Cálculo Numérico

Resolução numérica de EDO's

Prof.: Roberto F. Ausas
Cálculo Numérico
e-mail: rfausas@icmc.usp.br

ICMC-USP - São Carlos
Instituto de Ciências Matemáticas e de Computação

Semestre 2 - 2017

Tópicos

1. Método de Euler
2. Métodos de Taylor
3. Métodos de Runge-Kutta
4. Métodos tipo previsor-corretor

Introdução

Vamos estudar **métodos numéricos** para resolver o problema de valor inicial para equações diferenciais ordinarias (EDO's).

$$y'(t) = \frac{dy(t)}{dt} = f\left(t, y(t)\right) \quad \forall\ t \in I = [t_0, T] \subset \mathbb{R}$$

junto com a condição inicial

$$y(t_0) = y_0$$

Exemplos de EDO's: População

Consideremos o equação de primeira ordem

$$\begin{cases} \dfrac{dy(t)}{dt} &= \lambda\, y \\ \\ y(0) &= y_0 \end{cases}$$

$\lambda \in \mathbb{R}$.

Neste caso, a função f é linear:

$$f(t, y(t)) = \lambda\, y$$

mas, não tem dependência explícita na variável independente t.

Exemplos de EDO's: Exemplo não linear

Consideremos o problema de primeira ordem

$$\begin{cases} \dfrac{dy(t)}{dt} &= t^2 + y^2(t) \\ \\ y(0) &= \dfrac{1}{2} \end{cases}$$

Neste caso, a função f é não linear e tem dependência explicíta na variável independente t:

$$f(t, y(t)) = t^2 + y^2(t)$$

Exemplos de EDO's: Lotka-Volterra (Ecologia)

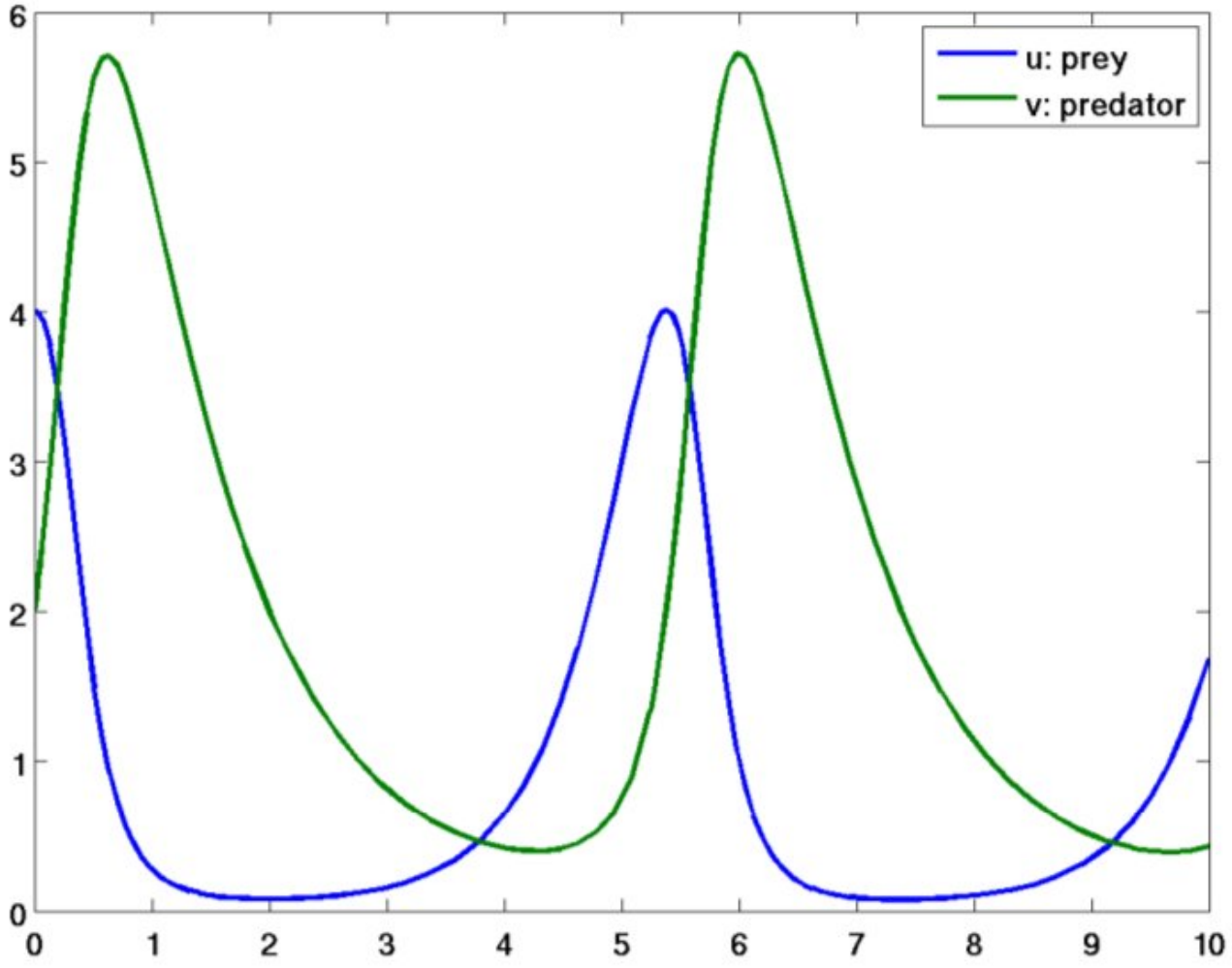
Agora, consideremos um sistema de duas equações de primeira ordem: Modelo de interação entre presas e predadores:

$$\begin{cases} \dfrac{du(t)}{dt} = f_1(u,v) = (\alpha - \beta\, v)\, u & \text{Presa} \\ \dfrac{dv(t)}{dt} = f_2(u,v) = (\delta\, u - \gamma)\, v & \text{Predador} \end{cases}$$

Com condições iniciais $u(0) = u_0$ e $v(0) = v_0$.

$$\mathbf{f}(\mathbf{y}) = \begin{pmatrix} f_1(\mathbf{y}) \\ \\ f_2(\mathbf{y}) \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} u \\ \\ v \end{pmatrix}$$

Exemplos de EDO's: Lotka-Volterra



Exemplos de EDO's: Pêndulo

Aplicamos a lei de Newton para uma massa pendurada:

$$m\,\ell^2 \frac{d^2\theta(t)}{dt^2} = m\,\ell^2\theta''(t) = -m\,g\,\ell\,\sin\left(\theta(t)\right)$$

em que

- m é a massa;
- ℓ é o comprimento
- θ é o ângulo e θ'' é a aceleração angular
- $g = 9.81\ \mathrm{m/s^2}$

Esta é uma equação de segunda ordem, que podemos transformar en duas equações de primeira ordem

Exemplos de EDO's: Pêndulo

Definimos:

$$y_1(t) = \theta(t), \qquad y_2(t) = \theta'(t)$$

Então, podemos escrever:

$$\begin{aligned} y_1'(t) &= \theta'(t) = \dots = y_2(t) \\ y_2'(t) &= \theta''(t) = -\underbrace{\frac{g}{\ell}}_{\omega^2} \sin(\theta(t)) = -\omega^2 \sin(y_1(t)) \end{aligned}$$

Exemplos de EDO's: Pêndulo

Então, podemos escrever:

$$\begin{aligned} y_1'(t) &= y_2(t) \\ y_2'(t) &= -\omega^2 \sin(y_1(t)) \end{aligned}$$

ou usando notação vetorial

$$\mathbf{y}'(\mathbf{t}) = \begin{pmatrix} y_1'(t) \\ \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} y_2(t) \\ \\ -\omega^2 \sin(y_1(t)) \end{pmatrix} = \mathbf{f}(t, \mathbf{y}(t))$$

Que é um sistema de duas equações ordinarias de primeira ordem!

Exemplos de EDO's: Pêndulo

Com notação vetorial escreveremos em geral

$$\mathbf{y}'(\mathbf{t}) = \frac{d\mathbf{y}(t)}{dt} = \mathbf{f}(t, \mathbf{y}(t))$$

em que precisamos uma condição inicial $\mathbf{y}(0) = \mathbf{y}_0$.

Por exemplo, para o caso do pêndulo precisamos:

$$\mathbf{y}(\mathbf{0}) = \begin{pmatrix} y_1(0) \\ \\ y_2(0) \end{pmatrix} = \begin{pmatrix} \theta(0) \\ \\ \theta'(0) \end{pmatrix}$$

Isto é o ângulo e a velocidade inicial da massa.

Exemplos de EDO's: Movimento de planetas

O problema de interação gravitacional de um planeta com o sol é descrito por:

$$\begin{cases} x''(t) &= -\gamma \dfrac{x(t)}{r^3(t)} \\ \\ y''(t) &= -\gamma \dfrac{y(t)}{r^3(t)} \end{cases}$$

em que $r(t) = \sqrt{x^2(t) + y^2(t)}$.

Este sistema pode ser transformado em um sistema de 4 equações de primeira ordem $\rightarrow$ Fazer como execício!

EDO's: Existência e unicidade de soluções

Antes de começar resolver equações numericamente, precisamos saber se o problema que temos está matemáticamente **bem posto!**

Vamos considerar o problema

$$(*)\begin{cases} y'(t) = f(t, y(t)) \quad \forall\ t \in I \\ y(0) = y_0 \end{cases}$$

Temos um teorema clássico que garante a existência e a unicidade de soluções.

EDO's: Existencia e unicidade de soluções

Teorema de Picard

Vamos supor que a função $f(t, y(t))$ é

1. Continua com respeito aos dois argumentos.
2. Lipschitz continua com respeito ao seu segundo argumento, i.e., existe uma constante positiva L tal que

$$|f(t, y_a) - f(t, y_b)| \leq L\,|y_a - y_b| \quad \forall\ t \in I,\ \forall\ y_a, y_b \in \mathbb{R}$$

Então a solução $y(t)$ do problema (*) **existe**, é **unica** e pertence a $C^1(I)$.

EDO's: Existencia e unicidade de soluções

Teorema

Vamos supor que a função $f(t, y)$ seja definida num conjunto convexo $\mathcal{D} \subset \mathbb{R}^2$, se existir uma constante L tal que

$$\left| \frac{\partial f}{\partial y}(t, y) \right| \leq L,$$

$\forall\ (t, y) \in \mathcal{D}$, então f satisfaz uma condição de Lipschitz em $\mathcal{D}$ na variável y e com constante de Lipschitz L.

Resolução numérica de EDO's

- A idéia é gerar uma sequência de valores $t_0, t_1, \ldots,$ e uma sequência correspondente de valores $y_0, y_1, \ldots,$, tais que y_n aproxima a solução exata do problema em t_n, i.e.,

$$y_n \approx y(t_n), \quad n = 0, 1, \ldots$$

Introdução

- Definimos o tamanho do passo como

$$h = t_{n+1} - t_n$$

As vezes, o passo h é chamado Δt ou δt.

Na maiora dos métodos que estudaremos, h será considerado constante, embora, em métodos mais sofisticados, o passo h é variável, de tal forma de controlar a precisão dos resultados de maneira mais eficiente.

Método de Euler explícito

- É o método mais simples para resolver numericamente uma EDO
- A ideia é definir pontos no intervalo de interesse: $a = t_0 < t_1 < t_2 < \cdots < t_N = b$, considerando um passo fixo $h = t_{n+1} - t_n$
- Vamos avançar desde um ponto t_n ao próximo t_{n+1}, passo por passo.

Método de Euler explícito

- Escrevemos:

$$y'(t_n) = \frac{y(t_{n+1}) - y(t_n)}{h} + \mathcal{O}(h)$$

Mas, lembremos que $y'(t) = f(t, y(t))$, então:

$$f(t_n, y(t_n)) = \frac{y(t_{n+1}) - y(t_n)}{h} + \mathcal{O}(h)$$

$$y(t_{n+1}) = y(t_n) + h\, f(t_n, y(t_n)) + \mathcal{O}(h^2)$$

Método de Euler explícito

- O que motiva o método numérico:

$$y_{n+1} = y_n + h\, f(t_n, y_n)$$

$$t_{n+1} = t_n + h$$

Com condição inicial $y(t_0) = y_0$.

Método de Euler explícito

- Para um sistema de várias equações de primeira ordem escreveriamos

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + h\,\mathbf{f}(t_n, \mathbf{y}_n) \\ t_{n+1} &= t_n + h \end{aligned}$$

Com condição inicial $\mathbf{y}(t_0) = \mathbf{y}_0$.

Método de Euler explícito: Exemplo

Vamos resolver o problema mais simples:

$$y'(t) = f(t, y) = \lambda\, y$$

$\lambda \in \mathbb{R}$ e com condição inicial, $y(0) = y_0$.

Neste caso a solução exata é conhecida

$$y(t) = y_0\, e^{\lambda t}$$

Podemos testar o método numérico e comparar com a solução exata!

Método de Euler explícito: Exemplo

Dados $t_0, y_0,$ para $n = 0, 1, \ldots$

$$\begin{aligned} y_{n+1} &= y_n + h\, f(t_n, y_n) = y_n + h\, \lambda y_n = (1 + h\, \lambda)\, y_n \\ t_{n+1} &= t_n + h \end{aligned}$$

Método de Euler explícito: Exemplo 1

```
clear ;
#Time interval
t0 = 0.0; tf =  10.0;
#Initialization
y0 = 4.0; h = 0.02; lambda = -1;
t(1) = t0; y(1) = y0; yexact(1) = y0;
n = 1;
while t(n) <= tf
  y(n+1) = (1.0 + h*lambda)*y(n);
  t(n+1) = t(n) + h;
  yexact(n+1) = y0*exp(lambda*t(n+1));
  n = n + 1;
end

plot(t,yexact,"-*1;Exact;", t,y, "-*4;Explicit Euler;");
```

Exemplos de EDO's: Pêndulo

Lembremos o sistema de duas equações ordinarias de primeira ordem para o pêndulo:

$$\mathbf{y}'(\mathbf{t}) = \begin{pmatrix} y_1'(t) \\ \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} y_2(t) \\ \\ -\omega^2 \sin(y_1(t)) \end{pmatrix} = \mathbf{f}(t, \mathbf{y}(t))$$

Método de Euler explícito: Exemplo 2

```
clear ;
%Time interval
t0 = 0.0; tf = 15.0;
%Initialization
y0(1) = pi/2.1; % Initial angle
y0(2) = 0.0; % Initial velocity
%Step size
h = 0.00025;
n = 1;
t(1) = t0;
y(:,1) = y0;
while t(n) <= tf
  y(:,n+1) = EulerExplicit(t(n), y(:,n), h, @funcpend);
  t(n+1) = t(n) + h;
  n = n + 1;
end
plot(t,y(1,:), "-5;Explicit Euler;");
```

Método de Euler explícito: Exemplo 2

```
function [ynew] = EulerExplicit(t, yold, h, funcyp)

  ynew = yold + h*funcyp(t, yold)';

end
```

```
function [yp] = funcpend(t, y)
  g = 9.81;
  length = 1.0;
  omega2 = g/length;

  yp(1) = y(2);
  yp(2) = -omega2 * sin(y(1));
end
```

Método de Euler implícito

- Este método é mais "estável" como veremos depois.

- Agora, escrevemos:

$$y'(t_{n+1}) = \frac{y(t_{n+1}) - y(t_n)}{h} + \mathcal{O}(h)$$

Mas, lembremos que $y'(t) = f(t, y(t))$, então:

$$f(t_{n+1}, y(t_{n+1})) = \frac{y(t_{n+1}) - y(t_n)}{h} + \mathcal{O}(h)$$

$$\Rightarrow y(t_{n+1}) = y(t_n) + h\, f(t_{n+1}, y(t_{n+1})) + \mathcal{O}(h^2)$$

Método de Euler implícito

- O que motiva o método numérico:

$$y_{n+1} = y_n + h\, f(t_{n+1}, y_{n+1})$$

$$t_{n+1} = t_n + h$$

Com condição inicial y_0.

Método de Euler implícito

Mas, agora qual é o problema?

A incógnita y_{n+1} aparece tanto do lado esquerdo como do lado direito ...

Se a função $f(t, y)$ for não linear no seu segundo argumento $\Rightarrow$ precisaremos resolver, em cada passo, a equação não linear:

$$y_{n+1} - y_n - h\, f(t_{n+1}, y_{n+1}) = 0$$

para achar $y_{n+1}, \quad n = 0, 1, \ldots \rightarrow$ Isto é mais costoso!

Método de Euler implícito

- Para um sistema de várias equações de primeira ordem escreveriamos

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\,\mathbf{f}(t_{n+1}, \mathbf{y}_{n+1})$$

$$t_{n+1} = t_n + h$$

Com condição inicial $\mathbf{y}(t_0) = \mathbf{y}_0$. Neste caso precisaremos resolver um sistema de equações não lineares!

$$\mathbf{y}_{n+1} - \mathbf{y}_n - h\,\mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}) = 0$$

Método de Euler implícito: Exemplo 1

Por enquanto, voltemos ao exemplo simples:

$$y'(t) = f(t, y) = \lambda\, y$$

$\lambda \in \mathbb{R}$ e com condição inicial, $y(0) = y_0$,

cuja solução exata era:

$$y(t) = y_0\, e^{\lambda t}$$

Método de Euler implícito: Exemplo 1

Dados $t_0, y_0,$ para $n = 0, 1, \ldots$

$$y_{n+1} = y_n + h\, f(t_{n+1}, y_{n+1}) = y_n + h\, \lambda\, y_{n+1}$$

$$t_{n+1} = t_n + h$$

Afortunadamente, este problema é linear, pois $f = \lambda\, y$ é uma função linear na variável y

Método de Euler implícito: Exemplo 1

Dados $t_0, y_0,$ para $n = 0, 1, \ldots$

$$y_{n+1} = y_n + h\,\lambda\,y_{n+1} \quad \Rightarrow \quad y_{n+1} = \frac{y_n}{1 - h\,\lambda}$$

$$t_{n+1} = t_n + h$$

Afortunadamente, este problema é linear, pois $f = \lambda\,y$ é uma função linear na variável y

Método de Euler implícito: Exemplo 1

```
clear ;
#Time interval
t0 = 0.0; tf =  10.0;
#Initialization
y0 = 4.0; h = 0.02; lambda = -1;
t(1) = t0; y(1) = y0; yexact(1) = y0;
n = 1;
while t(n) <= tf
  y(n+1) = y(n) / (1.0 - h*lambda);
  t(n+1) = t(n) + h;
  yexact(n+1) = y0*exp(lambda*t(n+1));
  n = n + 1;
end

plot(t,yexact,"-*1;Exact;", t,y, "-*3;Implicit Euler;");
```

Método de Euler implícito: Exemplo 2

Para o sistema:

$$\mathbf{y}'(\mathbf{t}) = \begin{pmatrix} y_1'(t) \\ \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} y_2(t) \\ \\ -\omega^2 \sin(y_1(t)) \end{pmatrix} = \mathbf{f}(t, \mathbf{y}(t))$$

Após discretizar com Euler implícito, vamos resolver um sistema de duas equações não lineares

$$\mathbf{y}_{n+1} - \mathbf{y}_n - h\,\mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}) = 0$$

$$\begin{aligned} (y_1)_{n+1} &- (y_1)_n - h\,(y_2)_{n+1} &&= 0 \\ (y_2)_{n+1} &- (y_2)_n + h\,\omega^2 \sin((y_1)_{n+1}) &&= 0 \end{aligned}$$

Método de Euler implícito: Exemplo 2

```
clear ;
%Time interval
t0 = 0.0; tf = 3.0;
%Initialization
y0(1) = pi/2.1; % Initial angle
y0(2) = 0.0; % Initial velocity
%Step size
h = 0.25;
n = 1;
t(1) = t0;
y(:,1) = y0;
while t(n) <= tf
  y(:,n+1) = EulerImplicit(t(n), y(:,n), h, @funcpendjac);
  t(n+1) = t(n) + h;
  n = n + 1;
end
plot(t,y(1,:), "-*1;Implicit Euler;");
```

Método de Euler explícito: Exemplo 2

```
function [ynew] = EulerImplicit(t, yold, h, funcyp)

  TOLF = 1.0E-12;
  MAXIT = 3;
  error = TOLF + 1.0;
  iter = 0;

  y = yold;
  while(error > TOLF && iter < MAXIT)
    [res, jac] = funcyp(t, h, yold, y);
    ynew = y - jac\res;
    error = norm(res);
    y = ynew;
    iter = iter + 1;
  end

end
```

Método de Euler explícito: Exemplo 2

```
function [res, jac] = funcpendjac(t, h, yold, y)
  g = 9.81;
  length = 1.0;
  omega2 = g/length;

  res = zeros(2,1);
  res(1) = y(1) - yold(1) - h * y(2);
  res(2) = y(2) - yold(2) + h * omega2 * sin(y(1));

  jac = zeros(2,2);
  jac(1,1) = 1.0;
  jac(1,2) = -h;
  jac(2,1) =  h * omega2 * cos(y(1));
  jac(2,2) = 1;

end
```

Método Trapezoidal implícito

Uma forma interessante de obter ele é assim:

$$\frac{dy}{dt} = f(t, y(t)) \quad \Rightarrow \quad y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))\, dt$$

que usando a regra do trapézio para integrar, motiva o método numérico

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right]$$

já que $h = t_{n+1} - t_n$.

Este método é uma combinação dos anteriores.

Método α

Uma generelização do anterior seria, definindo um parámetro α, $0 \leq \alpha \leq 1$:

$$y_{n+1} = y_n + h\,[(1-\alpha)f(t_n, y_n) + \alpha\, f(t_{n+1}, y_{n+1})]$$

Então,

- Euler explícito $\rightarrow \alpha = 0$
- Método Trapezoidal implícito $\rightarrow \alpha = \frac{1}{2}$
- Euler implícito $\rightarrow \alpha = 1$

Exemplo

Apliquemos o método α novamente para o problema simples:

$$y'(t) = f(t, y) = \lambda\, y, \qquad y(0) = y_0$$

$$\begin{aligned} y_{n+1} &= y_n + h\,[(1-\alpha) f(t_n, y_n) + \alpha\, f(t_{n+1}, y_{n+1})] = \\ &= y_n + h\,[(1-\alpha)\lambda y_n + \alpha\,\lambda y_{n+1}] \end{aligned}$$

$$\Rightarrow \quad y_{n+1} - \alpha\, h\lambda\, y_{n+1} = y_n + (1-\alpha)\, h\,\lambda\, y_n$$

$$y_{n+1} = y_n \frac{1 + (1-\alpha)\, h\,\lambda}{1 - \alpha\, h\,\lambda}$$

Método α

```
clear ;
#Time interval
t0 = 0.0; tf =  10.0;
#Initialization
y0 = 4.0; h = 0.5; lambda = -1;
alpha = 0.5;
t(1) = t0; y(1) = y0; yexact(1) = y0;
n = 1;
while t(n) <= tf
  y(n+1) = y(n) * (1.0 + (1.0-alpha)*h*lambda) /
                  (1.0 - alpha*h*lambda);
  t(n+1) = t(n) + h;
  yexact(n+1) = y0*exp(lambda*t(n+1));
  n = n + 1;
end

plot(t,yexact,"-*1;Exact;", t,y, "-*4;alpha = 0.5;");
```

Método Trapezoidal explícito

Partimos do método Trapezoidal implícito:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right]$$

Agora, fazemos a aproximação: $y_{n+1} = y_n + h\, f(t_n, y_n)$, o que leva ao método:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_n + h\, f(t_n, y_n))\right]$$

Análise dos métodos

Temos que definir certos conceitos importantes:

- Convergência;
- Consistência;
- Estabilidade;

Análise dos métodos

Consideremos um método geral de um passo:

$$y_{n+1} = y_n + h\,\Phi(t_n, y_n, y_{n+1}, h)$$

Se $\Phi(\dots)$ depende de y_{n+1}, então, o método é chamado de **implícito**, em caso contrário, de **explícito.**

Análise dos métodos: Exemplos

- Método de Euler Explícito:
  $\Phi(t_n, y_n, y_{n+1}, h) = f(t_n, y_n)$
- Método Trapezoidal explícito:
  $\Phi(t_n, y_n, y_{n+1}, h) = \frac{1}{2}[f(t_n, y_n) + f(t_n + h, y_n + hf(t_n, y_n))]$
- Método Trapezoidal implícito:
  $\Phi(t_n, y_n, y_{n+1}, h) = \frac{1}{2}[f(t_n, y_n) + f(t_n + h, y_{n+1})]$

Erro de truncamento

Definimos o erro de truncamento como:

$$\tau_n(t) = \frac{y(t_{n+1}) - y(t_n)}{h} - \Phi(t, y(t_n), y(t_{n+1}), h)$$

em que, lembremos, $y(t)$ é a solução exata!

Ou seja, é o erro feito ao inserir a solução exata no esquema numérico proposto!

Erro de truncamento

Para um método *razoável*, esperamos que $|\tau_n| \to 0$ quando $h \to 0$.

Já que:

$$\lim_{h\to 0} \frac{1}{h}(y(t_{n+1}) - y(t_n)) = f(t, y(t_n))$$

Isto motiva a definição de consistência:

Consistência

Um método de um passo diz-se consistente com o problema (*) se:

$$\lim_{h\to 0}\Phi(t, y(t_n), y(t_{n+1}), h) = f(t, y(t_n))$$

$\forall\ n, y \in \mathbb{R}$ e toda função f com $\partial f/\partial y$ limitada.

Ordem de consistência de um método

Um método de um passo tem ordem de consistência p se

$$\tau_n = \mathcal{O}(h^p)$$

$\forall\ n, y \in \mathbb{R}$ e toda função f com derivadas com respeito a y até ordem p limitadas.

Definições de Erro

- Erro global: é a diferença entre a solução exata num determinado tempo e a solução do método numérico:

$$\epsilon_n = y(t_n) - y_n$$

- Erro num passo: é o erro feito avançando um passo do método numérico a partir de uma condição inicial que coincide com a solução exata:

$$\varepsilon_n = y(t_n) - [y(t_{n-1}) + h\,\Phi(t_{n-1}, y(t_{n-1}), h)]$$

Exemplo 1: Ordem do método de Euler explícito

Calculamos o erro de truncamento:

$$\tau_n = \frac{y(t_n + h) - y(t_n)}{h} - \Phi(t_n, y(t_n), y(t_{n+1}), h)$$

Mas, considerando o desenvolvimento de Taylor:

$$y(t_n + h) = y(t_n) + h\,y'(t_n) + \frac{h^2}{2}\,y''(t_n) + \frac{h^3}{6}\,y'''(t_n) + \ldots$$

Resulta

$$\tau_n = \frac{y(t_n) + h\,y'(t_n) + \frac{h^2}{2}\,y''(t_n) + \frac{h^3}{6}\,y'''(t_n) + \ldots - y(t_n)}{h} - \underbrace{\Phi(t_n, y(t_n), h)}_{f(t_n, y(t_n))}$$

$$\Rightarrow \tau_n = \underbrace{y'(t_n) - f(t_n, y(t_n))}_{0} + \frac{h}{2}\,y''(t_n) + \text{T.O.S.} = \frac{h}{2}\,y''(t_n) + \text{T.O.S.}$$

Exemplo 1: Ordem do método de Euler explícito

Finalmente, resulta:

$$\| \tau_n \| \leq \frac{\| y''(t_n) \|}{2} h + \text{T.O.S.} \leq \frac{1}{2} \max_{t \in (a,b)} \| y''(t_n) \| h$$

i.e.

$$\| \tau_n \| \leq c\, h$$

Sendo c uma constante positiva independente de h e t.

Teorema

O método de Euler Explícito é consistente de ordem 1.

Exemplo 2: Ordem do método α

Calculamos o erro de truncamento:

$$\tau_n = \frac{y(t_n + h) - y(t_n)}{h} - \Phi(t_n, y(t_n), y(t_{n+1}), h)$$

Exemplo 2: Ordem do método α

Calculamos o erro de truncamento:

$$\tau_n = \frac{y(t_n + h) - y(t_n)}{h} - \Phi(t_n, y(t_n), y(t_{n+1}), h)$$

Agora temos

$$\Phi(t_n, y(t_n), y(t_{n+1}), h) = (1 - \alpha)\, f(t_n, y(t_n)) + \alpha\, f(t_{n+1}, y(t_{n+1}))$$

Exemplo 2: Ordem do método α

Calculamos o erro de truncamento:

$$\tau_n = \frac{y(t_n + h) - y(t_n)}{h} - \Phi(t_n, y(t_n), y(t_{n+1}), h)$$

Agora temos

$$\Phi(t_n, y(t_n), y(t_{n+1}), h) = (1 - \alpha)\, f(t_n, y(t_n)) + \alpha\, f(t_{n+1}, y(t_{n+1}))$$

Já que $y'(t) = f(t, y(t))$:

$$\Phi(t_n, y(t_n), y(t_{n+1}), h) = (1 - \alpha)\, y'(t_n) + \alpha\, y'(t_{n+1})$$

Exemplo 2: Ordem do método α

Calculamos o erro de truncamento:

$$\tau_n = \frac{y(t_n + h) - y(t_n)}{h} - \Phi(t_n, y(t_n), y(t_{n+1}), h)$$

Agora temos

$$\Phi(t_n, y(t_n), y(t_{n+1}), h) = (1 - \alpha)\, f(t_n, y(t_n)) + \alpha\, f(t_{n+1}, y(t_{n+1}))$$

Já que $y'(t) = f(t, y(t))$:

$$\Phi(t_n, y(t_n), y(t_{n+1}), h) = (1 - \alpha)\, y'(t_n) + \alpha\, y'(t_{n+1})$$

Consideramos os desenvolvimentos de Taylor:

$$y(t_n + h) = y(t_n) + h\, y'(t_n) + \frac{h^2}{2}\, y''(t_n) + \frac{h^3}{6}\, y'''(t_n) + \ldots$$

$$y'(t_n + h) = y'(t_n) + h\, y''(t_n) + \frac{h^2}{2}\, y'''(t_n) + \ldots$$

Exemplo 2: Ordem do método α

Inserimos o desenvolvimento para $y(t_n + h)$ e a definição de Φ:

$$\tau_n = \frac{y(t_n) + h\,y'(t_n) + \frac{h^2}{2}\,y''(t_n) + \frac{h^3}{6}\,y'''(t_n) + \ldots - y(t_n)}{h} - \; (1-\alpha)\,y'(t_n) - \alpha\,y'(t_{n+1})$$

Exemplo 2: Ordem do método α

Inserimos o desenvolvimento para $y(t_n + h)$ e a definição de Φ:

$$\tau_n = \frac{y(t_n) + h\,y'(t_n) + \frac{h^2}{2}\,y''(t_n) + \frac{h^3}{6}\,y'''(t_n) + \ldots - y(t_n)}{h} - \left[(1-\alpha)\,y'(t_n) - \alpha\,y'(t_{n+1})\right]$$

$$\tau_n = y'(t_n) + \frac{h}{2}\,y''(t_n) + \frac{h^2}{6}\,y'''(t_n) + \ldots - \left[(1-\alpha)\,y'(t_n) - \alpha\,y'(t_{n+1})\right]$$

Exemplo 2: Ordem do método α

Inserimos o desenvolvimento para $y(t_n + h)$ e a definição de Φ:

$$\tau_n = \frac{y(t_n) + h\,y'(t_n) + \frac{h^2}{2}\,y''(t_n) + \frac{h^3}{6}\,y'''(t_n) + \ldots - y(t_n)}{h} - \left[(1-\alpha)\,y'(t_n) - \alpha\,y'(t_{n+1})\right]$$

$$\tau_n = y'(t_n) + \frac{h}{2}\,y''(t_n) + \frac{h^2}{6}\,y'''(t_n) + \ldots - \left[(1-\alpha)\,y'(t_n) - \alpha\,y'(t_{n+1})\right]$$

Inserimos o desenvolvimento para $y'(t_n + h)$

$$\tau_n = y'(t_n) + \frac{h}{2}\,y''(t_n) + \frac{h^2}{6}\,y'''(t_n) + \ldots - \left[(1-\alpha)\,y'(t_n) - \right.$$
$$\left. \alpha\left[y'(t_n) + h\,y''(t_n) + \frac{h^2}{2}\,y'''(t_n) + \ldots\right]\right]$$

Exemplo 2: Ordem do método α

Agrupamos

$$\tau_n = (1 - (1 - \alpha) - \alpha)\, y'(t_n) + h \left(\frac{1}{2} - \alpha\right) y''(t_n) + \frac{h^2}{2} \left(\frac{1}{3} - \alpha\right) y'''(t_n) + \text{T.O.S.}$$

Exemplo 2: Ordem do método α

Agrupamos

$$\tau_n = (1 - (1 - \alpha) - \alpha)\, y'(t_n) + h \left(\frac{1}{2} - \alpha\right) y''(t_n) + \frac{h^2}{2} \left(\frac{1}{3} - \alpha\right) y'''(t_n) + \text{T.O.S.}$$

Finalmente

$$\tau_n = h \left(\frac{1}{2} - \alpha\right) y''(t_n) \; + \; \frac{h^2}{2} \left(\frac{1}{3} - \alpha\right) y'''(t_n) \; + \; \text{T.O.S.}$$

Exemplo 2: Ordem do método α

Agrupamos

$$\tau_n = (1-(1-\alpha)-\alpha)\,y'(t_n) + h\left(\frac{1}{2}-\alpha\right)y''(t_n) + \frac{h^2}{2}\left(\frac{1}{3}-\alpha\right)y'''(t_n) + \text{T.O.S.}$$

Finalmente

$$\tau_n = h\left(\frac{1}{2}-\alpha\right)y''(t_n) \;+\; \frac{h^2}{2}\left(\frac{1}{3}-\alpha\right)y'''(t_n) \;+\; \text{T.O.S.}$$

Teorema

O método α é consistente de ordem 1 se $\alpha \neq \frac{1}{2}$ e é consistente de ordem 2 se $\alpha = \frac{1}{2}$.

Convergência

Um método numérico diz-se **convergente** se:

$$|y(t_n) - y_n| \leq C(h)$$

$\forall\ n = 0, \ldots, N$, em que $C(h)$ é infinitessimal com respeito a h quando h tende a zero.

Ordem de convergência

Teorema

Um método de um passo que é consistente de ordem p, em que Φ é Lipschitz em y e é zero-estável, é convergente de ordem p, i.e., $\exists c > 0$, tal que $|e_n| \leq \underbrace{c\, h^p}_{\mathcal{O}(h^p)}$.

Temos que estudar o conceito de estabilidade, mas, antes verifiquemos se os métodos estudados tem a ordem teórica esperada.

Ordem de convergência

```
# set up variables (see complete code)
alpha = 0.5;
h(1) = 1.0;
for k=1:10 % Loop over values of h
  t(1) = t0; y(1) = y0; % Initial condition
  n = 1;
  while t(n) <= tf % Loop over time steps
    y(n+1) = y(n) * (1.0 + (1.0-alpha)*h(k)*lambda)
                    / (1.0 - alpha*h(k)*lambda);
    t(n+1) = t(n) + h(k);
    n = n + 1;
  end
  yexact(n) = y0*exp(lambda*t(n));
  error(k) = abs(yexact(n) - y(n));
  h(k+1) = h(k) / 2;
end
xlabel("h"); ylabel("Error");
loglog (h(1:10),error, "-*3;alpha=0.5;")
```

Estabilidade absoluta

Que acontece quando consideramos intervalos não limitados?

Estabilidade absoluta

Que acontece quando consideramos intervalos não limitados?

Voltamos ao problema simples de decaimento exponencial:

$$\begin{cases} \dfrac{dy(t)}{dt} &= -\lambda\, y, \quad \lambda > 0 \\ \\ y(0) &= y_0 \end{cases}$$

cuja solução exata era $y(t) = y_0\, e^{-\lambda\, t}$. Claramente, quando $t \to \infty$ a solução $y(t) \to 0$

O conceito de estabilidade absoluta está relacionado com a possibilidade, de que o método numérico consiga reproduzir esse comportamento.

Estabilidade absoluta

Aplicamos novamente o método de Euler explícito:

$$\text{Dados } t_0, y_0, \qquad \text{para } n = 0, 1, \ldots$$

$$\begin{aligned} y_{n+1} &= y_n + h\, f(t_n, y_n) = y_n - h\,\lambda y_n \\ \\ \Rightarrow y_{n+1} &= (1 - h\,\lambda)\, y_n = \\ &= (1 - h\,\lambda)^2\, y_{n-1} = \\ &= (1 - h\,\lambda)^3\, y_{n-2} = \ldots \\ &= (1 - h\,\lambda)^{n+1}\, y_0 \end{aligned}$$

Estabilidade absoluta

⇒ para reproduzir o comportamento esperado, i.e., que a solução tenda para zero quando $n \to \infty$, precisamos que

$$|1 - h\lambda| < 1 \quad \Rightarrow \quad -1 < 1 - h\lambda < 1$$

de onde surge a condição

$$0 < h\lambda < 2$$

$$\Rightarrow \quad y_{n+1} \xrightarrow[n\to\infty]{} 0 \quad \text{se} \quad h < \frac{2}{\lambda}.$$

que é a definição de estabilidade absoluta para este caso. Para outros métodos a condição poderá ser diferente.

Estabilidade absoluta

Como fazer com outras equações e no caso de sistemas, para escolher h em métodos explícitos?

$$\begin{cases} \mathbf{y}' &= \mathbf{f}(t, \mathbf{y}) \\ \mathbf{y}(0) &= \mathbf{y}_0 \end{cases}$$

Regra prática: Escolher $h < \dfrac{2}{\lambda}$, λ sendo o autovalor de maior módulo de $\dfrac{\partial \mathbf{f}}{\partial \mathbf{y}}$.

Método de Euler explícito: Exemplo 1

```
clear ;
#Time interval
t0 = 0.0; tf =  10.0;
#Initialization
y0 = 4.0; h = 0.02; lambda = -1;
t(1) = t0; y(1) = y0; yexact(1) = y0;
n = 1;
while t(n) <= tf
  y(n+1) = (1.0 + h*lambda)*y(n);
  t(n+1) = t(n) + h;
  yexact(n+1) = y0*exp(lambda*t(n+1));
  n = n + 1;
end

plot(t,yexact,"-*1;Exact;", t,y, "-*4;Explicit Euler;");
```

Métodos de Taylor

Métodos de Taylor

Lembrando o teorema de Taylor, que de fato já temos usado muitas vezes:

$$y(t_n + h) = y(t_n) + h\, y'(t_n) + \frac{h^2}{2}\, y''(t_n) + \dots$$

definindo

$$T_p(t_n, y(t_n)) = \sum_{j=1}^{p} \frac{y^{(j)}(t_n)}{j!}\, h^{j-1}$$

resulta

$$y(t_n + h) = y(t_n) + h\, T_p(t_n, y(t_n)) + \mathcal{O}(h^{p+1})$$

Método de Taylor

Notemos que $y'(t_n) = f(t_n, y(t_n))$, então:

Método de Taylor

Notemos que $y'(t_n) = f(t_n, y(t_n))$, então:

$$y'(t) = f$$

Método de Taylor

Notemos que $y'(t_n) = f(t_n, y(t_n))$, então:

$$\begin{aligned} y'(t) &= f \\ y''(t) &= f' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} = f_t + f_y\, y' = f_t + f_y\, f \end{aligned}$$

Método de Taylor

Notemos que $y'(t_n) = f(t_n, y(t_n))$, então:

$$\begin{aligned}
y'(t) &= f \\
y''(t) &= f' = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} = f_t + f_y\, y' = f_t + f_y\, f \\
y'''(t) &= f'' = f_{tt} + 2\, f_{ty}\, y' + f_y\, y'' + f_{yy}\, (y')^2 \\
&= f_{tt} + 2\, f_{ty}\, y' + f_{yy}\, f^2 + f_y\, (f_t + f_y\, f)
\end{aligned}$$

Definindo o operador $D = (\frac{\partial}{\partial t} + f\,\frac{\partial}{\partial y})$ ...

Método de Taylor

Podemos escrever

$$y(t_n + h) = y(t_n) + h\, f(t_n, y(t_n)) + \frac{h^2}{2} f'(t_n, y(t_n)) \ + \ \ldots \ + \ + \frac{h^{p+1}}{(p+1)!} f^{(p)}(\xi_n, y(\xi_n))$$

Se consideramos o método numérico:

$$y_{n+1} = y_n + h\, T_p(t_n, y_n)$$

em que

$$T_p(t_n, y_n) = f(t_n, y_n) + \frac{h}{2} f'(t_n, y_n) \ + \ \ldots \ + \ \frac{h^{p-1}}{p!} f^{(p-1)}(t_n, y_n)$$

O erro de truncamento será: $\mathcal{O}(h^p)$.

Método de Taylor

- O método de Euler explícito é o método de Taylor de ordem 1.
- Estes métodos são explícitos, então existirá uma restrição de estabilidade a ser respeitada.
- Uma desvantagem, é a necessidade de calcular todas as derivadas da função f.

Método de Taylor: Exemplo

$$\begin{cases} \dfrac{dy(t)}{dt} &= y - t^2 + 1 \\ \\ y(0) &= 0.5 \end{cases}$$

A função $f(t, y(t)) = y - t^2 + 1$

Vamos calcular as derivadas:

Método de Taylor: Exemplo

$$
\begin{aligned}
f &= y - t^2 + 1 \\
f' &= \left(\frac{\partial}{\partial t} + f\,\frac{\partial}{\partial y}\right)(f) = y - t^2 - 2t + 1 \\
f'' &= \left(\frac{\partial}{\partial t} + f\,\frac{\partial}{\partial y}\right)(f') = y - t^2 - 2t - 1 \\
f''' &= \left(\frac{\partial}{\partial t} + f\,\frac{\partial}{\partial y}\right)(f'') = y - t^2 - 2t - 1
\end{aligned}
$$

Como exercício, fazer as contas e verificarlo!

Método de Taylor: Exemplo

- O método de Taylor de ordem 2 seria:

$$y_{n+1} = y_n + h\, T_2(t_n, y_n) = y_n + h\,[f(t_n, y_n) + \frac{h}{2} f'(t_n, y_n)]$$

Método de Taylor: Exemplo

- O método de Taylor de ordem 2 seria:

$$\begin{aligned} y_{n+1} &= y_n + h\, T_2(t_n, y_n) = y_n + h\,[f(t_n, y_n) + \frac{h}{2} f'(t_n, y_n)] \\ &= y_n + h\,(y_n - t_n^2 + 1) + \frac{h^2}{2}(y_n - t_n^2 - 2t_n + 1) \end{aligned}$$

Método de Taylor: Exemplo

- O método de Taylor de ordem 4 seria:

$$y_{n+1} = y_n + h\, T_4(t_n, y_n) = y_n + h\,[f(t_n, y_n) + \frac{h}{2} f'(t_n, y_n) + \frac{h^2}{6} f''(t_n, y_n) + \frac{h^3}{24} f'''(t_n, y_n)] =$$

Método de Taylor: Exemplo

- O método de Taylor de ordem 4 seria:

$$
\begin{aligned}
y_{n+1} &= y_n + h\, T_4(t_n, y_n) = y_n + h\,[f(t_n, y_n) + \frac{h}{2} f'(t_n, y_n) \\
&\quad + \frac{h^2}{6} f''(t_n, y_n) + \frac{h^3}{24} f'''(t_n, y_n)] = \\
&= y_n + h\,(y_n - t_n^2 + 1) + \frac{h^2}{2}\,(y_n - t_n^2 - 2t_n + 1) \\
&\quad + \frac{h^3}{6}(y_n - t_n^2 - 2t_n - 1) + \frac{h^4}{24}(y_n - t_n^2 - 2t_n - 1)
\end{aligned}
$$

Métodos de Runge-Kutta

Métodos de Runge-Kutta

Um método explícito de Runge-Kutta de r estagios é um método da forma:

$$y_{n+1} = y_n + h\,\Phi_r$$

em que

$$\Phi_r = c_1\,k_1 + c_2\,k_2 + \cdots + c_r\,k_r$$

Métodos de Runge-Kutta

Um método explícito de Runge-Kutta de r estagios é um método da forma:

$$y_{n+1} = y_n + h\,\Phi_r$$

em que

$$\Phi_r = c_1\,k_1 + c_2\,k_2 + \cdots + c_r\,k_r$$

$$\begin{aligned}
k_1 &= f(t, y) \\
k_2 &= f(t + \alpha_2\,h, y + h\,\beta_{21}\,k_1) \\
k_3 &= f(t + \alpha_3\,h, y + h\,(\beta_{31}\,k_1 + \beta_{32}\,k_2)) \\
k_4 &= f(t + \alpha_4\,h, y + h\,(\beta_{41}\,k_1 + \beta_{42}\,k_2 + \beta_{43}\,k_3)) \\
&\vdots \\
k_r &= f(t + \alpha_r\,h, y + h\,(\beta_{r1}\,k_1 + \cdots + \beta_{r,r-1}\,k_{r-1}))
\end{aligned}$$

Métodos de Runge-Kutta

Dependendo do valor dos coeficientes:

- $\alpha_i,\ i = 1, 2, 3, \ldots, r$
- $\beta_{ij},\ 2 \leq i \leq r-1,\ 1 \leq j < i.$

Teremos diferentes métodos de Runge-Kutta (RK).

Estes métodos estão disenhados para ter a mesma ordem de convergência que os métodos de Taylor, mas, a sua implementação é mais simples, já que não precisamos calcular as derivadas de f.

Métodos de Runge-Kutta: RK1

Este coincide com o método de Euler explícito ou com o método de Taylor de ordem 1, i.e.,

$$y_{n+1} = y_n + h\, c_1\, k_1$$

- $r = 1$
- $k_1 = f(t_n, y_n), \quad c_1 = 1$

$$y_{n+1} = y_n + h\, f(t_n, y_n)$$

Métodos de Runge-Kutta: RK2

Este método de ordem 2, coincide com o método Trapezoidal explícito:

$$y_{n+1} = y_n + h\,(c_1\,k_1 + c_2\,k_2)$$

- $r = 2$
- $k_1 = f(t_n, y_n), \quad c_1 = \dfrac{1}{2}$
- $k_2 = f(t_n + h, y_n + h\,k_1), \quad c_2 = \dfrac{1}{2}$

$$y_{n+1} = y_n + \frac{h}{2}\,[f(t_n, y_n) + f(t_n + h, y_n + h\,f(t_n, y_n))]$$

Métodos de Runge-Kutta: RK2

Este método de ordem 2, coincide com o método Trapezoidal explícito:

$$y_{n+1} = y_n + h\,(c_1\,k_1 + c_2\,k_2)$$

- $r = 2$
- $k_1 = f(t_n, y_n), \quad c_1 = \dfrac{1}{2}$
- $k_2 = f(t_n + h, y_n + h\,k_1), \quad c_2 = \dfrac{1}{2}$

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_n + h, y_n + h\,f(t_n, y_n))\right]$$

Métodos de Runge-Kutta: RK2 - Exemplo

Consideramos $y' = f(t, y) = t^2 + y^2, \quad y(0) = y_0$:
$n = 0, 1, \ldots$

$$
\begin{aligned}
k_1 &= t_n^2 + y_n^2 \\
k_2 &= (t_n + h)^2 + (y_n + h\,k_1)^2 \\
y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2)
\end{aligned}
$$

Métodos de Runge-Kutta: RK4

Este método é de ordem 4 e se escreve:

$$y_{n+1} = y_n + h\,(c_1\,k_1 + c_2\,k_2 + c_3\,k_3 + c_4\,k_4)$$

- $r = 4$
- $k_1 = f(t_n, y_n), \quad c_1 = \dfrac{1}{6}$
- $k_2 = f(t_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,h\,k_1), \quad c_2 = \dfrac{1}{3}$
- $k_3 = f(t_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,h\,k_2), \quad c_3 = \dfrac{1}{3}$
- $k_4 = f(t_n + h, y_n + h\,k_3), \quad c_4 = \dfrac{1}{6}$

$$y_{n+1} = y_n + \frac{h}{6}\,(k_1 + 2\,k_2 + 2\,k_3 + k_4)$$

Métodos de Runge-Kutta

Em geral se verifica:

- O método é consistente se $c_1 + c_2 + \cdots + c_r = 1$
- Para os métodos de ordem maior do que 1, se deve verificar: $\alpha_i = \sum_{\ell=1}^{i-1} \beta_{i\ell}$

Métodos de Runge-Kutta

Para estudar a estabilidade, vamos considerar o exemplo do decaimento exponencial $y'(t) = -\lambda\, y,\ \lambda > 0$ e o método RK4

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{6}\,(k_1 + 2\,k_2 + 2\,k_3 + k_4) \\ &= y_n \left(1 - h\,\lambda + \frac{1}{2}\,(h\,\lambda)^2 - \frac{1}{6}\,(h\,\lambda)^3 + \frac{1}{24}\,(h\,\lambda)^4\right) \\ &= F(h\,\lambda) y_n \end{aligned}$$

Precisa-se estudar os valores de $h\,\lambda$ para os quais $|F(h\,\lambda)| < 1$, no caso, se verifica $h \lesssim \dfrac{2.78}{\lambda}$.

Métodos de Runge-Kutta: RK4

```
clear ;
#Time interval
t0 = 0.0; tf =  10.0;
#Initialization
y0 = 4.0; h = 1.0; lambda = 1;
t(1) = t0; y(1) = y0; yexact(1) = y0;
n = 1;
myfile = fopen ("rk4.dat", "w");
while t(n) <= tf
  k1 = -lambda*y(n);
  k2 = -lambda*(y(n) + (h/2)*k1);
  k3 = -lambda*(y(n) + (h/2)*k2);
  k4 = -lambda*(y(n) + h*k3);;
  y(n+1) = y(n) + (h/6.0)*(k1 + 2*k2 + 2*k3 + k4);
  t(n+1) = t(n) + h;
  yexact(n+1) = y0*exp(-lambda*t(n+1));
fprintf(myfile, "%f %f \n", t(n), y(n));
  n = n + 1;
end
```

Métodos Previsor-Corretor

Métodos tipo Previsor-Corretor

A idéia é reduzir a complexidade que surge ao utilizar métodos implícitos, em que precisa-se aplicar um método para resolver equações não lineares.

Para ver isto, consideramos o exemplo do método trapezoidal implícito:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right]$$

Métodos tipo Previsor-Corretor

Consideremos:

- **Previsor:** Usar o método de Euler explícito para gerar uma aproximação:

$$\tilde{y}_{n+1} = y_n + h\, f(t_n, y_n)$$

Métodos tipo Previsor-Corretor

Consideremos:

- **Previsor:** Usar o método de Euler explícito para gerar uma aproximação:

$$\tilde{y}_{n+1} = y_n + h\, f(t_n, y_n)$$

- **Corretor:** Usar essa aproximação no método trapezoidal:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right]$$

Métodos tipo Previsor-Corretor

Consideremos:

- **Previsor:** Usar o método de Euler explícito para gerar uma aproximação:

$$\tilde{y}_{n+1} = y_n + h\, f(t_n, y_n)$$

- **Corretor:** Usar essa aproximação no método trapezoidal:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right]$$

Este é o método que foi introduzido anteriormente como método Trapezoidal explícito!

Métodos tipo Previsor-Corretor

Isto poderia ser feito de maneira iterativa também:

1. **Previsor:** $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$

Métodos tipo Previsor-Corretor

Isto poderia ser feito de maneira iterativa também:

1. **Previsor:** $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$
2. $k = 0$

Métodos tipo Previsor-Corretor

Isto poderia ser feito de maneira iterativa também:

1. **Previsor:** $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$
2. $k = 0$
3. **Corretor:** $y_{n+1}^{(k+1)} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1}^{(k)})\right]$

Métodos tipo Previsor-Corretor

Isto poderia ser feito de maneira iterativa também:

1. **Previsor:** $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$
2. $k = 0$
3. **Corretor:** $y_{n+1}^{(k+1)} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_{n+1}^{(k)})\right]$
4. Se $\dfrac{|y_{n+1}^{(k+1)} - y_{n+1}^{(k)}|}{|y_{n+1}^{(k+1)}|} > \text{TOL} \Rightarrow k \leftarrow k+1$ e voltar para 3. Em caso contrario, aceitar $y_{n+1}^{(k+1)}$ como y_{n+1} e passar ao próximo passo.

Métodos tipo Previsor-Corretor

Usando diferentes métodos explícitos como **previsores** (p.e., Euler explícito, Taylor, etc.) ...

e diferentes métodos implícitos como **corretores:** (p.e. Método do Trapézio, Simpson, multipasso, etc),

... teremos diferentes métodos de tipo **Previsor-Corretor**

Métodos tipo Previsor-Corretor

Vários métodos de tipo previsor-corretor aparecem no contexto dos chamados métodos multipasso:

A idéia de um método multipasso é melhorar a precisão também utilizando os valores $y_n,\ y_{n-1},\ y_{n-2}, \ldots,\ y_{n-p}$ e calculando os polinômios interpoladores que passam por esses pontos.

Métodos Multipasso

- Adams-Bashforth de 3 passos: Este é explícito e de ordem 3
  Dados y_0, y_1, y_2, para $n = 2, 3, \ldots$

$$y_{n+1} = y_n + \frac{h}{12}\left[23\, f(t_n, y_n) - 16\, f(t_{n-1}, y_{n-1}) + 5\, f(t_{n-2}, y_{n-2})\right]$$

Métodos Multipasso

- Adams-Bashforth de 3 passos: Este é explícito e de ordem 3
  Dados y_0, y_1, y_2, para $n = 2, 3, \ldots$

$$y_{n+1} = y_n + \frac{h}{12}\left[23\, f(t_n, y_n) - 16\, f(t_{n-1}, y_{n-1}) + 5\, f(t_{n-2}, y_{n-2})\right]$$

- Adams-Bashforth de 4 passos: Este é explícito e de ordem 4
  Dados y_0, y_1, y_2, y_3, para $n = 3, 4, \ldots$

$$y_{n+1} = y_n + \frac{h}{12}\left[55\, f(t_n, y_n) - 59\, f(t_{n-1}, y_{n-1}) + 37\, f(t_{n-2}, y_{n-2}) - 9\, f(t_{n-3}, y_{n-3})\right]$$

Métodos Multipasso

- Adams-Moulton de 2 passos: Este é implícito e de ordem 3
  Dados y_0, y_1, para $n = 1, 2, \ldots$

$$y_{n+1} = y_n + \frac{h}{12}\left[5f(t_{n+1}, y_{n+1}) + 8\,f(t_n, y_n) - f(t_{n-1}, y_{n-1})\right]$$

Métodos Multipasso

- Adams-Moulton de 2 passos: Este é implícito e de ordem 3
  Dados y_0, y_1, para $n = 1, 2, \ldots$

$$y_{n+1} = y_n + \frac{h}{12}\left[5f(t_{n+1}, y_{n+1}) + 8\,f(t_n, y_n) - f(t_{n-1}, y_{n-1})\right]$$

- Adams-Moulton de 3 passos: Este é implícito e de ordem 4
  Dados y_0, y_1, y_2, para $n = 2, 3, \ldots$

$$y_{n+1} = y_n + \frac{h}{24}\left[9\,f(t_{n+1}, y_{n+1}) + 19\,f(t_n, y_n) - 5\,f(t_{n-1}, y_{n-1}) + \right.$$
$$\left. + f(t_{n-2}, y_{n-2})\right]$$

Métodos Multipasso

- Adams-Moulton de 2 passos: Este é implícito e de ordem 3
  Dados y_0, y_1, para $n = 1, 2, \ldots$

$$y_{n+1} = y_n + \frac{h}{12}\left[5f(t_{n+1}, y_{n+1}) + 8\,f(t_n, y_n) - f(t_{n-1}, y_{n-1})\right]$$

- Adams-Moulton de 3 passos: Este é implícito e de ordem 4
  Dados y_0, y_1, y_2, para $n = 2, 3, \ldots$

$$y_{n+1} = y_n + \frac{h}{24}\left[9\,f(t_{n+1}, y_{n+1}) + 19\,f(t_n, y_n) - 5\,f(t_{n-1}, y_{n-1}) + \right.$$
$$\left. + f(t_{n-2}, y_{n-2})\right]$$

Já que estes métodos são implícitos, podemos aplicar o estudado sobre métodos previsores-corretores.

Métodos Multipasso Previsor-Corretor

1. **Previsor** : $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$

Métodos Multipasso Previsor-Corretor

1. **Previsor** : $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$
2. $k = 0$

Métodos Multipasso Previsor-Corretor

1. **Previsor** : $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$
2. $k = 0$
3. **Corretor:**

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{12}\left[5f(t_{n+1}, y_{n+1}^{(k)}) + 8\, f(t_n, y_n) - f(t_{n-1}, y_{n-1})\right]$$

Métodos Multipasso Previsor-Corretor

1. **Previsor** : $y_{n+1}^{(0)} = y_n + h\, f(t_n, y_n)$
2. $k = 0$
3. **Corretor:**

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{12}\left[5f(t_{n+1}, y_{n+1}^{(k)}) + 8\, f(t_n, y_n) - f(t_{n-1}, y_{n-1})\right]$$

4. Se $\frac{|y_{n+1}^{(k+1)} - y_{n+1}^{(k)}|}{|y_{n+1}^{(k+1)}|} > \text{TOL} \Rightarrow k \leftarrow k+1$ e voltar para 3. Em caso contrario, aceitar $y_{n+1}^{(k+1)}$ como y_{n+1} e passar ao próximo passo.