

Computação Científica II

Métodos numéricos para resolver equações diferenciais

2020-PLE

Método de Runge-Kutta

- O método clássico RK da ordem 4 ($e_k = \mathcal{O}(h^4)$) é

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Onde

$$k_1 = f(t_n, u_n) = f_n$$

$$k_2 = f\left(t_n + \frac{h}{2}, u_n + \frac{hk_1}{2}\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, u_n + \frac{hk_2}{2}\right)$$

$$k_4 = f(t_n + h, u_n + hk_3) \approx f_{n+1}$$

Método de Runge-Kutta

- O método clássico RK da ordem 4 ($e_k = O(h^4)$) é

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_n, u_n) = f_n$$

$$k_2 = f\left(t_n + \frac{h}{2}, u_n + \frac{hk_1}{2}\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, u_n + \frac{hk_2}{2}\right)$$

$$k_4 = f(t_n + h, u_n + hk_3)$$

- Em geral, um método RK de estágio-s (s-stage) é dado por

$$u_{n+1} = u_k + h \sum_{i=1}^s b_i k_i$$

Onde

$$k_1 = f(t_n, u_n)$$

$$k_2 = f(t_n + c_2 h, u_n + h a_{21} k_1)$$

...

$$k_s = f\left(t_n + c_s h, u_n + \sum_{j=1}^{s-1} h a_{sj} k_j\right)$$

Método de Runge-Kutta

- Em geral, um método RK de estágio-s (s-stage) é dado por

$$u_{n+1} = u_k + h \sum_{i=1}^s b_i k_i$$

Onde

$$k_1 = f(t_n, u_n)$$

$$k_2 = f(t_n + c_2 h, u_n + h a_{21} k_1)$$

...

$$k_s = f\left(t_n + c_s h, u_n + \sum_{j=1}^{s-1} h a_{sj} k_j\right)$$

- Podemos encapsular todos esses coeficientes no tableau do Butcher como

$$\begin{pmatrix} \mathbf{c} & A \\ & \mathbf{b}^T \end{pmatrix} = \begin{pmatrix} 0 & & & & & \\ c_2 & a_{21} & & & & \\ . & . & . & & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss-1} & \\ & b_1 & b_2 & \dots & b_{s-1} & b_s \end{pmatrix}$$

Método de Runge-Kutta

- Por exemplo, o metodo RK4 neste formulario será

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_n, u_n) = f_n$$

$$k_2 = f\left(t_n + \frac{h}{2}, u_n + \frac{hk_1}{2}\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, u_n + \frac{hk_2}{2}\right)$$

$$k_4 = f(t_n + h, u_n + hk_3)$$

$$\begin{pmatrix} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ & 1/6 & 1/3 & 1/3 & 1/6 \end{pmatrix}$$

Método de Runge-Kutta

- Ou o método com o seguinte quadro será

$$\begin{pmatrix} 0 & & \\ 1/2 & 1/2 & \\ & 0 & 1 \end{pmatrix}$$

$$u_{n+1} = u_n + h(0 \times k_1 + 1 \times k_2)$$

$$k_1 = f(t_n, u_n)$$

$$k_2 = f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_1\right)$$

$$\begin{pmatrix} 0 & & & & \\ c_2 & a_{21} & & & \\ \cdot & \cdot & \cdot & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss-1} \\ & b_1 & b_2 & \dots & b_{s-1} & b_s \end{pmatrix}$$

$$u_{n+1} = u_n + h \sum_{i=1}^s b_i k_i$$

$$k_1 = f(t_n, u_n)$$

$$k_s = f\left(t_n + c_s h, u_n + \sum_{j=1}^{s-1} h a_{sj} k_j\right)$$

Exemplo 1

-
- Resolve

$$\begin{cases} \frac{du_1}{dt} = 1.2u_1 - 0.6u_1u_2 \\ \frac{du_2}{dt} = 0.8u_2 + 0.3u_1u_2 \end{cases} \quad \begin{matrix} u_1(0) = 2 \\ u_2(0) = 1 \end{matrix} \quad T_f = 20$$

Com método RK4.

Métodos RK incorporados (embedded)

- Um método RK incorporados usa o mesmo \mathbf{c} e A , mas consiste em dois vetores \mathbf{b} .

$$\begin{pmatrix} \mathbf{c} & A \\ \mathbf{b}^T & \hat{\mathbf{b}}^T \end{pmatrix} = \begin{pmatrix} 0 & & & & & \\ c_2 & a_{21} & & & & \\ \cdot & \cdot & \cdot & & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss-1} & \\ & b_1 & b_2 & \dots & b_{s-1} & b_s \\ & \hat{b}_1 & \hat{b}_2 & \dots & \hat{b}_{s-1} & \hat{b}_s \end{pmatrix}$$

- O ponto é que o primeiro \mathbf{b} nos dá um método de ordem p para encontrar u_{n+1} .
- O segundo $\hat{\mathbf{b}}$ nos dá um método de ordem $p + 1$, para calcular \tilde{u}_{n+1} , que pode ser usado para estimar o erro local.

Método de Bogacki-Shampine de ordem 2/3

- Por exemplo

$$\begin{pmatrix} 0 & & & & \\ 1 & \frac{1}{2} & & & \\ \frac{3}{2} & \frac{1}{2} & & & \\ 3 & 0 & \frac{3}{4} & & \\ \frac{4}{3} & \frac{2}{9} & \frac{3}{9} & \frac{4}{9} & \\ 1 & \frac{2}{9} & \frac{3}{9} & \frac{4}{9} & 0 \\ & \frac{7}{24} & \frac{6}{24} & \frac{8}{24} & \frac{3}{24} \end{pmatrix}$$

- Será

$$k_1 = f(t_n, u_n)$$

$$k_2 = f\left(t_n + \frac{h}{2}, u_n + \frac{hk_1}{2}\right)$$

$$k_3 = f\left(t_n + \frac{3h}{4}, u_n + \frac{3hk_2}{4}\right)$$

$$k_4 = f\left(t_n + h, u_n + \frac{h}{9}(2k_1 + 3k_2 + 4k_3)\right)$$

$$u_{n+1} = u_n + \frac{h}{9}(2k_1 + 3k_2 + 4k_3 + 0)$$

$$\tilde{u}_{n+1} = u_n + \frac{h}{24}(7k_1 + 6k_2 + 8k_3 + 3k_4)$$

Métodos RK incorporados (embedded)

- Um método RK incorporados usa o mesmo \mathbf{c} e A , mas consiste em dois vetores \mathbf{b} .
- O ponto é que o primeiro \mathbf{b} nos dá um método de ordem p (global) para encontrar u_{n+1} .
- O segundo $\hat{\mathbf{b}}$ nos dá um método de ordem $p + 1$, para calcular \tilde{u}_{n+1} , que pode ser usado para estimar o erro local.
- Podemos usar a diferença entre u_{n+1} e \tilde{u}_{n+1} para estimar o erro de truncamento. Se assumirmos que o \tilde{u}_{n+1} é muito mais preciso que u_{n+1} , então o erro local é

$$l(t_k, h) \approx \delta = |\tilde{u}_{n+1} - u_{n+1}|$$

Controle de tamanho de passo adaptável

- Se assumirmos que o \tilde{u}_{n+1} é muito mais preciso que u_{n+1} , então o erro local é

$$l(t_k, h) \approx \delta = |\tilde{u}_{n+1} - u_{n+1}|$$

- Agora considere o seguinte algoritmo, para ir de u_n a u_{n+1} com tamanho de passo h .
 1. Resolva usando ambos os métodos (ordem 2 e 3), e calcule \tilde{u}_{n+1} , u_{n+1} e δ .
 2. Se $\delta \leq tol$, aceite u_{n+1} como a solução no próximo passo de tempo. Podemos também aumentar o tamanho de passo na próxima etapa como $h_{nex} = h \left(\frac{tol}{\delta} \right)^{\frac{1}{p+1}}$.
 3. Caso contrario, escolha um novo tamanho de passo como $h_{new} = \frac{h}{2}$. Em seguida, recalcule novamente \tilde{u}_{n+1} , u_{n+1} e δ para a etapa atual e retorne à segunda etapa.

Exemplo 2

Resolva a seguinte equação usando o método adaptativo.

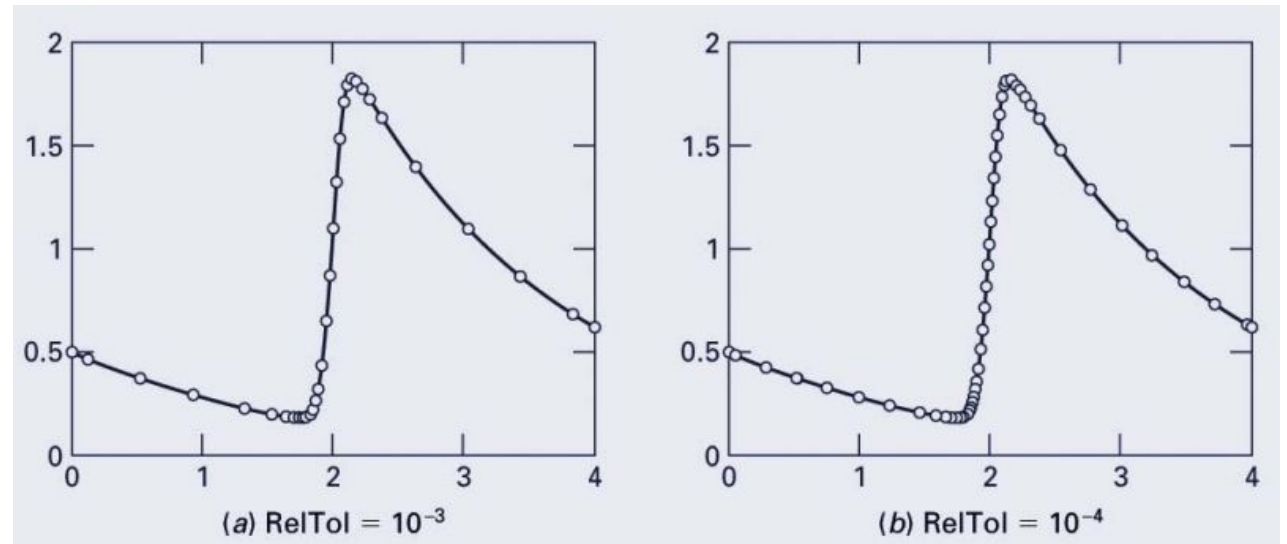
$$\frac{du}{dt} = e^u - u e^t$$
$$u(0) = 0$$

Exercício 1

-
- Resolve

$$\frac{du}{dt} = 10e^{-\frac{(t-2)^2}{2(0.075)^2}} - 0.6u$$

Com método RK2/3 adaptativo. Compare a solução para $tol = 10^{-3}$ e 10^{-4} .



Métodos de múltiplos passos

- Até agora, em todos os métodos que estudamos, a única informação necessária era (t_n, u_n) . Nós chamamos esses métodos de **passo único**.
- Existem métodos em que precisamos de mais informações da solução no passado. Por exemplo

$$u_{n+1} = u_{n-1} + 2h \cdot f(t_n, u_n)$$

- Esse método é chamado “Leap-frog”. Podemos obtê-lo usando a aproximação da diferença central.
- Isso é chamado de método de duas passos, porque precisamos u_n e u_{n-1} para encontrar u_{n+1} .
- Preste atenção que não podemos iniciar este método para $n = 1$. para este ponto, geralmente usamos métodos de passo único, como Euler avançado.

Métodos de múltiplos passos

- A forma geral de um método de p-passo linear é

$$u_{n+1} = \alpha_1 u_n + \cdots + \alpha_p u_{n-p+1} + h(\beta_0 f_{n+1} + \beta_1 f_n + \cdots + \beta_p f_{n-p+1})$$

- Aqui se $\beta_0 = 0$, o método é explícito. Caso contrario, é implícito.
- Para iniciar este método, além de u_0 , precisamos u_1, \dots, u_{p-1} .
- Esses valores podem ser calculados com um método de passo único, como RK4.
- Alguns métodos clássicos de múltiplos passos são
 - Adams-Bashforth (explícito)
 - Adams-Moulton (implícito)
 - Backward Differentiation Formulas (BDF-p)

Métodos de Adams-Bashforth

- A forma geral do método de Adams-Bashforth é

$$u_{n+1} = u_n + h \left(f_n + \frac{1}{2} \nabla f_n + \frac{5}{12} \nabla^2 f_n + \dots \right)$$

$$\nabla f_n = f_n - f_{n-1}$$

$$\nabla^2 f_n = f_n - 2f_{n-1} + f_{n-2}$$

$$\nabla^3 f_n = f_n - 3f_{n-1} + 3f_{n-2} - f_{n-3}$$

- Por exemplo

$$u_{n+1} = u_n + \frac{h}{2} [3f_n - f_{n-1}]$$

$$u_{n+1} = u_n + \frac{h}{12} [23f_n - 16f_{n-1} + 5f_{n-2}]$$

Métodos de Adams-Bashforth

- A forma geral do método de Adams-Bashforth é

$$u_{n+1} = u_n + h \left(f_n + \frac{1}{2} \nabla f_n + \frac{5}{12} \nabla^2 f_n + \dots \right)$$

$$\nabla f_n = f_n - f_{n-1}$$

$$\nabla^2 f_n = f_n - 2f_{n-1} + f_{n-2}$$

$$\nabla^3 f_n = f_n - 3f_{n-1} + 3f_{n-2} - f_{n-3}$$

- Em geral, um p-passo método de Adams-Bashforth tem ordem p .
- A forma geral do método de Adams-Moulton é

$$u_{n+1} = u_n + h \left(f_{n+1} - \frac{1}{2} \nabla f_{n+1} - \frac{1}{12} \nabla^2 f_{n+1} + \dots \right)$$

Métodos de Adams-Moulton

- A forma geral do método de Adams-Moulton é

$$u_{n+1} = u_n + h \left(f_{n+1} - \frac{1}{2} \nabla f_{n+1} - \frac{1}{12} \nabla^2 f_{n+1} + \cdots \right)$$

$$\nabla f_n = f_n - f_{n-1}$$

$$\nabla^2 f_n = f_n - 2f_{n-1} + f_{n-2}$$

$$\nabla^3 f_n = f_n - 3f_{n-1} + 3f_{n-2} - f_{n-3}$$

- Por exemplo

$$u_{n+1} = u_n + \frac{h}{12} [5f_{n+1} + 8f_n - f_{n-1}]$$

Em geral, um p-passo método de Adams-Moulton tem ordem $p + 1$.

Métodos BDF

- O método de Adams-Bashforth $u_{n+1} = u_n + h \left(f_n + \frac{1}{2} \nabla f_n + \frac{5}{12} \nabla^2 f_n + \dots \right)$
- O método de Adams-Moulton

$$u_{n+1} = u_n + h \left(f_{n+1} - \frac{1}{2} \nabla f_{n+1} - \frac{1}{12} \nabla^2 f_{n+1} + \dots \right)$$

- A forma geral do método BDF é

$$\nabla u_{n+1} + \frac{1}{2} \nabla^2 u_{n+1} + \dots + \frac{1}{p} \nabla^p u_{n+1} = h f_{n+1}$$

$$\nabla f_n = f_n - f_{n-1}$$

$$\nabla^2 f_n = f_n - 2f_{n-1} + f_{n-2}$$

$$\nabla^3 f_n = f_n - 3f_{n-1} + 3f_{n-2} - f_{n-3}$$

Métodos BDF

- O método de Adams-Bashforth

$$u_{n+1} = u_n + h \left(f_n + \frac{1}{2} \nabla f_n + \frac{5}{12} \nabla^2 f_n + \dots \right)$$

- O método de Adams-Moulton

$$u_{n+1} = u_n + h \left(f_{n+1} - \frac{1}{2} \nabla f_{n+1} - \frac{1}{12} \nabla^2 f_{n+1} + \dots \right)$$

- A forma geral do método BDF é

$$\nabla u_{n+1} + \frac{1}{2} \nabla^2 u_{n+1} + \dots + \frac{1}{p} \nabla^p u_{n+1} = h f_{n+1}$$

- Esse método é implícito e adequado para os problemas rígidos.

$$u_{n+1} = \frac{4}{3} u_n - \frac{1}{3} u_{n-1} + \frac{2}{3} h f_{n+1}$$

Métodos de múltiplos passos

-

- Adams-Bashforth

$$u_{n+1} = u_n + h \left(f_n + \frac{1}{2} \nabla f_n + \frac{5}{12} \nabla^2 f_n + \dots \right)$$

- Adams-Moulton

$$u_{n+1} = u_n + h \left(f_{n+1} - \frac{1}{2} \nabla f_{n+1} - \frac{1}{12} \nabla^2 f_{n+1} + \dots \right)$$

- BDF

$$\nabla u_{n+1} + \frac{1}{2} \nabla^2 u_{n+1} + \dots + \frac{1}{p} \nabla^p u_{n+1} = h f_{n+1}$$

- O erro local desses métodos é

$$l(t_k, h) = C \frac{d^{p+1}u}{dt} (t_k) h^{p+1}$$

- Podemos estimar a derivada usando alguma aproximação de diferença.
- Portanto, o algoritmo de tamanho de passo adaptável pode ser aplicado aqui também.

Estabilidade de métodos de múltiplos passos

- Vamos considerar a estabilidade do método “Leap-frog” para nosso problema de teste.

$$\frac{du}{dt} = \lambda u \quad u(0) = 1$$
$$u_{n+1} = u_{n-1} + 2h\lambda u_n$$

- Geralmente, para métodos de múltiplos passos, formamos uma equação característica.

$$\mu^2 = 1 + 2h\lambda\mu$$

- Considere μ_0 e μ_1 como raízes dessa equação. Então observe que μ_0^n e μ_1^n são a solução da principal diferença de equação também.

Estabilidade de métodos de múltiplos passos

- $u_{n+1} = u_{n-1} + 2h\lambda u_n \rightarrow \mu^2 = 1 + 2h\lambda\mu$
- Considere μ_0 e μ_1 como raízes dessa equação. Então observe que μ_0^n e μ_1^n são a solução da principal diferença de equação também.
- Porque a equação original é linear, qualquer combinação linear dessas raízes é também a solução dela.

$$u_n = \alpha_0 \mu_0^n + \alpha_1 \mu_1^n$$

- Os coeficientes α_0 e α_1 podem ser encontrados usando u_0 e u_1 . Para o nosso exemplo, temos $u_0 = 1$. Portanto

$$u_0 = 1 = \alpha_0 + \alpha_1 \rightarrow \alpha_1 = -\alpha_0$$

Estabilidade de métodos de múltiplos passos

- $$u_{n+1} = u_{n-1} + 2h\lambda u_n \quad \rightarrow \quad \mu^2 = 1 + 2h\lambda\mu$$

$$u_n = \alpha_0 \mu_0^n + \alpha_1 \mu_1^n = \alpha(\mu_0^n - \mu_1^n)$$

- Como mencionei, podemos usar outros métodos de passo único para encontrar as etapas de tempo iniciais de que precisamos. Por exemplo, podemos usar o método de Euler avançado.

$$u_1 = \alpha(\mu_0 - \mu_1) = 1 + h\lambda \quad \rightarrow \quad \alpha = \frac{1 + h\lambda}{\mu_0 - \mu_1}$$

- Para este caso, as raízes são

$$\mu_0 = h\lambda - \sqrt{h^2\lambda^2 + 1} \qquad \mu_1 = h\lambda + \sqrt{h^2\lambda^2 + 1}$$

Estabilidade de métodos de múltiplos passos

- Finalmente, a solução no passo de tempo n será

$$u_n = \frac{1 + h\lambda}{-2\sqrt{h^2\lambda^2 + 1}} \left(\left[h\lambda - \sqrt{h^2\lambda^2 + 1} \right]^n - \left[h\lambda + \sqrt{h^2\lambda^2 + 1} \right]^n \right)$$

- Agora, se o problema principal é analiticamente estável ($Re(\lambda) \leq 0$), então, para ter um método numericamente estável, precisamos ter

$$\lim_{n \rightarrow \infty} u_n = 0$$

- Se pudéssemos encontrar um h_{max} , em que para qualquer $h \leq h_{max}$ essa condição é válida, o método é numericamente estável.

Encontre h_{max} para o método Leapfrog

Estabilidade de métodos de múltiplos passos

- Se a equação característica tem múltiplas raízes com multiplicidade de m , então a solução consistirá de

$$\mu_j^n, n\mu_j^n, n^2\mu_j^n, \dots, n^{m-1}\mu_j^n$$

- Como você viu, verificar a condição de estabilidade é muito difícil. Portanto, estamos procurando condições mais simples, porém mais restritivas, para impor a estabilidade.
- Por exemplo, considere que todas as raízes são simples e $|\mu_j| < 1$. É óbvio que tal condição restritiva reforçará

$$\lim_{n \rightarrow \infty} u_n = 0$$