

CZU: 510.5:517:004.4/.6

DOI: 10.36120/2587-3636.v37i3.17-28

MODELAREA MATEMATICĂ ȘI SIMULAREA ASISTATĂ DE CALCULATOR A SISTEMELOR MECANICE CU MULTE GRADE DE LIBERTATE

Nicolae BALMUȘ, dr., conf. univ.

<https://orcid.org/0000-0002-0491-2918>

Catedra Informatică și Tehnologii Informaționale
Universitatea Pedagogică de Stat „Ion Creangă” din Chișinău

Rezumat. Marea majoritate a fenomenelor și proceselor din natură pentru care simulare asistată de calculator prezintă interes științific și metodic-didactic, din punct de vedere matematic se descriu prin sisteme de ecuații diferențiale de ordinul doi. În acest context, în lucrare este descris și implementat în limbajul de programare Object Pascal un algoritm original de rezolvare numerică a sistemelor de ecuațiilor diferențiale de ordinul doi. Aplicațiile interactive de simulare sunt realizate în mediul de programare Delphi (variante FMX). În baza exemplelor, descrise detaliat în lucrare, utilizatorii pot realiza activități interactive de simulare pentru orice procese, care din punct de vedere matematic se descriu prin sisteme de ecuații diferențiale de gradul doi.

Cuvinte cheie: algoritmi de calcul numeric, modelare matematică, programare vizuală, simulare asistată de calculator, fenomene oscilatorii.

MATHEMATICAL MODELING AND COMPUTER-AIDED SIMULATION OF MECHANICAL SYSTEMS WITH MULTIPLE DEGREES OF FREEDOM

Abstract. The vast majority of phenomena and processes in nature for which computer-assisted simulation is of scientific and methodical-didactic interest, from a mathematical point of view, are described by systems of differential equations of the second order. In this context, the paper describes and implements in the Object Pascal programming language an original algorithm for the numerical solution of systems of differential equations of the second order. Interactive simulation applications are made in the Delphi programming environment (FMX variant). Based on the examples, described in detail in the paper, users can perform interactive simulation activities for any processes, which from a mathematical point of view are described by systems of differential equations of the second degree.

Keywords: numerical computing algorithms, mathematical modeling, visual programming, computer-aided simulation, oscillatory phenomena.

1. Introducere

Din punct de vedere matematic, sistemele mecanice cu multe grade de libertate pot fi descrise printr-un sistem implicit de ecuații diferențiale ordinare de gradul doi, care în cazul utilizării formalismului Lagrange are forma :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i, \quad i = 0 \dots (n-1), \quad (1)$$

unde $q_i, \dot{q}_i = \frac{dq_i}{dt}, Q_i(t, q_0, q_1, \dots, q_{n-1}, \dot{q}_0, \dot{q}_1, \dots, \dot{q}_{n-1})$ - sunt coordonatele, vitezele și forțele neconservative generalizate, $L = T(q_0, q_1, \dots, q_{n-1}, \dot{q}_0, \dot{q}_1, \dots, \dot{q}_{n-1}) - V(t, q_0, q_1, \dots, q_{n-1})$,

T -energia cinetică, V -energia potențială, L -funcția Lagrange, n -numărul gradelor de libertate.

În caz general sistemul de ecuații (1) poate fi reprezentat în următoarea formă implicită:

$$\sum_{j=0}^{n-1} a_{i,j} \ddot{q}_j = b_i, \quad i = 0 \dots (n-1), \quad (2)$$

unde coeficienții $a_{i,j}$ sunt funcții de tipul $f(q_0, q_1, \dots, q_{n-1}, \dot{q}_0, \dot{q}_1 \dots \dot{q}_{n-1})$. Coeficienții b_i suplimentar depind și de timp.

Rezolvând sistemul de ecuații liniare (2) față de necunoscutele $\ddot{q}_j$, obținem următorul sistem de ecuații diferențiale, în formă explicită:

$$\ddot{q}_i = \frac{d^2 q_i}{dt^2} = F_i(t, q_0, q_1, \dots, q_{n-1}, \dot{q}_0, \dot{q}_1 \dots \dot{q}_{n-1}), \quad i = 0 \dots (n-1), \quad (3)$$

Pentru simularea asistată de calculator a comportamentului sistemelor mecanice cu multe grade de libertate în funcție de timp sunt necesare metode analitice sau numerice de rezolvare a sistemului de ecuații diferențiale (3), în baza condițiilor inițiale care, de regulă sunt formulate prin intermediul coordonatelor și vitezelor generalizate ale sistemului în momentul inițial de timp (problema Cauchy):

$$t = 0, q_i = q_{0i}, \dot{q}_i = \dot{q}_{0i}, \quad i = 0 \dots (n-1) \quad (4)$$

2. Metode directe pentru rezolvarea numerică a sistemelor de ecuații diferențiale ordinare de ordinul doi

În lucrarea [2] a fost descris și implementat un algoritm de rezolvare numerică directă a unei ecuații diferențiale ordinare de ordinul doi. Prezentăm în continuare generalizarea acestui algoritm pentru rezolvarea numerică a unui sistem de ecuații diferențiale ordinare de ordinul doi, implementat în procedura RSED2RK4_M, cu ajutorul căreia, în baza condițiilor inițiale (4) se determină soluțiile numerice ale sistemului de ecuații diferențiale (3) pentru momentul de timp t , cu precizia 10^{-12} , setată în mod predefinit. Dacă această precizie nu se atinge pe parcursul intervalului de timp <0.01 secundă, procesul iterativ de ameliorare a soluțiilor se stopează forțat.

```

procedure RSED2RK4_M(t0:extended;x0,v0:array of extended;
                    pfs:pfsxv;t:extended;var x,v:array of extended);
//x0,v0, x,v- tablouri dinamice care conțin condițiile inițiale în momentul t0 și
//soluțiile sistemului de ecuații diferențiale, calculate în momentul //de timp t;
//pfsxv- tip procedural care se declară în afara procedurii;
//type pfsxv=procedure(t:extended;x,v:array of extended;var f:array of //extended);
var n,i,j,ned:integer;h,tt0,ti,eps:extended; p:boolean;
xc,xcp,vc,vcp,xx0,vv0,k0,k1,k2,k3,f,x00,v00, xh2,vh2, xh,
vh:array[0..100] of extended;
procedure RSED2RK4(t0:extended;x0,v0:array of extended;pfs:pfsxv;t:extended;
                    var x,v:array of extended);
//procedura rezolvă sistemul de ecuații diferențiale fără iterații:

```

```

var n,i,ned:integer;h:extended; p:boolean;
xc,vc,k0,k1,k2,k3,f,xh2,vh2, xh,vh:array[0..100] of extended;
begin //începutul procedurii RSED2RK4
    ned:=high(x0); h:=t-t0; n:=1;
    pfs(t0,x0,v0,f);
    for i:=0 to ned do begin
        xh2[i]:=x0[i]+v0[i]*h/2+f[i]*sqr(h/2)/2;
        xh[i]:=x0[i]+v0[i]*h+f[i]*sqr(h)/2;
        k0[i]:=h*f[i];   vh2[i]:=v0[i]+k0[i]/2; end;
    pfs(t0+h/2,xh2,vh2,f); for i:=0 to ned do
        begin k1[i]:=h*f[i];   vh2[i]:=v0[i]+k1[i]/2 end;
    pfs(t0+h/2,xh2,vh2,f); for i:=0 to ned do
        begin k2[i]:=h*f[i];   vh2[i]:=v0[i]+k2[i] end;
    pfs(t0+h,xh,vh2,f); for i:=0 to ned do
        begin k3[i]:=h*f[i];
            vc[i]:=v0[i]+(k0[i]+2*k1[i]+2*k2[i]+k3[i])/6;
            xc[i]:=x0[i]+v0[i]*h+f[i]*sqr(h)/2 ;
        end;
    for i:=0 to ned do begin x[i]:=xc[i];v[i]:=vc[i];end;
end; //sfârșitul procedurii RSED2RK4
begin //începutul procedurii RSED2RK4_M
    ned:=high(x0); eps:=1E-12;// precizia predefinită
    h:=t-t0; n:=1;
    RSED2RK4(t0,x0,v0,pfs,t,xc,vc);// determină soluțiile în primă aproximație
    Repeat // etapa iterativă de ameliorare a soluțiilor
        ti:=now;
        for i:=0 to ned do begin
            xp[i]:=xc[i];vp[i]:=vc[i];   xx0[i]:=x0[i];vv0[i]:=v0[i];end;
            h:=h/2;n:=n*2;   tt0:=t0; // intervalul de integrare se divizează în
                                //subintervale 2,4,8,...

        for j:=1 to n do begin
            RSED2RK4(tt0,xx0,vv0,pfs,tt0+h,xc,vc);//
            for i:=0 to ned do begin   xx0[i]:=xc[i]; vv0[i]:=vc[i];   end;
            tt0:=tt0+h;
        end;
        p:=true;
        for i:=0 to ned do
            p:=p and (abs(xc[i]-xp[i])<eps)and (abs(vc[i]-vp[i])<eps);
        //se verifică precizia (diferența dintre soluția curente și cea precedentă)
        if (now-ti)*24*3600>0.01 then p:=true
        //în cazul când timpul de calcul în procedură este mai mare de 0.01 secunde,
        //procesul de verificare a preciziei se finalizează forțat
    until p;
    for i:=0 to ned do begin x[i]:=xc[i];v[i]:=vc[i];end;
end; //sfârșit RSED2RK4_M

```

3. Validarea, prin experiment numeric, a algoritmului de rezolvare numerică a sistemelor de ecuații diferențiale de ordinul doi

Pentru validarea algoritmului de rezolvare numerică a sistemelor de ecuații diferențiale ordinare de ordinul doi, implementat în procedura RSED2RK4_M vom realiza câteva activități interactive de modelare-simulare a unor sisteme mecanice cu multe grade de libertate.

3.1. Pendulul dublu

Pendul dublu (două puncte materiale articulate prin intermediul a două bare rigide fără masă) este un sistem mecanic cu două grade de libertate. Dacă în calitate de coordonate generalizate se aleg unghiurile de abatere de la poziția de echilibru (fig. 1), relativ simplu se obțin expresiile pentru energia cinetică și potențială [1], în baza cărora se scriu în formă explicită expresiile pentru energia totală E (4) și funcția Lagrange (5) a sistemului.

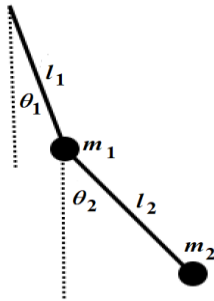


Figura1.

$$T = \frac{1}{2}(m_1 + m_2)l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2l_2^2\dot{\theta}_2^2 + l_1l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2) \quad (4)$$

$$V = -(m_1 + m_2)gl_1\cos(\theta_1) - m_2gl_2\cos(\theta_2) \quad (5)$$

$$E = T + V \quad (6)$$

$$L = T - V \quad (7)$$

Pendulul dublu

Cunoscând funcția Lagrange a sistemului (7), cu ajutorul expresiei (1) se scriu ecuațiile diferențiale care guvernează mișcările pendulului dublu:

$$(m_1 + m_2)l_1\ddot{\theta}_1 + m_2l_2\ddot{\theta}_2\cos(\theta_1 - \theta_2) = -m_2l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2) - (m_1 + m_2)g\sin(\theta_1) \quad (8)$$

$$m_2l_2\ddot{\theta}_2 + m_2l_1\ddot{\theta}_1\cos(\theta_1 - \theta_2) = m_2l_1\dot{\theta}_1^2\sin(\theta_1 - \theta_2) - m_2g\sin(\theta_2) \quad (9)$$

Rezolvând acest sistem de ecuații liniare în raport cu necunoscutele $\ddot{\theta}_1$ și $\ddot{\theta}_2$ obținem:

$$\ddot{\theta}_1 = (b_1a_{22} - b_2a_{12})/(a_{11}a_{22} - a_{21}a_{12}) \quad (10)$$

$$\ddot{\theta}_2 = (b_2a_{11} - b_1a_{21})/(a_{11}a_{22} - a_{21}a_{12}) \quad (11)$$

unde

$$a_{11} = (m_1 + m_2)l_1; \quad a_{12} = m_2l_2\cos(\theta_1 - \theta_2);$$

$$a_{21} = m_2l_1\cos(\theta_1 - \theta_2); \quad a_{22} = m_2l_2;$$

$$b_1 = -m_2l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2) - (m_1 + m_2)g\sin(\theta_1);$$

$$b_2 = m_2l_1\dot{\theta}_1^2\sin(\theta_1 - \theta_2) - m_2g\sin(\theta_2);$$

3.1.1. Designul și codul aplicației "Pendulul dublu" (aplicație Delphi FMX)

Pentru simularea oscilațiilor pendulului dublu, în fereastra Form1, din componentele Delphi FMX se realizează designul reprezentat în fig. 2.

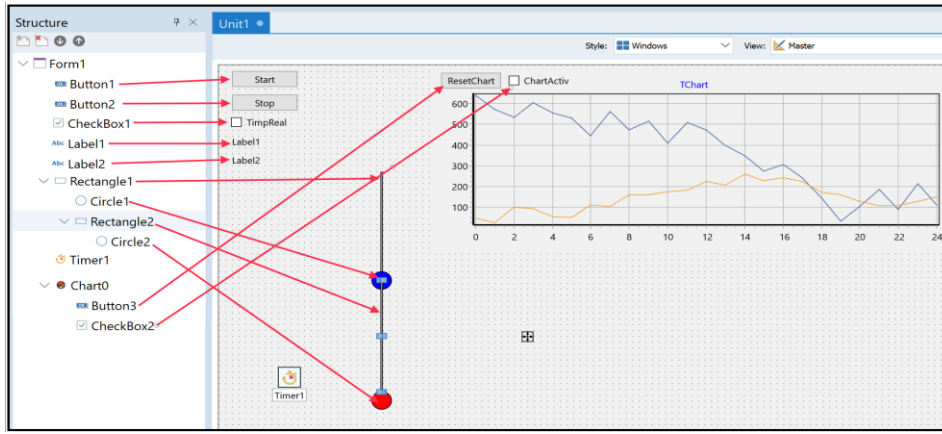


Figura 2. Structura și designul aplicației „Pendul dublu”

Codul aplicației se implementează în unitatea Unit1, respectând următoarele etape:

1. Se declară tipul procedural de date:

```
type pfsxv=procedure(t:extended;x,v:array of extended;var f:array of extended);
```

și variabilele globale:

```
var m1,m2,l1,l2,g,t0,ET0,ET,ti,t:extended;
var xx0,vv0,xx,vv:array of extended;
```

2. Se implementează procedura pfpd în care se descrie sistemul de ecuații diferențiale

(8)-(9), utilizând notațiile: $\theta_1=x[0]$, $\theta_2=x[1]$, $\dot{\theta}_1=v[0]$, $\dot{\theta}_2=v[1]$.

```
procedure pfpd(t:extended;x,v:array of extended;var f:array of extended);
```

```
var a11,a12,b1,a21,a22,b2:extended;
```

```
begin //pendul dublu
```

```
    a11:=(m1+m2)*l1; a12:=m2*l2*cos(x[0]-x[1]);
    b1:=-m2*l2*sqr(v[1])*sin(x[0]-x[1])-(m1+m2)*g*sin(x[0]);
    a21:=m2*l1*cos(x[0]-x[1]); a22:=m2*l2;
    b2:=m2*l1*sqr(v[0])*sin(x[0]-x[1])-m2*g*sin(x[1]);
    f[0]:=(b1*a22-b2*a12)/(a11*a22-a12*a21);
    f[1]:=(a11*b2-b1*a21)/(a11*a22-a12*a21);
```

```
end;
```

3. Se implementează procedura RSED2RK4 codul căreia este descris mai sus.

4. Se implementează codul Button1.Click (butonul Start):

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin g:=9.81;// accelerația căderii libere
```

```
    setlength(xx0,2); setlength(vv0,2); setlength(xx,2); setlength(vv,2);
```

```
    m1:=1;m2:=1;l1:=1;l2:=1;// valorile parametrilor pendulului dublu
```

```
    t0:=0; xx0[0]:=pi/14; vv0[0]:=0; xx0[1]:=-3*pi/4; vv0[1]:=0;
```

```
    // condițiile inițiale:
```

```
    ET0:=0.5*(m1+m2)*sqr(l1*vv0[0])+ 0.5*m2*sqr(l2*vv0[1])+
```

```
    m2*l1*l2*vv0[0]*vv0[1]*cos(xx0[0]-xx0[1])- // se calculează energia
```

```
    //totală (m1+m2)*g*l1*cos(xx0[0])- m2*g*l2*cos(xx0[1]); în momentul inițial
```

```
    if CheckBox1.IsChecked then ti:=now else t:=0;
```

```
    // dacă CheckBox1 bifat/debifat setează animația în timp real/scalat
```

```

chart0.Series[0].Clear;    //inițializează obiectul chart0 pentru construirea
chart0.Series[0].Color:=TAlphacolors.Blue;// graficelor
chart0.Series[1].Clear;
chart0.Series[1].Color:=TAlphacolors.Red;
timer1.Enabled:=true;// Porneste timer-ul în care se realizează animația
end;

5. Se implementează codul pentru procedura OnTimer
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    if CheckBox1.IsChecked then t:=(now-ti)*24*3600 else t:=t+0.01;
    //dacă CheckBox1 este bifat timpul t este real, în caz contrar timpul este scalat
    label1.Text:='t='+t.ToString;// se afișează timpul în secunde
    RSED2RK4_M(t0,xx0,vv0,pfpd,t,xx,vv);//în baza condițiilor inițiale se
    //determină soluțiile ecuațiilor diferențiale pentru momentul de timp t
    rectangle1.RotationAngle:=xx[0]*180/pi;
    Rectangle2.RotationAngle:=xx[1]*180/pi;
    // se realizează rotirea pendulelor
    t0:=t; xx0[0]:=xx[0]; vv0[0]:=vv[0]; xx0[1]:=xx[1]; vv0[1]:=vv[1];
    //se modifică condițiile inițiale: soluțiile obținute la etapa curentă
    //devin condiții inițiale pentru următoarea etapă
    if checkbox2.IsChecked then
    begin
        chart0.series[0].AddXY(t,xx[0]); chart0.series[1].AddXY(t,xx[1]); end;
        ET:=0.5*(m1+m2)*sqr(l1*vv[0])+ 0.5*m2*sqr(l2*vv[1])+
        m2*l1*l2*vv[0]*vv[1]*cos(xx[0]-xx[1])-(m1+m2)*g*l1*cos(xx[0])-
        m2*g*l2*cos(xx[1]);
        //Se calculează energia totală a sistemului la momentul curent
        label1.Text:=(et-et0).ToString;
    end;
end;

```

După implementarea acestor etape, aplicația poate fi utilizată pentru realizarea experimentelor numerice:

1. Deoarece problema pendulului dublu nu are soluții analitice, vom verifica corectitudine soluțiilor numerice indirect, verificând diferența dintre energia totală a sistemului în momentul curent de timp și cel inițial. Deoarece sistemul este conservativ, această diferență trebuie să fie egală cu 0.

În rezultatul experimentului numeric, pe parcursul a 60 minute, diferență $E(t)-E(t_0)$ nu a depășit valoarea 10^{-6} (rezultatul se afișează în obiectul Label1).

Acest rezultat demonstrează, indirect, corectitudinea algoritmului de rezolvare numerică a sistemelor de ecuații diferențiale ordinare de ordinul doi.

2. Conform teoriei, în cazul oscilațiilor mici $\theta_1, \theta_2 \ll 1$, $m_1 \gg m_2$, $l_1 = l_2$ are loc fenomenul de transfer al energiei de la un pendul la altul (fenomenul de bătai). Pentru $m_1 = 1; m_2 = 0.05; l_1 = 0.25; l_2 = 0.25; \theta_1 = 0, \theta_2 = 0.2$,

rezultatul experimentului numeric este reprezentat în fig.3.

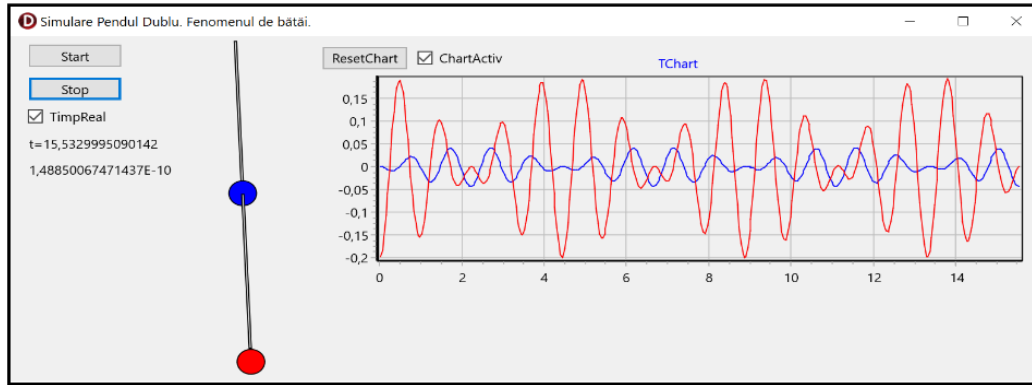


Figura 3. Pendulul dublu (observarea fenomenului de bătăi)

3. Conform teoriei oscilațiile pendulului dublu sunt exemple naturale de fenomene haotice. Pentru verificarea acestei afirmații în designul aplicație se mai adaugă încă 3 obiecte de tip TChart. În Chart0 și Chart1 vom afișa graficele $\theta_1 = f(t)$ și $\theta_2 = f(t)$, iar în Chart2 și Chart3, graficele $\theta_1 = f(\dot{\theta}_1)$ și $\theta_2 = f(\dot{\theta}_2)$ – portretele de fază. Aceste grafice se construiesc în procedura TForm1.Timer1Timer(Sender: TObject), prin adăugarea următoarei porțiuni de cod:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
... ..
... ..
if checkbox2.IsChecked then begin
chart0.series[0].AddXY(t,xx[0]);
chart2.series[0].AddXY(xx[0],vv[0]);
chart1.series[0].AddXY(t,xx[1]);
chart3.series[0].AddXY(xx[ngl-1],vv[1]);
end;
end;
```

Pentru parametrii $m1:=1kg; m2:=1kg; l1:=1m; l2:=2m; \theta_1=3\pi/4, \theta_2=0$, rezultatul experimentului numeric, realizat pe parcursul a 20 secunde este reprezentat în fig.4. Reprezentările grafice din fig.1 demonstrează caracterul haotic al oscilațiilor pentru ambele pendule (trajectoriile acestor oscilații în spațiul de fază nu sunt figuri închise).

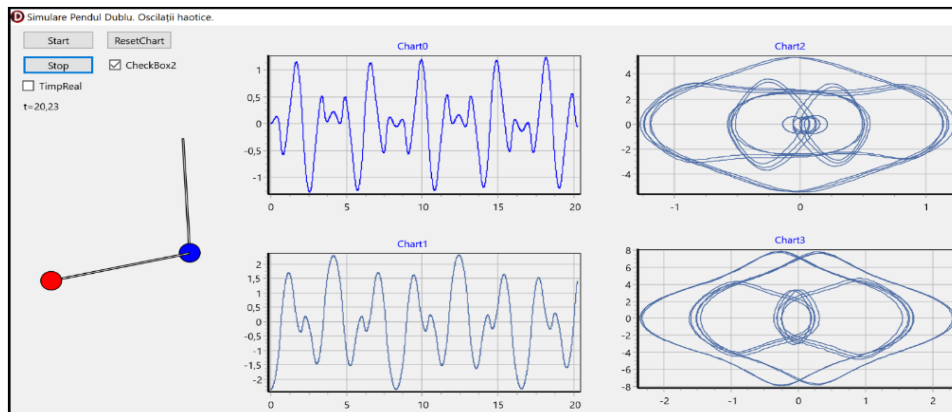


Figura 4. Pendulul dublu (demonstrarea fenomenului de haos)

3.2. Oscilatori liniari cuplați

Sistemele de oscilatori cuplați sunt utilizate pentru modelare diferitor structuri și fenomene fizice: cristale, molecule, procesul de formare și propagare a undelor elastice etc. Considerăm următorul sistem unidimensional de n corpuri identice, de masă m cuplate între ele cu ajutorul a $(n+1)$ resorturi, de asemenea identice cu coeficientul de elasticitate k . Resorturile marginale au coeficientul de elasticitate k' . Schema acestui sistem este reprezentată în fig.5.



Figura 5. Schema sistemului de oscilatori cuplați

Ecuatiile diferențiale care descriu mișcările corpurilor din acest sistem se obțin simplu, identificând forțele care acționează asupra fiecărui corp și aplicând legea a doua a lui Newton:

$$\begin{aligned}\ddot{x}_0 &= k(x_1 - x_0) - k'x_0 \\ m\ddot{x}_i &= k(x_{i+1} - 2x_i + x_{i-1}), \quad i = 1, 2, \dots, n-2 \\ m\ddot{x}_{n-1} &= -k'x_{n-1} + k(x_{n-2} - x_{n-1})\end{aligned}\quad (12)$$

3.2.1. Designul și codul aplicației "Oscilatori cuplați" (aplicație Delphi FMX)

Designul aplicației este reprezentat în fig.5.

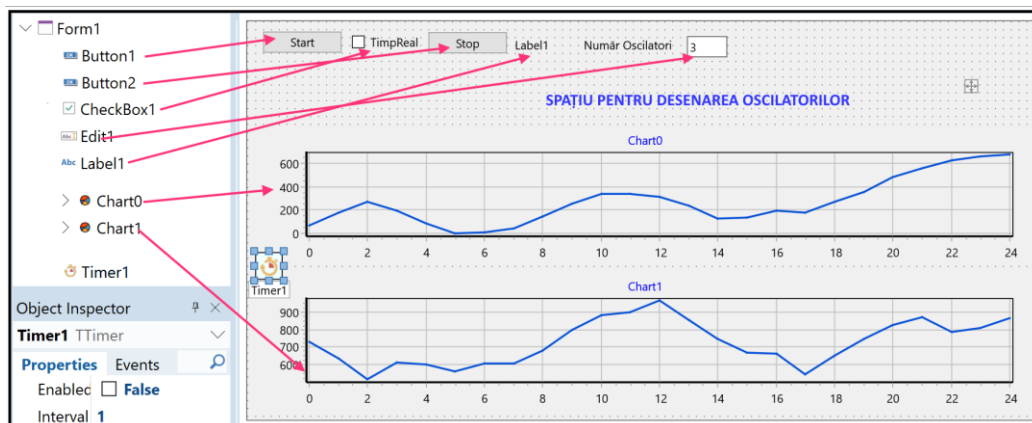


Figura 5. Designul aplicației „Oscilatori Cuplați”

La etapa de design, utilizatorul include în fereastra Form1 setul de componente, necesare pentru realizarea interactivă a animației: Butonul "Start" demarează animația; CheckBox1 setează modul de realizare a animației (timp real/scalat); în Label1 se afișează timpul curent al animației; în Edit 1 se setează numărul de oscilator; Valorile celorlalți parametri care descriu animația sunt definiți în Button1.Click. În componentele Chart0 și Chart1 se afișează graficele oscilațiilor pentru primul și ultimul corp al sistemului.

Codul aplicației se realizează în compartimentul **implemetation** al unității Unit1.pas, parcurgând următoarele etape:

- Se implementează compartimentul declarațiilor globale:
`const na=51; ngl=na-1; larc=80;`
`type pfsxv=procedure(t:extended;x,v:array of extended;`

var f:array of extended);

```
var m,k,kf,t0,ti,t,E0,Et:extended;
var xx0,vv0,xx,vv:array of extended;
imgr:array[0..(na-1)] of Timage; //obiecte dinamice pentru reprezentarea
//grafică
cr:array[0..(ngl-1)] of TCircle; // pentru desenarea dinamică a oscilatorilor
lb:array[0..(ngl-1)] of tlabel; // cuplați
```

2. Se implementează procedura DrawArch:

```
DrawArch(img:tImage;n:integer;dd:extended);
//Desenează un arc cu n spire; dd-grosimea (<=10)
var i:integer; h,h2,L,p,ll:extended;
ap:array of TPointF;
begin
  setlength(ap,2*n+4); l:=Img.Width; ll:=10;
  if l>=2*ll then
    begin
      p:=(1-2*ll)/(n)/4; if p<=0 then p:=1;
      h:=Img.Height;h2:=h/2; ;
      ap[0].X:=0;ap[0].Y:=h2;
      ap[1].X:=ll;ap[1].Y:=h2;
      ap[2].X:=ap[1].x+p;ap[2].Y:=0;
      for i:=3 to 2*n+1 do
        begin
          ap[i].X:=ap[i-1].x+2*p;
          if i mod 2 =1 then ap[i].Y:=h else ap[i].Y:=0;
        end;
      ap[2*n+2].x:=ap[2*n+1].x +p;ap[2*n+2].y:=h2;
      ap[2*n+3].x:=ap[2*n+2].x +ll;ap[2*n+3].y:=h2;
      Img.Bitmap.SetSize(Round(Img.Width), Round(Img.Height));
      img.Bitmap.BitmapScale:=1;
      Img.Bitmap.Canvas.BeginScene;//
      Img.Bitmap.Canvas.Stroke.Thickness:=dd;
      for i := 1 to 2*n+3 do Img.Bitmap.Canvas.DrawLine(ap[i-1],ap[i],1);
      Img.Bitmap.Canvas.EndScene;
    end;
  end;
```

3. Se implementează procedura în care se definesc ecuațiile diferențiale în conformitate cu expresiile (12):

```
procedure pfpd(t:extended;x,v:array of extended;var f:array of extended);
var i,n:integer;
begin
  n:=high(x);
  f[0]:=(-k*x[0]-kf*(x[0]-x[1]))/m;
  for i := 1 to n-1 do f[i]:=(k*(x[i+1]-2*x[i]+x[i-1]))/m);
```

- ```

 f[n]:=(-kf*x[n]-k*(x[n]-x[n-1]))/m;
end;

```
4. Se implementează codul pentru butonul „Start”:
- ```

procedure TForm1.Button1Click(Sender: TObject);
var i:integer;
begin k:=30; kf:=30; m:=1; dc:=10; sdc:=5;
    na:=edit1.Text.ToInteger; ngl:=na-1; larc:=80;
    for i:=0 to High (cr) do begin lb[i].Destroy; cr[i].Destroy; end;
    for i:=0 to High (imgr) do imgr[i].Destroy;
    setlength(cr, ngl); setlength(lb, ngl); setlength(imgr, na);
    setlength(xx0, ngl); setlength(vv0, ngl); setlength(xx, ngl);
    setlength(vv, ngl);
    t0:=0;
    for i:=0 to ngl-1 do begin
        if not odd(i) then xx0[i]:=20 else xx0[i]:=-20 ; vv0[i]:=0; end;
    //for i:=0 to ngl-1 do begin if not odd(i) then
        //xx0[i]:=0 else xx0[i]:=-0 ; vv0[i]:=0; end;
    //xx0[0]:=30 ; xx0[7]:=-0 ;
    for i := 0 to na-1 do begin
        imgr[i]:=TImage.Create(self); imgr[i].Parent:=self;
        imgr[i].Height:=30;
        imgr[i].Position.Y:=50; //100-imgr[i].Height/2+dc/2;
    end;
    for i:=0 to ngl-1 do begin
        cr[i]:=TCircle.Create(self) ; cr[i].Parent:=self;
        cr[i].Width:=dc; cr[i].Height:=dc;
        cr[i].Position.y:=50+imgr[i].Height/2-sdc;
        cr[i].Position.x:=larc+Larc*i+xx0[i]-sdc;
        lb[i]:=TLabel.Create(self) ; lb[i].Parent:=cr[i];
        lb[i].Position.x:=2; lb[i].Position.y:=-15; lb[i].Text:=i.ToString;
    end;
    for i:=0 to na-1 do DrawArcH(imgr[i], 8, 1);
    chart0.Series[0].Clear; chart1.Series[0].Clear;
    if CBox1.IsChecked then ti:=now else t:=0;
    timer1.Enabled:=true;
end;

```
5. Se implementează codul pentru procedura Timer1Timer(Sender: TObject):
- ```

procedure TForm1.Timer1Timer(Sender: TObject);
var i:integer;
begin
 if CBox1.IsChecked then t:=(now-ti)*24*3600 else t:=t+0.01;
 label1.text:='t='+t.ToString;
 RSED2RK4_M(t0, xx0, vv0, pfpd, t, xx, vv);
 t0:=t; for i:=0 to high(xx0) do begin xx0[i]:=xx[i]; vv0[i]:=vv[i]; end;

```

```

for i:=0 to ngl-1 do cr[i].Position.x:=larc+Larc*i+xx[i]-sdc;
for i:=1 to na-2 do
begin
 imgr[i].Position.x:=cr[i-1].Position.x+sdc;
 imgr[i].Width:=cr[i].Position.X-cr[i-1].Position.X;
end;
imgr[na-1].Position.x:=cr[ngl-1].Position.X+sdc;
imgr[ngl].Width:=80*na-cr[ngl-1].Position.X;
imgr[0].Width:=cr[0].Position.X;{ +xx[0]+sdc;}
imgr[0].Position.x:=cr[0].Position.x-imgr[0].Width+sdc;
for i:=0 to na-1 do DrawArch(imgr[i],8,1);
chart0.series[0].AddXY(t,xx[0]);
chart1.series[0].AddXY(t,xx[7]);
end;

```

După implementarea acestor etape, aplicația poate fi utilizată pentru realizarea experimentelor numerice în timp real sau scalat.

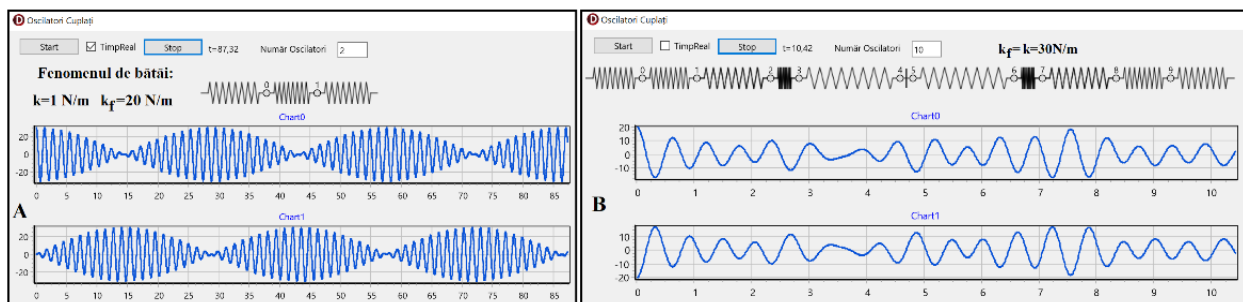
### 3.2.2. Experimente numerice cu oscilatori cuplați

Prezentăm în continuare câteva capturi de ecran obținute în cadrul experimentelor numerice.

În fig.6A se simulează fenomenul de bătăi în sistemul format din doi oscilatori identici  $m_1=m_2=1\text{kg}$ ,  $k_f=20\text{N/m}$  cuplați prin intermediul resortului cu rigiditatea  $k=1\text{N/m}$ . Conform teoriei [3], perioada bătăilor trebuie să fie:

$$T = 2\pi\sqrt{m}/\left(\sqrt{k_f + 2k} - \sqrt{k_f}\right) \approx 28.78$$

Acest rezultat se confirmă în fig.6A.



**Figura 6. Oscilatori cuplați: exemple de oscilații**

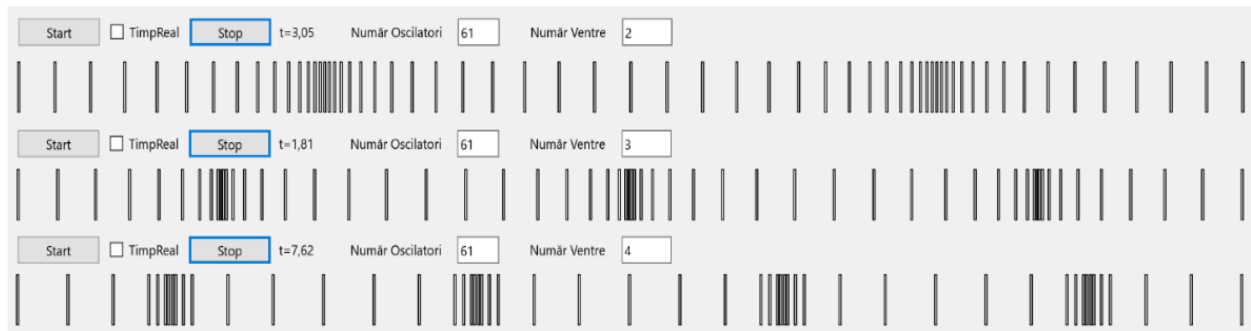
În fig.6B sunt reprezentate oscilațiile unui sistem format din 10 oscilatori identici, care inițial au fost excitați antisimetric. Pentru acest caz, conform teoriei primului, și ultimul oscilator trebuie să oscileze identic, dar faze opuse. Acest rezultat se confirmă în fig.6B.

Aplicația descrisă în acest paragraf, ușor poate fi modificată pentru simularea proceselor de formare a undelor staționare. Pentru aceasta:

- se exclud din codul aplicației instrucțiunile care desenează resorturile (obiectele `imgr`: array of `TImage`);
- obiectele `TCircle` se înlocuiesc cu `TRectangle`;

- condițiile inițiale se setează în conformitate cu profilul inițial al unde (sinusoidă cu numărul dorit de maxime (ventre)).

În fig.7 sunt prezentate trei capturi de ecran în care se observă unde staționare cu 2, 3 și 4 ventre.



**Figura 7. Unde staționare în sistemul de oscilatori cuplați**

## Concluzii

În baza investigațiilor realizate și a codurilor de programare Pascal, prezente în lucrare, utilizatorul final (elevii, studenții, masteranzii, cadrele didactice), cu eforturi minimale și cunoștințe medii de programare Delphi poate restabili aplicațiile descrise și dezvolta în continuare alte aplicații de modelare-simulare de concepție proprie pentru diverse sisteme dinamice, modelul matematic al cărora conține sisteme de ecuații diferențiale de ordinul doi.

## Bibliografie

1. NECULAI, A. *Eseu despre Fundamentele Modelarii Matematice*. București: Editura Academiei Romane, 2012.
2. BALMUȘ, N. Modelarea matematică și simularea asistată de calculator a sistemelor mecanice cu un grad de libertate. În: *Acta et Commentationes (Științe ale Educației)*, 2024, nr.2(36), pp. 43-52.
3. COURTEAUD, J. Cours sur les oscillateurs couplés. [https://www.sciencesalecole.org/wp-content/uploads/2018/01/IPhO-COURTEAUD-Cours\\_oscillateurs\\_couples.pdf](https://www.sciencesalecole.org/wp-content/uploads/2018/01/IPhO-COURTEAUD-Cours_oscillateurs_couples.pdf)