neste exemplo representamos a temperatura a cada 20 segundos, representando a distribuição inicial e final a vermelho e a verde, respectivamente.

```

%Dados do Problema
k=0.835
L=20; %cm
dt=2; %seg
dx=5; %cm
t=0:dt:5*60; %4min
x=0:dx:L;
N=length(x);

%Condicao Inicial; Temperatura 20 em toda a barra
Tinicial=20*ones(N,1);

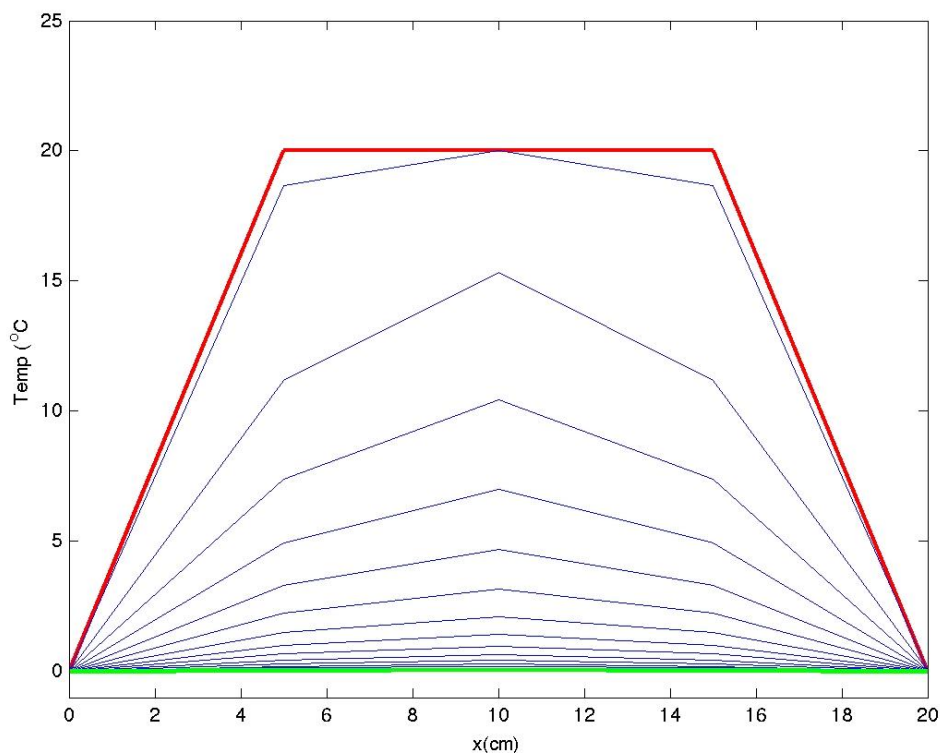
%Condiçoes Fronteira; Temperatura a 0 nas duas extremidades
Te=0; Td=0;
Tinicial(1)=Te; Tinicial(N)=Td;

%Representacao da temperatura inicial
figure(1)
plot(x,Tinicial,'-r','LineWidth',2)
xlabel('x(cm)')
ylabel('Temp (^oC)')
ylim([-1 25])

%Coeficientes da Discretizacao
C1=k*dt/(dx^2);

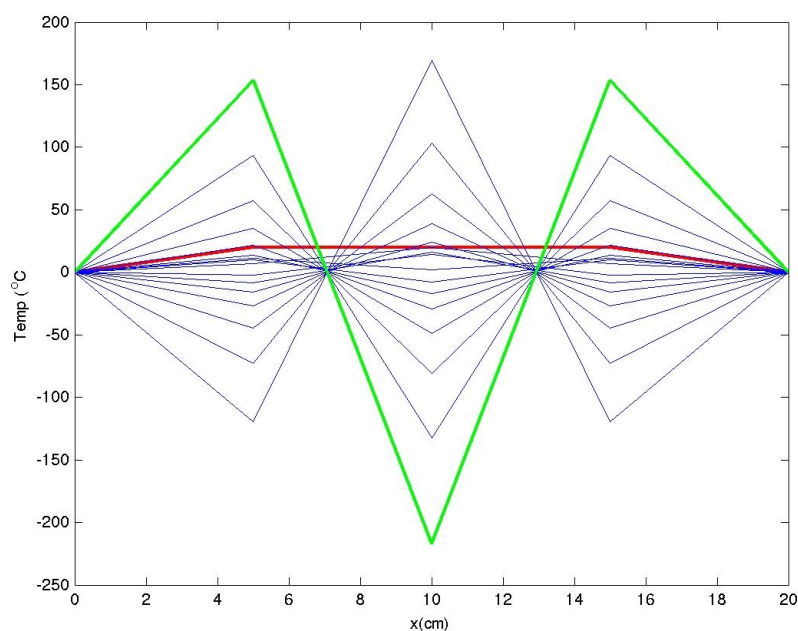
Tanterior=Tinicial;
for l=1:length(t)
    Tnovo(1) = 0;
    for i = 2:N-1
        Tnovo(i) = C1*Tanterior(i-1)+(1-
2*C1)*Tanterior(i)+C1*Tanterior(i+1);
    end
    Tnovo(N) = 0;
    Tanterior=Tnovo;
    if mod(t(l),20)==0
        hold on
        plot(x,Tnovo)
    end
end
Tfinal=Tnovo;
plot(x,Tfinal,'-g','LineWidth',2)
  
```

Obtém-se a seguinte evolução da temperatura:



Na figura verifica-se que inicialmente a barra estava a 20°C (linha vermelha), com a exceção das extremidades que estavam a 0°C. Com o evoluir do tempo (azuis) a temperatura no interior da barra vai diminuindo, encontrando-se totalmente a 0°C ao fim dos 5 minutos (verde).

Se alterarmos o  $\Delta t$  de 2 para 20 segundos o resultado obtido é:



que tem um comportamento instável, e a solução numérica diverge da solução exata.

Isto demonstra que neste método, o  $\Delta t$  e o  $\Delta x$  não podem ser escolhidos de forma arbitrária, mas têm de satisfazer uma condição de estabilidade, dada por:

$$r = \frac{k\Delta t}{\Delta x^2} \leq \frac{1}{2}$$

No caso do problema,  $k=0.835\text{cm}^2/\text{s}$  e  $\Delta x=5\text{cm}$ , pelo que para  $\Delta t=2\text{s}$ ,  $r=0.0668<0.5$ , ou seja, satisfaz o critério de estabilidade.

Quando considerado o  $\Delta t=20\text{s}$ ,  $r=0.668>0.5$ , portanto o critério de estabilidade não é satisfeito, e como tal o método não converge para a solução certa. É necessário ter sempre o critério de estabilidade em consideração quando se utiliza o método explícito.

## MÉTODO IMPLÍCITO SIMPLES

Como foi discutido em cima, os métodos explícitos têm problemas de estabilidade, além de não utilizarem toda a informação disponível para a solução. Como alternativa existem os métodos implícitos, cuja única desvantagem é terem algoritmos um pouco mais complicados.

No método implícito, a cada passo de tempo, resolve-se o sistema de equações constituído por uma equação para cada ponto da malha. De forma resumida, utiliza-se iterativamente, para cada passo de tempo, o processo descrito para os problemas de valor de fronteira.

Para resolver a equação da transferência de calor utilizando o método implícito, começa-se também por discretizar a equação.

As derivadas no tempo são aproximadas por diferenças avançadas (como no método explícito) e as derivadas no espaço por diferenças centradas. Este método tem também uma precisão de primeira ordem no tempo, e de segunda ordem no espaço. Discretizando os vários termos da equação:

$$\frac{\partial T}{\partial t} \approx \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

onde  $\Delta t$  é o passo de tempo, e a letra  $l$  é utilizada para definir o índice no tempo. Para a discretização no espaço utilizam-se diferenças centradas, no passo de tempo  $l+1$  (atenção que no método explícito era em  $l$ ). Neste caso, como temos uma segunda derivada, a aproximação é:

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i-1}^{l+1} - 2T_i^{l+1} + T_{i+1}^{l+1}}{\Delta x^2}$$

Substituindo as duas aproximações, temos:

$$k \frac{T_{i-1}^{l+1} - 2T_i^{l+1} + T_{i+1}^{l+1}}{\Delta x^2} = \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

Reorganizando a equação de forma aos termos em  $(l+1)$  ficarem do lado esquerdo da equação, temos:

$$-C_1 T_{i-1}^{l+1} + (2C_1 + 1)T_i^{l+1} - C_1 T_{i+1}^{l+1} = T_i^l$$

em que  $C_1 = k \frac{\Delta t}{\Delta x^2}$

Para simplificar, vamos escrever a equação na forma

$$AT_{i-1}^{l+1} + BT_i^{l+1} + CT_{i+1}^{l+1} = DT_i^l$$

onde  $A=-C_1$ ,  $B=2C_1+1$ ,  $C=-C_1$  e  $D=1$ .

Pode-se então construir um sistema com uma equação para cada ponto da malha, com as devidas alterações nos pontos onde se aplicam as condições de fronteira.

$$\left\{ \begin{array}{l} \text{condição fronteira 1} \\ AT_1^{l+1} + BT_2^{l+1} + CT_3^{l+1} = T_2^l \\ AT_2^{l+1} + BT_3^{l+1} + CT_4^{l+1} = T_3^l \\ \dots \\ AT_{nx-3}^{l+1} + BT_{nx-2}^{l+1} + CT_{nx-1}^{l+1} = T_{nx-2}^l \\ AT_{nx-2}^{l+1} + BT_{nx-1}^{l+1} + CT_{nx}^{l+1} = T_{nx-1}^l \\ \text{condição fronteira 2} \end{array} \right.$$

Considerando, a título de exemplo, uma malha de 4 pontos, com condições de fronteira de Dirichlet, o sistema seria dado por:

$$\left\{ \begin{array}{l} T_1^{l+1} = \text{condição fronteira 1} \\ AT_1^{l+1} + BT_2^{l+1} + CT_3^{l+1} = T_2^l \\ AT_2^{l+1} + BT_3^{l+1} + CT_4^{l+1} = T_3^l \\ T_4^{l+1} = \text{condição fronteira 2} \end{array} \right.$$

ou na forma matricial:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ A & B & C & 0 \\ 0 & A & B & C \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_1^{l+1} \\ T_2^{l+1} \\ T_3^{l+1} \\ T_4^{l+1} \end{bmatrix} = \begin{bmatrix} \text{cond. fronteira 1} \\ T_2^l \\ T_3^l \\ \text{cond. fronteira 2} \end{bmatrix}$$

A temperatura para o novo instante e para todos os pontos da barra é obtida a partir dos valores do instante anterior, de uma só vez, através da resolução deste sistema. Este processo repete-se iterativamente, para cada passo de tempo.

### Exercício 5.2

Resolva o exercício 5.1 utilizando o método implícito simples.

Resolução do Exercício 5.2:

Para resolver este problema temos de começar por discretizar a equação da difusão segundo o método implícito, o que já foi feito em cima. Portanto, partimos da equação discretizada:

$$AT_{i-1}^{l+1} + BT_i^{l+1} + CT_{i+1}^{l+1} = DT_i^l$$

em que  $C_1 = k \frac{\Delta t}{\Delta x^2}$  e  $A=-C_1$ ,  $B=2C_1-1$ ,  $C=-C_1$  e  $D=1$ .

Podemos escrever um código no matlab para resolver esta equação.

```
%Dados do Problema
k=0.835;
L=20; %cm
dt=2; %seg % ou dt=20;
dx=5; %cm
t=0:dt:5*60; %4min
x=0:dx:L;
N=length(x);

%Condicao Inicial; Temperatura 20 em toda a barra
Tinicial=20*ones(N,1);

%Condicoes Fronteira; Temperatura a 0 nas duas extremidades
Te=0; Td=0;
Tinicial(1)=Te; Tinicial(N)=Td;

%Representacao da temperatura inicial
figure(1)
plot(x,Tinicial,'-r','LineWidth',2)
xlabel('x(cm)'); ylabel('Temp (^oC)'); ylim([-1 25])

%Coeficientes da Discretizacao
C1=k*dt/(dx^2);
A=-C1; B=2*C1+1; C=-C1; D=1;

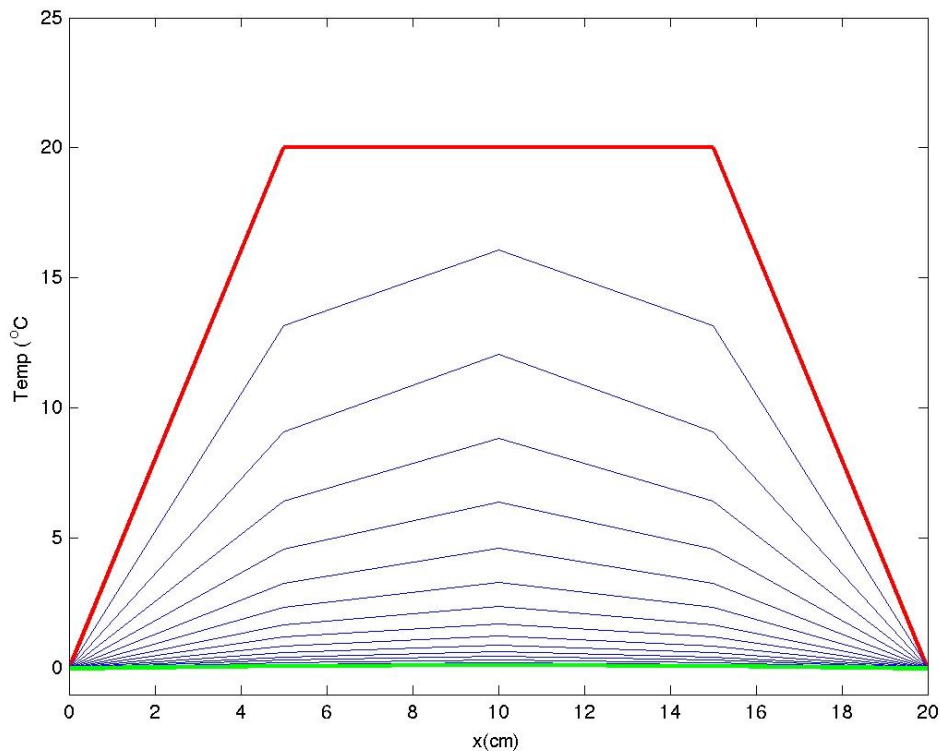
% Ciclo para percorrer os tempos
Tanterior=Tinicial;
for l=1:length(t)

    %Inicializacao das matrizes
    M = zeros(N,N);
    b = zeros(N,1);
    %Condicoes Fronteira
    M(1,1) = 1; b(1)=Te;
    M(N,N)=1; b(N)=Td;

    %Interior
    for i = 2:N-1
        M(i,i-1) = A;
        M(i,i) = B;
        M(i,i+1) = C;
        b(i) = D*Tanterior(i);
    end

    %Resolucao do Sistema
    Tnovo = M\b;
    %Tnovo passa a ser o anterior, para a iteracao seguinte
    Tanterior=Tnovo;
    if mod(t(l),20)==0
        hold on
        plot(x,Tnovo)
    end
end
Tfinal=Tnovo;
plot(x,Tfinal,'-g','LineWidth',2)
```

Neste exemplo representamos a temperatura a cada 20 segundos, representando a distribuição inicial e final a vermelho e a verde, respectivamente. O resultado é igual ao obtido utilizando o método explícito:



Neste caso o resultado obtido é o mesmo, quer se considere um  $\Delta t$  de 2 ou de 20 segundos. Ou seja, no método implícito a solução converge, mesmo quando  $r > 1/2$ , o que permite mais liberdade na escolha dos valores de  $\Delta x$  e  $\Delta t$ .

## MÉTODO DE CRANK-NICOLSON

Finalmente apresentamos o último método, que é uma alternativa ao método implícito e que têm uma precisão de segunda ordem, tanto no espaço como no tempo. O que permite esta precisão é o facto das aproximações das derivadas serem calculadas no meio do passo de tempo. A derivada em ordem ao tempo é aproximada por diferenças avançadas:

$$\frac{\partial T}{\partial t} \approx \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

mas as discretizações no espaço são centradas e são calculadas no ponto médio, fazendo a média das aproximações obtidas em  $l$  e em  $l+1$ :

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{1}{2} \left( \frac{T_{i-1}^l - 2T_i^l + T_{i+1}^l}{\Delta x^2} + \frac{T_{i-1}^{l+1} - 2T_i^{l+1} + T_{i+1}^{l+1}}{\Delta x^2} \right)$$

Substituindo as duas aproximações na equação da difusão,  $k \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t}$ , temos:

$$k \frac{1}{2} \left( \frac{T_{i-1}^l - 2T_i^l + T_{i+1}^l}{\Delta x^2} + \frac{T_{i-1}^{l+1} - 2T_i^{l+1} + T_{i+1}^{l+1}}{\Delta x^2} \right) = \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

Reorganizando a equação de forma aos termos em  $(l+1)$  ficarem do lado esquerdo da equação, temos:

$$\frac{C_1}{2} T_{i-1}^{l+1} + (-1 - C_1) T_i^{l+1} + \frac{C_1}{2} T_{i+1}^{l+1} = -\frac{C_1}{2} T_{i-1}^l + (-1 + C_1) T_i^l - \frac{C_1}{2} T_{i+1}^l$$

em que  $C_1 = k \frac{\Delta t}{\Delta x^2}$

Para simplificar, vamos escrever a equação na forma

$$AT_{i-1}^{l+1} + BT_i^{l+1} + CT_{i+1}^{l+1} = DT_{i-1}^l + ET_i^l + FT_{i+1}^l$$

onde  $A=C_1/2$ ,  $B=(-1-C_1)$ ,  $C=C_1/2$ ,  $D=-C_1/2$ ,  $E=(-1+C_1)$  e  $F=-C_1/2$ .

Pode-se então construir um sistema com uma equação para cada ponto da malha, com as devidas alterações nos pontos onde se aplicam as condições de fronteira.

$$\left\{ \begin{array}{l} \text{condição fronteira 1} \\ AT_1^{l+1} + BT_2^{l+1} + CT_3^{l+1} = DT_1^l + ET_2^l + FT_3^l \\ AT_2^{l+1} + BT_3^{l+1} + CT_4^{l+1} = DT_2^l + ET_3^l + FT_4^l \\ \dots \\ AT_{nx-3}^{l+1} + BT_{nx-2}^{l+1} + CT_{nx-1}^{l+1} = DT_{nx-3}^l + ET_{nx-2}^l + FT_{nx-1}^l \\ AT_{nx-2}^{l+1} + BT_{nx-1}^{l+1} + CT_{nx}^{l+1} = DT_{nx-2}^l + ET_{nx-1}^l + FT_{nx}^l \\ \text{condição fronteira 2} \end{array} \right.$$

Considerando, a título de exemplo, uma malha de 4 pontos, com condições de fronteira de Dirichlet, o sistema ficaria:

$$\left\{ \begin{array}{l} T_1^{l+1} = \text{condição fronteira 1} \\ AT_1^{l+1} + BT_2^{l+1} + CT_3^{l+1} = DT_1^l + ET_2^l + FT_3^l \\ AT_2^{l+1} + BT_3^{l+1} + CT_4^{l+1} = DT_2^l + ET_3^l + FT_4^l \\ T_4^{l+1} = \text{condição fronteira 2} \end{array} \right.$$

ou na forma matricial:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ A & B & C & 0 \\ 0 & A & B & C \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_1^{l+1} \\ T_2^{l+1} \\ T_3^{l+1} \\ T_4^{l+1} \end{bmatrix} = \begin{bmatrix} \text{cond. fronteira1} \\ DT_1^l + ET_2^l + FT_3^l \\ DT_1^l + ET_2^l + FT_3^l \\ \text{cond. fronteira2} \end{bmatrix}$$

Da mesma forma que no método implícito simples, a temperatura para o novo instante e para todos os pontos da barra é obtida a partir dos valores do instante anterior, de uma só vez, através da resolução deste sistema. Este processo repete-se iterativamente, para cada passo de tempo.

Exercício 5.3

Resolva o exercício 5.1 utilizando o método de Crank-Nicolson.

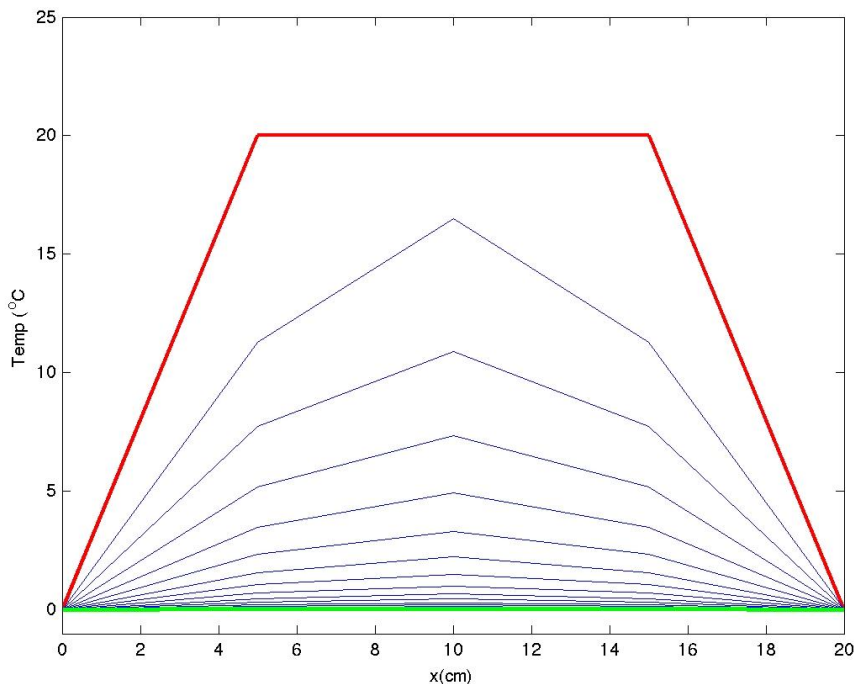
Resolução do Exercício 5.3:

Para resolver este problema temos de começar por discretizar a equação da difusão segundo o método de Crank-Nicolson, o que já foi feito em cima. Portanto, partimos da equação discretizada:

$$AT_{i-1}^{l+1} + BT_i^{l+1} + CT_{i+1}^{l+1} = DT_{i-1}^l + ET_i^l + FT_{i+1}^l$$

onde  $C_1 = k \frac{\Delta t}{\Delta x^2}$  e  $A=C_1/2, B=(-1-C_1), C=C_1/2, D= -C_1/2, E=(-1+C_1)$  e  $F=-C_1/2$ .

Utilizando o código de MATLAB apresentado na página seguinte, obtém-se novamente a solução:





```

%Dados do Problema
k=0.835;
L=20; %cm
dt=20; %seg
dx=5; %cm
t=0:dt:5*60; %4min
x=0:dx:L;
N=length(x);

%Condicao Inicial; Temperatura 20 em toda a barra
Tinicial=20*ones(N,1);

%Condiçoes Fronteira; Temperatura a 0 nas duas extremidades
Te=0; Td=0;
Tinicial(1)=Te; Tinicial(N)=Td;

%Representacao da temperatura inicial
figure(1)
plot(x,Tinicial,'-r','LineWidth',2)
xlabel('x(cm)')
ylabel('Temp (^oC)')
ylim([-1 25])

%Coeficientes da Discretizacao
C1=k*dt/(dx^2);
A=C1/2; B=-1-C1; C=C1/2; D=-C1/2; E=-1+C1; F=-C1/2;

% Ciclo para percorrer os tempos
Tanterior=Tinicial;
for l=1:length(t)

    %Inicializacao das matrizes
    M = zeros(N,N);
    b = zeros(N,1);

    %Condiçoes Fronteira
    M(1,1) = 1; b(1)=Te;
    M(N,N)=1; b(N)=Td;

    %Interior
    for i = 2:N-1
        M(i,i-1) = A;
        M(i,i) = B;
        M(i,i+1) = C;
        b(i) = D*Tanterior(i-1)+E*Tanterior(i)+F*Tanterior(i+1);
    end

    %Resolucao do Sistema
    Tnovo = M\b;
    %Tnovo passa a ser o anterior, para a proxima iteracao
    Tanterior=Tnovo;
    if mod(t(l),20)==0
        hold on
        plot(x,Tnovo)
    end
end
Tfinal=Tnovo;
plot(x,Tfinal,'-g','LineWidth',2)

```