# Mathematical Modeling in Delphi

Andreï V. Serghienko
Yaroslavl P.G. Demidov State University
Yaroslavl, Russia

The literature on Delphi [1] numbers many manuals. However among them there are few books, oriented to the solution of scientific and technical problems. The work contains the examples of programs in Delphi.

Depending on the type of problems it is convenient to use or graphic (usual), or console applications of Delphi. Graphic applications are applicable for the plotting of functions. Console applications are applicable especially, when we needn't the visualization, and it is necessary to introduce by hand lots of data. One can tell the console applications from the graphic ones with the presence of the next line in the text of program:

```
{$APPTYPE CONSOLE}
```

## 1 Systems of Linear Algebraic Equations

The next program solves the system of linear algebraic equations ($a_{00} \neq 0$):

$$\begin{aligned}
&a_{00}x_0 + a_{01}x_1 + ... + a_{0n}x_n = b_0,\\
&a_{10}x_0 + a_{11}x_1 + ... + a_{1n}x_n = b_1,\\
&...\\
&a_{n0}x_0 + a_{n1}x_1 + ... + a_{nn}x_n = b_n;
\end{aligned}$$

using the compact elimination scheme [2]:

$$\gamma_{i0} = a_{i0} \;\; (i = 0, 1, ..., n)\,, \;\; \alpha_{0k} = \frac{a_{0k}}{a_{00}} \;\; (k = 1, 2, ..., n)\,,$$

$$\gamma_{ik} = a_{ik} - \sum_{j=0}^{k-1} \gamma_{ij}\alpha_{jk} \;\; (i, k = 1, 2, ..., n;\; i \geqslant k)\,,$$

$$\alpha_{ik} = \frac{1}{\gamma_{ii}} \left( a_{ik} - \sum_{j=0}^{i-1} \gamma_{ij}\alpha_{jk} \right) \;\; (i = 1, 2, ..., n;\; k = 2, 3, ..., n;\; i < k)\,,$$

$$\beta_0 = \frac{b_0}{a_{00}},\;\; \beta_i = \frac{1}{\gamma_{ii}} \left( b_i - \sum_{j=0}^{i-1} \gamma_{ij}\beta_j \right) \;\; (i = 1, 2, ..., n)\,;$$

$$x_n = \beta_n,$$

$$...$$

$$x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n = \beta_1,$$

$$x_0 + \alpha_{01}x_1 + \alpha_{02}x_2 + ... + \alpha_{0n}x_n = \beta_0.$$

```
program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var
  aa: array[0..9000,0..9000] of real;
  bb: array[0..9000] of real;
  gamma: array[0..9000,0..9000] of real;
  alpha: array[0..9000,0..9000] of real;
  beta: array[0..9000] of real;
  xx: array[0..9000] of real;

var
  i,j,k,l,m,n:integer;
begin
  { TODO -oUser -cConsole Main : Insert code here }
  writeln('Compact Elimination Scheme.');
  write('A system rank m = '); readln(m);
  n:=m-1;
```

```
for i:=0 to n do
    begin
       write('Elements '); writeln('a['+IntToStr(i)+',0–'+IntToStr(n)+']:');
       for k:=0 to n do read(aa[i,k]);
    end;
write('A column '); writeln('b[0–'+IntToStr(n)+']:');
for i:=0 to n do read(bb[i]); readln;
for i:=0 to n do gamma[i,0]:=aa[i,0];
for k:=1 to n do alpha[0,k]:=aa[0,k]/aa[0,0];
for l:=1 to n–1 do
    begin
       for i:=l to n do
           begin
              gamma[i,l]:=aa[i,l];
              for j:=0 to l–1 do gamma[i,l]:=gamma[i,l]–gamma[i,j]*alpha[j,l];
           end;
       for k:=l+1 to n do
           begin
              alpha[l,k]:=aa[l,k];
              for j:=0 to l–1 do alpha[l,k]:=alpha[l,k]–gamma[l,j]*alpha[j,k];
              alpha[l,k]:=alpha[l,k]/gamma[l,l];
           end;
    end;
gamma[n,n]:=aa[n,n];
for j:=0 to n–1 do gamma[n,n]:=gamma[n,n]–gamma[n,j]*alpha[j,n];
beta[0]:=bb[0]/aa[0,0];
for i:=1 to n do
    begin
       beta[i]:=bb[i];
       for j:=0 to i–1 do beta[i]:=beta[i]–gamma[i,j]*beta[j];
       beta[i]:=beta[i]/gamma[i,i];
    end;
xx[n]:=beta[n];
for k:=n–1 downto 0 do
    begin
       xx[k]:=beta[k];
       for j:=k+1 to n do xx[k]:=xx[k]–alpha[k,j]*xx[j];
    end;
```

```
  writeln('The solution of system:');
  for k:=0 to n do
     begin
       write('x['+IntToStr(k)+']='); writeln(xx[k]:4:2);
     end;
  readln;
end.
```

## 2 Method of Tangents

The method of tangents [2] is a powerful iterative method of solution of transcendental equations. The algorithm of method of tangents:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

where $x_i$ is the i-th approximation of solution of the equation

$$f(x) = 0.$$

The next program plots a function, which we give to it. Before the launch of the program it is necessary to create a file 'In.bmp' with the dimensions 455×405 square pixels (one can more, but not less!) and put it in the same folder, that the program. This file will represent a canvas, on which the program will draw a function. In the text of program indicate the function

$$f(x) = x^3 - 5x^2 + 5x + 1$$

and the segment of values of argument

$$-0.5 \leqslant x \leqslant 3.8.$$

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
```

Figure 1: The file 'In.bmp'.

```
  TForm1 = class(TForm)
    Go: TButton;
    procedure GoClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  STEP = 1E–4;
  x0 = –0.5;
  xfin = 3.8;

var
  Form1: TForm1;
  Figure1: TBitMap;

implementation

{$R *.dfm}

function f(x:real):real;
begin
```

```
  f:=x*x*x–5*x*x+5*x+1;
end;

procedure TForm1.GoClick(Sender: TObject);
var
  i:integer;
  x,y,min,max:real;
begin
  Figure1:=TBitMap.Create;
  Figure1.LoadFromFile('In.bmp');
  with Figure1.Canvas do
     begin
       MoveTo(60,360); LineTo(360,360);
       MoveTo(60,360); LineTo(60,60);
       for i:=0 to 10 do
          begin
            MoveTo(60+30*i,365); LineTo(60+30*i,355);
          end;
       for i:=0 to 50 do
          begin
            MoveTo(60+6*i,363); LineTo(60+6*i,357);
          end;
       for i:=0 to 10 do
          begin
            MoveTo(55,360–30*i); LineTo(65,360–30*i);
          end;
       for i:=0 to 50 do
          begin
            MoveTo(57,360–6*i); LineTo(63,360–6*i);
          end;
       Pen.Style:=psDot;
       for i:=1 to 10 do
          begin
            MoveTo(60+30*i,355); LineTo(60+30*i,60);
          end;
       for i:=1 to 10 do
          begin
            MoveTo(65,360–30*i); LineTo(360,360–30*i);
          end;
```

```
        Pen.Style:=psSolid;
        x:=x0;
        min:=f(x);
        max:=f(x);
        while x<xfin do
          begin
            x:=x+STEP; y:=f(x);
            if y<min then min:=y;
            if y>max then max:=y;
          end;
        x:=x0; y:=f(x);
        while x<=xfin do
          begin
            Pixels[60+Trunc(300*(x–x0)/(xfin–x0)),360–Trunc(300*(y–min)/(max–min
                ))]:=clGreen;
            x:=x+STEP; y:=f(x);
          end;
        x:=x0; y:=0;
        while x<=xfin do
          begin
            Pixels[60+Trunc(300*(x–x0)/(xfin–x0)),360–Trunc(300*(y–min)/(max–min
                ))]:=clRed;
            x:=x+STEP;
          end;
        Brush.Style:=bsClear;
        Font.Name:='MS Sans Serif';
        Font.Size:=10;
        TextOut(385,370,'x');
        for i:=0 to 5 do
          TextOut(53+60*i,370,FloatToStrF(x0+i*(xfin–x0)/5,ffGeneral,4,2));
        TextOut(20,20,'y');
        for i:=0 to 5 do
          TextOut(20,352–60*i,FloatToStrF(min+i*(max–min)/5,ffGeneral,4,2));
        TextOut(45,352–Trunc(300*(–min)/(max–min)),'0');
      end;
   Figure1.SaveToFile('Out.bmp');
end;

end.
```

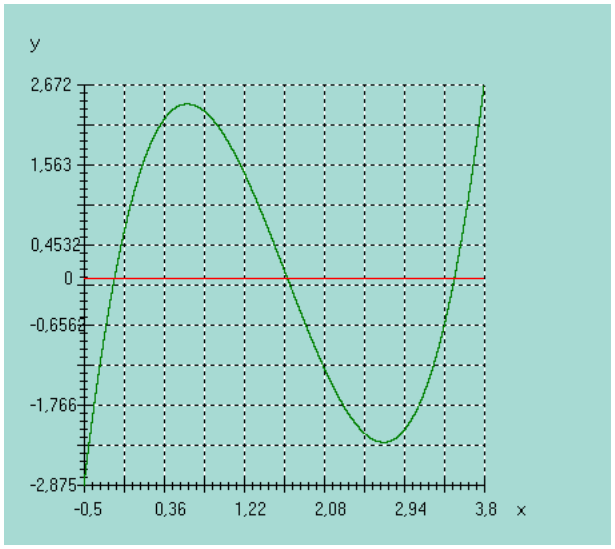

Figure 2: The file 'Out.bmp'. The graph of function $f(x) = x^3 - 5x^2 + 5x + 1$ at the segment $-0.5 \leqslant x \leqslant 3.8$.

Finally the program will create the file 'Out.bmp', which will contain the function

$$f(x) = x^3 - 5x^2 + 5x + 1,$$

plotted at the segment

$$-0.5 \leqslant x \leqslant 3.8.$$

From the obtained graph one can see, that the equation

$$x^3 - 5x^2 + 5x + 1 = 0$$

has three real roots at the segment

$$-0.5 \leqslant x \leqslant 3.8.$$

For the finding of roots we will use the method of tangents. In order to find the minimal root, choose the zeroth approximation as an arbitrary value of $x$ on the left of the maximum. For the finding of the maximal root act analogically: choose its zeroth approximation as an arbitrary value of $x$ on the right of the minimum. If we will choose the zeroth approximation in the interval between the extremums, then the method of tangents will bring us to one of three roots. For the finding of the middle root it is necessary to choose its zeroth approximation sufficiently nearly to it (one can choose the inflection point).

The next program realizes the method of tangents. In the text of program indicate the function

$$f(x) = x^3 - 5x^2 + 5x + 1.$$

```
program Project2;

{$APPTYPE CONSOLE}

uses
  SysUtils;

const
  APPROX = 1E–10;

var
  x:real;
  g:byte;
  h:char;
begin
  { TODO –oUser –cConsole Main : Insert code here }
  writeln('Method of Tangents.');
  g:=0;
  repeat
     write('x0='); readln(x);
     while abs(x*x*x–5*x*x+5*x+1)>=APPROX do
        x:=x–(x*x*x–5*x*x+5*x+1)/(3*x*x–10*x+5);
     write('x='); writeln(x);
     write('Once again? (y/n) '); readln(h);
     if h='n' then g:=1;
  until g=1;
end.
```

In the text of program there is a constant APPROX. When an absolute value of function in the next approximation of solution becomes less, than APPROX, the program finishes the cycle, realizing the algorithm of method of tangents. Decreasing a value of APPROX, we will increase an accuracy of calculation of the root.

Now let's consider another function. E.g.,

$$f\left(x\right) = \ln\left(x + 5.1\right) + \sin x.$$

Plot it at the segment

$$-5 \leqslant x \leqslant 5.$$

For that in the text of the first program instead of

```
  f:=x*x*x–5*x*x+5*x+1;
```

write

```
  f:=ln(x+5.1)+sin(x);
```

and instead of

```
  x0 = –0.5;
  xfin = 3.8;
```

write

```
  x0 = –5;
  xfin = 5;
```

Finally the obtained program plots this function (Figure 3). From the graph one can see, that the root of equation

$$\ln(x+5.1)+\sin x=0$$

is at the segment

$$-4.8 \leqslant x \leqslant -4.6.$$

In this case it is necessary to choose the zeroth approximation on the left of the root, in order that the method of tangents should work adequately. Make the corresponding changes in the text of the second program. Instead of

```
      while abs(x*x*x–5*x*x+5*x+1)>=APPROX do
         x:=x–(x*x*x–5*x*x+5*x+1)/(3*x*x–10*x+5);
```

write

```
      while abs(ln(x+5.1)+sin(x))>=APPROX do
         x:=x–(ln(x+5.1)+sin(x))/(1/(x+5.1)+cos(x));
```

The changed program will realize the method of tangents for the function

$$f\,(x)=\ln\,(x+5.1)+\sin x.$$

## 3 Enumerative Technique

The next program solves the transcendental equation

$$\arctan x-\sin xy-y=0$$

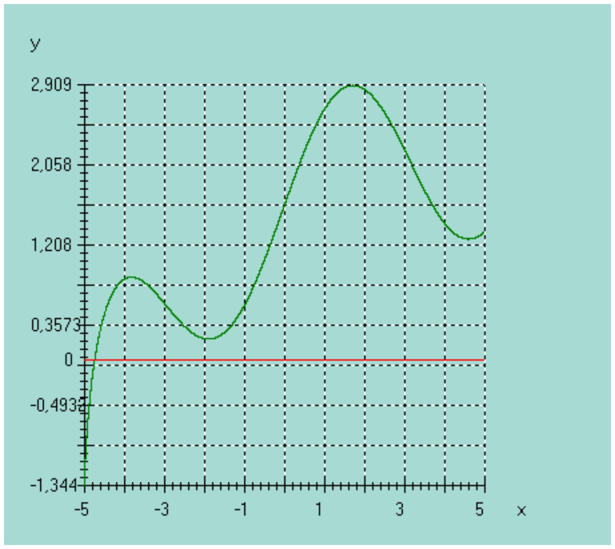

Figure 3: The graph of function $f(x) = \ln(x + 5.1) + \sin x$ at the segment $-5 \leqslant x \leqslant 5$.

at the segment

$$0 \leqslant x \leqslant 1.$$

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Go: TButton;
    procedure GoClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  Nx = 1E4;
  STEPy = 1E–4;
  x0 = 0;
  xfin = 1;

var
  Form1: TForm1;
  Figure1: TBitMap;

implementation

{$R *.dfm}

function f(x,y:real):real;
begin
  f:=arctan(x)–sin(x*y)–y;
end;

procedure TForm1.GoClick(Sender: TObject);
var
  i:integer;
  STEPx:real;
  x,y,min,max:real;
begin
  Figure1:=TBitMap.Create;
  Figure1.LoadFromFile('In.bmp');
  with Figure1.Canvas do
     begin
       MoveTo(60,360); LineTo(360,360);
       MoveTo(60,360); LineTo(60,60);
       for i:=0 to 10 do
          begin
            MoveTo(60+30*i,365); LineTo(60+30*i,355);
          end;
       for i:=0 to 50 do
          begin
            MoveTo(60+6*i,363); LineTo(60+6*i,357);
          end;
       for i:=0 to 10 do
          begin
            MoveTo(55,360–30*i); LineTo(65,360–30*i);
```

```
    end;
for i:=0 to 50 do
    begin
       MoveTo(57,360–6*i); LineTo(63,360–6*i);
    end;
Pen.Style:=psDot;
for i:=1 to 10 do
    begin
       MoveTo(60+30*i,355); LineTo(60+30*i,60);
    end;
for i:=1 to 10 do
    begin
       MoveTo(65,360–30*i); LineTo(360,360–30*i);
    end;
Pen.Style:=psSolid;
STEPx:=(xfin–x0)/Nx;
x:=x0;
min:=0;
max:=0;
while x<=xfin do
    begin
       y:=0;
       while f(x,y)>0 do y:=y+STEPy;
       if y<min then min:=y;
       if y>max then max:=y;
       x:=x+STEPx;
    end;
x:=x0;
while x<=xfin do
    begin
       y:=0;
       while f(x,y)>0 do y:=y+STEPy;
       Pixels[60+Trunc(300*(x–x0)/(xfin–x0)),360–Trunc(300*(y–min)/(max–min
           ))]:=clRed;
       x:=x+STEPx;
    end;
Brush.Style:=bsClear;
Font.Name:='MS Sans Serif';
```

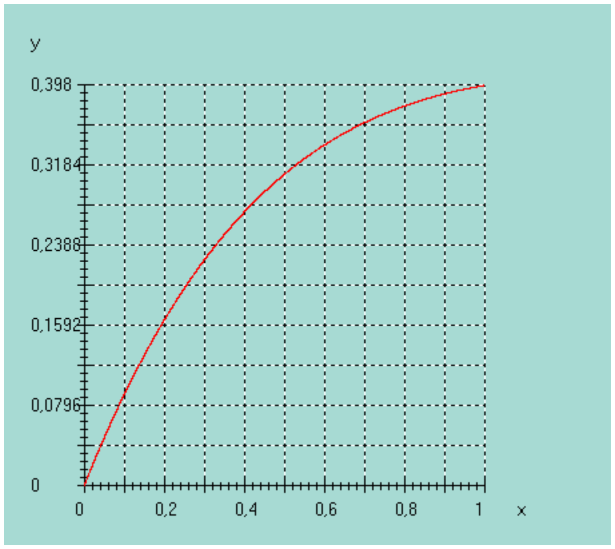

Figure 4: The solution of the transcendental equation $\arctan x - \sin xy - y = 0$ at the segment $0 \leqslant x \leqslant 1$.

```
         Font.Size:=10;
         TextOut(385,370,'x');
         for i:=0 to 5 do
             TextOut(53+60*i,370,FloatToStrF(x0+i*(xfin–x0)/5,ffGeneral,4,2));
         TextOut(20,20,'y');
         for i:=0 to 5 do
             TextOut(20,352–60*i,FloatToStrF(min+i*(max–min)/5,ffGeneral,4,2));
      end;
   Figure1.SaveToFile('Out.bmp');
end;

end.
```

## 4 System of Differential Equations

The next program solves the system of differential equations:

$$y''(x) + p_1(y)\, z(x)\, y'(x) - q_1(y) = 0,\ p_1(y) = \frac{2}{y},\ q_1(y) = 2y + \frac{1}{y},$$

$$z^3(x) = p_2(y)\, y'(x) + q_2(y),\ p_2(y) = \frac{1}{8}\left(y^2 + 3\right),\ q_2(y) = \frac{1}{8}\left(\frac{3}{y} + \frac{1}{y^3}\right);$$

which has the analytical solution

$$y\left(x\right)=e^{x},\ z\left(x\right)=\cosh x,\ z\left(y\right)=\frac{1}{2}\left(y+\frac{1}{y}\right).$$

The program solves the system at the segment

$$0\leqslant x\leqslant 3.$$

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Go: TButton;
    procedure GoClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  STEPx = 1E–4;
  STEPy = 1E–4;
  x0 = 0;
  xfin = 3;

var
  Form1: TForm1;
  Figure1: TBitMap;

implementation

{$R *.dfm}
```

```
function f(y:real):real;
begin
  f:=(1/2)*(y+1/y);
end;

function p1(y:real):real;
begin
  p1:=2/y;
end;

function q1(y:real):real;
begin
  q1:=2*y+1/y;
end;

function p2(y:real):real;
begin
  p2:=(1/8)*(y*y+3);
end;

function q2(y:real):real;
begin
  q2:=(1/8)*(3/y+1/(y*y*y));
end;

procedure TForm1.GoClick(Sender: TObject);
var
  i:integer;
  y0,Dy0:real;
  x,y,Dy,z,min,max,D2y,yfin:real;
begin
  Figure1:=TBitMap.Create;
  Figure1.LoadFromFile('In.bmp');
  with Figure1.Canvas do
     begin
       MoveTo(60,360); LineTo(360,360);
       MoveTo(60,360); LineTo(60,60);
       for i:=0 to 10 do
          begin
            MoveTo(60+30*i,365); LineTo(60+30*i,355);
```

```
    end;
for i:=0 to 50 do
    begin
       MoveTo(60+6*i,363); LineTo(60+6*i,357);
    end;
for i:=0 to 10 do
    begin
       MoveTo(55,360–30*i); LineTo(65,360–30*i);
    end;
for i:=0 to 50 do
    begin
       MoveTo(57,360–6*i); LineTo(63,360–6*i);
    end;
Pen.Style:=psDot;
for i:=1 to 10 do
    begin
       MoveTo(60+30*i,355); LineTo(60+30*i,60);
    end;
for i:=1 to 10 do
    begin
       MoveTo(65,360–30*i); LineTo(360,360–30*i);
    end;
Pen.Style:=psSolid;
y0:=exp(x0);
Dy0:=exp(x0);
x:=x0;
y:=y0;
Dy:=Dy0;
z:=exp((1/3)*ln(p2(y)*Dy+q2(y)));
min:=z;
max:=z;
D2y:=q1(y)–p1(y)*z*Dy;
repeat
    x:=x+STEPx;
    y:=y+Dy*STEPx;
    Dy:=Dy+D2y*STEPx;
    z:=exp((1/3)*ln(p2(y)*Dy+q2(y)));
    if z<min then min:=z;
```

```
            if z>max then max:=z;
            D2y:=q1(y)–p1(y)*z*Dy;
        until x>=xfin;
        yfin:=y;
        Brush.Style:=bsClear;
        Font.Name:='MS Sans Serif';
        Font.Size:=10;
        TextOut(400,370,'y');
        for i:=0 to 5 do TextOut(53+60*i,370,FloatToStrF(y0+i*(yfin–y0)/5,
            ffGeneral,4,2));
        TextOut(20,20,'z');
        for i:=0 to 5 do TextOut(20,352–60*i,FloatToStrF(min+i*(max–min)/5,
            ffGeneral,4,2));
        x:=x0;
        y:=y0;
        Dy:=Dy0;
        z:=exp((1/3)*ln(p2(y)*Dy+q2(y)));
        D2y:=q1(y)–p1(y)*z*Dy;
        Pixels[60,360–Trunc(300*(z–min)/(max–min))]:=clRed;
        repeat
            x:=x+STEPx;
            y:=y+Dy*STEPx;
            Dy:=Dy+D2y*STEPx;
            z:=exp((1/3)*ln(p2(y)*Dy+q2(y)));
            D2y:=q1(y)–p1(y)*z*Dy;
            Pixels[60+Trunc(300*(y–y0)/(yfin–y0)),360–Trunc(300*(z–min)/(max–min))
                ]:=clRed;
        until x>=xfin;
    end;
Figure1.SaveToFile('Num.bmp');
with Figure1.Canvas do
    begin
        y:=y0;
        z:=f(y);
        Pixels[60,360–Trunc(300*(z–min)/(max–min))]:=clPurple;
        repeat
            y:=y+STEPy;
            z:=f(y);
```

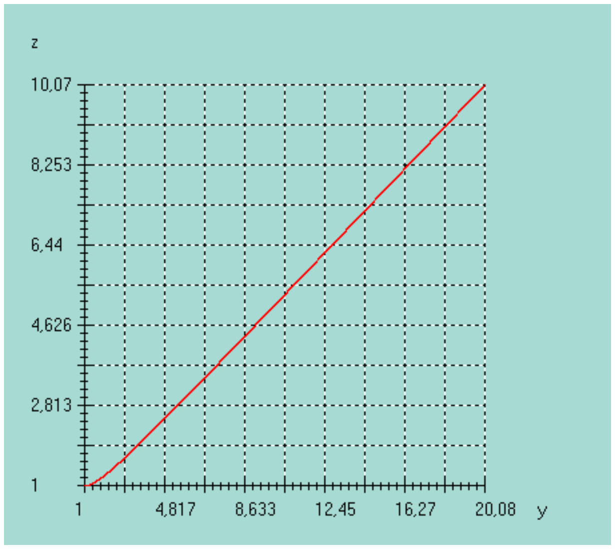

Figure 5: The numerical solution of system of differential equations at the segment $0 \leqslant x \leqslant 3$.

```
          Pixels[60+Trunc(300*(y–y0)/(yfin–y0)),360–Trunc(300*(z–min)/(max–min))
             ]:=clPurple;
       until y>=yfin;
     end;
  Figure1.SaveToFile('An.bmp');
end;

end.
```

The red line on the graph is the numerical solution of system of differential equations. The purple line – the analytical solution. The analytical solution is drawn on the graph with the numerical one. As one can see, the analytical solution covers the numerical one.

Now let's obtain the numerical and analytical solutions of the same system at the segment $-1 \leqslant x \leqslant 1$. For that in the text of program instead of

```
  x0 = 0;
  xfin = 3;
```

write

```
  x0 = –1;
  xfin = 1;
```

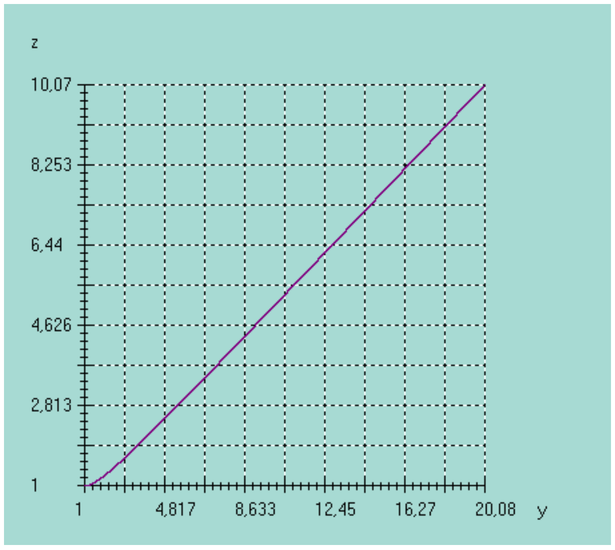

Figure 6: The analytical solution of system of differential equations at the segment $0 \leqslant x \leqslant 3$.

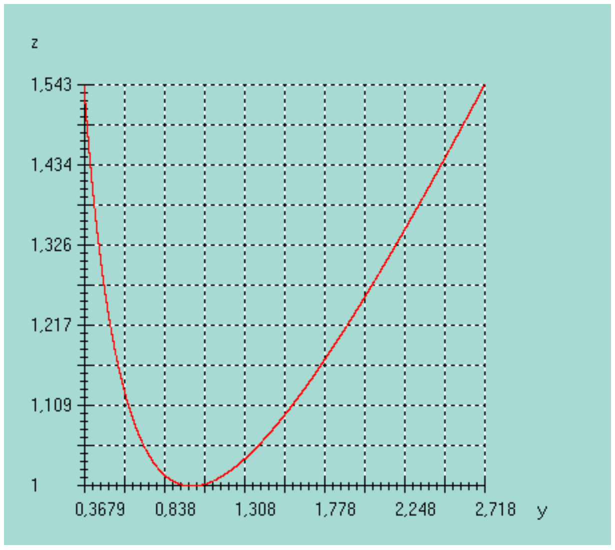

Figure 7: The numerical solution of system of differential equations at the segment $-1 \leqslant x \leqslant 1$.

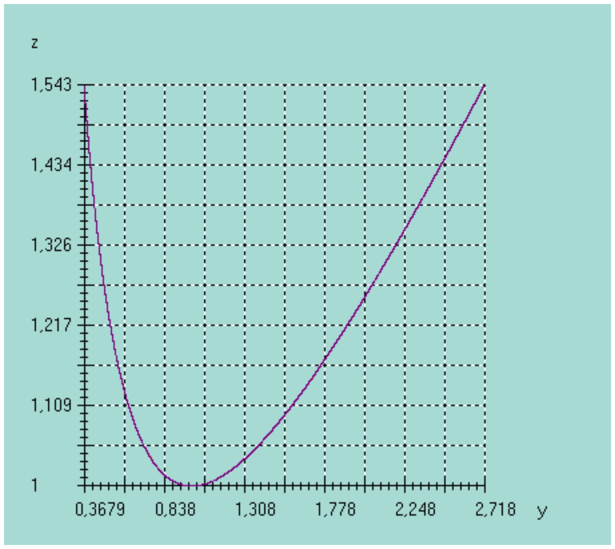

Figure 8: The analytical solution of system of differential equations at the segment $-1 \leqslant x \leqslant 1$.

## 5 Differential Equation

The next program solves the differential equation

$$y^{(3)}(x) + (\sin x)\, y''(x) + (\arctan x)\, y'(x) - (\cosh x)\, y(x) + 5 \sinh x = 0$$

with the initial conditions:

$$y(3) = 5,$$
$$y'(3) = -1,$$
$$y''(3) = 2.$$

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Go: TButton;
    procedure GoClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  STEP = 1E–4;
  lambda = 5;
  x0 = 3;
  y0 = 5;
  Dy0 = –1;
  D2y0 = 2;
  xfin = 4.5;

var
  Form1: TForm1;
  Figure1: TBitMap;

implementation

{$R *.dfm}

function cosh(x:real):real;
begin
  cosh:=(exp(x)+exp(–x))/2;
end;

function sinh(x:real):real;
begin
  sinh:=(exp(x)–exp(–x))/2;
end;

procedure TForm1.GoClick(Sender: TObject);
var
  i:integer;
  x,y,min,max,Dy,D2y,D3y:real;
begin
  Figure1:=TBitMap.Create;
  Figure1.LoadFromFile('In.bmp');
```

```
with Figure1.Canvas do
   begin
     MoveTo(60,360); LineTo(360,360);
     MoveTo(60,360); LineTo(60,60);
     for i:=0 to 10 do
        begin
          MoveTo(60+30*i,365); LineTo(60+30*i,355);
        end;
     for i:=0 to 50 do
        begin
          MoveTo(60+6*i,363); LineTo(60+6*i,357);
        end;
     for i:=0 to 10 do
        begin
          MoveTo(55,360–30*i); LineTo(65,360–30*i);
        end;
     for i:=0 to 50 do
        begin
          MoveTo(57,360–6*i); LineTo(63,360–6*i);
        end;
     Pen.Style:=psDot;
     for i:=1 to 10 do
        begin
          MoveTo(60+30*i,355); LineTo(60+30*i,60);
        end;
     for i:=1 to 10 do
        begin
          MoveTo(65,360–30*i); LineTo(360,360–30*i);
        end;
     Pen.Style:=psSolid;
     x:=x0;
     y:=y0;
     min:=y;
     max:=y;
     Dy:=Dy0;
     D2y:=D2y0;
     while x<xfin do
        begin
```

```
          D3y:=–sin(x)*D2y–arctan(x)*Dy+cosh(x)*y–lambda*sinh(x);
          x:=x+STEP;
          y:=y+Dy*STEP;
          if y<min then min:=y;
          if y>max then max:=y;
          Dy:=Dy+D2y*STEP;
          D2y:=D2y+D3y*STEP;
        end;
      Brush.Style:=bsClear;
      Font.Name:='MS Sans Serif';
      Font.Size:=10;
      TextOut(385,370,'x');
      for i:=0 to 5 do
        TextOut(53+60*i,370,FloatToStrF(x0+i*(xfin–x0)/5,ffGeneral,4,2));
      TextOut(20,20,'y');
      for i:=0 to 5 do
        TextOut(20,352–60*i,FloatToStrF(min+i*(max–min)/5,ffGeneral,4,2));
      x:=x0;
      y:=y0;
      Dy:=Dy0;
      D2y:=D2y0;
      Pixels[60,360–Trunc(300*(y0–min)/(max–min))]:=clRed;
      while x<xfin do
        begin
          D3y:=–sin(x)*D2y–arctan(x)*Dy+cosh(x)*y–lambda*sinh(x);
          x:=x+STEP;
          y:=y+Dy*STEP;
          Dy:=Dy+D2y*STEP;
          D2y:=D2y+D3y*STEP;
          Pixels[60+Trunc(300*(x–x0)/(xfin–x0)),360–Trunc(300*(y–min)/(max–min
              ))]:=clRed;
        end;
    end;
  Figure1.SaveToFile('Out.bmp');
end;

end.
```

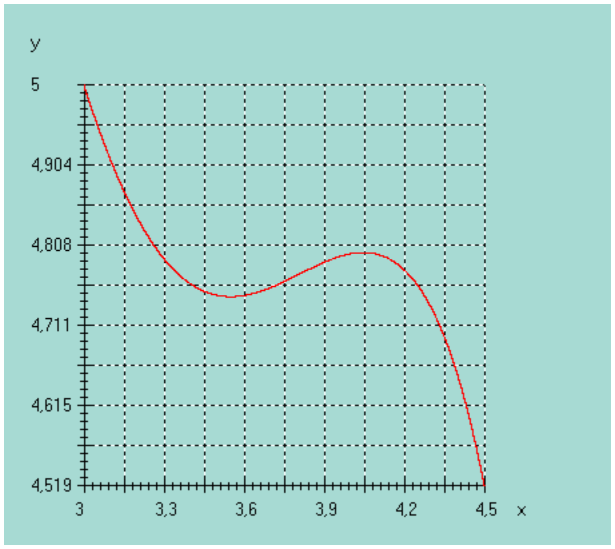

Figure 9: The solution of differential equation.

## 6 Definite Integral

The integral

$$\int_0^{\pi} \frac{\cos^4 \varphi}{1+\sin^2 \varphi} d\varphi = 2\pi \left(\sqrt{2} - \frac{5}{4}\right)$$

is calculated analytically by the use of complex analysis [3]. The next program calculates the integral

$$\int_{\mathrm{x0}}^{\mathrm{xfin}} \frac{\cos^4 \varphi}{1+\sin^2 \varphi} d\varphi.$$

```
program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;

function f(x:real):real;
begin
  f:=cos(x)*cos(x)*cos(x)*cos(x)/(1+sin(x)*sin(x));
end;
```

```
var
  STEP:real;
  x0,xfin,x,S:real;
begin
  { TODO -oUser -cConsole Main : Insert code here }
  writeln('Definite Integral.');
  write('x0='); readln(x0);
  write('xfin='); readln(xfin);
  write('A step = '); readln(STEP);
  x:=x0; S:=0;
  repeat
     S:=S+f(x);
     x:=x+STEP;
  until (x>=xfin);
  S:=STEP*S;
  write('The integral = '); write(S); readln;
end.
```

Choosing the limits of integration

$$\mathrm{x0} = 0,$$
$$\mathrm{xfin} = 3.14159265359$$

and the step

$$\mathrm{STEP} = 1\mathrm{E} - 7,$$

we obtain the value of integral

$$I = 1.031784.$$

If it is necessary to calculate another definite integral, then instead of

```
f:=cos(x)*cos(x)*cos(x)*cos(x)/(1+sin(x)*sin(x));
```

it is necessary to write a corresponding function.

## 7 Contour Integral

The next program calculates the integral

$$\oint\limits_{C\left(\sqrt{\frac{1}{e^2}-1}-\frac{1}{e},0\right)} \left(v\left(x,y\right)dx + u\left(x,y\right)dy\right) = \frac{\pi e^2}{2\left(1-e^2\right)^{3/2}} \ \left(0<e<1\right),$$

where

$$u(x,y) = \frac{xA(x,y) + yB(x,y)}{A^2(x,y) + B^2(x,y)},$$
$$v(x,y) = \frac{yA(x,y) - xB(x,y)}{A^2(x,y) + B^2(x,y)},$$
$$A(x,y) = x^4 - 6x^2y^2 + y^4 + \frac{4}{e}x^3 - \frac{12}{e}xy^2 + 2\left(\frac{2}{e^2}+1\right)\left(x^2-y^2\right) + \frac{4}{e}x + 1,$$
$$B(x,y) = 4\left(x+\frac{1}{e}\right)y\left(x^2-y^2+\frac{2}{e}x+1\right).$$

This integral is calculated analytically by the use of complex analysis [3]:

$$\oint\limits_{C_{\left(\sqrt{\frac{1}{e^2}-1}-\frac{1}{e},0\right)}} \left(v(x,y)\,dx + u(x,y)\,dy\right) =$$
$$= \mathrm{Re}\left(\frac{1}{i}\oint\limits_{|z-\sqrt{\frac{1}{e^2}-1}+\frac{1}{e}|=\varepsilon,\ 0<\varepsilon<2\sqrt{\frac{1}{e^2}-1}} \frac{zdz}{\left(z^2+\frac{2}{e}z+1\right)^2}\right)\ (0<e<1),$$
$$\frac{1}{i}\oint\limits_{|z-\sqrt{\frac{1}{e^2}-1}+\frac{1}{e}|=\varepsilon,\ 0<\varepsilon<2\sqrt{\frac{1}{e^2}-1}} \frac{zdz}{\left(z^2+\frac{2}{e}z+1\right)^2} = \frac{\pi e^2}{2\left(1-e^2\right)^{3/2}}\ (0<e<1).$$

```
program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;

function A(x,y,ec:real):real;
begin
  A:=x*x*x*x–6*x*x*y*y+y*y*y*y+(4/ec)*x*(x*x–3*y*y)+2*(2/(ec*ec)+1)*(x*x–y*y)+(4
     /ec)*x+1;
end;

function B(x,y,ec:real):real;
begin
  B:=4*(x+1/ec)*y*(x*x–y*y+(2/ec)*x+1);
```

```
end;

function u(x,y,ec:real):real;
begin
  u:=(x*A(x,y,ec)+y*B(x,y,ec))/(A(x,y,ec)*A(x,y,ec)+B(x,y,ec)*B(x,y,ec));
end;

function v(x,y,ec:real):real;
begin
  v:=(y*A(x,y,ec)–x*B(x,y,ec))/(A(x,y,ec)*A(x,y,ec)+B(x,y,ec)*B(x,y,ec));
end;

label
  bye;
var
  N:real;
  ec,eps,phi,S:real;
begin
  { TODO –oUser –cConsole Main : Insert code here }
  writeln('Contour Integral.');
  write('ec='); readln(ec);
  write('A quantity of partitions = '); readln(N);
  if (ec>=1) or (ec<=0) then
      begin
        write('ec must be 0<ec<1.');
        goto bye;
      end;
  eps:=sqrt(1/(ec*ec)–1);
  phi:=0; S:=0;
  repeat
      S:=S+u(sqrt(1/(ec*ec)–1)–1/ec+eps*cos(phi),eps*sin(phi),ec)*cos(phi)–v(sqrt
          (1/(ec*ec)–1)–1/ec+eps*cos(phi),eps*sin(phi),ec)*sin(phi);
      phi:=phi+2*PI/N;
  until (phi>=2*PI);
  S:=eps*(2*PI/N)*S;
  write('The integral = '); write(S);
  bye: readln;
end.
```

Choosing the value

$$e = 0.5$$

and the quantity of partitions of integration contour

$$\mathrm{N} = 1\mathrm{E}7,$$

we obtain the value of integral

$$I = 0.6045998.$$

## 8 Problems with Data on Characteristics

The next program solves the problem with data on characteristics [4]

$$u_{xy}(x, y) = \sin x + \cos y$$

in the rectangle

$$\begin{aligned} 1 \leqslant x \leqslant 3, \\ 2 \leqslant y \leqslant 4 \end{aligned}$$

with the boundary conditions

$$\begin{aligned} &u(x, 2) = \cosh x,\ 1 \leqslant x \leqslant 3, \\ &u(1, y) = \cosh(y - 1) + y \sinh(y - 2),\ 2 \leqslant y \leqslant 4. \end{aligned}$$

Finally we obtain the graphs of functions

$$\begin{aligned} &u(x, 4),\ 1 \leqslant x \leqslant 3, \\ &u(3, y),\ 2 \leqslant y \leqslant 4. \end{aligned}$$

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Go: TButton;
    procedure GoClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
  Nx = 10000;
  Ny = 10000;
  x0 = 1;
  xfin = 3;
  y0 = 2;
  yfin = 4;

var
  Form1: TForm1;
  Figure1: TBitMap;
  uu: array[0..10000,0..10000] of real;
  hh: array[0..10000,0..10000] of real;

implementation

{$R *.dfm}

function cosh(z:real):real;
begin
  cosh:=(exp(z)+exp(–z))/2;
end;

function sinh(z:real):real;
begin
  sinh:=(exp(z)–exp(–z))/2;
end;

function h(x,y:real):real;
begin
  h:=sin(x)+cos(y);
end;

function f(x:real):real;
```

```
begin
  f:=cosh(x);
end;

function g(y:real):real;
begin
  g:=cosh(y–1)+y*sinh(y–2);
end;

procedure TForm1.GoClick(Sender: TObject);
var
  i,j:integer;
  min,max:real;
begin
  for j:=0 to Ny–1 do
      for i:=0 to Nx–1 do
          hh[i,j]:=h(x0+i*(xfin–x0)/Nx,y0+j*(yfin–y0)/Ny);
  for i:=0 to Nx do
      uu[i,0]:=f(x0+i*(xfin–x0)/Nx);
  for i:=1 to Ny do
      uu[0,i]:=g(y0+i*(yfin–y0)/Ny);
  for j:=0 to Ny–1 do
      for i:=0 to Nx–1 do
          uu[i+1,j+1]:=hh[i,j]*(xfin–x0)*(yfin–y0)/(Nx*Ny)+uu[i,j+1]+uu[i+1,j]–uu[
              i,j];
  Figure1:=TBitMap.Create;
  Figure1.LoadFromFile('In.bmp');
  with Figure1.Canvas do
      begin
        MoveTo(60,360); LineTo(360,360);
        MoveTo(60,360); LineTo(60,60);
        for i:=0 to 10 do
            begin
              MoveTo(60+30*i,365); LineTo(60+30*i,355);
            end;
        for i:=0 to 50 do
            begin
              MoveTo(60+6*i,363); LineTo(60+6*i,357);
            end;
        for i:=0 to 10 do
            begin
              MoveTo(55,360–30*i); LineTo(65,360–30*i);
            end;
        for i:=0 to 50 do
            begin
              MoveTo(57,360–6*i); LineTo(63,360–6*i);
            end;
        Pen.Style:=psDot;
        for i:=1 to 10 do
            begin
              MoveTo(60+30*i,355); LineTo(60+30*i,60);
            end;
        for i:=1 to 10 do
            begin
              MoveTo(65,360–30*i); LineTo(360,360–30*i);
            end;
        Pen.Style:=psSolid;
        min:=uu[0,Ny];
        max:=uu[0,Ny];
        for i:=1 to Nx do
            begin
              if uu[i,Ny]<min then min:=uu[i,Ny];
              if uu[i,Ny]>max then max:=uu[i,Ny];
            end;
        Brush.Style:=bsClear;
        Font.Name:='MS Sans Serif';
        Font.Size:=10;
        TextOut(385,370,'x');
        for i:=0 to 5 do
            TextOut(53+60*i,370,FloatToStrF(x0+i*(xfin–x0)/5,ffGeneral,4,2));
        TextOut(20,20,'u(x,'+FloatToStrF(yfin,ffGeneral,4,2)+')');
        for i:=0 to 5 do
            TextOut(20,352–60*i,FloatToStrF(min+i*(max–min)/5,ffGeneral,4,2));
        for i:=0 to Nx do
            Pixels[60+Trunc(300*i/Nx),360–Trunc(300*(uu[i,Ny]–min)/(max–min))]:=
                clRed;
      end;
```

```
Figure1.SaveToFile('Outx.bmp');
Figure1.LoadFromFile('In.bmp');
with Figure1.Canvas do
   begin
     MoveTo(60,360); LineTo(360,360);
     MoveTo(60,360); LineTo(60,60);
     for i:=0 to 10 do
         begin
           MoveTo(60+30*i,365); LineTo(60+30*i,355);
         end;
     for i:=0 to 50 do
         begin
           MoveTo(60+6*i,363); LineTo(60+6*i,357);
         end;
     for i:=0 to 10 do
         begin
           MoveTo(55,360–30*i); LineTo(65,360–30*i);
         end;
     for i:=0 to 50 do
         begin
           MoveTo(57,360–6*i); LineTo(63,360–6*i);
         end;
     Pen.Style:=psDot;
     for i:=1 to 10 do
         begin
           MoveTo(60+30*i,355); LineTo(60+30*i,60);
         end;
     for i:=1 to 10 do
         begin
           MoveTo(65,360–30*i); LineTo(360,360–30*i);
         end;
     Pen.Style:=psSolid;
     min:=uu[Nx,0];
     max:=uu[Nx,0];
     for i:=1 to Ny do
         begin
           if uu[Nx,i]<min then min:=uu[Nx,i];
           if uu[Nx,i]>max then max:=uu[Nx,i];
```

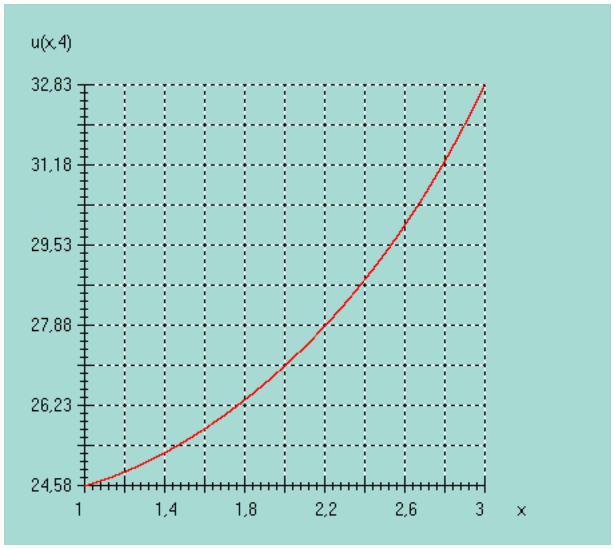

Figure 10: The graph of function $u\,(x,4)$ at the segment $1 \leqslant x \leqslant 3$.

```
            end;
          Brush.Style:=bsClear;
          Font.Name:='MS Sans Serif';
          Font.Size:=10;
          TextOut(385,370,'y');
          for i:=0 to 5 do
            TextOut(53+60*i,370,FloatToStrF(y0+i*(yfin–y0)/5,ffGeneral,4,2));
          TextOut(20,20,'u('+FloatToStrF(xfin,ffGeneral,4,2)+',y)');
          for i:=0 to 5 do
            TextOut(20,352–60*i,FloatToStrF(min+i*(max–min)/5,ffGeneral,4,2));
          for i:=0 to Ny do
            Pixels[60+Trunc(300*i/Ny),360–Trunc(300*(uu[Nx,i]–min)/(max–min))]:=
              clRed;
        end;
  Figure1.SaveToFile('Outy.bmp');
end;

end.
```

Now let's consider the problem with data on characteristics

$$u_{xy}\,(x,y) = \cos xy - xy \sin xy$$

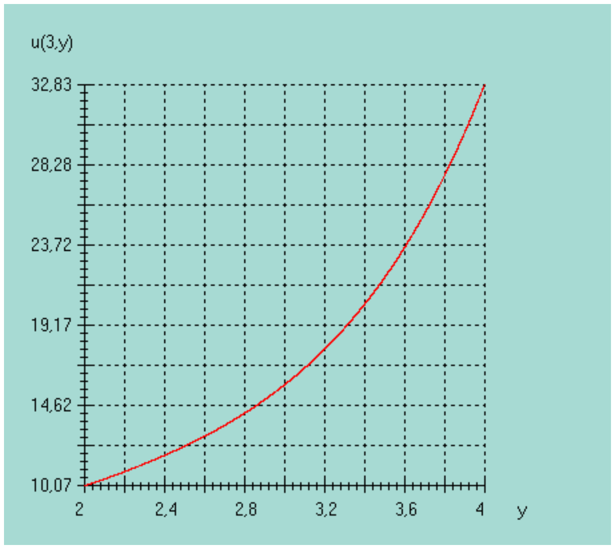

Figure 11: The graph of function $u\,(3,y)$ at the segment $2 \leqslant y \leqslant 4$.

in the rectangle

$$-\pi \leqslant x \leqslant \pi,$$
$$-1 \leqslant y \leqslant 1$$

with the boundary conditions

$$u\,(x,-1) = -\sin x, \quad -\pi \leqslant x \leqslant \pi,$$
$$u\,(-\pi,y) = -\sin \pi y, \quad -1 \leqslant y \leqslant 1.$$

This problem has the analytical solution

$$u\,(x,y) = \sin xy, \quad -\pi \leqslant x \leqslant \pi, \quad -1 \leqslant y \leqslant 1.$$

Make the corresponding changes in the text of initial program. Instead of

```
x0 = 1;
xfin = 3;
y0 = 2;
yfin = 4;
```

write

```
x0 = –PI;
xfin = PI;
y0 = –1;
```

```
   yfin = 1;
```

and instead of

```
function cosh(z:real):real;
begin
   cosh:=(exp(z)+exp(–z))/2;
end;

function sinh(z:real):real;
begin
   sinh:=(exp(z)–exp(–z))/2;
end;

function h(x,y:real):real;
begin
   h:=sin(x)+cos(y);
end;

function f(x:real):real;
begin
   f:=cosh(x);
end;

function g(y:real):real;
begin
   g:=cosh(y–1)+y*sinh(y–2);
end;
```

write

```
function h(x,y:real):real;
begin
   h:=cos(x*y)–x*y*sin(x*y);
end;

function f(x:real):real;
begin
   f:=–sin(x);
```

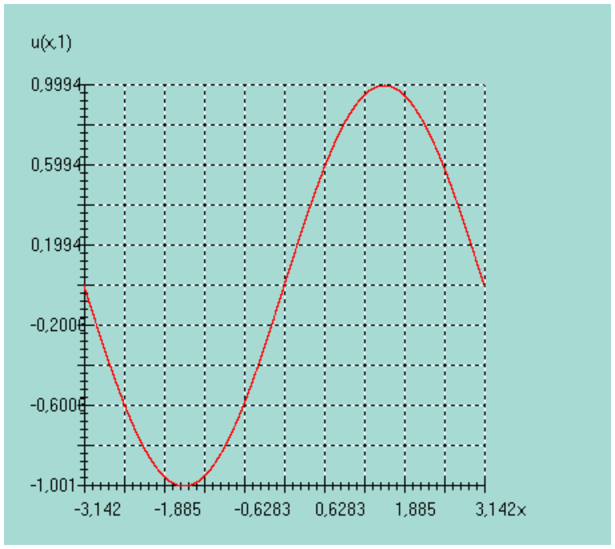

Figure 12: The graph of function $u\,(x,1)$ at the segment $-\pi \leqslant x \leqslant \pi$.

**end**;

**function** g(y:real):real;
**begin**
  g:=–sin(PI*y);
**end**;

Finally the changed program plots the functions

$$\begin{aligned} &u\,(x,1)\,, \quad -\pi \leqslant x \leqslant \pi,\\ &u\,(\pi,y)\,, \quad -1 \leqslant y \leqslant 1. \end{aligned}$$

## References

[1] Nikita Kul'tin. Foundations of Programming in Delphi 7 (in Russian). Saint-Petersburg, BKhV-Petersburg, 2006.

[2] G.A. Korn, T.M. Korn. Mathematical Handbook for Scientists and Engineers. Definitions, Theorems and Formulas for Reference and Review. McGraw-Hill Book Company; New York, Toronto, London; 1961.

[3] A.G. Sveshnikov, A.N. Tikhonov. Theory of Functions of Complex Variable (in Russian). Moscow, Fizmatlit, 2010.

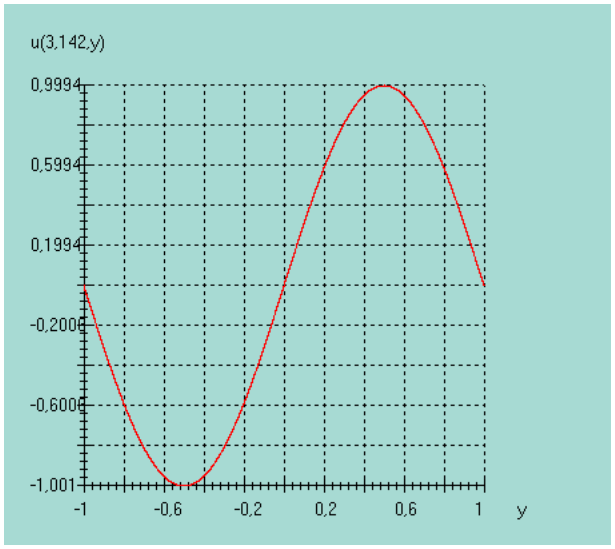

Figure 13: The graph of function $u\,(\pi, y)$ at the segment $-1 \leqslant y \leqslant 1$.

[4] A.N. Tikhonov, A.A. Samarskiĭ. Equations of Mathematical Physics (in Russian). Moscow, Nauka, 2004.