Capítulo 2 - Problemas de Valor Inicial para Equações Diferenciais Ordinárias

Carlos Balsa

balsa@ipb.pt

Departamento de Matemática
Escola Superior de Tecnologia e Gestão de Bragança

Matemática Aplicada - Mestrados Eng. Química e Industrial

INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e de Gestão

Outline

Equações Diferenciais

- Equações diferenciais envolvem derivadas de uma função desconhecida
- *Equação Diferencial Ordinária*(EDO): todas as derivadas são relativas a uma única variável independente, por vezes representando o tempo
- Solução numérica de equações diferencias é baseada numa aproximação de dimensão finita
- Equação Diferencial é substituída por uma equação algébrica cuja solução aproxima a solução da equação diferencial

Ordem de uma EDO

- Ordem de uma EDO é determinada pela ordem da mais alta derivada da função solução que ocorre na EDO
- Exemplos

  $y'' + 3y' + 6y = \sin(t)$ é de ordem 2
  $y'' + 3yy' = e^t$ é de ordem 2
  $(y')^3 + 6y = -1$ é de ordem 1
- EDO de ordem superior pode ser transformada num sistema equivalente de equações de primeira ordem
- Analisaremos apenas métodos numéricos para EDOs de primeira ordem
- Grande parte do software para EDOs foi desenhado apenas para a resolução de EDOs de primeira ordem

EDOs de Ordem Superior

- Para uma EDO de ordem k

$$y^{(k)}(t) = f(t, y, y', \dots, y^{(k-1)})$$

define-se k novas funções incógnitas

$$u_1(t) = y(t), \;\; u_2(t) = y'(t), \;\; \dots \;\;, \;\; u_k(t) = y^{(k-1)}(t)$$

- A EDO original é então equivalente ao sistema de primeira ordem

$$\begin{bmatrix} u_1'(t) \\ u_2'(t) \\ \vdots \\ u_{k-1}'(t) \\ u_k'(t) \end{bmatrix} = \begin{bmatrix} u_2(t) \\ u_3(t) \\ \vdots \\ u_k(t) \\ f(t, u_1, u_2, \dots, u_k) \end{bmatrix}$$

Exemplo: Segunda lei de Newton

- A lei do movimento, $F = ma$, é uma EDO de 2ª ordem, pois a aceleração a é a segunda derivada da posição, que designamos por y
- Assim, a EDO tem a forma

$$y'' = F/m$$

- Definindo $u_1 = y$ e $u_2 = y'$ obtemos o sistema equivalente de duas EDO de primeira ordem

$$\begin{bmatrix} u_1' \\ u_2' \end{bmatrix} = \begin{bmatrix} u_2 \\ F/m \end{bmatrix}$$

Exemplo (continuação)

- Podemos agora usar métodos para equações de primeira ordem para resolver este sistema
- A primeira componente da solução u_1 é a solução y da equação de 2ª ordem original
- A segunda componente da solução u_2 é a velocidade y'

Exercício

Transforme a seguinte EDO de 3ª ordem num sistema equivalente de equações de primeira ordem

$$2y''' - y'' + 5y = 0$$

Solução:

$$\begin{bmatrix} u_1' \\ u_2' \\ u_3' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{5}{2} & 0 & \frac{1}{2} \end{bmatrix} . \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \Leftrightarrow \mathbf{u}' = A\mathbf{u}$$

Equações Diferenciais Ordinárias

- Sistemas genéricos de EDOs de primeira ordem têm a seguinte forma
$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$$
em que $\mathbf{y} : \mathbb{R} \to \mathbb{R}^n$, $\mathbf{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, e $\mathbf{y}' = d\mathbf{y}/dt$ designa a derivada relativa a t,
$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \\ \vdots \\ y_n'(t) \end{bmatrix} = \begin{bmatrix} dy_1(t)/dt \\ dy_2(t)/dt \\ \vdots \\ dy_n(t)/dt \end{bmatrix}$$
- Função $\mathbf{f}$ é dada e queremos determinar a função incógnita $\mathbf{y}$ que verifica a EDO
- Para simplificar, vamos apenas considerar casos especiais de uma única equação (ODE escalar)

Problemas de Valor Inicial

- Por si só a EDO $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ não determina uma função solução única
- Isto porque a EDO apenas especifica o declive $\mathbf{y}'(t)$ da função solução em cada ponto, mas não especifica o valor de $\mathbf{y}(t)$ para algum ponto
- Em geral, existe uma infinidade de funções que satisfazem a ODE.
- Para obter uma solução particular, o valor $\mathbf{y}_0$ da função solução tem de ser conhecido para algum ponto t_0

Problemas de Valor Inicial (continuação)

- É necessário que os dados do problema indiquem $\mathbf{y}(t_0) = \mathbf{y}_0$, o que determina a solução única da EDO
- Se considerarmos a variável independente t como o tempo, podemos pensar em t_0 como o tempo inicial e em $\mathbf{y}_0$ como o valor inicial da função incógnita
- Por isso, é designado por *Problema de Valor Inicial*, ou *PVI*
- A EDO governa a evolução do sistema ao longo do tempo desde o seu estado inicial $\mathbf{y}_0$ no tempo t_0, e nós procuramos uma função $\mathbf{y}(t)$ que descreve o estado do sistema em função do tempo

Exemplo: Problema de Valor Inicial

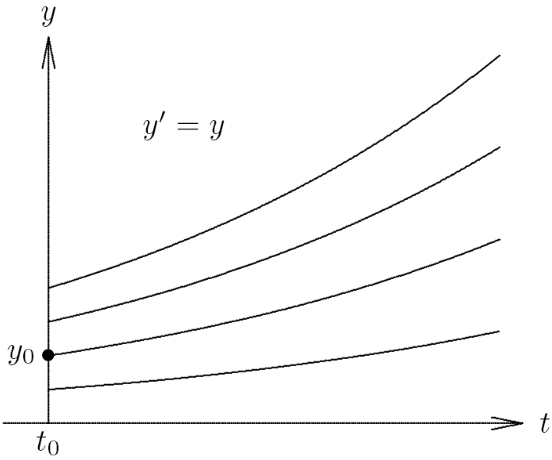
- Considere a seguinte EDO escalar

$$y' = y$$

- O conjunto das soluções tem a forma geral $y = ce^t$, em que c é uma constante real qualquer
- Impondo a condição inicial $y(t_0) = y_0$ permite obter a solução única correspondente a este caso particular
- Para este exemplo, se $t_0 = 0$, então $c = y_0$, significando que a solução é $y(t) = y_0 e^t$

Exemplo: Problema de Valor Inicial

Família das soluções para a EDO $y' = y$



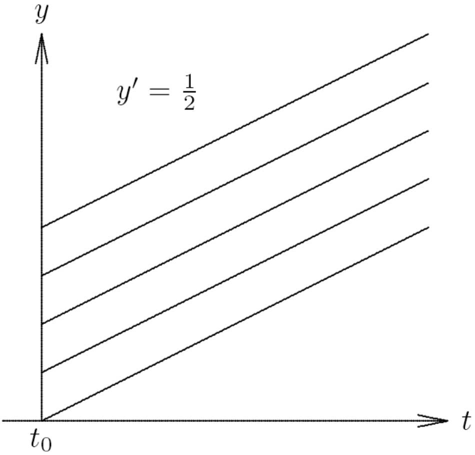
Estabilidade das Soluções

A solução de uma EDO é

- *Estável* se soluções resultantes da perturbação do valor inicial se mantiverem próximas da solução original
- *Assimptoticamente estável* se soluções resultantes da perturbação do valor inicial convergem para a solução original
- *Instável* se soluções resultantes da perturbação do valor inicial divergem da solução original sem limites

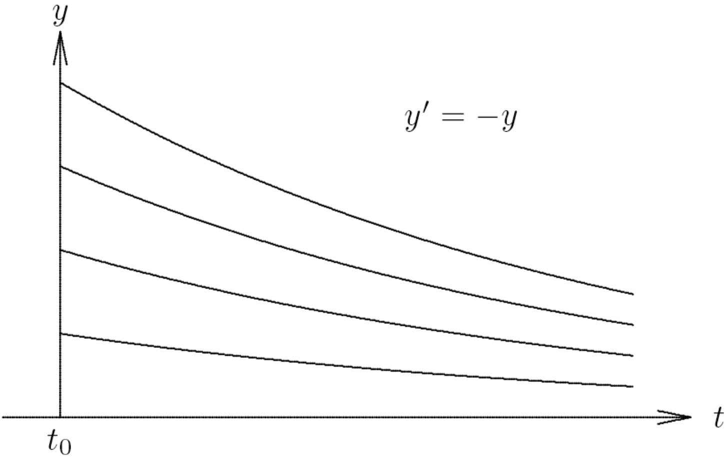
Exemplo: Soluções Estáveis

Família das soluções para a EDO $y' = \frac{1}{2}$



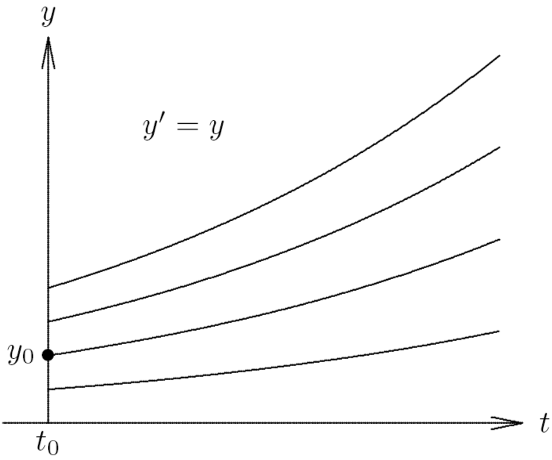
Exemplo: Soluções Assimptoticament Estáveis

Família das soluções para a EDO $y' = -y$



Exemplo: Soluções Instáveis

Família das soluções para a EDO $y' = y$



Exemplo: Estabilidade das Soluções

- Considere uma EDO escalar $y' = \lambda y$, com λ constante
- Considerando $t_0 = 0$ o tempo inicial e $y(0) = y_0$ o valor inicial, a solução é dada por $y(t) = y_0 e^{\lambda t}$
- Para λ real
  - $\lambda > 0$: todas as soluções não nulas crescem exponencialmente, logo cada solução é instável
  - $\lambda < 0$: todas as soluções não nulas decaem exponencialmente, logo cada solução para além de ser estável é também assimptoticamente estável
- Para λ complexo
  - Re$(\lambda) > 0$: instável
  - Re$(\lambda) < 0$: assimptoticamente estável
  - Re$(\lambda) = 0$: estável mas não assimptoticamente estável

Exemplo: Sistema linear de EDOs

- Um sistema linear de EDOs homogéneo com coeficientes constantes tem a forma
$$\mathbf{y}' = \mathbf{A}\mathbf{y}$$
em que $\mathbf{A}$ é uma matriz $n \times n$ e a condição inicial é $\mathbf{y}(0) = \mathbf{y}_0$
- Supondo que $\mathbf{A}$ é diagonizável com valores próprios λ_i e correspondentes vectores próprios $\mathbf{v}_i$, $i = 1, 2, \ldots, n$
- Exprimindo $\mathbf{y}_0$ como combinação linear dos $\mathbf{v}_i$: $\mathbf{y}_0 = \sum_{i=1}^{n} \alpha_i v_i$
- Então
$$\mathbf{y}(t) = \sum_{i=1}^{n} \alpha_i v_i e^{\lambda_i t}$$
é a solução da EDO que satisfaz a condição inicial $\mathbf{y}(0) = \mathbf{y}_0$

Exemplo: Sistema linear de EDOs (continuação)

- Valores próprios de **A** com parte real positiva conduzem a um crescimento exponencial dos componentes da solução
- Valores próprios de **A** com parte real negativa conduzem a um decrescimento exponencial dos componentes da solução
- Valores próprios de **A** com parte real nula (i.e. imaginários puros) conduzem a oscilações dos componentes da solução
- A solução é
  - Estável se Re(λ_i)$\leq$ 0 para todos os valores próprios
  - Assimptoticamente estável se Re(λ_i)$<$ 0 para todos os valores próprios
  - Instável se Re(λ_i)$>$ 0 para pelo menos um dos valores próprios

Exercício

Determine a solução e analise a estabilidade do seguinte problema de valor inicial

$$\begin{cases} \mathbf{y}' = \mathbf{A}\mathbf{y} \\ \mathbf{y}_0 = [1 \;\; 0 \;\; 0]^T \end{cases} \quad \text{com} \quad \mathbf{A} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Solução geral é

$$\mathbf{y}(t) = \alpha_1 v_1 e^{\lambda_1 t} + \alpha_2 v_2 e^{\lambda_2 t} + \alpha_3 v_3 e^{\lambda_3 t}$$

com $\lambda_1, \lambda_2, \lambda_3$ os valores próprios de **A** e $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ os vectores próprios de **A**

Resolução:

1. Calcular os valores próprios: $|\lambda I - \mathbf{A}| = 0$
2. Calcular os vectores próprios: **v** a equação $(\mathbf{A} - \lambda I)\,\mathbf{v} = 0$

Exercício, resolução

1. Valores próprios:

$$
\begin{aligned}
&|\lambda I - \mathbf{A}| = 0 \\
\Leftrightarrow &\lambda^3 - 2\lambda^2 + \lambda - 2 = 0 \\
\Leftrightarrow &\lambda = 2 \vee \lambda = -i \vee \lambda = i
\end{aligned}
$$

$\longrightarrow$ Sistema instável, existe um valor próprio real positivo

2. Vectores próprios:

$$
\begin{aligned}
(\mathbf{A} - 2I)\,\mathbf{v_1} = 0 \Leftrightarrow \mathbf{v_1} = [0 \;\; 0 \;\; 1]^T \\
(\mathbf{A} + iI)\,\mathbf{v_2} = 0 \Leftrightarrow \mathbf{v_2} = [1 \;\; i \;\; 0]^T \\
(\mathbf{A} - iI)\,\mathbf{v_3} = 0 \Leftrightarrow \mathbf{v_3} = [1 \;\; -i \;\; 0]^T
\end{aligned}
$$

Resolução (continuação)

Solução geral:

$$\mathbf{y}(t) = \alpha_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} e^{2t} + \alpha_2 \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} e^{-it} + \alpha_3 \begin{bmatrix} 1 \\ -i \\ 0 \end{bmatrix} e^{it} \Leftrightarrow \begin{cases} y_1(t) = \alpha_2 e^{-it} + \alpha_3 e^{it} \\ y_2(t) = \alpha_2 i e^{-it} - \alpha_3 i e^{it} \\ y_3(t) = \alpha_1 e^{2t} \end{cases}$$

Solução original para $\mathbf{y}(0) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$

$$\begin{cases} y_1(t) = \frac{1}{2}\left(e^{it} + e^{-it}\right) \\ y_2(t) = -\frac{i}{2}\left(e^{it} - e^{-it}\right) \\ y_3(t) = 0 \end{cases} \Leftrightarrow \begin{cases} y_1(t) = \cos(t) \\ y_2(t) = \sin(t) \\ y_3(t) = 0 \end{cases}$$

Solução Numérica de EDOs

- Solução analítica de EDO é uma função bem definida que pode ser avaliada para qualquer valor de t
- Solução numérica de EDO é uma tabela de valores aproximados da função solução para um conjunto discretos de pontos
- Começando em t_0 com o valor dado y_0, seguimos a função ditada pela EDO
- Calculo de $\mathbf{f}(t_0, \mathbf{y}_0)$ indica o declive da trajectória nesse ponto
- Usamos esta informação para prever o valor $\mathbf{y}_1$ da solução no tempo futuro $t_1 = t_0 + h$ para um determinado incremento de tempo h

Solução Numérica de EDOs, continuação

- Valores aproximados da solução são gerados passo a passo em incrementos dentro do intervalo no qual procuramos a solução
- Ao deslocar-nos de um ponto discreto para o outro, cometemos um erro, significando que o valor da próxima solução aproximada pertence a uma outra solução diferente daquela de onde partimos
- Estabilidade ou instabilidade da solução determina, em parte, se tais erros são ampliados ou reduzidos com o tempo

Método de Euler

- Sistema genérico de EDOs $\mathbf{f}(t, \mathbf{y})$, consideramos a série de Taylor

$$\begin{aligned} \mathbf{y}(t+h) &= \mathbf{y}(t) + h\mathbf{y}'(t) + \frac{h^2}{2}\mathbf{y}''(t) + \dots \\ &= \mathbf{y}(t) + h\mathbf{f}(t, \mathbf{y}(t)) + \frac{h^2}{2}\mathbf{y}''(t) + \dots \end{aligned}$$

- Método de Euler consite em eliminar os termos de ordem maior ou igual a dois

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h_k \mathbf{f}(t_k, \mathbf{y}_k)$$

- Método de Euler prevê solução através da extrapolação ao longo de uma linha recta cujo declive é $\mathbf{f}(t_k, \mathbf{y}_k)$
- Método de Euler é de passo-simples porque depende apenas da informação num único ponto do tempo para avançar para o próximo

Exemplo: Método de Euler

- Aplicando o método de Euler à EDO $y' = y$ com um passo h, prevemos a solução no tempo $t_1 = t_0 + h$ a partir da solução em t_0

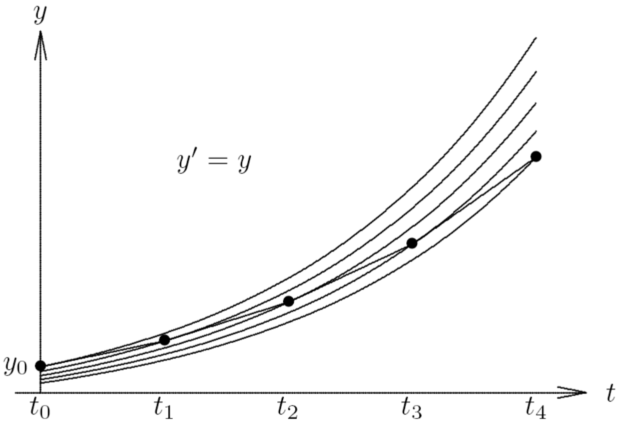
$$y_1 = y_0 + hy_0' = y_0 + hy_0 = (1 + h)\, y_0$$

- O valor da solução obtido em t_1 não é exacto, $y_1 \neq y(t_1)$
- Por exemplo, se $t_0 = 0$, $y_0 = 1$ e $h = 0.5$, então $y_1 = 1.5$, enquanto que a solução exacta para esta condião inicial é $y(0.5) = \exp(0.5) \approx 1.649$
- Consequentemente, y_1 pertence a uma outra solução diferente daquela de onde partimos

Exemplo, continuação

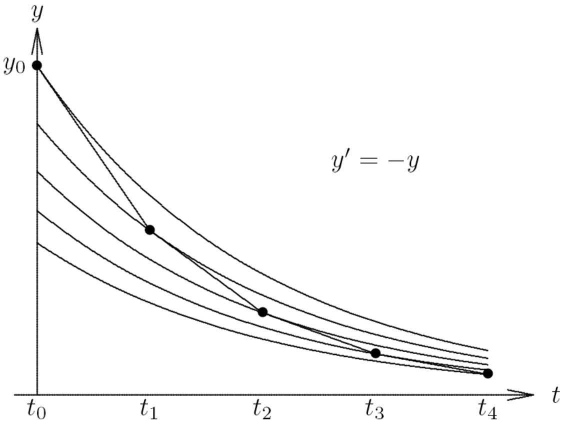
- Para continuar o processo numérico, damos um novo passo de t_1 para $t_2 = t_1 + h = 1.0$, obtendo $y_2 = y_1 + hy'_1 = 1.5 + (0.5)(1.5) = 2.25$
- Agora y_2 difere não só da solução exacta do problema original em $t = 1$, $y(1) = \exp(1) \approx 2.718$, mas também difere da solução que passa no ponto anterior (t_1, y_1), que tem o valor aproximado 2.473 em $t = 1$
- Pelo que nos deslocamos novamente para uma outra solução desta EDO

Exemplo, continuação



Para soluções instáveis os erros numéricos aumentam com o tempo

Exemplo, continuação



Para soluções estáveis os erros numéricos podem diminuir com o tempo

Erros na solução numérica de EDOs

- Métodos numéricos para resolver EDOs incorrem em dois tipos de erros distintos
  - Erros de arredondamento devidos à precisão finita da aritmética de ponto flutuante
  - Erros de truncatura (discretização) devidos aos métodos de aproximação usados e que permaneceriam mesmo que se usasse uma aritmética exacta
- Na prática os erros de truncatura são o factor dominante e determinam a exactidão da solução numérica de uma EDO, pelo que vamos focar a nossa atenção nos erros de truncatura

Erro Global e Erro Local

Erro de truncatura em qualquer ponto t_k pode ser decomposto em

- Erro global: diferença entre solução calculada y_k e solução verdadeira $y(t_k)$ que passa pelo ponto (t_0, y_0)

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{y}(t_k)$$

- Erro local: erro efectuado num único passo do método numérico

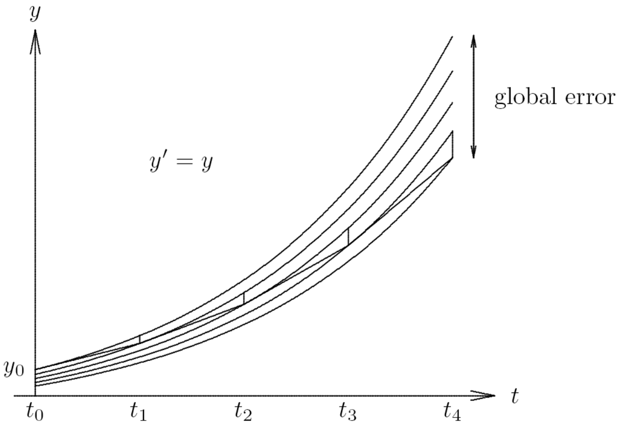
$$\ell_k = \mathbf{y}_k - \mathbf{u}_{k-1}(t_k)$$

em que $\mathbf{u_{k-1}}(t)$ é a solução verdadeira que passa pelo ponto $(t_{k-1}, \mathbf{y}_{k-1})$

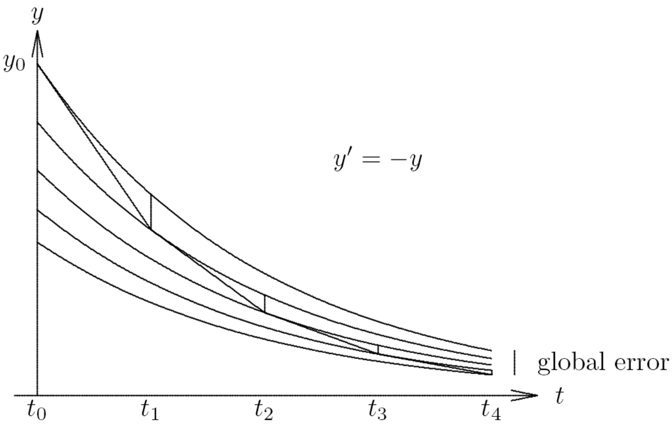
Erro Global e Erro Local, continuação

- Erro global não é necessariamente a soma dos erros locais
- Erro global é geralmente maior do que a soma dos erros locais se a solução for instável, mas pode ser inferior à soma se a solução for estável
- Queremos ter um pequeno erro global mas apenas podemos controlar o erro local directamente

Erro Global e Erro Local, continuação



Erro Global e Erro Local, continuação



Ordem de Exactidão de um Método Numérico

- Ordem de exactidão: de um método numérico é p se

$$\ell_k = \mathcal{O}\left(h_k^{p+1}\right)$$

- Então o erro local por unidade de passo: $\ell_k / h_k = \mathcal{O}\left(h_k^p\right)$
- Sob certas condições razoáveis

$$e_k = \mathcal{O}\left(h^p\right)$$

em que h é o comprimento médio do passo.

Ordem de Exactidão do Método de Euler

- Para o sistema genérico de EDOs $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$, consideramos a série de Taylor

$$\begin{aligned}\mathbf{y}(t+h) &= \mathbf{y}(t) + h\mathbf{y}'(t) + \mathcal{O}\left(h^2\right)\\ &= \mathbf{y}(t) + h\mathbf{f}(t, \mathbf{y}(t)) + \mathcal{O}\left(h^2\right)\end{aligned}$$

- Se tomarmos $t = t_k$ e $h = h_k$ obtemos

$$\mathbf{y}\left(t_{k+1}\right) = \mathbf{y}\left(t_k\right) + h_k\mathbf{f}\left(t_k, \mathbf{y}_k\right) + \mathcal{O}\left(h_k^2\right)$$

- Subtraindo esta expressão ao método de Euler ($\mathbf{y}_{k+1} = \mathbf{y}_k + h_k\mathbf{f}\left(t_k, \mathbf{y}_k\right)$) obtemos

$$\begin{aligned}\ell_{k+1} &= \mathbf{y}_{k+1} - \mathbf{y}\left(t_{k+1}\right)\\ &= \left[\mathbf{y}_k - \mathbf{y}\left(t_k\right)\right] + h_k\left[\mathbf{f}\left(t_k, \mathbf{y}_k\right) - \mathbf{f}\left(t_k, \mathbf{y}\left(t_k\right)\right)\right] - \mathcal{O}\left(h_k^2\right)\end{aligned}$$

Método de Euler, continuação

- Não havendo erros anteriores temos que $\mathbf{y}_k = \mathbf{y}(t_k)$, a diferença entre parêntesis rectos do lado direito também será nula, ficará apenas o termo $\mathcal{O}\left(h_k^2\right)$ representando o erro local
- Isto significa que o método de Euler é de primeira ordem

Estabilidade de um Método Numérico

- Método numérico é estável se pequenas perturbações resultarem em soluções numéricas diferentes mas não divergentes dentro certos limites
- Tais diferenças nas soluções numéricas podem ser provocadas pela instabilidade da solução da EDO, mas pode também ser devida ao próprio método numérico, mesmo quando a solução da EDO é estável
- O erro global resulta da acumulação dos erros locais e dos erros propagados

Estabilidade do Método de Euler

- Pelo teorema do valor médio

$$\begin{aligned}\mathbf{f}(t_k, \mathbf{y}_k) - \mathbf{f}(t_k, \mathbf{y}(t_k)) &= \mathbf{J}_f(t_k, \xi)(\mathbf{y}_k - \mathbf{y}(t_k)) \\ &= \mathbf{J}_f(t_k, \alpha \mathbf{y}_k + (1-\alpha)\mathbf{y}(t_k))(\mathbf{y}_k - \mathbf{y}(t_k))\end{aligned}$$

  em que $\mathbf{J}_f$ é a matriz Jacobiana de f e $\alpha \in [0, 1]$
- Podemos então expressar o erro global no passo $k+1$ como

$$e_{k+1} = (I + h_k \mathbf{J}_f)\, e_k + \ell_{k+1}$$

- Erro global é multiplicado em cada passo pelo factor de amplificação $I + h_k \mathbf{J}_f$
- Erros não vão crescer se o raio espectral $\rho(I + h_k \mathbf{J}_f) \leq 1$, o que equivale a dizer que todos os valores próprios de $h_k \mathbf{J}_f$ pertencem a um circulo do plano complexo com raio 1 e centrado em -1

Método de Euler, continuação

Em geral o factor de crescimento do erro global depende de

- Método numérico, determina a forma do factor de amplificação
- Passo h
- Jacobiana $\mathbf{J}_f$, que é determinada pela EDO

Métodos Implícitos

- Método de Euler é explícito porque apenas usa a informação no tempo t_k para avançar a solução no tempo t_{k+1}
- Embora posso parecer desejável, o método de Euler apresenta uma região de estabilidade muito limitada
- Maiores regiões de estabilidade podem ser obtidas usando informação do t_{k+1}, o que torna o método implícito
- O exemplo mais simples é o método de Euler implícito

$$y_{k+1} = y_k + h_k f\left(t_{k+1}, y_{k+1}\right)$$

- O método é implícito porque necessitamos de avaliar f com argumento y_{k+1} antes de conhecer o seu valor

Exemplo: Método de Euler Implícito

- Considere a EDO escalar e não linear $y' = -y^3$ com condição inicial $y(0) = 1$
- Usando o método de Euler Implícito com passo $h = 0.5$, obtemos a equação implícita

$$y_1 = y_0 + hf(t_1, y_1) = 1 - 0.5y_1^3$$

  do valor da solução no proximo ponto
- Esta equação não linear em y_1 pode ser resolvida através de um método como o das bisseções, ponto fixo ou de Newton
- Como aproximação inicial pode usar-se a solução anterior $y_0 = 1$, ou um método explícito como o de Euler, com o qual obtemos $y_1 = y_0 - 0.5y_0^3 = 0.5$
- As iterações poderão então convergir para o valor final $y_1 = 0.7709$

Estabilidade e exactidão do Método de Euler Implícito

- Factor de crescimento do método de Euler implícito para um sistema de EDOs genérico $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ é $(\mathbf{I} - h\mathbf{J}_f)^{-1}$ cujo raio espectral é menor do que 1 se todos os valores próprios de $h\mathbf{J}_f$ pertencerem à região do plano complexo fora do circulo de raio 1 centrado em 1
- Região de estabilidade de uma EDO é constituída por todo o semiplano complexo esquerdo: valores próprios de $\mathbf{J}_f$ com parte real negativa
- Para EDOs com soluções estáveis este método é estável para qualquer dimensão do passo, isto é incondicionalmente estável
- Grande vantagem dos métodos incondicionalmente estáveis é que a exactidão pretendida é apenas condicionada pela escolha do passo
- Apesar do método de Euler implícito ser incondicionalmente estável, é um método de primeira ordem de exactidão ($e_k = \mathcal{O}(h)$) o que limita muito a sua utilidade

Método de Euler Modificado

- Ordem de exactidão superior pode ser obtida fazendo a média das soluções previstas pelos métodos de Euler e Euler implícito para obter o método de Euler modificado

$$y_{k+1} = y_k + h_k \left(f\left(t_k, y_k\right) + f\left(t_{k+1}, y_{k+1}\right)\right)/2$$

- Factor de crescimento do método de Euler modificado para um sistema de EDOs genérico $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ é $\left(\mathbf{I} + \frac{h}{2}\mathbf{J}_f\right)\left(\mathbf{I} - \frac{h}{2}\mathbf{J}_f\right)^{-1}$ cujo raio espectral é menor do que 1 se todos os valores próprios de $h\mathbf{J}_f$ estiverem do lado esquerdo do plano complexo
- Método de Euler modificado é incondicionalmente estável e é de segunda ordem de exactidão ($e_k = \mathcal{O}(h^2)$) o que faz dele um método mais eficiente do que os dois anteriores

Métodos Numéricos para EDOs

- Existem muitos métodos diferentes para resolver EDOs, a maior parte pertence a um das seguintes tipos
  - Série de Taylor
  - Runge-Kutta
  - Múltiplos Passos

  Vamos considerar de forma breve cada um destes tipos de métodos

Métodos da Série de Taylor

- O método de Euler pode ser derivado do desenvolvimento em série de Taylor
- Retendo mais termos na série de Taylor podemos gerar métodos de passo simples de ordem superior ao de Euler
- Por exemplo, retendo um termo adicional na série de Taylor

$$\mathbf{y}(t+h) = \mathbf{y}(t) + h\mathbf{y}'(t) + \frac{h^2}{2}\mathbf{y}''(t) + \frac{h^3}{6}\mathbf{y}'''(t) + \cdots$$

obtemos um método de segunda ordem

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h_k\mathbf{y}'_k + \frac{h_k^2}{2}\mathbf{y}''_k$$

Métodos da Série de Taylor, continuação

- Esta aproximação requer o cálculo de derivadas de ordem superior de $\mathbf{y}$, o que pode ser obtido derivando $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ pela regar da cadeia:

$$\begin{aligned} \mathbf{y}'' &= \mathbf{f}_t(t, \mathbf{y}) + \mathbf{f}_y(t, \mathbf{y})\,\mathbf{y}' \\ &= \mathbf{f}_t(t, \mathbf{y}) + \mathbf{f}_y(t, \mathbf{y})\,\mathbf{f}(t, \mathbf{y}) \end{aligned}$$

  em que os subscritos indicam derivadas parciais relativamente à variável dada
- À medida que a ordem das derivadas aumenta também a complexidade das expressões para as derivadas, sendo cada vez mais difícil calculá-las, pelo que os métodos baseados na serie de taylor de ordem superior não tem sido muito usados na prática

Métodos de Runge-Kutta

- Em vez de calcular as derivadas, os métodos de Runge-Kutta simulam o efeito das derivadas de ordem superior determinando o valor de **f** várias vezes entre t_k e t_{k+1}
- O exemplo mais simples é o método de Euler modificado

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h_k}{2}\left(\mathbf{k}_1 + \mathbf{k}_2\right)$$

em que

$$\begin{aligned}\mathbf{k}_1 &= \mathbf{f}\left(t_k, \mathbf{y}_k\right) \\ \mathbf{k}_2 &= \mathbf{f}\left(t_k + h_k, \mathbf{y}_k + h_k\mathbf{k}_1\right)\end{aligned}$$

Métodos de Runge-Kutta, continuação

- O método de Euler modificado é um método de Runge-Kutta de segunda ordem
- O método de Runge-Kutta mais conhecido é o de quarta ordem

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h_k}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right)$$

em que

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}\left(t_k, \mathbf{y}_k\right) \\
\mathbf{k}_2 &= \mathbf{f}\left(t_k + h_k/2, \mathbf{y}_k + (h_k/2)\,\mathbf{k}_1\right) \\
\mathbf{k}_3 &= \mathbf{f}\left(t_k + h_k/2, \mathbf{y}_k + (h_k/2)\,\mathbf{k}_2\right) \\
\mathbf{k}_4 &= \mathbf{f}\left(t_k + h_k, \mathbf{y}_k + h_k\mathbf{k}_3\right)
\end{aligned}$$

Métodos de Runge-Kutta, continuação

- Para avançar para o tempo t_{k+1}, os métodos de Runge-Kutta não precisam de informações sobre as soluções anteriores a t_k, o que faz deles métodos "auto-iniciantes" no principio da integração e torna fácil a mudança do comprimento do passo ao longo da integração
- Estes factos tornam também estes métodos relativamente fáceis de programar, daí também a sua grande popularidade
- Infelizmente os métodos de Runge-Kutta não propiciam uma estimativa do erro sobre a qual se possa basear a escolha do comprimento do passo

Métodos de Múltiplos Passos

- Métodos de Múltiplos Passos usam a informação em mais do que um ponto anterior para estimar a solução no proximo ponto
- Multi-Passos Lineares têm a forma

$$\mathbf{y}_{k+1} = \sum_{i=1}^{m} \alpha_i \mathbf{y}_{k+1-i} + h \sum_{i=0}^{m} \beta_i \mathbf{f}\left(t_{k+1-i}, \mathbf{y}_{k+1-i}\right)$$

- Parâmetros α_i e β_i são determinados por interpolação polinomial
- Se $\beta_0 = 0$ o método é explícito, se $\beta_0 \neq 0$ o método é implícito
- Método implícitos são normalmente mais correctos e estáveis do que os explícitos, mas requerem uma estimativa inicial de $\mathbf{y}_{k+1}$

Métodos de Múltiplos Passos

- Como a estimativa inicial pode ser convenientemente fornecida por um método explícito, então os dois podem ser usados como um par preditor-corrector
- Em cada passo pode-se efectuar a correcção do valor previsto uma ou várias vezes até se obter determinada tolerância
- Em alternativa ao esquema previsão-correcção pode usar-se métodos de resolução de equações não lineares, como o de Newton, para resolver a equação não-linear em $\mathbf{y}_{k+1}$ resultante do método implícito

Exemplos: Métodos de Múltiplos Passos

- O mais simples dos métodos multi-passos explícito de segunda ordem é

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h}{2}\left(3\mathbf{y}'_k - \mathbf{y}'_{k-1}\right)$$

- O mais simples dos métodos multi-passos implícito de segunda ordem é o método de Euler modificado

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h}{2}\left(\mathbf{y}'_{k+1} + \mathbf{y}'_k\right)$$

- Um dos mais conhecidos pares de métodos multi-passo consiste no Adams-Bashforth, preditor de quarta ordem

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h}{24}\left(55\mathbf{y}'_k - 59\mathbf{y}'_{k-1} + 37\mathbf{y}'_{k-2} - 9\mathbf{y}'_{k-3}\right)$$

  e o implícito de quarta ordem Adams-Moulton corrector

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h}{24}\left(9\mathbf{y}'_{k+1} + 19\mathbf{y}'_k - 5\mathbf{y}'_{k-1} + \mathbf{y}'_{k-2}\right)$$

Propriedades dos Métodos de Múltiplos Passos

- Não são auto-iniciantes pelo que é necessário obter várias estimativas do valor de $\mathbf{y}_k$ para iniciar o processo de integração
- Difícil de mudar o comprimento do passo de ponto para ponto
- Boa estimativa do erro local pode ser obtida à partir da diferença entre valor predito e corrigido
- Relativamente mais difíceis de programar
- Métodos implícitos têm maiores regiões de estabilidade do que os explícitos, mas necessitam de serem iterados para beneficiar plenamente dessa propriedade
- Apesar dos métodos implícitos serem mais estáveis do que os explícitos, eles não são necessariamente incondicionalmente estáveis

Métodos Disponíveis no Octave e na NMLibforOctave

Octave

- Método de Runge-Kutta de 4ª e 5ª ordem: `[] = ODE45()`

NMLibforOctave

- Método de Euler: `[] = ODE_EULER()`
- Método de Euler Implícito: `[] = ODE_BEULER()`
- Método de Euler Modificado: `[] = ODE_MEULER()`
- Método de Preditor de 4ª Ordem: `[] = ODE_FOP()`

Bibliografia

Exposição baseada essencialmente no capítulo 9 de

- Michael T. Heath. "Scientific Computing an Introductory Survey". McGraw-Hill, 2002, New York.

e no capítulo 7 de

- Alfio quarteroni e Fausto Saleri. "Cálculo Científico com MATLAB e Octave". Springer, 2006, Milão.