

SME0602 - 2017
Prof. Gustavo Buscaglia

Lista - Solução numérica de EDOs

Material complementar de estudo: Slides do Prof. Roberto Ausas.

A seguinte função retorna o resultado de uma simulação de tempo $t = t_0$ até tempo $t_0 + n_t \delta t$ do sistema

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

onde a função $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ está codificada na função f .

```
function [y time]=umpasso(f,y0,t0,dt,nt)
time(1)=t0;
y(:,1)=y0;
for n=1:nt
    fn(:,1)=feval(f,y(:,n),time(n));
    y(:,n+1)=y(:,n)+dt*fn;
    time(n+1)=time(n)+dt;
endfor
```

Sendo

```
function f = fpend(y,t)
ome=1.;
f(1)=y(2);
f(2)=-ome*ome*sin(y(1));
end
```

ela resolve um pêndulo geométrico, $\theta'' = -\omega^2 \sin(\theta)$, pelo método de Euler explícito com os parâmetros escolhidos. Pode ser usada, por exemplo, da seguinte maneira:

```
[y1 time1]=umpasso("fpend",[-pi+1;0],0,0.1,600);
plot(time1,y1(1,:),"-b","linewidth",2)
```

-
1. A função f pode ser incorporada dentro do código, como na função `pendulo`,

```
function [y time]=pendulo(y0,t0,dt,nt)
ome=1.;
time(1)=t0;
y(:,1)=y0;
for n=1:nt
    ## fn(:,1)=[y(2,n);-ome*ome*sin(y(1,n))];
    ## y(:,n+1)=y(:,n)+dt*fn;
    ### ou equivalentemente
    y(1,n+1)=y(1,n)+dt*y(2,n);
    y(2,n+1)=y(2,n)-ome*ome*sin(y(1,n));
    ###
    time(n+1)=time(n)+dt;
endfor
```

que pode ser usada fazendo

```
[y2 time2]=pendulo([-pi+1;0],0,3,20);
plot(time2,y2(1,:), "or", "linewidth", 2)
```

Alguém propõe usar informação mais atualizada de y quando é possível, substituindo a linha

```
y(2,n+1)=y(2,n)-ome*ome*sin(y(1,n));
```

por

```
y(2,n+1)=y(2,n)-ome*ome*sin(y(1,n+1));
```

Qual é a ordem de convergência do método original e do modificado? Estime qual o máximo passo de tempo em que os métodos são estáveis. Rode os códigos e compare os resultados.

Ajuda: Chamando $\theta = y_1$ e $\zeta = y_2$, o método original é

$$\begin{aligned}\theta_{n+1} &= \theta_n + \delta t \zeta_n \\ \zeta_{n+1} &= \zeta_n - \omega^2 \delta t \sin \theta_n\end{aligned}$$

Que, escrito de maneira que seja mais parecido da equação,

$$\begin{aligned}\frac{\theta_{n+1} - \theta_n}{\delta t} - \zeta_n &= 0 \\ \frac{\zeta_{n+1} - \zeta_n}{\delta t} + \omega^2 \sin \theta_n &= 0\end{aligned}$$

nos permite escrever o erro de truncamento como

$$\tau_{n+1} = \left(\frac{\frac{\theta(t_{n+1}) - \theta(t_n)}{\delta t} - \zeta(t_n)}{\frac{\zeta(t_{n+1}) - \zeta(t_n)}{\delta t} + \omega^2 \sin(\theta(t_n))} \right)$$

Agora por desenvolvimento de Taylor se mostra que $\tau_{n+1} = O(\delta t)$.

Com raciocínio parecido é obtido o erro de truncamento para o método modificado

$$\tau_{n+1} = \left(\frac{\frac{\theta(t_{n+1}) - \theta(t_n)}{\delta t} - \zeta(t_n)}{\frac{\zeta(t_{n+1}) - \zeta(t_n)}{\delta t} + \omega^2 \sin(\theta(t_{n+1}))} \right)$$

e com séries de Taylor se calcula a ordem, que poderia ser $O(\delta t^2)$, mas não tenho certeza.

Quanto à pergunta do máximo passo de tempo, devemos voltar as equações do problema, e ver que

$$f(t, y) = \begin{pmatrix} y_2 \\ -\omega^2 \sin y_1 \end{pmatrix}$$

e portanto a matriz jacobiana é

$$J(t, y) = \begin{pmatrix} 0 & 1 \\ -\omega^2 \cos y_1 & 0 \end{pmatrix}$$

cujos autovalores são $\lambda = \pm i \omega \sqrt{\cos y_1}$ e assim esperamos que o método seja estável se

$$\delta t < \frac{2}{\omega} .$$

Também podemos calcular o passo de tempo limite de maneira exata se consideramos a equação linearizada,

$$\begin{aligned}\theta_{n+1} &= \theta_n + \delta t \zeta_n \\ \zeta_{n+1} &= \zeta_n - \omega^2 \delta t \theta_n\end{aligned}$$

escrita como

$$\begin{pmatrix} \theta_{n+1} \\ \zeta_{n+1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \delta t \\ -\omega^2 \delta t & 1 \end{pmatrix}}_S \begin{pmatrix} \theta_n \\ \zeta_n \end{pmatrix}$$

e vemos para que valor de $\omega\delta t$ a matriz S tem um autovalor em módulo > 1 .

Os autovalores satisfazem

$$(\lambda - 1)^2 + (\omega\delta t)^2 = 0$$

então,

$$\lambda = 1 \pm i\omega\delta t$$

isto é, sempre tem um autovalor que leva a crescimento exponencial da solução numérica, como já foi mostrado na aula.

Para o método modificado, por ser

$$\begin{aligned}\theta_{n+1} &= \theta_n + \delta t \zeta_n \\ \zeta_{n+1} &= \zeta_n - \omega^2 \delta t \theta_{n+1} = (1 - \omega^2 \delta t^2) \zeta_n - \omega^2 \delta t \theta_n\end{aligned}$$

resulta

$$\begin{pmatrix} \theta_{n+1} \\ \zeta_{n+1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \delta t \\ -\omega^2 \delta t & 1 - \omega^2 \delta t^2 \end{pmatrix}}_S \begin{pmatrix} \theta_n \\ \zeta_n \end{pmatrix}$$

A matriz S tem determinante 1, de fato os autovalores são um par de complexos conjugados de módulo 1 se $\omega\delta t \leq 2$ (verificar isto), e se $\omega\delta t > 2$ passam a ser dois números reais, um deles maior que 1 em módulo. Então esse método é estável se $\delta t \leq 2/\omega$, como tínhamos previsto.

-
2. Por interpolação linear no tempo se obtem o método

$$y_{n+1} = y_{n-1} + 2\delta t f(t_n, y_n)$$

Esse método é explícito ou implícito? De um passo ou de mais passos? De que ordem de consistência? É estável? Qual é o seu passo de tempo limite?

Ajuda: Reescrevemos o método levando-o na forma que assemelha a equação diferencial

$$\frac{y_{n+1} - y_{n-1}}{2\delta t} - f(t_n, y_n) = 0$$

Assim, o erro de truncamento desse passo é

$$\tau_{n+1} = \frac{y(t_{n+1}) - y(t_{n-1})}{2\delta t} - f(t_n, y(t_n))$$

agora por desenvolvimento de Taylor

$$y(t_{n\pm 1}) = y(t_n) \pm y'(t_n) \delta t + y''(t_n) \frac{\delta t^2}{2} + \dots$$

se chega ao cálculo da ordem, deveria ser $O(\delta t^2)$.

Para calcular o passo de tempo limite deve ser considerado o método quando $f(t, y) = \lambda y$, assim,

$$y_{n+1} = y_{n-1} + 2\delta t \lambda y_n$$

Aqui se usa o truque de definir $z_n = y_{n-1}$, de maneira que

$$\begin{aligned}y_{n+1} &= z_n + 2\delta t \lambda y_n \\ z_{n+1} &= y_n\end{aligned}$$

ou, equivalentemente,

$$\begin{pmatrix} y_{n+1} \\ z_{n+1} \end{pmatrix} = \underbrace{\begin{pmatrix} 2\delta t \lambda & 1 \\ 1 & 0 \end{pmatrix}}_S \begin{pmatrix} y_n \\ z_n \end{pmatrix}$$

É fácil ver que dos autovalores da matriz S sempre tem um que é de valor absoluto > 1 , já que o produto deles deve ser -1 (igual a $\det S$). Verifique isto.

Portanto o método é bonito, de segunda ordem, mas incondicionalmente instável, isto é, explode para qualquer δt .

-
3. Modifique a função **pendulo** para que utilize o método de Euler implícito. Resolva a equação que aparece por Newton-Raphson.

-
4. Escreva o código Octave de uma função **f** tal que usando a função **umpasso** seja resolvida a equação

$$I''(t) + 2\alpha I'(t) + k I(t) = \sin(2t)$$

com condição inicial $I(0) = 0$, $I'(0) = 1$. Os parâmetros α e k são dados (por exemplo, iguais a 1).

Calcule o passo de tempo limite de estabilidade.

Escreva como seria utilizada a função **umpasso** na linha de comando de Octave.

Boa prática!