

Tutorial para integrar equações de movimento utilizando o Scilab

Raul Gonzalez Lima

6 de Maio de 2011

Conteúdo

1	Introdução	5
1.1	O integrador	5
1.2	Leituras recomendadas	6
1.3	Autoria dos exemplos	6
2	Exemplo: duas massas, duas molas e dois amortecedores	7
2.1	Definindo o conjunto de equações diferenciais no Scilab	8
2.2	<i>Script</i> com o conjunto de comandos de integração	9
2.3	Para executar o <i>script</i>	11
3	Exemplo: veículo 2D passando por lombada	13
3.1	Definindo o conjunto de equações diferenciais no Scilab	15
3.2	<i>Script</i> com o conjunto de comandos de integração	16
3.3	Para executar o <i>script</i>	20
4	Exemplo: partícula que colide contra o solo	23
4.1	As equações do movimento	24
4.2	O critério algébrico para encerrar a integração numérica	25
4.3	<i>Script</i> com o conjunto de comandos de integração	26
4.4	Detalhamento da função <i>integra_ate_choque</i>	29
4.5	Para executar o <i>script</i>	29

Capítulo 1

Introdução

Nesta apostila descreve-se, através de alguns exemplos, como integrar no tempo equações de movimento de sistemas de corpos rígidos. Recomenda-se a elaboração de dois arquivos, *scripts*, diferentes. Um deles funciona como um programa principal que invoca o integrador (a função *ode(.)*), que por sua vez, invoca uma função que contém as equações do movimento. O programa principal será um arquivo ASCII com um nome arbitrário mas que deve ter terminação *.sci*. A função com as equações do movimento será descrita em um arquivo ASCII com nome arbitrário mas que deve ter terminação *.sce*. A recomendação do uso de dois arquivos é questão de estilo e pode ser abandonada.

Na disciplina PME2200 Mecânica B o aluno desenvolve habilidades para determinar as equações do movimentos de sistemas de corpos rígidos, seja através dos teoremas do movimento do centro de massa e da quantidade de movimento angular ou através das equações de Lagrange. A integração numérica das equações do movimento permite visualizar a resposta destes sistemas, trazendo significado físico a cada termo de cada equação. É, portanto, poderoso instrumento didático para transmitir o *poder informativo* das equações do movimento.

1.1 O integrador

O integrador no Scilab para problemas de valores iniciais chama-se ODE. A maneira mais simples de invocar o integrador é

```
y=ode(t0,y0,t,g);
```

onde t_0 é o instante inicial de integração, y_0 é o vetor de estado inicial, t é o vetor de instantes em que se deseja a resposta do sistema e g é a função a ser

integrada. Noutras palavras, t_0 e y_0 são as condições iniciais de um sistema equações diferenciais do tipo

$$\frac{dy}{dt} = g(t, y) \quad (1.1)$$

onde a função g pode ser uma função escalar ou um vetor de funções conforme a eq. 1.2,

$$g(t, y) = \begin{bmatrix} g_1(t, y_1, y_2, \dots, y_n) \\ g_2(t, y_1, y_2, \dots, y_n) \\ \dots \\ g_n(t, y_1, y_2, \dots, y_n) \end{bmatrix}, \quad (1.2)$$

quando $g(t, y)$ é um vetor de funções, y é um vetor de mesma dimensão. Para poder invocar este integrador é preciso definir as condições iniciais, o vetor t e a função que será integrada.

1.2 Leituras recomendadas

Informações adicionais podem ser encontradas nos seguintes sítios

www.dca.ufrn.br/~pmotta/sciport-3.0.pdf

www.dca.ufrn.br/~pmotta/scilab-slides3.pdf

http://cermics.enpc.fr/scilab_new/site/Liens/intro/intro.html

1.3 Autoria dos exemplos

O exemplo do veículo passando por uma lombada foi desenvolvido por diversas pessoas, entre as pessoas que formularam o problema encontram-se o Prof. Dr. Roberto Martins de Souza, a Dra. Sara Aida Rodriguez Pulecia e o autor deste tutorial. O exemplo da partícula que colide com solo foi inspirado da apostila tutorial do ambiente gráfico de simulação e com a ajuda do doutorando Marcos Rabelo. A equipe de professores de Mecânica B agradece a possibilidade de reprodução destes exemplos.

Capítulo 2

Exemplo: duas massas, duas molas e dois amortecedores

Problemas de valores iniciais definidos por conjuntos de equações diferenciais ordinárias podem ser integrados utilizando o Scilab e linhas de comandos. Neste tutorial desenvolve-se um exemplo de integração no tempo de um sistema com duas massas, duas molas e dois amortecedores, conforme a fig. 2.1.

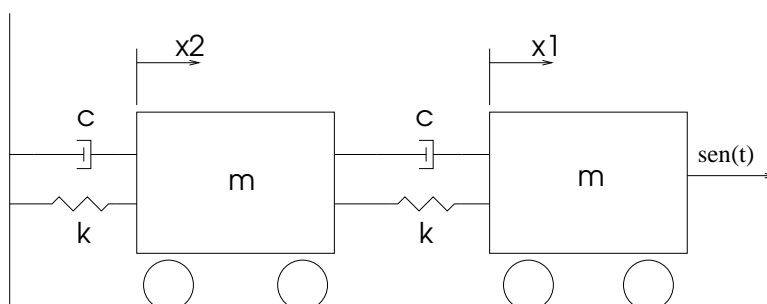


Figura 2.1: Sistema com duas massas, duas molas e dois amortecedores

Decorre da Segunda Lei de Newton e do modelo de amortecimento viscoso,

$$m\ddot{x}_1 = -k(x_1 - x_2) - c(\dot{x}_1 - \dot{x}_2) + \sin(t) \quad (2.1)$$

e

$$m\ddot{x}_2 = -kx_2 - cx_2 + k(x_1 - x_2) + c(\dot{x}_1 - \dot{x}_2) \quad (2.2)$$

Rearrajando estas duas equações para isolar as acelerações do lado esquerdo, temos,

$$\ddot{x}_1 = -\frac{k}{m}(x_1 - x_2) - \frac{c}{m}(\dot{x}_1 - \dot{x}_2) + \sin(t) \quad (2.3)$$

e

$$\ddot{x}_2 = \frac{k}{m}(x_1 - 2x_2) + \frac{c}{m}(\dot{x}_1 - 2\dot{x}_2). \quad (2.4)$$

Este sistema de duas equações diferenciais de segunda ordem deve ser transformado em um sistema de quatro equações diferenciais de primeira ordem para que possam ser utilizados integradores clássicos. Para promover esta transformação define-se um vetor y em função dos deslocamentos e das derivadas,

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{pmatrix} \quad (2.5)$$

e sua derivada no tempo \dot{y} torna-se

$$\dot{y} = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{pmatrix} = \begin{pmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{pmatrix} \quad (2.6)$$

Em termos deste vetor de estado, a eq. 2.3 e a eq. 2.4 podem ser representadas por

$$\begin{pmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{pmatrix} = \begin{pmatrix} \dot{x}_1 \\ -\frac{k}{m}(x_1 - x_2) - \frac{c}{m}(\dot{x}_1 - \dot{x}_2) + \sin(t) \\ \dot{x}_2 \\ \frac{k}{m}(x_1 - 2x_2) + \frac{c}{m}(\dot{x}_1 - 2\dot{x}_2) \end{pmatrix} \quad (2.7)$$

2.1 Definindo o conjunto de equações diferenciais no Scilab

Uma das maneiras de definir um conjunto de equações diferenciais no Scilab é através de um arquivo texto, preparado em um editor de textos comum, e que recebe o sufixo *.sci*, por convenção. Assim, o conjunto de equações do nosso exemplo, eq. 2.7, foi definido pelo arquivo *exemplo.sci*, reproduzido nas linhas que seguem

2.2. SCRIPT COM O CONJUNTO DE COMANDOS DE INTEGRAÇÃO

```
function [dy]=g(t,y)
    y1=y(2);
    y2=-k*(y(1)-y(3))/m-c*(y(2)-y(4))/m+sin(t);
    y3=y(4);
    y4=k*(y(1)-2*y(3))/m+c*(y(2)-2*y(4))/m;
    dy=[y1;y2;y3;y4];
endfunction
```

Resta ainda escrever um *script* onde se definem as constantes m , k , c , o vetor de instantes em que se deseja a resposta e como deve ser o gráfico que mostra a resposta.

2.2 *Script* com o conjunto de comandos de integração

Nesta seção descreve-se um arquivo com os comandos necessários à integração da função $g(t, y)$. Este arquivo de comandos, *exemplo.sce*, tem, por convenção, sufixo *.sce*. Inicialmente definem-se as variáveis m , k e c . Segue uma leitura da função $g(t, y)$. Definem-se as condições iniciais $t_{inicial}$ e y_0 . Definem-se ainda o incremento de tempo de integração Dt , o instante final de integração t_{final} e o vetor de instantes em que se deseja a resposta do sistema $y(t)$.

```
// Rigidez em N/m
k=1.0;

// Coeficiente de amortecimento em Ns/m
c=0.001;

// Massa em kg
m=1.0;

// Carrega a definicao da funcao dy=g(t,y)
getf exemplo.sci;

// Vetor estado inicial
y0=[0;0;0;0];

// Instante inicial de integracao
```

10CAPÍTULO 2. EXEMPLO: DUAS MASSAS, DUAS MOLAS E DOIS AMORTECEDORES

```
t_inicial=0.0;

// Incremento de tempo desejado na resposta
Dt=0.01;

// Instante final de integracao
t_final=16.0;

// Vetor de instantes desejados na resposta
t=[t_inicial:Dt:t_final];

// Rotina de integracao da funcao dy=g(t,y)
y=ode(y0,t_inicial,t,g);

// Comando para gerar grafico dos deslocamentos
scf(0)
plot2d(t',[y(1,:)',y(3,:)]',leg="x1@x2");
xtitle('Deslocamento das massas','tempo (s)','Deslocamento (m)');

// Guarda o grafico em arquivo eps
filename='deslocamento';
xs2eps(0,filename);
```

Os últimos comandos promovem a integração numérica propriamente dita, geram um gráfico dos deslocamentos das duas massas, fig. 2.2, e criam um arquivo em Encapsulated Postscript, *deslocamento.eps*.¹

¹No comando `plot2d` o sinal de apóstrofe indica transposta de vetor ou matriz, ou seja, t' é a transposta de t . Outro detalhe que merece explicações é o termo `y(1,:)`, leia-se *tome todas as colunas da primeira linha da matriz y*.

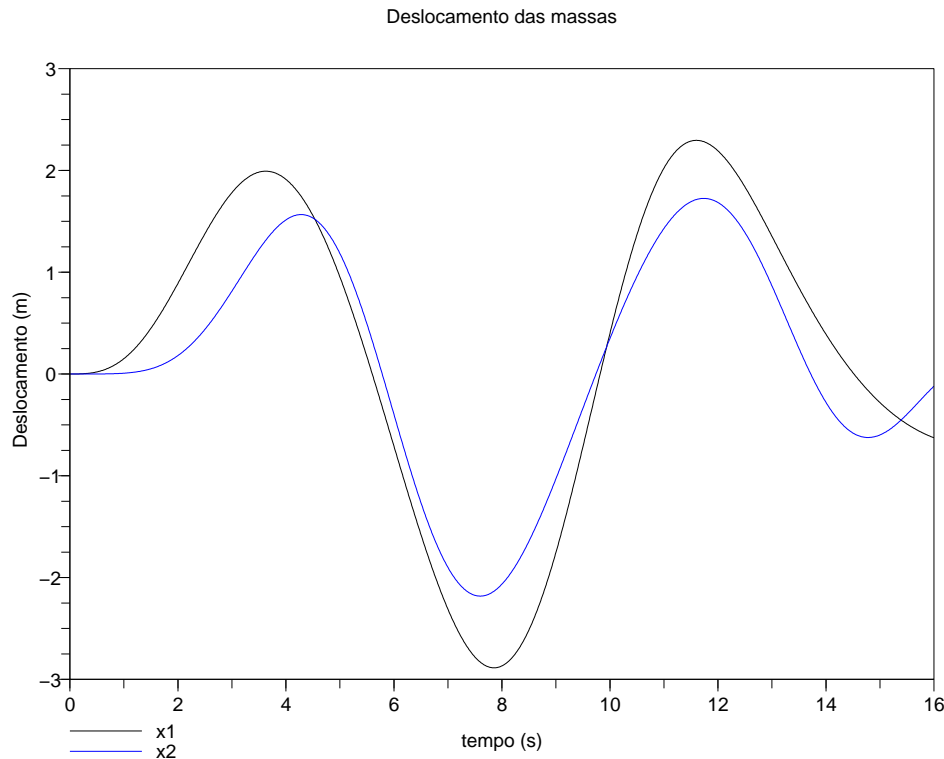


Figura 2.2: Deslocamento calculado das duas massas

2.3 Para executar o *script*

Após invocar o Scilab através de seu ícone, aparece uma tela com um *prompt* que é o símbolo `-->`. A primeira providência a ser tomada é descobrir qual é o diretório de trabalho atual. Para conhecer o diretório de trabalho atual deve-se digitar no *prompt* `pwd` e pressionar a tecla `<enter>`. Se acaso os arquivos de comandos e de definição da função, `exemplo.sce` e `exemplo.sci`, não se encontram no diretório de trabalho, deve-se alterar o diretório de trabalho através do comando `chdir()`. Utilize o *help* do scilab para entender o comando `chdir()` no particular sistema operacional que você estiver utilizando.

Finalmente, para executar o *script* de comandos que se encontra no diretório de trabalho, `exemplo.sce`, basta digitar

```
exec('exemplo.sce');
```

ou simplesmente

12CAPÍTULO 2. EXEMPLO: DUAS MASSAS, DUAS MOLAS E DOIS AMORTECEDORES

```
exec exemplo.sce;
```


14CAPÍTULO 3. EXEMPLO: VEÍCULO 2D PASSANDO POR LOMBADA

Após percorrer uma distância L_1 , o veículo tem de suplantare um obstáculo em sua trajetória, como indicado na fig. 3.2. Durante a passagem pelo obstáculo, o veículo permanece com velocidade constante . O pavimento tem altura definida por:

$$h(x) = \left\{ \begin{array}{ll} 0 & \text{se } x \leq L_1 + L/2 \\ \frac{H}{2} \left(1 - \cos \frac{2\pi}{L_2} x \right) & \text{se } L_1 + L/2 < x < L_1 + L_2 + L/2 \\ 0 & \text{se } x \geq L_1 + L_2 + L/2 \end{array} \right\} \quad (3.1)$$

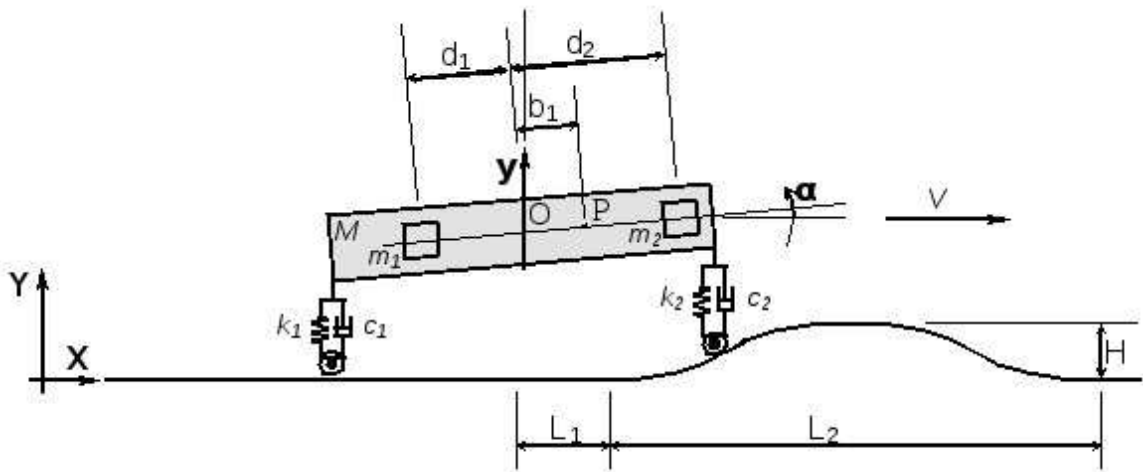


Figura 3.2: Veículo sobre a lombada

Admitem-se como dados $X_0 = 0$; $L = 5$ m; $L_1 = 50$ m; $L_2 = 3,7$ m; $H = 0,1$ m; $b = 0,1$ m; $b_1 = 0,7$ m; $M = 400$ kg; $V = 18$ m/s e $g = 9,8$ m/s². Sugere-se utilizar 0,001 s como intervalo de integração. Considera-se ainda $m_1 = m_2 = 120$ kg; $d_1 = d_2 = 2,5$ m; $k_1 = k_2 = 30.000$ N/m; $c_1 = c_2 = 2.100$ Ns/m ; $y_O(0) = l_0 = 0,3$ m e $\alpha(0) = 0$.

As equações do movimento foram determinadas para o movimento de arfagem α , ver eq. 3.2. A dedução pode ser feita através dos teoremas do movimento do centro de massa e do teorema da quantidade de movimento angular ou através da Mecânica Analítica.

3.1. DEFININDO O CONJUNTO DE EQUAÇÕES DIFERENCIAIS NO SCILAB15

$$\begin{aligned}
& \text{sen}(\alpha)(M_2d_2 - M_1d_1)\ddot{x}_O \\
& + \text{cos}(\alpha)(M_1d_1 - M_2d_2)\ddot{y}_O \\
& \quad + (M_1d_1^2 + M_2d_2^2)\ddot{\alpha} \\
& + \text{cos}(\alpha)(M_1d_1 - M_2d_2)g \\
& - k_1L \text{cos}(\alpha)(l_0 - y_O - L\text{sen}(\alpha) + y_A(t)) \\
& + k_2L \text{cos}(\alpha)(l_0 - y_O + L\text{sen}(\alpha) + y_B(t)) = \\
& - c_1L \text{cos}(\alpha)(\dot{y}_O + L\dot{\alpha} \text{cos}(\alpha)) - \dot{y}_A(t) \\
& + c_2L \text{cos}(\alpha)(\dot{y}_O - L\dot{\alpha} \text{cos}(\alpha)) - \dot{y}_B(t)
\end{aligned} \tag{3.2}$$

e para a oscilação vertical y_O , ver eq. 3.3.

$$\begin{aligned}
& (M_1 + M_2 + M + m)\ddot{y}_O \\
& + \text{cos}(\alpha)(M_1d_1 - M_2d_2)\ddot{\alpha} \\
& + \text{sen}(\alpha)(M_2d_2 - M_1d_1)\dot{\alpha}^2 \\
& + (M_1 + M_2 + M + m)g \\
& - k_1(l_0 - y_O - L\text{sen}(\alpha) + y_A(t)) \\
& - k_2(l_0 - y_O + L\text{sen}(\alpha) + y_B(t)) = \\
& - c_1(\dot{y}_O + L\dot{\alpha} \text{cos}(\alpha)) - \dot{y}_A(t) \\
& - c_2(\dot{y}_O - L\dot{\alpha} \text{cos}(\alpha)) - \dot{y}_B(t)
\end{aligned} \tag{3.3}$$

3.1 Definindo o conjunto de equações diferenciais no Scilab

Uma das maneiras de definir um conjunto de equações diferenciais no Scilab é através de um arquivo texto, preparado em um editor de textos comum, e que recebe o sufixo *.sci*, por convenção. Assim, o conjunto de equações do nosso exemplo, eq. 3.3 e eq. 3.2, foi definido pelo arquivo *emsc.sci*, reproduzido no *script* que segue. Observa-se que a equação, eq. 3.1, que descreve a altura da via $h(x)$ foi codificada no início deste arquivo, junto com as equações diferenciais, ver linhas 2 a 25 do arquivo *emsc.sci*. A última linha define o vetor \dot{x} , que no script recebe o nome de *dy*.

```

function [dy]=car(t,y)
if t<tA1 then
  yA=0.0;
  dyA=0.0;
else
  if t<tA2 then
    yA=H*(1-cos(2*pi*(v*t-L1)/L2))/2;
    dyA=H*(sin(2*pi*(v*t-L1)/L2))*pi*v/L2;
  
```

16CAPÍTULO 3. EXEMPLO: VEÍCULO 2D PASSANDO POR LOMBADA

```
    else
        yA=0.0;
        dyA=0.0;
    end
end
if t<tB1 then
    yB=0.0;
    dyB=0.0;
else
    if t<tB2 then
        yB=H*(1-cos(2*pi*(v*t-L1-L)/L2))/2;
        dyB=H*(sin(2*pi*(v*t-L1-L)/L2))*pi*v/L2;
    else
        yB=0.0;
        dyB=0.0;
    end
end
end
F=-A*g+B*sin(y(3))*y(4)*y(4)+k1*(L0-y(1)
-L*sin(y(3))/2+yA)+k2*(L0-y(1)+L*sin(y(3))/2+yB)
-c1*(y(2)+L*y(4)*cos(y(3))/2-dyA)-c2*(y(2)-
L*y(4)*cos(y(3))/2-dyB);
H=-B*g*cos(y(3))+k1*L*cos(y(3))*(L0-y(1)-L*sin(y(3))/2+yA)/2
-k2*L*cos(y(3))*(L0-y(1)+L*sin(y(3))/2+yB)/2
-c1*L*cos(y(3))*(y(2)+L*y(4)*cos(y(3))/2-dyA)/2
+c2*L*cos(y(3))*(y(2)-L*y(4)*cos(y(3))/2-dyB)/2;
determinante=A*C-B*B*cos(y(3))*cos(y(3));
dy1=y(2);
dy2=(C*F-B*cos(y(3))*H)/determinante;
dy3=y(4);
dy4=(A*H-B*cos(y(3))*F)/determinante;
dy5=dy2;
dy=[dy1;dy2;dy3;dy4;dy5];
endfunction
```

3.2 *Script* com o conjunto de comandos de integração

Nesta seção descreve-se um arquivo com os comandos necessários à integração da função $g(t, y)$. Este arquivo de comandos, *exemplo.sce*, tem, por

3.2. SCRIPT COM O CONJUNTO DE COMANDOS DE INTEGRAÇÃO 17

convenção, sufixo *.sce*. Inicialmente definem-se as variáveis, por exemplo, m , m_1 , m_2 , k_1 e c_1 . Segue uma leitura da função $g(t, y)$. Definem-se as condições iniciais $t_{inicial}$ e y_0 . Definem-se ainda o incremento de tempo de integração Dt , o instante final de integração t_{final} e o vetor de instantes em que se deseja a resposta do sistema $y(t)$.

```
clear all
k1=30000;
k2=30000;
d1=2.50;
d2=2.50;
M=400;
M1=120;
M2=120;
L=5.0;
L1=90.0;
L2=3.7;
L0=0.3;
Y_0=L0;
b=0.1;
c1=2100.0;
c2=2100.0;
v=48.0;
pi=3.1416;
g=9.8;
H=0.1;
b1=0.7;

y0=[L0; 0.0; 0.0; 0.0; 0.0];

J0=M*(b*b+L*L)/12;

A=M1+M2+M;
B=(M1*d1-M2*d2);
C=M1*d1*d1+M2*d2*d2+M*(b*b+L*L)/12;

tA1=L1/v
tA2=(L1+L2)/v
tB1=(L1+L)/v
tB2=(L1+L2+L)/v
```

18CAPÍTULO 3. EXEMPLO: VEÍCULO 2D PASSANDO POR LOMBADA

```
t_inicial=0.0;
Dt=0.001;
t_final=10.0;

t=[t_inicial:Dt:t_final];

getf emsc3.sci;

ys=ode('stiff',y0,t_inicial,t,car);

scf(1)
plot2d(t',[ys(1,:)']);
xlabel('Deslocamento vertical do ponto 0 vs. tempo','t','Yo');

scf(2)
plot2d(t',[ys(3,:)']);
xlabel('Arfagem vs. tempo','t','Angulo alfa');

for i=tA1/Dt:t_final/Dt
    yb1(i)=ys(1,i)+b1*sin(ys(3,i));
end

for i=1:tA1/Dt
    yb1(i)=ys(1,tA1/Dt)+b1*sin(ys(3,tA1/Dt));
end

desconforto=max(abs(yb1(:)-yb1(t_final/Dt-1)))

scf(3)
plot2d(yb1);
xlabel('Altura na posição do piloto vs. tempo','t','Yb1 (m)');
```

Três gráficos são gerados, na primeira tela *scf(1)* está a história temporal do deslocamento vertical variável y_o , na segunda tela *scf(2)* está a história temporal do ângulo de arfagem α e na terceira tela *scf(3)* está o deslocamento vertical no assento do piloto y_{b1} , que depende tanto de α quanto de y_o . A fig. 3.3 apresenta o gráfico de $y_o = y_o(t)$ e a fig. 3.4 apresenta o gráfico de $\alpha_o = \alpha_o(t)$ para esta condição. Observe que o gráfico de y_o permite identificar dois instantes distintos do movimento, um deles, inicial, associado à estabilização de y_o em um valor de equilíbrio e outro associado à passagem do veículo pelo obstáculo.

3.2. SCRIPT COM O CONJUNTO DE COMANDOS DE INTEGRAÇÃO19

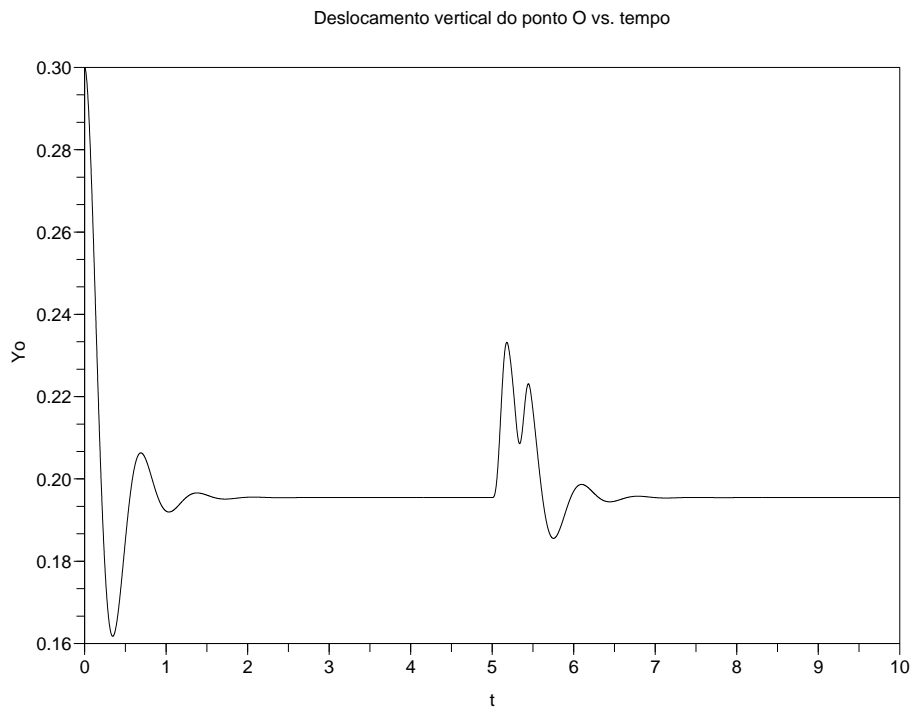


Figura 3.3: Veículo sobre a lombada

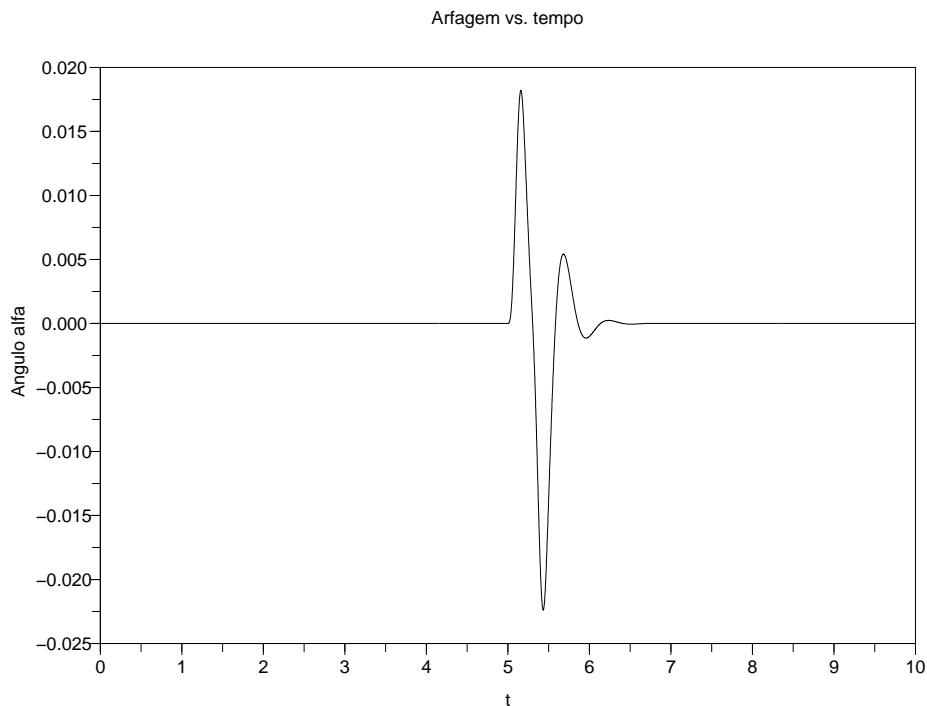


Figura 3.4: Veículo sobre a lombada

Desta forma é possível obter informações sobre conforto ou segurança de um veículo na fase de projeto, lembrando que as hipóteses de modelagem restringem o campo de perguntas que podemos dirigir à simulação numérica, a metodologia restringe o campo das conclusões pertinentes.

3.3 Para executar o *script*

Após invocar o Scilab através de seu ícone, aparece uma tela com um *prompt* que é o símbolo `-->`. A primeira providência a ser tomada é descobrir qual é o diretório de trabalho atual. Para conhecer o diretório de trabalho atual deve-se digitar no *prompt* `pwd` e pressionar a tecla `<enter>`. Se acaso os arquivos de comandos e de definição da função, `emsc.sce` e `emsc.sci`, não se encontram no diretório de trabalho, deve-se alterar o diretório de trabalho através do comando `chdir()`. Utilize o *help* do scilab para entender o comando `chdir()` no particular sistema operacional que você estiver utilizando.

Finalmente, para executar o *script* de comandos que se encontra no diretório de trabalho, `emsc.sce`, basta digitar

```
exec('emsc.sce');
```

ou simplesmente

```
exec emsc.sce;
```

22CAPÍTULO 3. EXEMPLO: VEÍCULO 2D PASSANDO POR LOMBADA

Capítulo 4

Exemplo: partícula que colide contra o solo

Considere uma partícula de massa m que é abandonada de uma altura h_0 do solo com velocidade vertical v_0 , esta velocidade é considerada positiva quando a partícula se afasta do solo. Admite-se ainda que a aceleração da gravidade é $g = 9,81 \text{ m/s}^2$ e que o coeficiente de restituição da colisão partícula/solo é $e = 0,85$. Nestas condições, a partícula irá colidir diversas vezes contra solo conforme a figura 4.1

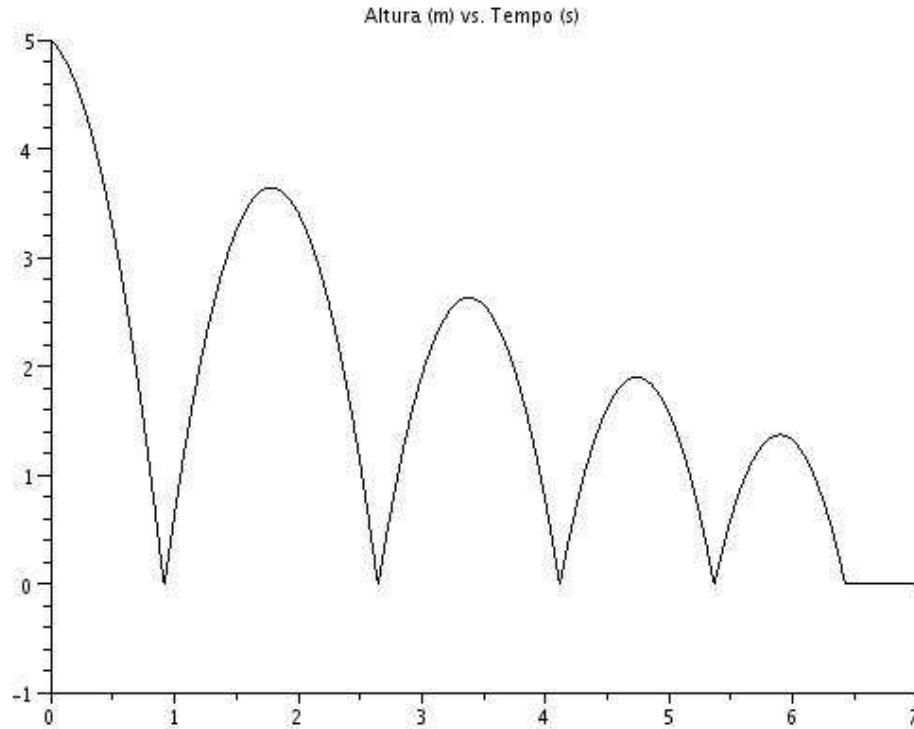


Figura 4.1: Partícula que se choca contra o solo

4.1 As equações do movimento

Do teorema do movimento do baricentro, podemos afirmar sobre a partícula,

$$m\ddot{y} = -mg \quad (4.1)$$

que pode ser reduzida a

$$\ddot{y} = -g. \quad (4.2)$$

Na maioria dos integradores numéricos o sistema de equações de movimento deve ser informado na forma,

$$\dot{x} = f(x, t) \quad (4.3)$$

portanto, como um sistema de equações diferenciais ordinárias de primeira ordem. A eq. 4.2, que é de segunda ordem, pode ser transformada em um

sistema de equações de primeira ordem através de um artifício. Define-se um vetor estado, por exemplo,

$$x = \begin{Bmatrix} y \\ \dot{y} \end{Bmatrix}. \quad (4.4)$$

E a primeira derivada temporal do vetor estado é

$$\dot{x} = \begin{Bmatrix} \dot{y} \\ \ddot{y} \end{Bmatrix}. \quad (4.5)$$

A equação 4.2 pode ser escrita em termos do vetor estado e sua derivada como

$$\begin{Bmatrix} \dot{y} \\ \ddot{y} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} y \\ \dot{y} \end{Bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} g \quad (4.6)$$

ou em forma matricial mais compacta

$$\dot{x} = Ax + Bg \quad (4.7)$$

onde a matriz A é

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (4.8)$$

e a matriz B é

$$B = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad (4.9)$$

A função de Scilab com as equações do movimento é representada no arquivo *choque_teste.sci* reproduzido abaixo.

```
function dy=f(t,y)
  dy = A*y+B;
endfunction
```

4.2 O critério algébrico para encerrar a integração numérica

O integrador de linha de comando que o Scilab oferece chama-se *ode()*. Ele permite interromper a integração numérica caso uma ou mais funções

algébricas se tornem zero. Estas funções algébricas são informadas ao integrador *ode()* através da definição de uma função de Scilab, aqui neste exemplo, no arquivo *choque_condicao.sci*, reproduzido a seguir. Este arquivo informa que no instante em que a altura $z1 = y$ for pequena e a velocidade $z2 = \dot{y}$ for negativa, a condição de colisão foi satisfeita e a integração deve ser interrompida para que novas condições de velocidade sejam impostas como resultado do choque.

```
function z=g(t,y)
  z1=[1 0]*y;
  z2=[0 1]*y;
  z3=z1+sqrt(z2*z2)+z2;
  z=[z3];
endfunction
```

4.3 *Script* com o conjunto de comandos de integração

Sempre que inicializado o integrador *ode()* responde com uma lista de informações, uma matriz com a história de estados Y nos instantes de tempo definidos previamente pelo vetor *temp* e com uma matriz linha *rd*, cujo primeiro elemento contém o instante de tempo em que a condição de colisão foi satisfeita e o segundo elemento contém o número de condições algébricas que resultaram satisfeitas e, portanto, têm valor zero. No nosso exemplo há apenas uma condição algébrica definida, a função g é escalar.

Na primeira parte do *script* os parâmetros do problema são informadas, como variáveis globais. Define-se o tempo inicial, o tempo final de integração, o intervalo de tempo em que gostaríamos de conhecer o estado do sistema Dt , a altura inicial da partícula $h0$, a velocidade inicial da partícula $v0$, o estado inicial $y0$, o coeficiente de restituição e a aceleração da gravidade *grav*.

```
// Exemplo de Choque de uma partícula com o solo

clc
global temp estado

//Instante inicial de integracao [s])
t_inicial = 0.0;

//Incremento de tempo na resposta [s]
```

4.3. SCRIPT COM O CONJUNTO DE COMANDOS DE INTEGRAÇÃO 27

```
Dt = 0.01;

//Instante final de integração [s]
t_final = 7.0;

//Altura inicial [m]
h0 = 5;

//Velocidade inicial [m/s]
v0 = -1;

//Estado Inicial
y0=[h0; v0];

//Coeficiente de restituição
e = 0.85;

//Aceleração da gravidade [m/s^2]
grav = 9.81;
```

A seguir são introduzidas as matrizes A e B que pertencem às equações do movimento, eq. 4.7 e através do comando *getd*, todas as funções do diretório especificado são lidas e incorporadas para serem utilizadas na sessão de Scilab. As funções definidas no diretório são f , g e *integra_ate_choque* (esta última função será abordada em breve).

```
//Carrega a definição da função dy = f(t,y)
A=[0 1; 0 0];
B=[ 0; -grav];

getd("/home/raul/disciplinas/pme2200/choque");
```

Como o integrador deve ser interrompido na condição de colisão, $g(x, t) = 0$, é necessário ir armazenando os estados já calculados em uma matriz *perene*, que conterà a história dos estados de diversos períodos sem colisão. Definem-se então dois vetores, *temp* e a matriz *estado*, onde cada coluna está associada a um instante múltiplo do intervalo de resposta Dt . A matriz é inicializada com zeros, tanto na primeira linha que representa a história do deslocamento y , quanto na segunda linha que representa a história da velocidade \dot{y} .

28CAPÍTULO 4. EXEMPLO: PARTÍCULA QUE COLIDE CONTRA O SOLO

```
//inicia o tempo
t0=t_inicial;

// aloca memoria para vetor tempo e matriz historia do estado
n_tempo=floor(t_final/Dt);
for i=1:n_tempo
    temp(i)=(i-1)*Dt;
    vetor_zero(i)=0.0;
end
estado=[vetor_zero'; vetor_zero'];
```

Procede-se, então, uma sequência de integrações numéricas que são interrompidas pelo critério de colisão. Após cada colisão, as condições de velocidade *pós-choque* são impostas pela função *integra_ate_choque*. A sequência de integrações numéricas são gerenciadas pelo laço (*loop*) de programação *while* que segue. Se a integração for interrompida por outro motivo, diferente da condição de colisão, então a altura da partícula é diferente de zero e interrompe-se o laço de programação *while*. Neste laço, os estados integrados numericamente são armazenados na matriz *estado*.

```
t_choque=0.0;
```

```
while t_choque<t_final do
    [t_choque,y_choque,estado]=integra_ate_choque(t_choque,t_final,Dt,y0,estado)
    y0=y_choque
    if abs(y_choque(1,1))>1.0e-8 then
        t_choque=t_final
    end
end
```

Quando a sequência de integrações numéricas estiver terminada, elaborase um gráfico da história da altura e arquiva-se este gráfico no formato jpg, neste exemplo.

```
scf(0);
xtitle('Altura (m) vs. Tempo (s)');
plot2d(temp',estado(1,:));
//Exportando para jpg
filename='choque_resultado.jpg';
xs2jpg(0,filename);
```

Resta detalhar como opera a função *integra_ate_choque*, que é uma função definida pelo usuário.

4.4 Detalhamento da função *integra_ate_choque*

Esta função invoca o integrador *ode* do Scilab, de um certo instante t_{inic} , até que ocorra uma colisão ou até que seja atingido o tempo final t_{final} . Deseja-se que os estados resultantes da integração numérica sejam reportados em instantes múltiplos de Dt . A função *ode* devolve uma matriz Y e uma matriz linha rd que contém no primeiro elemento o instante de colisão t_{choque} . O último estado (última coluna) da matriz Y contém o estado imediatamente antes do choque. O estado imediatamente antes do choque, $Y(:,coluna)$, tem sua velocidade modificada através de $[10; 0 - e] * Y(:,coluna)$. O vetor $dscrttime$ é um sub-vetor do vetor $temp$ e contém instantes de tempo múltiplos de Dt .

```
function [t_choque,y_choque,estado]=integra_ate_choque(t_inic,t_final,Dt,y0,estado)
    n=2; //numero de componentes do vetor estado
    t_inic_dt=t_inic-modulo(t_inic,Dt)+Dt;
    dscrttime = t_inic_dt:Dt:t_final;
    [Y,rd]=ode("roots",y0,t_inic_dt,dscrttime,f,1,g);
    t_choque=rd(1,1);
    [linha,coluna]=size(Y);
    y_choque=[1 0; 0 -e]*Y(:,coluna);
```

Finalmente, os estados obtidos nesta particular integração numérica são armazenados no vetor global e perene *estado* para que, posteriormente, seja possível elaborar um gráfico contendo múltiplas colisões.

```
    n_t_discreto=floor(t_choque/Dt);
    n_t_discreto_old=floor(t_inic/Dt);
for i=n_t_discreto_old+1:n_t_discreto+1
    for j=1:n
        estado(j,i)=Y(j,i-n_t_discreto_old);
    end
end
endfunction
```

4.5 Para executar o *script*

Após invocar o Scilab através de seu ícone, aparece uma tela com um *prompt* que é o símbolo `-- >`. A primeira providência a ser tomada é descobrir qual é o diretório de trabalho atual. Para conhecer o diretório de trabalho atual deve-se digitar no prompt `pwd` e pressionar a tecla `<enter>`. Se acaso os

30CAPÍTULO 4. EXEMPLO: PARTÍCULA QUE COLIDE CONTRA O SOLO

arquivos de comandos e de definição da função, *emsc.sce* e *emsc.sci*, não se encontram no diretório de trabalho, deve-se alterar o diretório de trabalho através do comando **chdir()**. Utilize o *help* do Scilab para entender o comando **chdir()** no particular sistema operacional que você estiver utilizando. Finalmente, para executar o script de comandos que se encontra no diretório de trabalho, *emsc.sce*, basta digitar

```
exec('teste_looping.sce');
```

ou simplesmente

```
exec teste_looping.sce;
```