Universidade do Minho
Escola de Ciências

Departamento de Matemática

Equações Diferenciais Ordinárias

- Problemas de Valores Iniciais
- Problemas de Valores de Fronteira

PROBLEMAS DE VALORES INICIAIS

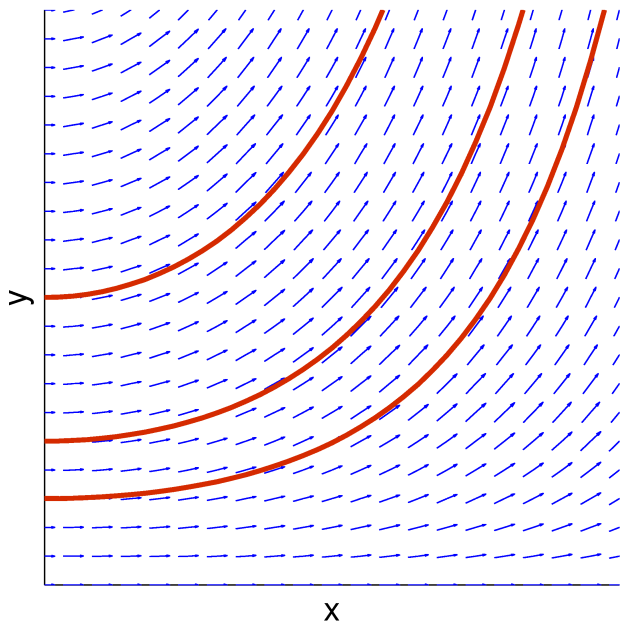
PVI de primeira ordem

PVI

$$y'(x) = f(x, y) \quad \text{EDO}$$

$$y(a) = \alpha \quad \text{condição inicial}$$

Problema Encontrar a curva integral que passa em (a, α).



Curvas integrais

PVI de ordem superior a 1

- Ordem 2:

$$y'' + f(x)y' + g(x)y = h(x) \quad \longrightarrow \quad \left\{ \begin{array}{l} y' = u \\ u' = h(x) - g(x)y - f(x)u \end{array} \right.$$

$$y'' = \underbrace{F(x, y, y')}_{h(x) - f(x)y' - g(x)y} \quad \longrightarrow \quad \underbrace{\mathbf{z}' = \mathbf{F}(x, \mathbf{z})}_{\mathbf{z}=(y,u);\ \mathbf{F}(x,\mathbf{z})=(u, h(x)-g(x)y-f(x)u)}$$

- Ordem k:

$$y^{(k)} = F(x, y', y'', \cdots, y^{(k-1)}) \underset{z_i = y^{(i-1)}}{\longrightarrow} \mathbf{z}' = \mathbf{F}(x, \mathbf{z})$$

Exemplo: Equação de **van der Pol**

$$y'' - \mu(1 - y^2)y' + y = 0$$

$$y(a) = \alpha \quad y'(a) = \beta$$

Reescrever como

$$\mathbf{z}' = \begin{bmatrix} y \\ u \end{bmatrix}' = \begin{bmatrix} u \\ \mu(1 - y^2)u - y \end{bmatrix} \qquad \mathbf{z}_a = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Existência e unicidade de solução

Teorema

Seja $f(x, y)$ definida e contínua para todos os pontos (x, y) tais que $a \leq x \leq b$, $y \in \mathbb{R}$ (a e b finitos).
Se f satisfaz uma condição de Lipschitz relativamente à segunda variável, i.e., se existe uma constante $0 < L < \infty$ tal que

$$|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|,$$

para $a \leq x \leq b$ e quaisquer $y_1, y_2 \in \mathbb{R}$, então, dado qualquer valor α existe uma e uma só solução do problema

$$y'(x) = f(x, y), \quad y(a) = \alpha.$$

Métodos de variável discreta

Ideia: como não conseguimos determinar $y(x)$ para todo o $x \in [a, b]$, calculamos **aproximações** y_k para os valores $y(x_k)$ num **conjunto discreto de pontos** $(x_k)_{k=0}^{N}$, nesse intervalo. Por simplificação, consideramos pontos equidistantes, i.e., $x_k = a + kh;\ k = 0, 1, \ldots, N$, onde o **passo** h é dado por

$$h = \frac{b-a}{N},$$

para um determinado inteiro N. O valor inicial dá-nos

$$y_0 = y(x_0) = y(a) = \alpha.$$

A ideia é tentar, a partir deste valor, determinar y_1 como aproximação para $y(x_1)$, y_2 como aproximação para $y(x_2)$ etc.

Os métodos que determinam aproximações para a solução do problema num conjunto discreto de pontos (x_k) da variável independente são chamados **métodos de variável discreta** .

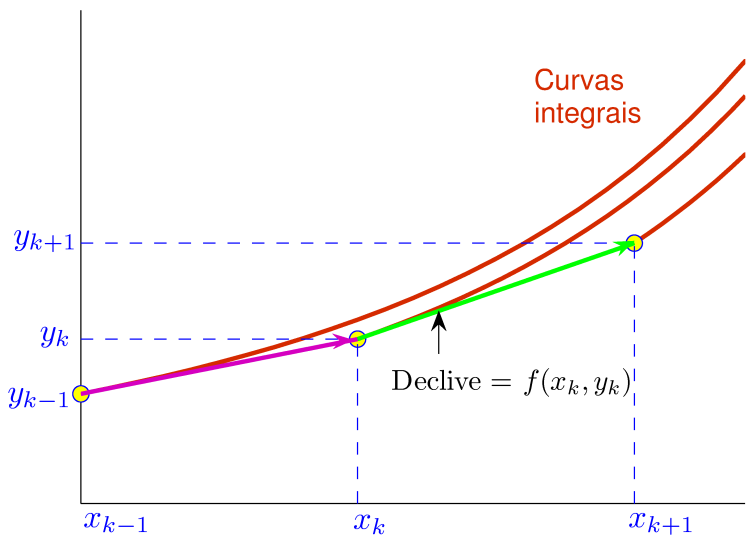
Métodos de passo único

Método de Euler

$$y_{k+1} = y_k + hf(x_k, y_k); k = 0, 1, \ldots, N-1,$$
$$y_0 = \alpha.$$

Geometricamente,

o Método de Euler consiste em aproximar a solução em x_{k+1}, seguindo a tangente à curva integral em x_k



Exemplo: Método de Euler

$y' = xy, \quad y(0) = 1$

Solução exata: $y(x) = e^{x^2/2}$.

$h = 0.2$

k	x_k	y_k	$y(x_k)$	Erro
1	0.2	1.000	1.020	0.020
2	0.4	1.040	1.083	0.043

$h = 0.1$

k	x_k	y_k	$y(x_k)$	Erro
1	0.1	1.000	1.005	0.005
2	0.2	1.010	1.020	0.010
3	0.3	1.030	1.046	0.016
4	0.4	1.062	1.083	0.022

Método de Euler no MATLAB

```
function y=metEuler(f,a,b,alfa,N)
h=(b-a)/N;
x=a:h:b;
y=zeros(1,N+1);
y(1)=alfa;
for k=1:N
    y(k+1)=y(k)+h*f(x(k),y(k));
end

>> f=@(x,y) x.*y;
>> y=metEuler(f,0,0.4,1,4)
y =
1.0000 1.0000 1.0100 1.0302 1.0611
```

Erro de discretização

Erro de discretização local

Chama-se **erro de discretização local** no ponto x_{k+1} e denota-se por $\mathcal{L}_h(x_{k+1})$, ao erro produzido na aplicação do passo k do método (isto é, ao erro cometido ao passar de y_k para y_{k+1}), *supondo que partimos da solução exata, isto é, supondo que* $y_k = y(x_k)$. Por outras palavras,

$\mathcal{L}_h(x_{k+1})$ é a diferença entre y_{k+1} e o valor em x_{k+1} da curva integral $x \mapsto \tilde{y}(x)$ que passa em (x_k, y_k).

$$\left.\begin{array}{l} y \in C^2[a,b] \\ \\ M = \max\limits_{a \le x \le b} |y''(x)| \end{array}\right| \quad \Rightarrow \quad |\mathcal{L}_h(x_{k+1})| \le \frac{h}{2}^2 M$$

O erro de discretização local do método de Euler é $\mathcal{O}(h^2)$.

Erro de discretização global

Chama-se **erro de discretização global** no ponto x_{k+1} e denota-se por $\epsilon_h(x_{k+1})$ ao erro acumulado nos diversos passos até se chegar a y_{k+1}. Por outras palavras,

$\epsilon_h(x_{k+1})$ é a diferença entre y_{k+1} e o valor em x_{k+1} da curva integral $x \mapsto y(x)$ que passa em (x_0, y_0).

Erro de discretização do método de Euler

Se $y \in C^2[a,b]$, $M = \max_{a \leq x \leq b} |y''(x)|$ e , além disso, f é diferenciável em relação à segunda variável com derivada parcial $\frac{\partial f}{\partial y}$ limitada, i.e. $|\frac{\partial f}{\partial y}| \leq L$, então as aproximações obtidas pelo método de Euler convergem para a solução exata, quando $h \to 0$ e

$$\epsilon_h(x_{k+1})| \leq \frac{hM}{2L}(e^{(x_k - x_0)L} - 1),$$

i.e. **o erro de discretização global do método de Euler é** $\mathcal{O}(h)$.

Estimativa para o erro no método de Euler

- y_k aproximação para $y(x_k)$ obtida usando passo h
- $\tilde{y}_k$ aproximação para $y(x_k)$ obtida usando passo $2h$

Estimativa para o erro de discretização no ponto x_k:

(correspondente ao uso do passo "mais fino")

$$|\epsilon_h(x_k)| = |y(x_k) - y_k| \approx |y_k - \tilde{y}_k|$$

Métodos baseados na série de Taylor

O método de Euler pode ser deduzido truncando a expansão em série de Taylor de $y(x_{k+1})$ em torno de x_k antes do termo $\mathcal{O}(h^2)$.

Métodos de ordem superior poderiam ser obtidos de modo análogo, retendo mais termos da série de Taylor. Por exemplo, retendo os três primeiros termos da série, obtém-se o método

$$y_{k+1} = y_k + hf(x_k, y_k) + \frac{h^2}{2}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f\right)(x_k, y_k),$$

o qual tem ordem de convergência local $\mathcal{O}(h^3)$ e ordem de convergência global $\mathcal{O}(h^2)$.

Este tipo de métodos baseados na expansão em série de Taylor têm, no entanto, a desvantagem de necessitarem do cálculo das derivadas parciais da função f , o que os pode tornar bastante trabalhosos.

Métodos de Runge-Kutta

Os chamados métodos de Runge-Kutta foram desenvolvidos com o objetivo de produzir resultados com o mesmo grau de precisão dos métodos obtidos pela expansão em série de Taylor, mas evitando o cálculo das várias derivadas da função f.

Método de Runge-Kutta de ordem 2

$$\begin{aligned} y_{k+1} &= y_k + \tfrac{1}{2}(k_1 + k_2), \\ k_1 &= hf(x_k, y_k), \\ k_2 &= hf(x_k + h, y_k + k_1), \\ y_0 &= \alpha. \end{aligned}$$

Este método corresponde a substituir $f(x_k, y_k)$ no método de Euler, por uma média pesada do valor de f nos pontos (x_k, y_k) e $(x_{k+1}, y_k + hf(x_k, y_k))$.

A ordem de convergência global deste método é $\mathcal{O}(h^2)$.

Método de Runge-Kutta de ordem 4

$$
\begin{aligned}
y_{k+1} &= y_k + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\
k_1 &= hf(x_k, y_k), \\
k_2 &= hf(x_k + \tfrac{h}{2}, y_k + \tfrac{k_1}{2}), \\
k_3 &= hf(x_k + \tfrac{h}{2}, y_k + \tfrac{k_2}{2}), \\
k_4 &= hf(x_k + h, y_k + k_3), \\
y_0 &= \alpha.
\end{aligned}
$$

Este método corresponde a substituir $f(x_k, y_k)$ no método de Euler, por uma média pesada do valor de f em 4 pontos.

A ordem de convergência global deste método é $\mathcal{O}(h^4)$.

Métodos de passo único

Métodos de **passo único** podem ser escritos na forma geral

$$y_{k+1} = y_k + h\phi(x_k, y_k),$$

para alguma função ϕ. Nestes métodos, o passo $k+1$ baseia-se apenas no uso de informação obtida no passo k .

Euler: $\phi(x, y) = f(x, y)$

R-K ordem 2: $\phi(x, y) = \frac{1}{2}\left(f(x, y) + f(x + h, y + h(f(x, y))\right)$.

Pode provar-se, sob certas condições em ϕ e f que se o erro de discretização local de um método é $\mathcal{O}(h^p)$, então o erro de discretização global é $\mathcal{O}(h^{p-1})$.

Erros de arredondamento

$$\text{Erro} \begin{cases} \text{Erro discretização ou truncatura} \\ \\ \text{Erro de arredondamento} \end{cases}$$

Consideremos um método de passo único da forma

$$y_{k+1} = y_k + h\phi(x_k, y_k),$$

com erro de discretização $\mathcal{O}(h^p)$.

$$h \textbf{ diminui} \begin{cases} \text{Erro discretização } \textbf{diminui} \\ \\ \text{Erro de arredondamento } \textbf{aumenta} \end{cases}$$

Na prática, isto significa que, a partir de determinado valor do passo, os erros de arredondamento podem ser dominantes relativamente aos erros de discretização e nada se ganha com a diminuição do passo.

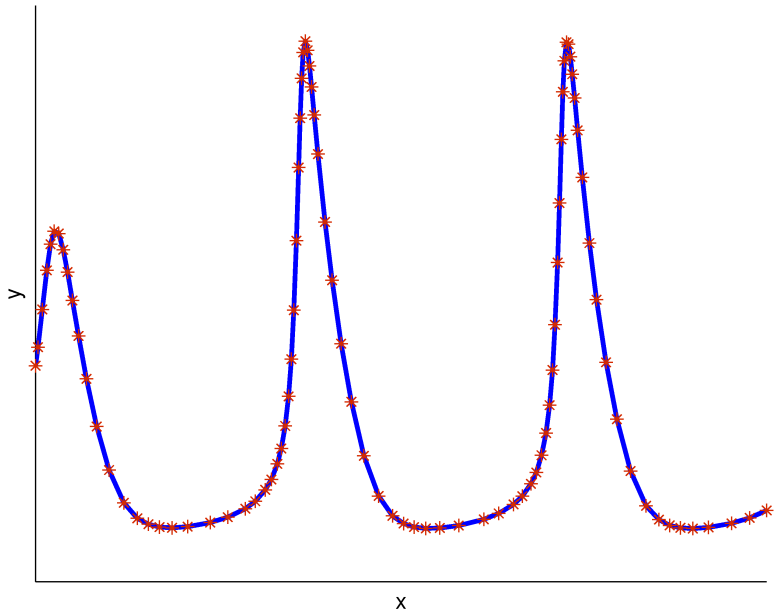
Métodos adaptativos

Relembremos que, para o Método de Euler, o erro de discretização local satisfaz

$$|\mathcal{L}_h(x_{k+1})| \leq \frac{h}{2}^2 M,$$

onde $M = \max_{a \leq x \leq b} |y''(x)|$.

Isto significa que o erro será, em princípio, pequeno se a segunda derivada da solução for pequena. Em regiões onde isto acontece, pode obter-se a precisão desejada, usando um valor do passo h relativamente grande. Em contrapartida, se M é grande, é necessário usar um passo mais pequeno para controlar o erro. Esta é a ideia do controle adaptativo do passo: usar uma malha que tem mais pontos (h menor) nas regiões onde a solução tiver uma variação mais rápida e menos pontos (h maior) nas zonas em que y variar lentamente.



No caso dos métodos de Runge-Kutta, a estratégia é análoga, mas devem ser tidas em linha de conta derivadas da solução de ordem superior. Note-se que, geralmente, estas derivadas não são conhecidas. No entanto, em algumas situações são conhecidas as regiões onde a solução varia mais e onde varia menos, podendo o passo ser adaptado em função desse facto.

Atualmente todos os bons códigos que usam Métodos de Runge-Kutta empregam mecanismos para automaticamente alterar o valor do passo h à medida que o processo avança.

O procedimento usual é estimar o erro de discretização local e ajustar o passo adequadamente. Há duas formas simples de o fazer:

Processo 1

Usar um método de ordem p para calcular uma aproximação $\tilde{y}_k$ com passo h e repetir o processo com passo $h/2$ para obter uma outra aproximação (em princípio diferente da anterior) y_k e comparar os resultados, usando a estimativa (ver exercício),

$$\epsilon_{h/2}(x_k)| = |y(x_k) - y_k| \approx \frac{|y_k - \tilde{y}_k|}{2^p - 1}$$

Processo 2

Usar 2 métodos de Runge-Kutta de ordem p e $p+1$: seja y_k o valor obtido com um método de ordem p e $\bar{y}_k$ o valor obtido com um método de ordem $p+1$. Então (ver exercício),

$$\epsilon_h(x_k)| = |y(x_k) - y_k| \approx |y_k - \bar{y}_k|$$

Em ambos os casos, admitimos que pretendemos distribuir uniformemente o erro ϵ no intervalo $[a, b]$. Então, se a condição $\epsilon_h(x_k) \leq \epsilon h$ for verdadeira, o valor obtido usando o passo correspondente é aceite. Caso contrário, deve usar-se um passo menor.

O segundo processo não teria vantagens sobre o primeiro se não fosse possível reaproveitar os valores de f utilizados no método de ordem p para o método de ordem $p+1$. Fehlberg mostrou que é possível desenvolver métodos, conhecidos como métodos de Runge-Kutta-Fehlberg , gozando desta propriedade.

BS(2,3) em `ode23`

O MATLAB tem implementada na função `ode23` a técnica de usar um método de Runge-Kutta de ordem 3 para estimar o erro associado ao uso do método de Runge-Kutta de ordem 2. O algoritmo usado é da autoria de Bogacki-Shampine (1989) BS(2,3).

$$
\begin{aligned}
k_1 &= f(x_k, y_k), \\
k_2 &= f(x_k + \tfrac{1}{2}h, y_k + \tfrac{1}{2}hk_1), \\
k_3 &= f(x_k + \tfrac{3}{4}h, y_k + \tfrac{3}{4}hk_2), \\
k_4 &= f(x_k + h, y_k + \tfrac{2}{9}hk_1 + \tfrac{1}{3}hk_2 + \tfrac{4}{9}hk_3), \\
&\quad k_4 \text{ pode ser reutilizado no passo seguinte}
\end{aligned}
$$

$$
\begin{aligned}
y_{k+1} &= y_k + \tfrac{7}{24}hk_1 + \tfrac{1}{4}hk_2 + \tfrac{1}{3}hk_3 + \tfrac{1}{8}hk_4, && \text{ordem 2} \\
\bar{y}_{k+1} &= y_k + \tfrac{2}{9}hk_1 + \tfrac{1}{3}hk_2\,\tfrac{4}{9}hk_3, && \text{ordem 3}
\end{aligned}
$$

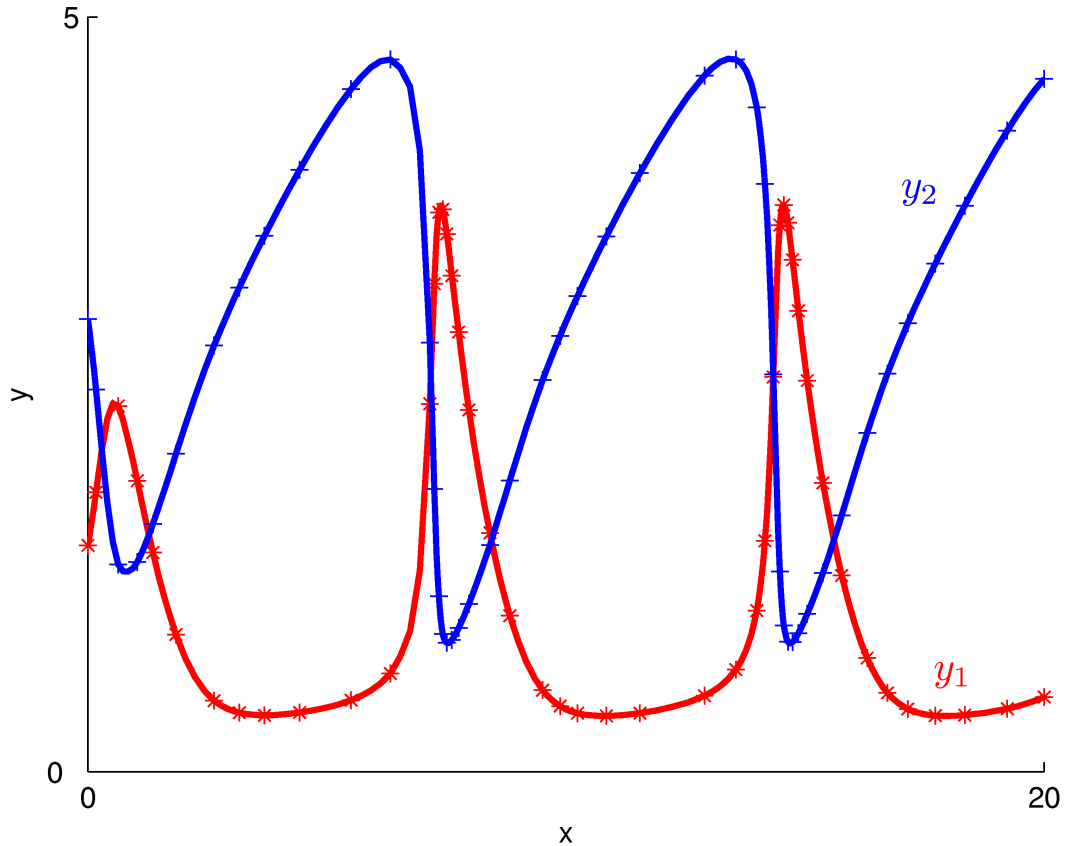
Este método precisa de 4 passos (3 avaliações da função, se o passo for aceite). Fehlberg (1979) desenvolveu o popular método F(4,5) que envolve apenas 6 passos (ver Pina, pag. 532). A função `ode45` do Matlab é baseada no algoritmo de 7 passos de Dormand e Prince (1980) DOPRI5. O passo adicional é usado para melhorar a fórmula.

Exemplo: Resolver o sistema

$$\begin{cases} y_1' = 1 + y_1^2 y_2 - 4y_1, & y_1(0) = 1.5 \\ y_2' = 3y_1 - y_1^2 y_2, & y_2(0) = 3 \end{cases}$$

usando a função `ode45` do Matlab.

```
>> f=@(x,y) [1+y(1)^2*y(2)-4*y(1);3*y(1)-y(1)^2*y(2)];
>> options=odeset('absTol',1e-6,'RelTol',1e-3,...
        'Stats','on');
>> [x y]=ode45(f,[0,20],[1.5;3],options);
 46 successful steps
 12 failed attempts
 349 function evaluations
>> i=1:4:length(x);
>> plot(x,y(:,1),'r',x(i),y(i,1),'*r',...
        x,y(:,2),'b',x(i),y(i,2),'+b')
```


5
0
y
0
20
x
y_2
y_1

PROBLEMAS DE VALORES DE FRONTEIRA

Diferenças finitas para aproximar derivadas

Sejam $x, x-h$ e $x+h$ três pontos de um intervalo onde uma determinada função y está definida.[1] Consideremos as expansões em série de Taylor de $y(x+h)$ e $y(x-h)$ em torno de x, até à segunda ordem:

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}y''(\xi), \quad \xi \in (x, x+h) \tag{1}$$

$$y(x-h) = y(x) - hy'(x) + \frac{h^2}{2}y''(\mu), \quad \mu \in (x-h, x) \tag{2}$$

De (1) decorre de imediato que

$$\frac{y(x+h)-y(x)}{h} = y'(x) + \frac{h}{2}y''(\xi), \quad \xi \in (x, x+h).$$

[1] No que se segue, admitimos sempre que a função em causa tem tantas derivadas contínuas quanto necessário.

Temos, então

$$y'(x) = \frac{y(x+h) - y(x)}{h} - \frac{h}{2}y''(\xi) = \frac{y(x+h) - y(x)}{h} + \mathcal{O}(h). \tag{3}$$

Assim, se h for pequeno, poderemos aproximar a derivada de y no ponto x pela combinação linear dos valores de y em x e $x+h$ dada por

$$\frac{y(x+h) - y(x)}{h}. \tag{4}$$

Essa expressão é conhecida por diferença finita progressiva de $1^{\underline{a}}$ ordem para aproximar $y'(x)$. De modo análogo se define a diferença finita regressiva de $1^{\underline{a}}$ ordem

$$\frac{y(x) - y(x-h)}{h}, \tag{5}$$

tendo-se, naturalmente,

$$y'(x) = \frac{y(x) - y(x-h)}{h} + \mathcal{O}(h). \tag{6}$$

Se y for três vezes continuamente diferenciável em $[x-h, x+h]$, tem-se

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{6}y'''(\xi_+), \xi_+ \in (x, x+h)$$
$$y(x-h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) - \frac{h^3}{6}y'''(\xi_-), \xi_- \in (x-h, x)$$

Subtraindo membro a membro, vem

$$y(x+h) - y(x-h) = 2hy'(x) + \frac{h^3}{3}\left(\frac{y'''(\xi_+) + y'''(\xi_-)}{2}\right).$$

Sendo y''' contínua, a média dos valores $y'''(\xi_+)$ e $y'''(\xi_-)$ será igual a $y'''(\xi)$ para algum $\xi \in (x-h, x+h)$.Tem-se, assim, a seguinte fórmula, conhecida por diferença finita centrada de segunda ordem para aproximar $y'(x)$:

$$\begin{aligned} y'(x) &= \frac{y(x+h) - y(x-h)}{2h} - \frac{h^2}{6}y'''(\xi) \\ &= \frac{y(x+h) - y(x-h)}{2h} + \mathcal{O}(h^2). \end{aligned} \tag{7}$$

Fórmulas de diferenças finitas para aproximar outras derivadas de y podem ser obtidas de modo análogo. Em particular, tem-se a seguinte fórmula centrada (de segunda ordem) para aproximar $y''(x)$:

$$y''(x) = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} + \mathcal{O}(h^2). \tag{8}$$

Problemas de Valores de Fronteira (PVF)

Consideremos, agora, uma equação diferencial de $2^{\underline{a}}$ ordem linear da forma

$$y''(x) + p(x)y'(x) + q(x)y(x) = r(x),\ x \in [a, b], \tag{9}$$

com p, q e r funções contínuas em $[a, b]$, sujeita a condições de fronteira do tipo[2]

$$y(a) = \alpha, \qquad y(b) = \beta. \tag{10}$$

As equações (9) e (10) definem um **problema de valores de fronteira (PVF)** para a função incógnita y, para a qual se pretende obter uma solução aproximada, numericamente.

Nota: No que se segue, suporemos sempre que a solução y existe, é única e suficientemente diferenciável em $[a, b]$.

[2] Condições de fronteira mais gerais serão consideradas posteriormente.

Método de diferenças finitas

O método que vamos descrever para a resolução do PVF (9)-(10) é chamado um **método de diferenças finitas**, pois baseia-se no uso de fórmulas de diferenças finitas para aproximar as derivadas envolvidas em (9).

Dividamos o intervalo $[a, b]$ em N subintervalos de igual amplitude h, pela introdução de $N + 1$ nós, i.e. sejam

$$x_k = a + (k - 1)h; k = 1, \ldots, N + 1; h = \frac{b - a}{N}. \tag{11}$$

Os nós $x_1 = a$ e $x_{N+1} = b$ são ditos nós fronteiros, sendo os restantes chamados nós interiores.

Em cada nó interior a equação diferencial

$$y''(x) + p(x)y'(x) + q(x)y(x) = r(x)$$

tem de ser satisfeita, isto é, tem-se

$$y''(x_k) + p(x_k)y'(x_k) + q(x_k)y(x_k) = r(x_k);\ k = 2, \ldots, N. \tag{12}$$

Se substituirmos as derivadas envolvidas na equação anterior por fórmulas de diferenças centradas (de $2^{\underline{a}}$ ordem), vem

$$\frac{y(x_{k+1}) - 2y(x_k) + y(x_{k-1})}{h^2} + p(x_k)\frac{y(x_{k+1}) - y(x_{k-1})}{2h} + \\ + q(x_k)y(x_k) = r(x_k) + \mathcal{O}(h^2); k = 2, \ldots, N.$$

Aproximações y_k para $y(x_k)$ poderão ser obtidas das equações anteriores "ignorando"os termos $\mathcal{O}(h^2)$, isto é, de

$$\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + p(x_k)\frac{y_{k+1} - y_{k-1}}{2h} + q(x_k)y_k = r(x_k);$$
$$k = 2, \ldots, N, \qquad (13)$$

sendo os valores y_1 e y_{N+1} conhecidos, das condições de fronteira:

$$y_1 = y(x_1) = \alpha, \quad y_{N+1} = y(x_{N+1}) = \beta. \qquad (14)$$

As equações (13) podem reescrever-se como

$$\underbrace{\left(1 - \tfrac{h}{2}p(x_k)\right)}_{a_k} y_{k-1} + \underbrace{\left(-2 + h^2 q(x_k)\right)}_{d_k} y_k + \underbrace{\left(1 + \tfrac{h}{2}p(x_k)\right)}_{c_k} y_{k+1} = \underbrace{h^2 r(x_k)}_{b_k};$$
$$k = 2, \ldots, N. \quad (15)$$

Estas equações, juntamente com (14), constituem um sistema linear de $N-1$ equações nas $N-1$ incógnitas $y_2, \ldots, y_N$, o qual pode ser escrito na forma matricial

$$T\mathbf{y} = \mathbf{b},$$

onde:

- T é a matriz tridiagonal

$$T = \begin{pmatrix} d_2 & c_2 & & & \\ a_3 & d_3 & c_3 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{N-1} & d_{N-1} & c_{N-1} \\ & & & a_N & d_N \end{pmatrix},$$

  em que

$$a_k = 1 - \tfrac{h}{2}\, p(x_k), \quad d_k = -2 + h^2 q(x_k), \quad c_k = 1 + \tfrac{h}{2}\, p(x_k),$$

- $\mathbf{y} = (y_2, \ldots, y_N)^{\mathrm{T}}$
- $\mathbf{b} = (h^2 r(x_2) - a_2\,\alpha,\ h^2 r(x_3), \ldots, h^2 r(x_{N-1}),\ h^2 r(x_N) - c_N\,\beta)^{\mathrm{T}}$.

O método que descrevemos é chamado método das diferenças finitas de $2^{\underline{\mathrm{a}}}$ ordem para resolver o PVF considerado.

Exemplo: Considere-se o seguinte PVF

$$\begin{cases} y''(x) + y'(x) - xy(x) = -xe^{2-x}, \quad x \in [1,2]\,, \\ y(1) = e, \\ y(2) = 1. \end{cases}$$

Vamos determinar a solução numérica para este problema, usando o método das diferenças finitas de $2^{\underline{a}}$ ordem, com passo $h = 0.25$.

Tem-se, neste caso, $p(x) = 1, q(x) = -x$ e $r(x) = -xe^{2-x}$. Então, as três equações correspondentes aos nós interiores são

$$(1 - \tfrac{h}{2})y_{k-1} + (-2 - h^2 x_k)y_k + (1 + \tfrac{h}{2})y_{k+1} = -h^2(x_k e^{2-x_k}); k = 2, 3, 4,$$

ou seja, são, atendendo a que $x_2 = 1.25, x_3 = 1.5, x_4 = 1.75$ e $h = 0.25$:

Por uma questão de simplicidade, trabalharemos com 4 c.d.

$$\begin{cases} 0.8750y_1 - 2.0781y_2 + 1.1250y_3 = -0.1654 \\ 0.8750y_2 - 2.0938y_3 + 1.1250y_4 = -0.1546 \\ 0.8750y_3 - 2.1094y_4 + 1.1250y_5 = -0.1404 \end{cases}.$$

Substituindo $y_1 = 2.7183$ (aproximação para $y(1) = e$ com 4 c.d. de precisão) na primeira equação e $y_5 = 1$ na terceira, obtém-se o seguinte sistema nas incógnitas y_2, y_3, y_4

$$\begin{cases} -2.0781y_2 + 1.1250y_3 = -2.5439 \\ 0.8750y_2 - 2.0938y_3 + 1.1250y_4 = -0.1546 \\ 0.8750y_3 - 2.1094y_4 = -1.2654 \end{cases},$$

o qual pode escrever-se matricialmente como $T\mathbf{y} = \mathbf{b}$, onde

$$T = \begin{pmatrix} -2.0781 & 1.1250 & 0 \\ 0.8750 & -2.0938 & 1.1250 \\ 0 & 0.8750 & -2.1094 \end{pmatrix},$$

$\mathbf{y} = (y_2, y_3, y_4)^{\mathrm{T}}$ e $\mathbf{b} = (-2.5439, -0.1546, -1.2654)^{\mathrm{T}}$.

Resolvendo o sistema anterior, obtém-se

$$\mathbf{y} = (2.1162, 1.6478, 1.2834)^{\mathrm{T}}.$$

A solução analítica do problema é $y(x) = e^{2-x}$, pelo que, nos pontos considerados, tem-se (com 4 c.d.):

$$\mathbf{y}_{\mathrm{exato}} = (2.1170, 1.6487, 1.2840)^{\mathrm{T}}.$$

Exemplo: [Resolução em Matlab]

Apresenta-se de seguida uma sequência de instruções em Matlab para resolver o problema anterior (a qual poderá adaptar-se facilmente para a resolução de problemas análogos).

```
>> p = @(x) 1; q = @(x) -x; r = @(x) -x.*exp(2-x);
>> h = 0.25;
>> x = 1:h:2; N = length(x)-1;
>> alfa = exp(1); beta = 1;
>> a=zeros(size(x));
>> b=a; c=a;d=a;
```

```
>> for k=2:N
a(k) = 1-(h/2)*p(x(k));
d(k) = -2+h^2*q(x(k));
c(k) = 1+(h/2)*p(x(k));
b(k) = h^2*r(x(k));
end
>> av = a(3:N);  dv = d(2:N);  cv = c(2:N-1);
>> T=diag(dv)+diag(av,-1)+diag(cv,1)

T =
-2.0781    1.1250         0
 0.8750   -2.0938    1.1250
 0         0.8750   -2.1094
```

```
>> b(2) = b(2) -a(2)*alfa;
>> b(N)= b(N)-c(N)*beta;
>> bv=b(2:N)'

bv =
-2.5439
-0.1546
-1.2654

>> y=(T\bv)'

y=
2.1162    1.6478    1.2834

>> yanal=@(x) exp(2-x);
>> yexato=yanal(x(2:N))

yexato= =
2.1170    1.6487    1.2840
```

Condições de fronteira envolvendo derivadas

Para ilustrar a aplicação de métodos de diferenças finitas para problemas cujas condições de fronteira não sejam tão simples como as condições (10), considere-se, por exemplo, um problema do tipo

$$\begin{cases} y''(x) + p(x)y'(x) + q(x)y(x) = r(x),\ x \in [a,b], \\ y'(a) = \gamma, \\ y(b) = \beta. \end{cases}$$

Supondo que são usadas as mesmas fórmulas de diferenças finitas, teremos então as mesmas equações (15) para obter $y_2, y_3, \ldots, y_N$.

A diferença agora, é que o valor de y_1 na primeira equação não é conhecido da condição de fronteira em $x = a$, sendo portanto uma nova incógnita a determinar.

Uma primeira possibilidade será aproximar condição de fronteira em a, $y'(a) = \gamma$, usando a fórmula de diferenças progressiva, obtendo, então uma nova equação

$$\frac{y_2 - y_1}{h} = \gamma,$$

a qual, adicionada às $N - 1$ equações (15), permitirá determinar as N incógnitas $y_1, \ldots, y_N$.

Note-se, no entanto, que

$$\frac{y(x_2) - y(x_1)}{h} = y'(x_1) + \mathcal{O}(h),$$

ou seja, esta fórmula tem ordem de precisão inferior às fórmulas usadas nos restantes pontos.

Se introduzirmos a notação $x_0 = x_1 - h = a - h$, tem-se, usando a forma centrada de $2^{\underline{a}}$ ordem:

$$\frac{y(x_2) - y(x_0)}{2h} = y'(x_1) + \mathcal{O}(h^2).$$

Assim, para que seja usada uma fórmula na fronteira com a mesma ordem de precisão $(\mathcal{O}(h^2))$ das usadas nos restantes pontos, dever-se-á usar esta última fórmula.
Nesse caso, obter-se-á uma nova equação

$$y_2 - y_0 = 2h\gamma, \tag{16}$$

a qual envolve, no entanto, uma nova incógnita $y_0 \approx y(x_0)$.

Uma equação adicional para (juntamente com as $N-1$ equações (15) e a equação (16)) permitir determinar $y_0, y_1, \ldots, y_{N-1}$ pode ser obtida aproximando também a equação diferencial no ponto fronteiro $x_1 = a$, isto é, exigindo que

$$\frac{y_2 - 2y_1 + y_0}{h^2} + p(x_1)\frac{y_2 - y_0}{2h} + q(x_1)y_1 = r(x_1),$$

o que corresponde à equação

$$\underbrace{\left(1 - \frac{h}{2}p(x_1)\right)}_{a_1} y_0 + \underbrace{\left(-2 + h^2 q(x_1)\right)}_{d_1} y_1 + \underbrace{\left(1 + \frac{h}{2}p(x_1)\right)}_{c_1} y_2 = \underbrace{h^2 r(x_1)}_{b_1} \qquad (17)$$

Exemplo: Considere-se agora o seguinte PVF, análogo ao anterior, mas em que a condição de fronteira no ponto $x = 1$ envolve o valor de y':

$$\begin{cases} y''(x) + y'(x) - xy(x) = -xe^{2-x}, \quad x \in [1, 2], \\ y'(1) = -e, \\ y(2) = 1. \end{cases}$$

Determinemos a solução numérica para este problema, usando novamente o método das diferenças finitas de $2^{\underline{a}}$ ordem, com passo $h = 0.25$.

Temos as mesmas equações resultantes dos nós interiores

$$\begin{cases} 0.8750y_1 - 2.0781y_2 + 1.1250y_3 = -0.1654 \\ 0.8750y_2 - 2.0938y_3 + 1.1250y_4 = -0.1546 \\ 0.8750y_3 - 2.1094y_4 + 1.1250y_5 = -0.1404 \end{cases}.$$

Substituindo $y_5 = 1$ na última equação, obtém-se

$$0.875y_3 - 2.1094y_4 = -1.2654.$$

Da condição de fronteira em $x = 1$, vem

$$\frac{y_2 - y_0}{2h} = -2.7183 \iff y_2 - y_0 = -1.3591.$$

Além disso, a equação relativa a x_1 é:

$$0.8750y_0 - 2.0625y_1 + 1.1250y_2 = -0.1699.$$

Tem-se, então, o sistema $T\mathbf{y} = \mathbf{b}$, onde $\mathbf{y} = (y_0, y_1, \ldots, y_4)^{\mathrm{T}}$,

$$T = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 \\ 0.8750 & -2.0625 & 1.1250 & 0 & 0 \\ 0 & 0.8750 & -2.0781 & 1.1250 & 0 \\ 0 & 0 & 0.8750 & -2.0938 & 1.1250 \\ 0 & 0 & 0 & 0.8750 & -2.1094 \end{pmatrix}$$

e

$$\mathbf{b} = (-1.3591, -0.1699, -0.1654, -0.1546, -1.2654)^{\mathrm{T}}.$$

Resolvendo esse sistema, obtém-se

$$\mathbf{y} = (3.4651, 2.7011, 2.1060, 1.6423, 1.2811)^{\mathrm{T}}.$$

A solução analítica do problema é a mesma do problema anterior,ou seja, é $y(x) = e^{2-x}$, pelo que, nos pontos considerados, tem-se (com 4 c.d.):

$$\mathbf{y}_{\mathrm{exato}} = (3.4903, 2.7183, 2.1170, 1.6487, 1.2840)^{\mathrm{T}}.$$

Exemplo: [Resolução em Matlab]

```
>> p = @(x) 1; q = @(x) -x; r = @(x) -x.*exp(2-x);
>> h = 0.25; x = 1:h:2; N = length(x)-1;
>> gama= -exp(1); beta = 1;
>> a = zeros(size(x));
>> b=a;c=a;d=a;
>> for k = 1:N % Note-se que aqui k começa em 1
a(k) = 1-(h/2)*p(x(k));
d(k) = -2+h^2*q(x(k));
c(k) = 1+(h/2)*p(x(k));
b(k) = h^2*r(x(k));
end
```

```
>> av = a(2:N); % Notar que os indices vão de 2 a N
>> dv = d(1:N); % Notar que os indices vão de 1 a N
>> cv = c(1:N-1); % Notar que os indices vão de 1 a N-1
>> T = zeros(5);
>> ST = diag(dv)+diag(av,-1)+diag(cv,1);% Submatriz de T
>> T(2:end,2:end)=ST;
>> T(1,1) = 1; T(1,3) = -1; T(2,1) = a(1);
>> T
```

```
T =
1.0000    0         -1.0000    0          0
0.8750   -2.0625     1.1250    0          0
0         0.8750    -2.0781    1.1250     0
0         0          0.8750   -2.0938     1.1250
0         0          0         0.8750    -2.1094
```

```
>> b(N) = b(N)-c(N)*beta;

>> bv = b(1:N)';

>> bv = [2*h*gama;bv]

bv =
-1.3591
-0.1699
-0.1654
-0.1546
-1.2654

>> y=(T\bv)'

y =
3.4651    2.7012    2.1060    1.6423    1.2812

>> yanal = @(x) exp(2-x);

>> xExt = [1-h, x(1:end-1)];

>> yexato = yanal(xExt)

yexato =
3.4903    2.7183    2.1170    1.6487    1.2840
```