

Métodos Numéricos para EDOs

2 de abril de 2012

- 1** Introdução
- 2** Métodos de Euler e do Trapézio
- 3** Métodos de Runge-Kutta
- 4** Métodos de Passo Variável
- 5** Representação em Espaço de Estados
- 6** Estabilidade
- 7** EDOs do Tipo Stiff
- 8** Sumário

Equações Diferenciais Ordinárias

Queremos resolver numericamente a equação

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \quad t \geq t_0$$

Existência de Solução

Teorema

Suponha que f satisfaz a condição de Lipschitz

$$\|f(t, x) - f(t, y)\| \leq L\|x - y\|$$

em $D = \{(x, t) : \|x - x_0\| < b, |t - t_0| < a\}$ e que $\|f(t, x)\| \leq B$ em D . Então, a equação diferencial

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \quad t \geq t_0$$

possui uma única solução no intervalo $|t - t_0| < \min(a, b/B)$.

Exemplo 1 - O que pode dar errado?

$$\dot{x} = \sqrt{x}, \quad x(0) = 0$$

- \sqrt{x} não é Lipschitz em $x = 0$.
- Solução não é única: $x = \frac{t^2}{4}$ ou $x = 0$.

Exemplo 2 - O que pode dar errado?

$$\dot{x} = x^2, \quad x(0) = 1$$

- x^2 não é Lipschitz em $(-\infty, \infty)$.
- Solução não existe para todos os tempos: $x(t) = \frac{1}{1-t}$.
- $x \rightarrow \infty$ quando $t \rightarrow 1$

Método de Euler

Podemos reescrever a EDO

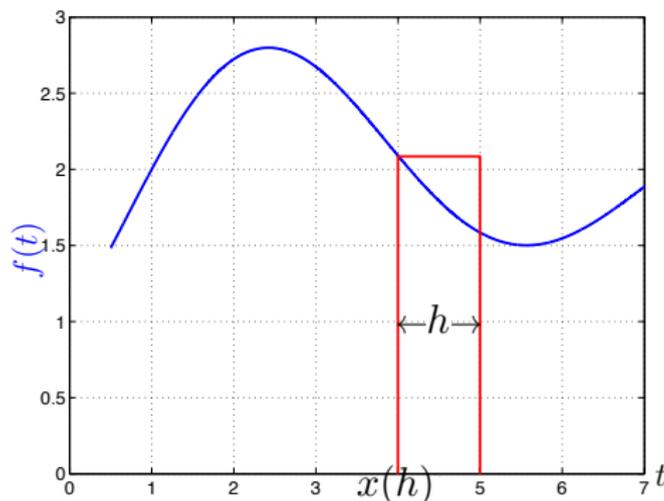
$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \quad t \geq t_0$$

como

$$x(t) = x_0 + \int_{t_0}^t f(s, x(s)) \, ds, \quad t \geq t_0$$

Método de Euler

Aproximamos a integral pela área do retângulo:



$$x(t+h) = x_0 + \int_{t_0}^t f(s, x(s)) ds + \int_t^{t+h} f(s, x(s)) ds \approx x(t) + hf(t, x(t))$$

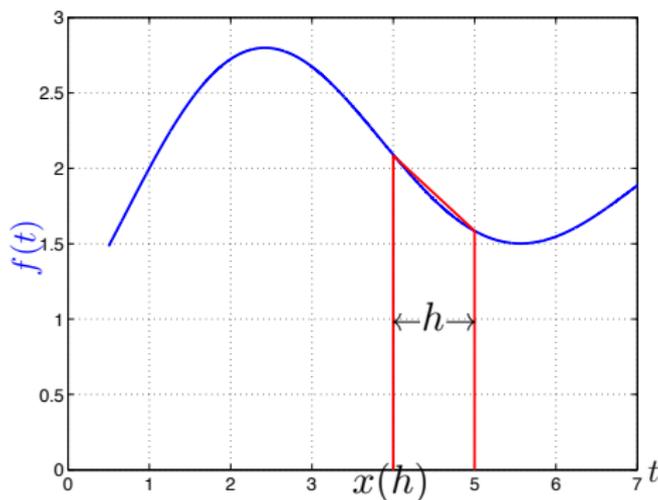
Método de Euler

Encrevendo $x(kh) = x_k$ e $t_k = kh$, obtemos a iteração correspondente ao método de Euler:

$$x_{k+1} = x_k + hf(t_k, x_k)$$

Método do Trapézio

Aproximamos a integral pela área do trapézio:



$$\int_t^{t+h} f(s, x(s)) ds \approx \frac{h}{2} [f(t, x(t)) + f(t+h, x(t+h))]$$

Método do Trapézio

Obtemos a seguinte iteração:

$$x_{k+1} = x_k + \frac{h}{2}[f(t_k, x_k) + f(t_{k+1}, x_{k+1})]$$

Note que x_{k+1} aparece em ambos os lados da equação. Por isso, precisamos resolver a equação não-linear implícita para obter x_{k+1} (o que exige calcular f em diversos pontos).

- O método de Euler é um exemplo de **método explícito**.
- O método do trapézio é um exemplo de **método implícito**.

Métodos de Runge-Kutta

Método clásico:

$$x_{k+1} = x_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_k, x_k)$$

$$k_2 = f(t_k + h/2, x_k + hk_1/2)$$

$$k_3 = f(t_k + h/2, x_k + hk_2/2)$$

$$k_4 = f(t_k + h, x_k + hk_3)$$

Erro de truncamento

Em geral, métodos explícitos podem ser escritos na forma

$$x_{k+1} = x_k + h\Phi(t_k, x_k; h)$$

Definimos o erro de truncamento

$$T_k = \frac{x(t_{k+1}) - x(t_k)}{h} - \Phi(t_k, x_k; h)$$

Erro de truncamento

$$T_k = \frac{x(t_{k+1}) - x(t_k)}{h} - \Phi(t_k, x_k; h)$$

Usando a série de Taylor:

$$x(t_{k+1}) = x(t_k) + hx'(t_k) + \frac{h^2}{2}x''(t_k) + \dots$$

obtemos

$$T_k = x'(t_k) + \frac{h}{2}x''(t_k) + \dots - \Phi(t_k, x_k; h)$$

Erro de truncamento

$$T_k = x'(t_k) + \frac{h}{2}x''(t_k) + \dots - \Phi(t_k, x_k; h)$$

Para o método de Euler, como $\Phi = x'$, temos que

$$T_k = \frac{h}{2}x''(t_k) + \frac{h^2}{6}x'''(t_k) \dots = O(h)$$

Erro Global

Se $T_k = O(h^p)$, então também o erro acumulado pelo método numérico é $O(h^p)$, isto é,

$$\|x(t_k) - x_k\| = O(h^p), \forall t_k \leq t_{final}$$

Neste caso, dizemos que o método de integração é de ordem p .

Métodos de Runge-Kutta

Em geral, métodos de Runge-Kutta são quaisquer métodos que podem ser escritos na forma:

$$x_{k+1} = x_k + h \sum_{i=1}^m \gamma_i k_i$$
$$k_i = f \left(t_k + \alpha_i h, x_k + h \sum_{j=1}^{i-1} \beta_j k_j \right), \quad i = 1, \dots, m$$

Note: o método acima requer m avaliações da função f .

Métodos de Runge-Kutta

- O método de Euler é de ordem 1 e tem $m = 1$.
- Para $p \leq 4$, um método de Runge-Kutta de ordem p requer $m = p$.
- Contudo, um método de Runge-Kutta de ordem 5 requer $m = 6$ (ou seja, o método torna-se menos vantajoso para $p > 4$).

Métodos de Passo Variável

- É possível que $x(t)$ varie muito rapidamente em certos intervalos de tempo de forma que é desejável que h seja pequeno.
- Por outro lado, pode haver intervalos de tempo em que $x(t)$ varie muito lentamente, sendo desejável que h seja grande.
- Tipicamente, desejamos que

$$erro_k = \|x_k - x(t_k)\| \leq TOL$$

- Como $erro_k \approx Ch^{p+1}$, o passo h ótimo será tal que

$$Ch_{OPT}^{p+1} \approx TOL$$

- Assim, obtemos

$$h_{OPT} = h \left(\frac{TOL}{erro_k} \right)^{1/p+1}$$

Método de Dormand-Prince (ode45)

- O método de Dormand-Prince é um método de Runge-Kutta de 4ª ordem com passo variável.
- Para ajustar o passo, usa-se um método de 5ª ordem para avaliar o erro de integração.

$$h_{k+1} = h_k \min \left(2, \max \left(0.5, 0.8 \left(\frac{TOL}{erro_k} \right)^{1/5} \right) \right)$$

$$erro_k = \|x_k^{\text{ordem } 4} - x_k^{\text{ordem } 5}\|$$

- Este método é implementado no Matlab sob o nome de ode45.

Equações Diferenciais de Ordem Superior

Até agora, vimos como resolver equações de 1ª ordem numericamente. Mas como resolver equações de ordem superior como a que descreve o movimento de um pêndulo?

$$\ddot{\theta} + \frac{g}{l} \text{sen}\theta = 0$$

Equações Diferenciais de Ordem Superior

Até agora, vimos como resolver equações de 1ª ordem numericamente. Mas como resolver equações de ordem superior como a que descreve o movimento de um pêndulo?

$$\ddot{\theta} + \frac{g}{l} \text{sen}\theta = 0$$

A chave está em converter a equação acima em uma equação de primeira ordem mas com múltiplas variáveis.

Equações Diferenciais de Ordem Superior

De fato, nossos métodos numéricos de integração permitem resolver

$$\dot{x} = f(t, x)$$

mesmo quando x é um vetor em \mathbb{R}^n .

Equações Diferenciais de Ordem Superior

De fato, nossos métodos numéricos de integração permitem resolver

$$\dot{x} = f(t, x)$$

mesmo quando x é um vetor em \mathbb{R}^n .

A expressão para o método de Euler, por exemplo, permanece a mesma para $x_k \in \mathbb{R}^n$:

$$x_{k+1} = x_k + hf(t_k, x_k).$$

Equações Diferenciais de Ordem Superior

Para o exemplo do pêndulo

$$\ddot{\theta} + \frac{g}{l} \text{sen}\theta = 0$$

definimos $x_1 = \theta$ e $x_2 = \dot{\theta}$. Dessa forma, temos

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{g}{l} \text{sen}x_1$$

Equações Diferenciais de Ordem Superior

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l}\text{sen}x_1\end{aligned}$$

Definindo

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ e } f(x) = \begin{bmatrix} x_2 \\ -\frac{g}{l}\text{sen}x_1 \end{bmatrix},$$

obtemos a EDO de primeira ordem

$$\dot{x} = f(x)$$

Representação em Espaço de Estados

$$\dot{x} = f(x)$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ e } f(x) = \begin{bmatrix} x_2 \\ -\frac{g}{l} \sin x_1 \end{bmatrix}$$

- Dizemos que a forma acima é uma **representação em espaço de estados** para a EDO do pêndulo.
- Dizemos que x_1 , o ângulo do pêndulo, e x_2 , a velocidade angular do pêndulo, são os estados do sistema dinâmico.

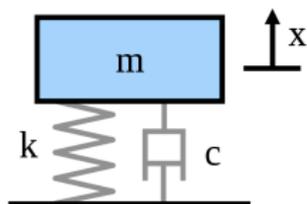
Representação em Espaço de Estados

$$\dot{x} = f(x)$$
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ e } f(x) = \begin{bmatrix} x_2 \\ -\frac{g}{l} \text{sen}x_1 \end{bmatrix}$$

- Dizemos que a forma acima é uma **representação em espaço de estados** para a EDO do pêndulo.
- Dizemos que x_1 , o ângulo do pêndulo, e x_2 , a velocidade angular do pêndulo, são os estados do sistema dinâmico.
- **Praticamente todo sistema dinâmico possui uma representação em espaço de estados (que não é única).**

Exercício

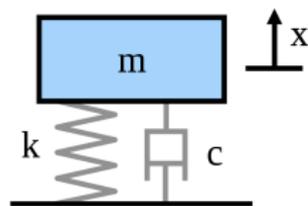
Obter a representação em espaço de estados para o sistema massa-mola-amortecedor.



$$F = m\ddot{x} = -c\dot{x} - kx$$

Exercício

Obter a representação em espaço de estados para o sistema massa-mola-amortecedor.



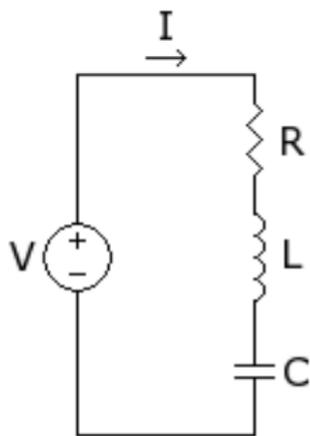
$$x_1 = x, \quad x_2 = \dot{x}$$

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -kx_1/m - cx_2/m \end{bmatrix}$$

$$F = m\ddot{x} = -c\dot{x} - kx$$

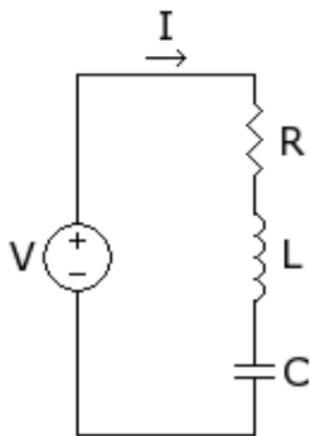
Exercício

Obter a representação em espaço de estados para a EDO que descreve o circuito RLC abaixo.



Exercício

Obter a representação em espaço de estados para a EDO que descreve o circuito RLC abaixo.

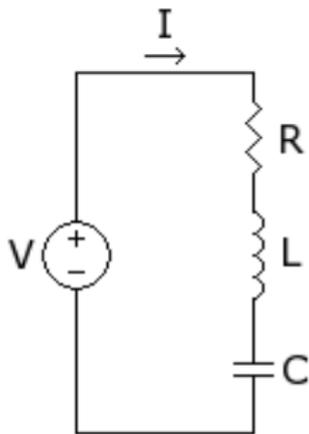


$$V = RI + L\dot{I} + V_C$$

$$I = C\dot{V}_C$$

Exercício

Obter a representação em espaço de estados para a EDO que descreve o circuito RLC abaixo.



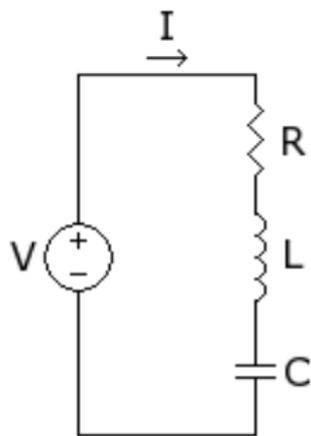
$$V = RI + L\dot{I} + V_C$$

$$I = C\dot{V}_C$$

$$\frac{d}{dt} \begin{bmatrix} V_C \\ I \end{bmatrix} = \begin{bmatrix} I/C \\ (V - V_C - RI)/L \end{bmatrix}$$

Exercício

Obter a representação em espaço de estados para a EDO que descreve o circuito RLC abaixo.



$$V = RI + L\dot{I} + V_C$$

$$I = C\dot{V}_C$$

$$\frac{d}{dt} \begin{bmatrix} V_C \\ I \end{bmatrix} = \begin{bmatrix} 0 & 1/C \\ -1/L & -R/L \end{bmatrix} \begin{bmatrix} V_C \\ I \end{bmatrix} + \begin{bmatrix} 0 \\ V/L \end{bmatrix}$$

Estabilidade

Considere o problema de resolver numericamente a equação

$$\dot{x} = \lambda x, \quad \lambda \in \mathbb{C}.$$

Sabemos que a solução exata é dada por

$$x(t) = x_0 e^{\lambda t}$$

e que $x(t) \rightarrow 0$ quando $t \rightarrow \infty$ se $\operatorname{Re}\{\lambda\} < 0$.

Estabilidade

$$\dot{x} = \lambda x, \quad \lambda \in \mathbb{C}.$$

Usando o método de Euler, obtemos a seguinte iteração

$$x_{k+1} = x_k + h\lambda x_k = (1 + \lambda h)x_k,$$

o que leva a

$$x_k = (1 + \lambda h)^k x_0.$$

Temos que $x_k \rightarrow 0$ quando $k \rightarrow \infty$ se $|1 + \lambda h| < 1$.

Estabilidade

- Dizemos que a integração numérica de $x(t)$ é estável se $x(t) \rightarrow 0$ implica $x_k \rightarrow 0$.
- Uma consequência prática da propriedade de estabilidade é que o efeito de erros numéricos diminui à medida que o tempo passa.

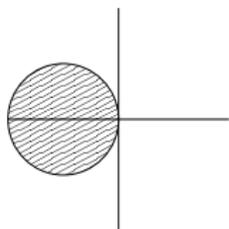
Estabilidade

- Dizemos que a integração numérica de $x(t)$ é estável se $x(t) \rightarrow 0$ implica $x_k \rightarrow 0$.
- Uma consequência prática da propriedade de estabilidade é que o efeito de erros numéricos diminui à medida que o tempo passa.
- Definimos a região de estabilidade como o conjunto de valores λ para os quais a integração é estável.
- No caso do método de Euler, a região de estabilidade é dada por

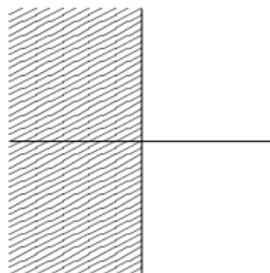
$$\{\lambda : |1 + \lambda h| < 1\}$$

que é caracterizada pelo interior do círculo de centro $-1/h$ e raio $1/h$.

Regiões de Estabilidade



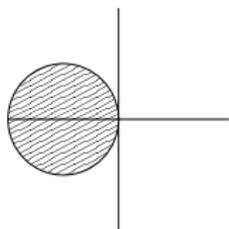
(a) Região de estabilidade para método de Euler



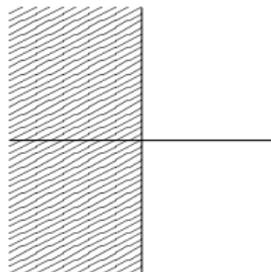
(b) Região de estabilidade para método do trapézio

- Para $|\lambda|$ grande, temos que fazer h muito pequeno para manter a estabilidade do método de Euler.
- Por outro lado, o método do trapézio é sempre estável.

Regiões de Estabilidade



(c) Região de estabilidade para método de Euler

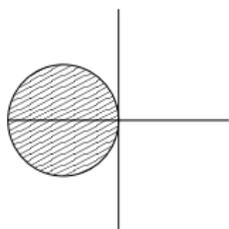


(d) Região de estabilidade para método do trapézio

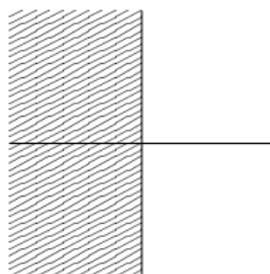
- Para $|\lambda|$ grande, temos que fazer h muito pequeno para manter a estabilidade do método de Euler.
- Por outro lado, o método do trapézio é sempre estável.

Em geral, métodos implícitos apresentam região de estabilidade muito maior que métodos explícitos.

Regiões de Estabilidade



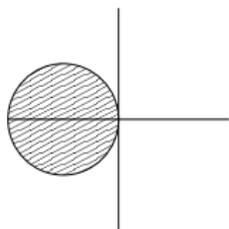
(e) Região de estabilidade para método de Euler



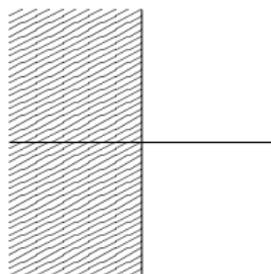
(f) Região de estabilidade para método do trapézio

- Se tivermos um processador muito rápido, qual o problema de fazer h muito pequeno e assim tornar o método de Euler estável?

Regiões de Estabilidade



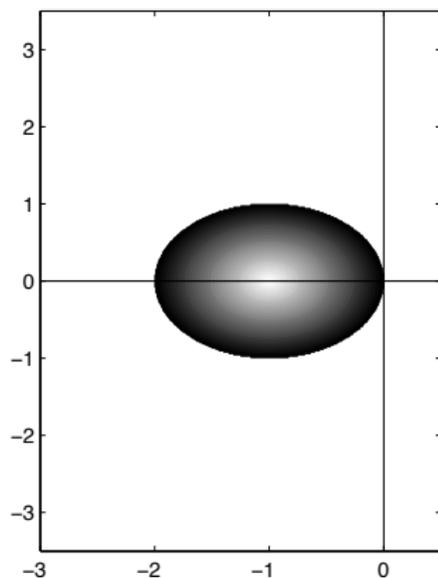
(g) Região de estabilidade para método de Euler



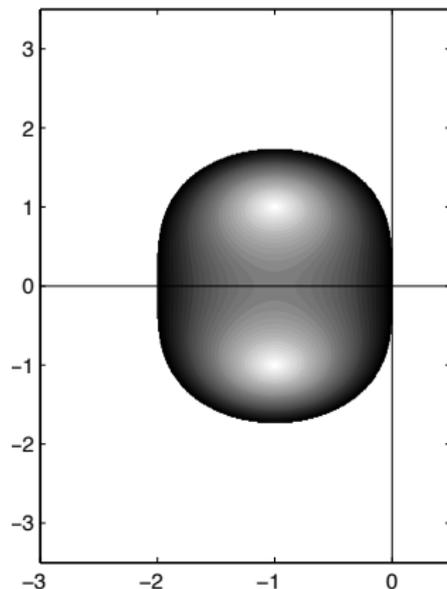
(h) Região de estabilidade para método do trapézio

- Se tivermos um processador muito rápido, qual o problema de fazer h muito pequeno e assim tornar o método de Euler estável?
- maiores erros (relativos) de arredondamento;
- mais erros de arredondamento são acumulados.

Regiões de Estabilidade

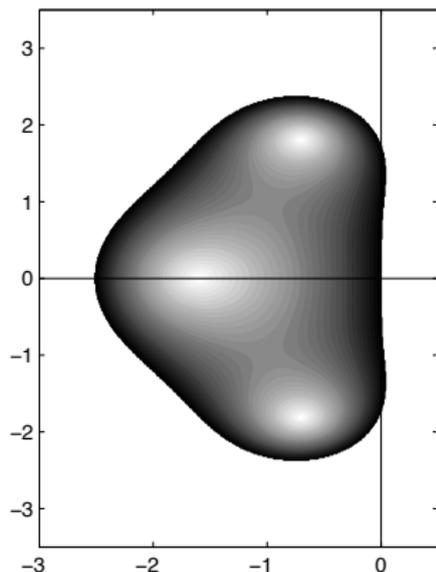


(i) Região de estabilidade para Runge-Kutta de ordem 1

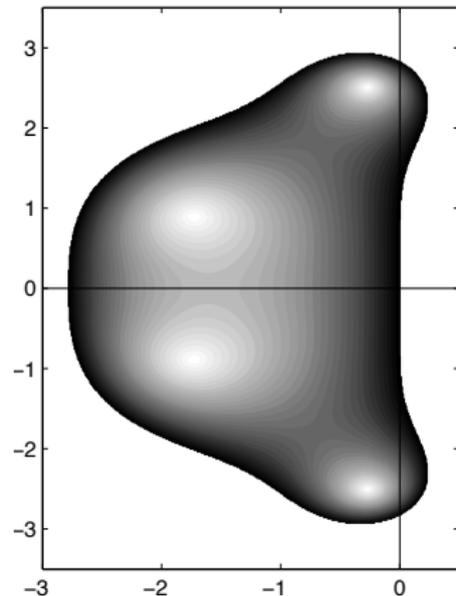


(j) Região de estabilidade para Runge-Kutta de ordem 2

Regiões de Estabilidade



(k) Região de estabilidade para Runge-Kutta de ordem 3



(l) Região de estabilidade para Runge-Kutta de ordem 4

EDOs do Tipo Stiff

Suponha que queiramos resolver a seguinte EDO numericamente:

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1000 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

A solução exata é dada por $x_1(t) = e^{-t}$ e $x_2(t) = e^{-1000t}$.

EDOs do Tipo Stiff

Se usarmos o método de Euler, a integração será estável se

$$|1 - h| < 1 \text{ e } |1 - 1000h| < 1.$$

- A primeira condição resulta em $h < 2$.
- Já a segunda resulta em $h < 0.002$.
- Para ter estabilidade, devemos escolher $h < 0.002$.

EDOs do Tipo Stiff

- Dizemos que uma EDO é **stiff** se ela apresenta modos com escalas de tempo separadas por diversas ordens de magnitude.
- Para esse tipo de EDO, é mais vantajoso usar métodos de integração implícitos pois
 - regiões de estabilidade são em geral muito maiores que a dos métodos explícitos
 - permitindo maiores passos de integração.
- **No Matlab: ode15s, ode23s**

Sumário

Método	Classificação	Acurácia	Uso
ode45	explícito	média	primeiro método a se tentar
ode23	explícito	baixa	maiores tolerâncias ou problemas moderadamente stiff
ode15s	implícito	baixa a média	se ode45 é lento devido a stiffness
ode23s	implícito	baixa	problemas stiff com altas tolerâncias. mais estável que ode15s
ode113	explícito (múltiplos passos)	alta	baixas tolerâncias ou problemas computacionalmente intensos