Tutorial: uso do Scilab para a integração numérica de equações diferenciais ordinárias

Prof. Dr. Renato Maia Matarazzo Orsino

Departamento de Engenharia Mecânica

Escola Politécnica – Universidade de São Paulo – Brasil

PME 3200 – Mecânica II

12 de Março de 2019

Conteúdo

- Ferramentas computacionais para integração numérica de EDOs
- Objetivo do tutorial
- Sintaxe do Scilab – aspectos básicos
- Integração numérica no Scilab – função `ode`
- Aplicação – dinâmica de um pêndulo

Ferramentas computacionais para integração numérica

Ferramentas gratuitas e de código aberto (open-source)

- Scilab (`www.scilab.org`)
  - Linha de comando e execução de *scripts* – função `ode`
  - Ambiente gráfico Xcos
- GNU Octave (`www.gnu.org/software/octave/`)
  - Linha de comando e execução de *scripts* – função `lsode`

Ferramentas comerciais

- MATLAB (`www.mathworks.com/products/matlab.html`)
  - Linha de comando e execução de *scripts* – função `ode45`
  - Ambiente gráfico Simulink
- Wolfram Mathematica[a] (`www.wolfram.com/mathematica/`)
  - Computação simbólica e numérica – função `NDSolve`

[a] Licença estudante gratuita via `https://atendimento-prod.sti.usp.br/otrs/public.pl?Action=PublicFAQZoom;ItemID=287`.

Objetivo

Atenção

- Este tutorial não visa prover um curso introdutório geral sobre programação em Scilab, tampouco sobre sua linguagem de programação.

- O objetivo deste tutorial é ilustrar o uso do Scilab no específico problema de integração de equações diferenciais ordinárias de segunda ordem.
- Ao final deste tutorial, espera-se que o aluno seja capaz de:
    1. Interpretar um código-exemplo desenvolvido especialmente para este problema.
    2. Modificar este código para a integração de outras equações diferenciais.

Sintaxe do Scilab – aspectos básicos

- Não é preciso declarar variáveis, basta definir seu valor:

```
j = 1;                 // inteiro
r0 = 1.0;              // ponto flutuante (1.0)
r1 = 210.E+9;          // ponto flutuante (210. * 10^9)
r2 = %pi/2;            // ponto flutuante (pi/2)
st = "hello";          // string
ml = [1, 0, 2.];       // matriz-linha ou vetor
mc = [1; 0; 2.];       // matriz-coluna ou vetor
M = [1, 2, 3.2; -4, -7.5, %pi]; // matriz 2 x 3
```

- Vetores com elementos igualmente espaçados:

```
v0 = 0.0;              // primeiro elemento
vn = 10.0;             // último elemento
dv = 0.1;              // espaçamento entre elementos
v = v0:dv:vn;          // vetor
```

Sintaxe do Scilab – aspectos básicos

- Índices de linhas e colunas iniciam em 1.
- Acessar ou modificar partes de um vetor:

```
ml = [-3., 0, 2.]; // matriz-linha ou vetor
q1 = ml(1);        // q1 recebe elemento 1 de ml (-3.)
ml(2) = 2 * %pi;   // modifica o elemento 2 de ml
```

- Acessar ou modificar partes de uma matriz:

```
M = [1, 2, 3.2; -4, -7.5, %pi]; // matriz 2 x 3
m23 = M(2, 3);    // m23 recebe linha 2, coluna 3 de M
r1 = M(1, :);     // r1 recebe linha 1 de M
c2 = M(:, 2);     // c2 recebe coluna 2 de M
M(2, 2) = -5.;    // modifica linha 2, coluna 2 de M
```

Sintaxe do Scilab – aspectos básicos

- Definição de funções:

```
function dy = pendulo(t, y)

    dy(1) = y(2);
    dy(2) = - (wn^2) * sin(y(1)) + A * sin(w * t);

    end
```

- O nome da função é `pendulo`.
- As variáveis de entrada (*input*) são t e y: t deve ser um escalar; y deve ser um vetor com ao menos dois elementos.
- A variável de saída (*output*) é dy, que será um vetor com exatamente dois elementos.
- As variáveis wn, A e w, vistas pela função como constantes, devem ser definidas externamente ao corpo da função.

Integração numérica no Scilab – função `ode`

```
Y = ode(y0, t0, t, f);
```

- Fornece uma solução numérica para a equação diferencial ordinária de primeira ordem com condição inicial $y = y0$ em $\tau = t0$:

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}\tau} = f(\tau, y) \\ y\big|_{\tau=t0} = y0 \end{cases}$$

- A variável y pode ser escalar ou vetor; τ deve ser um escalar (pois é uma equação diferencial *ordinária*).
- Assim, a entrada $y0$ da função `ode` deve ter as mesmas dimensões de y e a variável $t0$ deve ser um escalar.

Integração numérica no Scilab – função ode

```
Y = ode(y0, t0, t, f);
```

- A entrada t da função `ode` deve ser um vetor com um número finito de instantes de tempo τ em um intervalo $\tau \in [t0, tf]$ para os quais os valores de y serão numericamente computados. Exemplo:

```
t0 = 0.0;        // instante inicial
tf = 15.0;       // instante final
dt = 0.01;       // incremento de tempo
t = t0:dt:tf;    // vetor t
```

- A entrada f da função `ode` deve ser o nome de uma função que tem como saída o valor de $\dfrac{dy}{d\tau}$ e como entrada (τ, y).

Integração numérica no Scilab – função ode

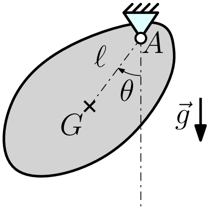
```
Y = ode(y0, t0, t, f);
```

- A saída Y da função ode é uma matriz da forma:

$$Y = \begin{bmatrix} y(1)|_{\tau=t(1)} & y(1)|_{\tau=t(2)} & \cdots & y(1)|_{\tau=t(k)} & \cdots \\ y(2)|_{\tau=t(1)} & y(2)|_{\tau=t(2)} & \cdots & y(2)|_{\tau=t(k)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ y(j)|_{\tau=t(1)} & y(j)|_{\tau=t(2)} & \cdots & y(j)|_{\tau=t(k)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

- Cada linha corresponde a uma variável do vetor y; cada coluna corresponde ao valor do vetor y no instante de tempo τ que ocupa a respectiva posição no vetor t.

Exemplo de aplicação - dinâmica de um pêndulo



Parâmetros

- g: aceleração da gravidade
- m: massa do pêndulo
- ℓ: distância do centro de massa à articulação.
- r: raio de giração com respeito ao eixo Az

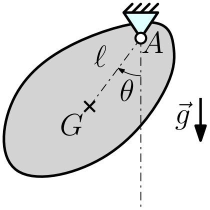
Teorema da Quantidade de Movimento Angular

$$\vec{H}_A = -mr^2\dot{\theta}\vec{k} \quad \text{e} \quad \vec{M}_A = mg\ell \sin\theta \vec{k}$$

$$\dot{\vec{H}}_A = \vec{M}_A \quad \Rightarrow \quad -mr^2\ddot{\theta} = mg\ell \sin\theta$$

$$\ddot{\theta} = -\omega_n^2 \sin\theta \quad \text{com} \quad \omega_n = \sqrt{\frac{g\ell}{r^2}}$$

Exemplo de aplicação - dinâmica de um pêndulo



Vetor de estados

$$y = \begin{bmatrix} y(1) \\ y(2) \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$

Equação de movimento – Forma de espaço de estados

$$\dot{y} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ -\omega_n^2 \sin\theta \end{bmatrix}$$

$$\dot{y} = \begin{bmatrix} \dot{y}(1) \\ \dot{y}(2) \end{bmatrix} = \begin{bmatrix} y(2) \\ -\omega_n^2 \sin(y(1)) \end{bmatrix} = \texttt{pendulo}(\tau, y)$$

Exemplo de aplicação - dinâmica de um pêndulo

```
clear;                                  // limpa workspace
g = 9.8; l = 1./2; r = 1./3;            // parâmetros
wn = sqrt(g*l/r^2);                     // frequência natural
T = 2*%pi/wn;                           // período (linear)

function dy = pendulo(t, y)
    dy(1) = y(2);
    dy(2) = - (wn^2) * sin(y(1));
    end

t = T * (0:0.01:10);                    // vetor tempo
y0 = [%pi/2; 0];                        // condições iniciais

Y = ode (y0, 0., t, pendulo);           // integração numérica

theta = Y(1,:);                         // linha 1 de Y
scf(1);                                 // janela gráfica 1
plot(t, theta);                         // plot theta vs. t
```