Universidade Estadual de Campinas
Instituto de Física Gleb Wataghin

Projeto Referente à disciplina F-809 - Instrumentação para Ensino

Sistema Caótico: Pêndulo Duplo

Aluno: Maurício Richartz

Orientador: Prof. Dr. Alberto Saa

Data de finalização do Relatório Final: 13/06/2005

28.1

1 Resumo

Este projeto de Instrumentação para Ensino consiste em analisar o movimento de um pêndulo duplo. Ele é dividido em duas partes: a primeira consiste em desenvolver uma applet (programa de computador na linguagem java) para simular o movimento de um pêndulo duplo; a segunda etapa, por sua vez, é comparar essa simulação com o movimento de um pêndulo real, discutindo a caoticidade do sistema.

2 Introdução

Caos é o estado de desordem e irregularidade de um sistema determinístico cuja evolução no tempo, apesar de ser governada por leis exatas e simples, é altamente sensível às condições iniciais. Pequenas variações nessas condições produzem resultados completamente diferentes, de tal forma que, a longo prazo, o comportamento de sistemas caóticos se torna tão imprevisível que pode parecer aleatório (no entanto, lembre-se que o sistema é determinístico, logo isso não pode ocorrer). Um pêndulo duplo é um dos exemplos mais simples de sistemas caóticos. Outros exemplos de sistemas caóticos incluem a atmosfera, o sistema solar, placas tectônicas, fluidos turbulentos, a econômia e o crescimento populacional.

3 Teoria

3.1 Runge-Kutta

Os métodos de Runge-Kutta permitem resolver equações diferenciais numericamente, sem a necessidade do cálculo de derivadas. Essa é a vantagem em relação aos métodos de série de Taylor. Nesse projeto, utiliza-se um método de Runge-Kutta de quarta ordem (ver apêndice **A**).

3.2 Equações de Movimento

3.2.1 Pêndulo Duplo

Considere um pêndulo duplo de massas m_1 e m_2 e comprimentos l_1 e l_2.

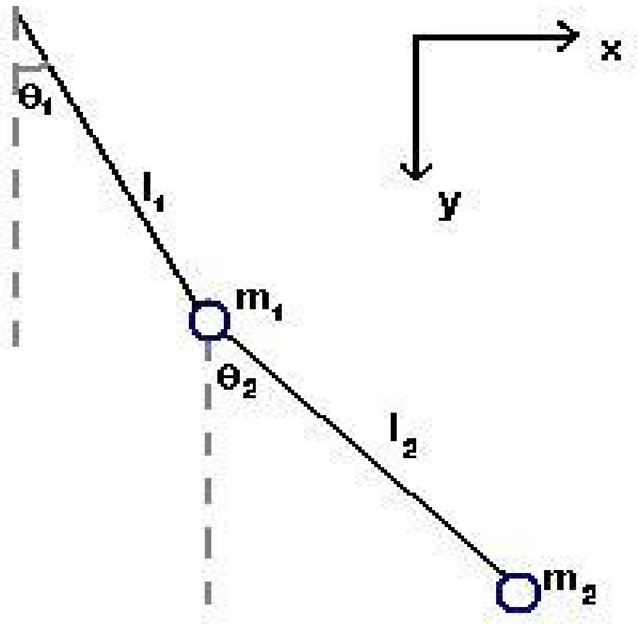

Figura 1: Pêndulo Duplo

Em termos das coordenada generalizadas θ_1 e θ_2, as coordenadas cartesianas são:

$$x_1 = l_1 \sin\theta_1 \tag{1}$$
$$y_1 = l_1 \cos\theta_1 \tag{2}$$
$$x_2 = l_1 \sin\theta_1 + l_2 \sin\theta_2 \tag{3}$$
$$y_2 = l_1 \cos\theta_1 + l_2 \cos\theta_2 \tag{4}$$

Dessa forma, a energia cinética é

$$T = \frac{1}{2}m_1(\dot{x_1}^2 + \dot{y_1}^2) + \frac{1}{2}m_2(\dot{x_2}^2 + \dot{y_2}^2) = \frac{1}{2}m_1[(l_1\cos\theta_1\dot{\theta}_1)^2 + (-l_1\sin\theta_1\dot{\theta}_1)^2]$$

$$+\frac{1}{2}m_2[(l_1\cos\theta_1\dot{\theta}_1 + l_2\cos\theta_2\dot{\theta}_2)^2 + (-l_1\sin\theta_1\dot{\theta}_1 - l_2\sin\theta_2\dot{\theta}_2)^2]$$

$$= \frac{1}{2}(m_1+m_2)l_1^2\dot{\theta_1}^2 + \frac{1}{2}m_2 l_2^2\dot{\theta_2}^2 + m_2 l_1 l_2 \dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2)$$

e a energia potencial

$$V = m_1 g y_1 + m_2 g y_2 = (m_1+m_2)g l_1\cos\theta_1 + m_2 g l_2 \cos\theta_2$$

Logo, o Lagrangiano é:

$$L = \frac{1}{2}(m_1+m_2)(l_1^2\dot{\theta_1}^2 - gl_1\cos\theta_1) + \frac{1}{2}m_2(l_2^2\dot{\theta_2}^2 - gl_2\cos\theta_2) + m_2 l_1 l_2 \dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2)$$

Usando as Equações de Euler-Lagrange, determina-se um sistema de equações diferenciais de primeira ordem que governam o movimento:

$$\frac{d}{dt}\begin{pmatrix}\theta_1\\ \theta_2\\ w_1\\ w_2\end{pmatrix} = \begin{pmatrix} w_1\\ w_2\\ \frac{(2m_1+m_2)g\sin\theta_1+m_2 g\sin(\theta_1-2\theta_2)-4m_2(l_2w_2^2+l_1w_1^2\cos(\theta_1-\theta_2))\sin(\theta_1-\theta_2)}{4l_1(m_1+m_2\sin^2(\theta_1-\theta_2))}\\ \frac{((m_1+m_2)(2l_1w_1^2-g\cos\theta_1)+2m_2l_2w_2^2\cos(\theta_1-\theta_2))\sin(\theta_1-\theta_2)}{2l_2(m_1+m_2\sin^2(\theta_1-\theta_2))}\end{pmatrix},$$

onde $w_1 = \dot{\theta_1}$ e $w_2 = \dot{\theta_2}$. Para maiores detalhes consulte o apêndice **B**.

4 Simulação

Uma applet é uma aplicação especial em Java que pode ser executada por um browser (um navegador WWW) quando é incorporada a uma página HTML. Foi construída uma applet para simular o movimento de um pêndulo duplo usando o editor JCreator 3.50 PRO. A applet permite definir, através de botões, as condições iniciais do pêndulo (posição e velocidade inicial de cada uma das massas) e alguns parâmetros (aceleração da gravidade, comprimento de ambas as hastes e o valor de ambas as massas). Os ângulos iniciais e os comprimentos das hastes podem ser alterados diretamente com o mouse. Conforme visto anteriormente, as equações que regem o movimento do pêndulo são não-lineares e acopladas e não podem ser resolvidas analiticamente. Para isso, implementou-se o algoritmo de Runge-Kutta em Java.

Foram desenvolvidas duas classes: Penduloduplo.class ($9,66$KB), que consiste no pêndulo propriamente dito, e runge.class ($2,22$KB), responsável por resolver o sistema de equações diferenciais usando o método de Runge-Kutta. Além disso, foi necessário desenvolver alguns métodos para executar ações específicas, como, por exemplo, passar parâmetros de uma classe para outra. O desenho na tela é feito utilizando funções simples como **fillOval** e **drawLine**. A interatividade com o mouse é possível com o uso dos métodos **mouseDown** e **mouseDrag**. Para maiores detalhes a respeito do código, incluindo comentários sobre os métodos desenvolvidos, consulte o apêndice **C**.

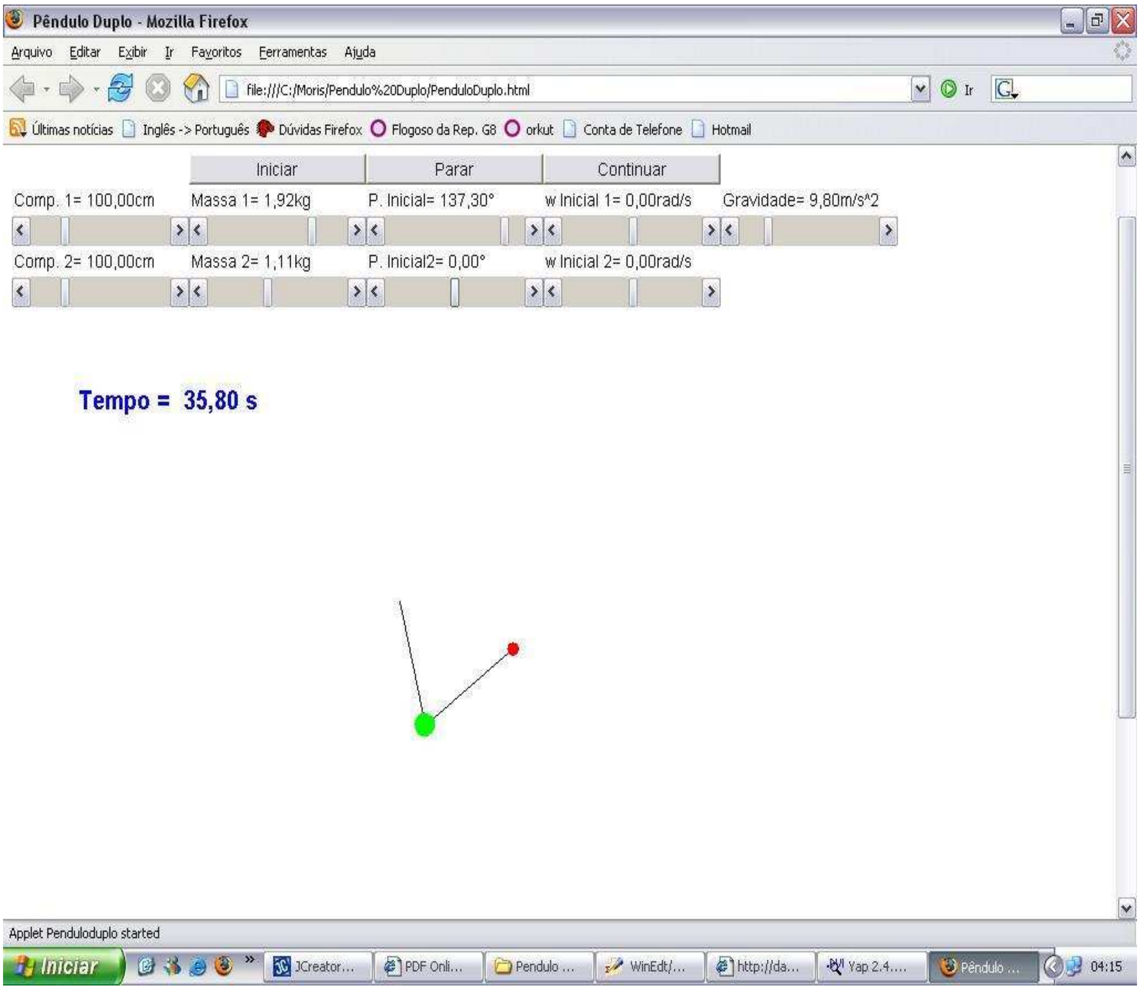

Figura 2: Applet desenvolvida que simula o movimento do pêndulo

5 Conclusão

Esse experimento é útil tanto a alunos de ensino médio como superior. Para os alunos de ensino médio interessa a demonstração experimental da caoticidade: eles podem ver na prática (simulação e experimento real) o conceito de caos em um sistema simples. Para alguns alunos, isso estimula e aumenta o gosto pela física, podendo inclusive motivar a seguir carreira nessa área. Já os alunos universitários, tem um interesse adicional: entender profundamente a teoria que explica o que ocorre na prática. Se esses alunos universitários já tiverem feito algum curso de sistemas dinâmicos, podem ver na prática alguns dos conceitos estudados em sala de aula, como pontos de equilíbrio, seções de Poincaré e órbitas densas.

6 Apêndices

6.1 Apêndice A - Runge-Kutta

Considere um sistema de n equações diferenciais de primeira ordem:

$$\dot{\vec{x}} = \vec{f}(t, \vec{x}), \quad \vec{x}(0) = \vec{x_0} \tag{5}$$

onde $\vec{x} = (x_1, ..., x_n)$ e $\vec{f}(t, \vec{x}) = (f_1(t, \vec{x}), ..., f_n(t, \vec{x}))$. Seja $\vec{x_n} = \vec{x}(t_n)$. O método utilizado permite aproximar $\vec{x}_{n+1}$ em $t_{n+1} = t_n + h$ a partir de $\vec{x_n}$ e t_n através da seguinte fórmula:

$$\vec{x}_{n+1} = \vec{x_n} + \frac{h}{6}(\vec{k_a} + 2\vec{k_b} + 2\vec{k_c} + \vec{k_d}), \tag{6}$$

onde

$$\vec{k_a} = \vec{f}(t_n, \vec{x_n}) \tag{7}$$

$$\vec{k_b} = \vec{f}(t_n + \frac{h}{2}, \vec{x_n} + \frac{h}{2}\vec{k_a}) \tag{8}$$

$$\vec{k_c} = \vec{f}(t_n + \frac{h}{2}, \vec{x_n} + \frac{h}{2}\vec{k_b}) \tag{9}$$

$$\vec{k_d} = \vec{f}(t_n + h, \vec{x_n} + h\vec{k_c}) \tag{10}$$

Para maiores informações consulte a referência [?].

6.2 Apêndice B - Pêndulo Duplo

A partir do lagrangiano determinado na seção (3.2.1), tem-se:

$$\frac{\partial L}{\partial \theta_1} = \frac{1}{2}(m_1 + m_2) g l_1 \sin\theta_1 - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2)$$

$$\frac{\partial L}{\partial \theta_2} = \frac{1}{2} m_2 g l_2 \sin\theta_2 + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2)$$

Além disso,

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2) l_1^2 \dot{\theta}_1 + m_2 l_1 l_2 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \dot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

Portanto,

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_1}) = (m_1+m_2) l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1-\theta_2) - m_2 l_1 l_2 \dot{\theta}_2 (\dot{\theta}_1 - \dot{\theta}_2) \sin(\theta_1-\theta_2)$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_2}) = m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 - \dot{\theta}_2) \sin(\theta_1 - \theta_2)$$

Assim, dividindo-se as equações de Euler-Lagrange $\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}_i}) - \frac{\partial L}{\partial \theta_i}$ por l_i, encontram-se as equações de movimento para o pêndulo duplo.A equação para θ_1 é:

$$(m_1+m_2) l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1-\theta_2) + m_2 l_2 \dot{\theta}_2^{\;2} \sin(\theta_1-\theta_2) - \frac{1}{2}(m_1+m_2) g \sin\theta_1 = 0 \tag{11}$$

Para θ_2, a equação é:

$$m_2 l_2 \ddot{\theta}_2 + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^{\;2} \sin(\theta_1 - \theta_2) - \frac{1}{2} m_2 g \sin\theta_2 = 0 \tag{12}$$

Para que seja possível utilizar o método de Runge-Kutta, é necessário transformar as equações diferenciais aclopadas de segunda ordem em um sistema de equações de primeira ordem. Introduzindo as variáveis $w_1 = \dot{\theta}_1$ e $w_2 = \dot{\theta}_2$ as equações de Euler-Lagrange se tornam:

$$\begin{pmatrix} (m_1 + m_2) l_1 & m_2 l_2 \cos(\theta_1 - \theta_2) \\ m_2 l_1 \cos(\theta_1 - \theta_2) & m_2 l_2 \end{pmatrix} \begin{pmatrix} \dot{w}_1 \\ \dot{w}_2 \end{pmatrix} = \begin{pmatrix} g(\theta_1, \theta_2, w_1, w_2) \\ h(\theta_1, \theta_2, w_1, w_2) \end{pmatrix}$$

Resolvendo-se o sistema para $\dot{w}_1$ e $\dot{w}_2$ em função de $\theta_1, \theta_2, w_1, w_2$ usando o software Mathematica 5.0, tem-se

$$\dot{w}_1 = \frac{(2m_1 + m_2) g \sin\theta_1 + m_2 g \sin(\theta_1 - 2\theta_2) - 4 m_2 (l_2 w_2^2 + l_1 w_1^2 \cos(\theta_1 - \theta_2)) \sin(\theta_1 - \theta_2)}{4 l_1 (m_1 + m_2 \sin^2(\theta_1 - \theta_2))}$$

$$\dot{w}_2 = \frac{((m_1 + m_2)(2 l_1 w_1^2 - g \cos\theta_1) + 2 m_2 l_2 w_2^2 \cos(\theta_1 - \theta_2)) \sin(\theta_1 - \theta_2)}{2 l_2 (m_1 + m_2 \sin^2(\theta_1 - \theta_2))}$$

6.3 Apêndice C – Códigos em Java

Penduloduplo.java:

```
import java.applet.*;
import java.awt.*;
import java.util.*;
import java.text.DecimalFormat;

public class Penduloduplo extends Applet implements Runnable
{
   Image m_Tela;
   Graphics m_Grafico;
   Graphics g_this;
   Dimension dim;
   Scrollbar B1,B2, B3, B4, B5, B6, B7, B8, B9;
   Thread th;

   double l=100, l2=100;
   double d=0.1, d2=0.1, tt=0;
   double gravidade=980;
   double thetazero=0, thetazero2=0;
   double wini1=0, wini2=0;
   double [] thetaaux;
   public Label Comprimento;
   public Label Gravidade;
   public Label Massa;
   public Label Pinicial;
   public Label Comprimento2;
   public Label Massa2;
   public Label Pinicial2;
   public Label winicial1;
   public Label winicial2;
   public DecimalFormat fd;
   public runge pduplo;
   public boolean m, m2;
   long tempo0, tempo1, tempo2;
   long tempotemp;
   double x=0,y=0,x2=0,y2=0;
   Point ponto0=new Point(300,300);
   int desenho=1;

   public void init(){
         pduplo = new runge(thetazero,thetazero2, gravidade,l,l2,d,
d2,wini1,wini2);
         Font f=new Font("Arial",Font.PLAIN,14);
      setFont(f);
      fd = new DecimalFormat(" 0.00;-0.00");  //serve para que os números
tenham apenas duas casa decimais
```

```
        setLayout(new BorderLayout());
          // Adiciona o painel com os botões e as barras de rolagem
            Panel p = new Panel();
            p.setLayout(new GridLayout(5,11));
            p.add(new Label("",Label.LEFT));
            p.add(new Button("Iniciar"));
            p.add(new Button("Parar"));
            p.add(new Button("Continuar"));
            p.add(new Label("",Label.LEFT));

            Comprimento = new Label("Comp. 1=" + fd.format(l) +
"cm",Label.LEFT);
            p.add(Comprimento);
            Massa= new Label("Massa 1=" +fd.format(d) + "kg" ,Label.LEFT);
            p.add(Massa);
            Pinicial = new Label("P. Inicial 1=" + fd.format(180*thetazero /
(Math.PI)) + "°" ,Label.LEFT);
            p.add(Pinicial);
            winicial1 = new Label("w Inicial 1=" + fd.format(wini1) + "rad/s"
,Label.LEFT);
            p.add(winicial1);
            Gravidade = new Label("Gravidade=" +fd.format(gravidade*0.01) +
"m/s^2",Label.LEFT);
            p.add(Gravidade);
            B1= new Scrollbar(Scrollbar.HORIZONTAL,1000,0,400,3000);
            B2= new Scrollbar(Scrollbar.HORIZONTAL,1,1,10,250);
            B3= new Scrollbar(Scrollbar.HORIZONTAL,1,0,-1800,1801);
            B4= new Scrollbar(Scrollbar.HORIZONTAL,1,0,-80,80);
            B5= new Scrollbar(Scrollbar.HORIZONTAL,1000,0,100,5000);

            p.add(B1);
            p.add(B2);
            p.add(B3);
            p.add(B4);
            p.add(B5);
            Comprimento2 = new Label("Comp. 2=" + fd.format(l2) +
"cm",Label.LEFT);
            p.add(Comprimento2);
            Massa2= new Label("Massa 2=" +fd.format(d2) + "kg" ,Label.LEFT);
            p.add(Massa2);
            Pinicial2 = new Label("P. Inicial 2=" + fd.format(180*thetazero2 /
(Math.PI)) + "°" ,Label.LEFT);
            p.add(Pinicial2);
            winicial2 = new Label("w Inicial 2=" + fd.format(wini2) + "rad/s"
,Label.LEFT);
            p.add(winicial2);
            p.add(new Label("",Label.LEFT));
            B6= new Scrollbar(Scrollbar.HORIZONTAL,1000,0,400,3000);
            B7= new Scrollbar(Scrollbar.HORIZONTAL,1,1,10,250);
```

```
        B8= new Scrollbar(Scrollbar.HORIZONTAL,1,0,-1800,1801);
        B9= new Scrollbar(Scrollbar.HORIZONTAL,1,0,-80,80);
        p.add(B6);
        p.add(B7);
        p.add(B8);
        p.add(B9);

        add("North", p);

        tempo0=new Date().getTime();
        calcpendulum();
        resize(800,800);
        setBackground(Color.white);

        dim=this.size();
        m_Tela=createImage(dim.width,dim.height);
        m_Grafico=m_Tela.getGraphics();
        Font f2=new Font("Arial",Font.BOLD,20);
        m_Grafico.setFont(f2);
    }

  // Cálculo das coordenas x1, x2, y1 e y2 do pêndulo duplo
    void calcpendulum(){
      tempotemp=new Date().getTime();
      thetaaux = pduplo.getangle();   //obtem o vetor de variáveis
      x=l*Math.sin(thetaaux[0]);
      y=l*Math.cos(thetaaux[0]);
              x2=l2*Math.sin(thetaaux[1]);
      y2=l2*Math.cos(thetaaux[1]);
      tt=pduplo.gettempo();  //obtem o tempo de simulação
    }

   // Configuração das barras de rolagem e dos botões:

    public boolean handleEvent(Event e) {
    if (e.target == B1) {
        l = (double)((Integer)e.arg).intValue() * 0.1;
        tempo0=new Date().getTime();
        pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
        calcpendulum();
        repaint();
        Comprimento.setText("Comp. 1=" +fd.format(l) + "cm");
        return true;
    }
    if (e.target == B2) {
        d = (double)((Integer)e.arg).intValue() * 0.1;
        tempo0=new Date().getTime();
        pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
```

```
            calcpendulum();
            repaint();
            Massa.setText("Massa 1=" +fd.format(d*0.1) + "kg");
            return true;
      }

      if (e.target == B3) {
            thetazero = (double)((Integer)e.arg).intValue() * 0.00174533 ;
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            Pinicial.setText("P. Inicial=" + fd.format(180*thetazero  / (Math.PI)) +
"°");
            return true;
      }
      if (e.target == B4) {
            wini1 = (double)((Integer)e.arg).intValue() * 0.1;
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            winicial1.setText("w Inicial 1=" + fd.format(wini1) + "rad/s");
            return true;
      }
      if (e.target == B5) {
            gravidade = (double)((Integer)e.arg).intValue() ;
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            Gravidade.setText("Gravidade=" + fd.format(gravidade*0.01) +
"m/s^2");
            return true;
      }
      if (e.target == B6) {
            l2 = (double)((Integer)e.arg).intValue() * 0.1;
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            Comprimento2.setText("Comp. 2=" +fd.format(l2) + "cm");
            return true;
      }
      if (e.target == B7) {
            d2 = (double)((Integer)e.arg).intValue() * 0.1;
```

```
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            Massa2.setText("Massa 2=" +fd.format(d2*0.1) + "kg");
            return true;
      }

      if (e.target == B8) {
            thetazero2 = (double)((Integer)e.arg).intValue() * 0.00174533 ;
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            Pinicial2.setText("P. Inicial2=" + fd.format(180*thetazero2  /
(Math.PI)) + "°");
            return true;
      }
       if (e.target == B9) {
            wini2 = (double)((Integer)e.arg).intValue() * 0.1 ;
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            winicial2.setText("w Inicial 2=" + fd.format(wini2) + "rad/s");
            return true;
      }
      if ("Iniciar".equals(e.arg)) {
            desenho = 2;
            tempo0=new Date().getTime();
            pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
            calcpendulum();
            repaint();
            desenho = 1;
            return true;
      }
      if ("Parar".equals(e.arg)) {
            desenho = 2;
            tempo1=new Date().getTime();
            return true;
      }
      if ("Continuar".equals(e.arg)) {
            desenho = 1;
            tempo2=new Date().getTime();
            tempo0=tempo0+(tempo2-tempo1);
            calcpendulum();
```

```
            repaint();
            return true;
        }
        return super.handleEvent(e);
}

       // Executa o pendulo:

        public void start(){
            if(th==null){
                th=new Thread(this);
               th.start();
            }
        }
        public void run(){
            while(th!=null){
              if(desenho==1){
                    try{
                        calcpendulum();
                       repaint();
                        th.sleep(100);
                    }catch(InterruptedException e){
                        ;
                    }
              }
            }
        }
        public void stop(){
            if(th !=null){
                th.stop();
                th=null;
            }
        }
        public void update(Graphics g){
            paint(g);
        }

        //registra o clique do mouse e calcula se ele é próximo de alguma das
massas
        public boolean mouseDown(Event evt,int c, int d){
          if((c-ponto0.x - x -50)*(c-ponto0.x-x-50)+(d-ponto0.y- y-50)*(d-ponto0.y-
y-50) < 1000){m=true;}
          else {m=false;}
          if((c-ponto0.x - x -x2 - 50)*(c-ponto0.x-x- x2 -50)+(d-ponto0.y - y - y2 -
50)*(d-ponto0.y - y - y2 -50) < 1000){m2=true;}
          else {m2=false;}
        return true;
        }
        //altera os comprimentos e os angulos iniciais de acordo com o mouse
        public boolean mouseDrag(Event evt,int a,int b){
```

```
            if(m){
             l=Math.sqrt((a-ponto0.x)*(a-ponto0.x)+(b-ponto0.y)*(b-ponto0.y));
             if (b < ponto0.y) {thetazero=Math.PI+Math.atan((double)(a-
ponto0.x)/(double)(b-ponto0.y));}
             else{thetazero=Math.atan((double)(a-ponto0.x)/(double)(b-ponto0.y));}
              if (Math.PI <thetazero && thetazero <= 1.5*Math.PI) {thetazero = -
2*Math.PI + thetazero;}
                    Comprimento.setText("Comp. 1=" +fd.format(l) + "cm");
                    Pinicial.setText("P. Inicial=" + fd.format(180*thetazero  / (Math.PI))
+ "°");
                    tempo0=new Date().getTime();
               pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
               calcpendulum();
               repaint();
            }
             if(m2){
             l2=Math.sqrt((a-ponto0.x - x)*(a-ponto0.x - x)+(b-ponto0.y - y)*(b-
ponto0.y-y));
             if (b < ponto0.y + y) {thetazero2=Math.PI+Math.atan((double)(a-
ponto0.x - x)/(double)(b-ponto0.y - y));}
             else{thetazero2=Math.atan((double)(a-ponto0.x - x)/(double)(b-
ponto0.y - y));}
              if (Math.PI <thetazero2 && thetazero2 <= 1.5*Math.PI) {thetazero2 = -
2*Math.PI + thetazero2;}
                    Comprimento2.setText("Comp. 2=" +fd.format(l2) + "cm");
                     Pinicial2.setText("P. Inicial 2=" + fd.format(180*thetazero2  /
(Math.PI)) + "°");
                    tempo0=new Date().getTime();
               pduplo = new runge(thetazero, thetazero2,
gravidade,l,l2,d,d2,wini1,wini2);
               calcpendulum();
               repaint();
            }

              return(true);
      }
        // Desenha o pendulo na tela:

    public void paint(Graphics g){
               m_Grafico.setColor(getBackground());
               m_Grafico.fillRect(0,0,dim.width,dim.height);
               m_Grafico.setColor(getForeground());
               m_Grafico.setColor(Color.black);

m_Grafico.drawLine(ponto0.x,ponto0.y,ponto0.x+(int)x,ponto0.y+(int)y);
               m_Grafico.setColor(Color.black);
               m_Grafico.drawLine(ponto0.x+(int)x,ponto0.y+(int)y,ponto0.x+(int)x +
(int)x2,ponto0.y+(int)y + (int)y2);
               m_Grafico.setColor(Color.green);
```

```
		m_Grafico.fillOval(ponto0.x+(int)(x-d/2.0),ponto0.y+(int)(y-
d/2.0),(int)d,(int)d);
		m_Grafico.setColor(Color.red);
		m_Grafico.fillOval(ponto0.x+(int)x + (int)(x2-d2/2.0),ponto0.y+(int)y +
(int)(y2-d2/2.0),(int)d2,(int)d2);
		m_Grafico.setColor(Color.blue);
		m_Grafico.drawString("Tempo = "+fd.format(tt) + " s",10,150);
		g.drawImage(m_Tela,50,50,this);
		}
}
```

runge.java:

```
public class runge

{
	public int  numVars = 4;
	public double[]    vars;
	double sim_time = 0;
   double g;
   double l1;
   double l2;
   double d;
   double d2;
   double m_time = -9999;


	public double diffeq0(double t, double[] x)   // t = tempo, x = vetor de
variáveis
  {
    return x[2];  // constitui a primeira equação diferencial do sistema: dtheta1 =
w1
  }

	public double diffeq1(double t, double[] x)   // t = tempo, x = vetor de
variáveis
  {
    return x[3];  // constitui a segunda equação diferencial do sistema:
dtheta2=w2
  }

  public double diffeq2(double t, double[] x)   // t = tempo, x = vetor de variáveis
  {
    // constitui a terceira equação diferencial do sistema: dw1 = ...
   double r = (d2*l1*x[2]*x[2]*Math.sin(x[1]-x[0])*Math.cos(x[1]-x[0]) +
d2*g*Math.sin(x[1])*Math.cos(x[1]-x[0]) + d2*l2*x[3]*x[3]*Math.sin(x[1]-x[0])-
(d+d2)*g*Math.sin(x[0]))/((d+d2)*l1 - d2*l1*Math.cos(x[1]-x[0])*Math.cos(x[1]-
x[0]) );
```

```
 return r;
}

public double diffeq3(double t, double[] x)   // t = tempo, x = vetor de variáveis
{
  // constitui a quarta equação diferencial do sistema:dw2 = ...
  double s = (-d2*l2*x[3]*x[3]*Math.sin(x[1]-x[0])*Math.cos(x[1]-x[0]) +
(d+d2)*(g*Math.sin(x[0])*Math.cos(x[1]-x[0]) - l1*x[2]*x[2]*Math.sin(x[1]-x[0]) -
g*Math.sin(x[1]) ))/((d+d2)*l2 - d2*l2*Math.cos(x[1]-x[0])*Math.cos(x[1]-x[0]) );
  return s;
}

 public double calculo(int i, double t, double[] x)
{
  switch (i)
  { case 0:  return diffeq0(t, x);
    case 1:  return diffeq1(t, x);
    case 2:  return diffeq2(t, x);
    case 3:  return diffeq3(t, x);
   default:
      System.out.println("Problema no cálculo");
      return 0;
  }
}

public void  solve(double t, double h)  //algoritmo de rungekutta
{
  int N = numVars;
  int i;
  double[] inp = new double[N];
  double[] k1 = new double[N];
  double[] k2 = new double[N];
  double[] k3 = new double[N];
  double[] k4 = new double[N];
  for (i=0; i<N; i++)
    k1[i] = calculo(i,t,vars);     // calcula no tempo t
  for (i=0; i<N; i++)
    inp[i] = vars[i]+k1[i]*h/2;
  for (i=0; i<N; i++)
    k2[i] = calculo(i,t+h/2,inp);  // calcula no tempo t+h/2
  for (i=0; i<N; i++)
    inp[i] = vars[i]+k2[i]*h/2;
  for (i=0; i<N; i++)
    k3[i] = calculo(i,t+h/2,inp);  // calcula no tempo t+h/2
  for (i=0; i<N; i++)
    inp[i] = vars[i]+k3[i]*h;
  for (i=0; i<N; i++)
    k4[i] = calculo(i,t+h,inp);    // calcula no tempo t+h
  for (i=0; i<N; i++)
    vars[i] = vars[i]+(k1[i]+2*k2[i]+2*k3[i]+k4[i])*h/6;
```

```
   }

    public runge(double theta, double theta2, double grav, double comp1,
double comp2, double de, double de2, double alfa1, double alfa2)
    // esse é o método responsável por passar as variáveis de uma classe para
outra
  {
        vars = new double[numVars];
        g= grav;
        l1 = comp1;
        l2=comp2;
   d=de;
   d2=de2;
   vars[0] = theta;
   vars[1] = theta2;
   vars[2] = alfa1;
   vars[3] = alfa2;

  }

   public double []  getangle()
  {
  // esse método efetua um passo do método de Runge-Kutta e retorna o vetor
de variáveis

      double h=0.11153;
      solve(sim_time, h);  // efetua um passo do método de runge-kutta
      sim_time += h;       // incrementa o tempo
      return vars;   // retorna o vetor de variáveis
   }
   public double gettempo() // esse método retorna o tempo de simulação
   {
        return sim_time;
    }

}
```

6.4 Apêndice D - Comentários do Coordenador do Curso

- (24/04/2005) Projeto aprovado. Aguardo para saber da construção pela oficina mecânica.
- (03/06/2005) O trabalho está sendo realizado. Será colocado na página da disciplina, quantos Mb ocupa?

Referências

[1] P.J. Junior, *Introdução ao Java*, Editora Berkeley, São Paulo, 2002.

[2] Arnold, V. I., Mathematical Methods of Classical Mechanics,Nova York: Springer-Verlag, 1989.

[3] M. A. G. Ruggiero e V.L.R. Lopes, *Cálculo Númerico - Aspectos Teóricos e Computacionais*, Makron Books, Segunda edição, 1997.

[4] Weisstein, Eric W. *Double Pendulum*, Eric Weisstein's World of Physics. http://scienceworld.wolfram.com/physics/DoublePendulum.html

[5] Goldstein, Herbert, Classical Mechanics, Addison-Wesley, EUA, 1980.

[6] http://www.myphysicslab.com/dbl__pendulum.html

[7] http://www.physics.usyd.edu.au/∼ wheat/dpend__html/

[8] http://www.sekine-lab.ei.tuat.ac.jp/ kanamaru/Chaos/e/DP