Требуется найти приближенное решение заданных дифференциальных уравнений

Использовать численный метод решения: Эйлера исправленный.

Выполнил: Хафизов А.И.

# Листинг программы

**Файл Unit1 .h:**

/\*на тему: "Маятник на тележке"

Задание: решить систему ОДУ:

d^2(x1)/dt^2=-(1/D)\*((m\*l\*l+Ip)\*temp1+m\*l\*cos(x[2])\*temp2)

d^2(x3)/dt^2=-(1/D)\*((m\*l\*cos(x[2]))\*temp1+(M+m)\*temp2)

где temp1=(B/Rm)\*V-B\*B/Rm\*x[1]-m\*l\*sin(x[2])\*x[3]\*x[3]-Bx\*x[1]-w1

temp2=m\*g\*l\*sin(x[2])-Bf\*x[3]-w2

t [a,b], x=(x1,x2,x3,x4)^T x

Пусть задано начальное условие: x1(0)=0, x2(0)=0, x3(0)=1.127, x4(0)=0

k1=0, k2=0, k3=0, k4=0

Пределы изменения аргумента: a=0, b=15.

Требуется найти приближенное решение заданных дифференциальных уравнений

Использовать численный метод решения: Эйлера исправленный.

Выполнил: Хафизов А.И., студент гр.3339

\*/

//---------------------------------------------------------------------------

#ifndef Unit1H

#define Unit1H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <Chart.hpp>

#include <ExtCtrls.hpp>

#include <Grids.hpp>

#include <TeEngine.hpp>

#include <TeeProcs.hpp>

#include <Series.hpp>

//---------------------------------------------------------------------------

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TEdit \*Edit1;

TEdit \*Edit2;

TEdit \*Edit3;

TEdit \*Edit4;

TEdit \*Edit5;

TEdit \*Edit6;

TLabel \*Label1;

TLabel \*Label2;

TLabel \*Label3;

TLabel \*Label4;

TLabel \*Label5;

TLabel \*Label6;

TLabel \*Label7;

TLabel \*Label8;

TButton \*Button1;

TButton \*Button2;

TStringGrid \*StringGrid1;

TChart \*Chart1;

TLineSeries \*Series1;

TLineSeries \*Series2;

TLabel \*Label9;

TLabel \*Label10;

TEdit \*Edit7;

TEdit \*Edit8;

TLabel \*Label11;

TLabel \*Label12;

TEdit \*Edit9;

TEdit \*Edit10;

TLabel \*Label13;

TLabel \*Label14;

TEdit \*Edit11;

TEdit \*Edit12;

TLabel \*Label15;

TLabel \*Label16;

TEdit \*Edit13;

TEdit \*Edit14;

TLabel \*Label17;

TLabel \*Label18;

TEdit \*Edit15;

TEdit \*Edit16;

TLabel \*Label19;

TEdit \*Edit17;

TLabel \*Label20;

TLabel \*Label21;

TEdit \*Edit18;

TEdit \*Edit19;

TLineSeries \*Series3;

TLineSeries \*Series4;

TLabel \*Label22;

TLabel \*Label23;

TLabel \*Label24;

TLabel \*Label25;

TEdit \*Edit20;

TEdit \*Edit21;

TEdit \*Edit22;

TEdit \*Edit23;

TLabel \*Label26;

TLabel \*Label27;

TLabel \*Label28;

TEdit \*Edit24;

TEdit \*Edit25;

TEdit \*Edit26;

void \_\_fastcall Button1Click(TObject \*Sender);

void \_\_fastcall Button2Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm1(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm1 \*Form1;

//---------------------------------------------------------------------------

#endif

**Файл Unit1.cpp:**

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

double x[N], z[N], k1[N];

double a,b,h,t,tp,tmax,hp;

double M,m,L,l,Ip,g,Ri,Lm,Km,r,Kg, ampl, Bx, Bf;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

//Обработчик событий кнопки "Таблица значений"

{

/\* Таблица значений \*/

StringGrid1->Visible = true;

StringGrid1->Cells[0][0] = "t";

StringGrid1->Cells[1][0] = "X1(t)";

StringGrid1->Cells[2][0] = "X2(t)";

StringGrid1->Cells[3][0] = "X3(t)";

StringGrid1->Cells[4][0] = "X4(t)";

/\* Начальное и конечное время \*/

a = StrToFloat(Edit3->Text);

b = StrToFloat(Edit4->Text);

/\* Шаг интегрирования и шаг печати \*/

h = StrToFloat(Edit5->Text);

hp = StrToFloat(Edit6->Text);

/\* Начальные значения \*/

M = StrToFloat(Edit7->Text);

m = StrToFloat(Edit8->Text);

L = StrToFloat(Edit9->Text);

l = StrToFloat(Edit10->Text);

Ip = StrToFloat(Edit11->Text);

g = StrToFloat(Edit12->Text);

Ri = StrToFloat(Edit13->Text);

Lm = StrToFloat(Edit14->Text);

Km = StrToFloat(Edit15->Text);

r = StrToFloat(Edit16->Text);

Kg = StrToFloat(Edit17->Text);

x[0] = StrToFloat(Edit1->Text);

x[1] = StrToFloat(Edit2->Text);

x[2] = StrToFloat(Edit18->Text);

x[3] = StrToFloat(Edit19->Text);

k1[0] = StrToFloat(Edit20->Text);

k1[1] = StrToFloat(Edit21->Text);

k1[2] = StrToFloat(Edit22->Text);

k1[3] = StrToFloat(Edit23->Text);

ampl = StrToFloat(Edit24->Text);

Bx = StrToFloat(Edit25->Text);

Bf = StrToFloat(Edit26->Text);

//---------------------------------------------------------------------------

f(N, a, x, z, M,m,L,l,Ip,g,Ri,Lm,Km,r,Kg,k1,ampl,Bx,Bf);

StringGrid1->Cells[0][1] = FloatToStr(a);

int k=1;

for(int i=0; i<N; i++)

StringGrid1->Cells[i+1][k] = FloatToStr(x[i]);

t=a;

tp = t+hp;

tmax = b+(h/2);

while(tp<=tmax) {

k=k+1;

Eyler(t, tp, h, N, x,M,m,L,l,Ip,g,Ri,Lm,Km,r,Kg,k1,ampl,Bx,Bf);

StringGrid1->Cells[0][k] = FloatToStr(tp);

for(int i=0; i<N; i++)

StringGrid1->Cells[i+1][k] = FloatToStr(x[i]);

t = tp;

tp = tp+hp;

}

StringGrid1->RowCount = k+1;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

//Обработчик событий кнопки "График"

{

/\* Метод для построение графика найденных функций \*/

int i,n;

double t, x1, x2, x3, x4;

Chart1->Visible=true;

Series1->Clear();

Series2->Clear();

Series3->Clear();

Series4->Clear();

n = StringGrid1->RowCount-1;

for(i=0; i<n; i++) {

t = StrToFloat(StringGrid1->Cells[0][i+1]);

x1 = StrToFloat(StringGrid1->Cells[1][i+1]);

x2 = StrToFloat(StringGrid1->Cells[2][i+1]);

x3 = StrToFloat(StringGrid1->Cells[3][i+1]);

x4 = StrToFloat(StringGrid1->Cells[4][i+1]);

Series1->AddXY(t, x1, "", clRed);

Series2->AddXY(t, x2, "", clNavy);

Series3->AddXY(t, x3, "");

Series4->AddXY(t, x4, "");

}

}

//---------------------------------------------------------------------------

**Файл Unit2.h**

//---------------------------------------------------------------------------

#ifndef EylerH

#define EylerH

//---------------------------------------------------------------------------

const int N=4;

void f(int n, double t, double x[], double z[], double M, double m, double L, double l, double Ip, double g, double Rm, double Lm, double Km, double r, double Kg, double k[], double ampl, double Bx, double Bf);

void Eyler(double a, double b, double h, int n, double x[], double M, double m, double L, double l, double Ip, double g, double Rm, double Lm, double Km, double r, double Kg, double k[], double ampl, double Bx, double Bf);

//---------------------------------------------------------------------------

#endif

**Файл Unit2.cpp**

//---------------------------------------------------------------------------

#pragma hdrstop

//---------------------------------------------------------------------------

#include <Math.h>

#include "Unit2.h"

//---------------------------------------------------------------------------

void f(int n, double t, double x[], double z[], double M, double m, double L, double l, double Ip, double g, double Rm, double Lm, double Km, double r, double Kg, double k[], double ampl, double Bx, double Bf)

{

double B,D,temp1,temp2;

double V,w1,w2;

w1=ampl\*sin(3\*t)/50;

w2=ampl\*cos(sin(5\*t))/50;

V=x[0]\*k[0]+x[1]\*k[1]+x[2]\*k[2]+x[3]\*k[3];

B=Kg\*Km/r;

D=m\*m\*l\*l\*sin(x[2])\*sin(x[2])+m\*M\*l\*l+(M+m)\*Ip;

temp1=((B/Rm)\*V-B\*B/Rm\*x[1]-m\*l\*sin(x[2])\*x[3]\*x[3]-Bx\*x[1]-w1);

temp2=m\*g\*l\*sin(x[2])-Bf\*x[3]-w2;

/\* x - массив значений искомых функций

z - массив значений производных \*/

z[0]=x[1];

z[1]=(1/D)\*((m\*l\*l+Ip)\*temp1+m\*l\*cos(x[2])\*temp2);

z[2]=x[3];

z[3]=(1/D)\*((m\*l\*cos(x[2]))\*temp1+(M+m)\*temp2);

}

//---------------------------------------------------------------------------

void Eyler(double a, double b, double h, int n, double x[], double M, double m, double L, double l, double Ip, double g, double Rm, double Lm, double Km, double r, double Kg, double k[], double ampl, double Bx, double Bf)

{/\* Метод Эйлера исправленный \*/

double t, t1, z[N], x2[N], fi1[N];

t=a;

while(t+h<=b) {

f(n, t, x, z,M,m,L,l,Ip,g,Rm,Lm,Km,r,Kg,k,ampl,Bx,Bf);

t1 = t+h;

for(int i=0; i<n; i++) x2[i]=x[i]+h\*z[i];

f(n, t1, x2, fi1,M,m,L,l,Ip,g,Rm,Lm,Km,r,Kg,k,ampl,Bx,Bf);

for(int i=0; i<n; i++) x[i]=x[i]+(fi1[i]+z[i])\*(h/2);

t=t+h;

}

}

//---------------------------------------------------------------------------

#pragma package(smart\_init)

# Контрольный пример

Отладка программы проводилась на примере из учебного пособия со следующими значениями параметров:



где 



при начальных условиях ,, с шагом h=0,1 и границами [a,b]=[0;15]

Интерфейс программы с исходными данными и результатами счета, представлены на рисунке 1.

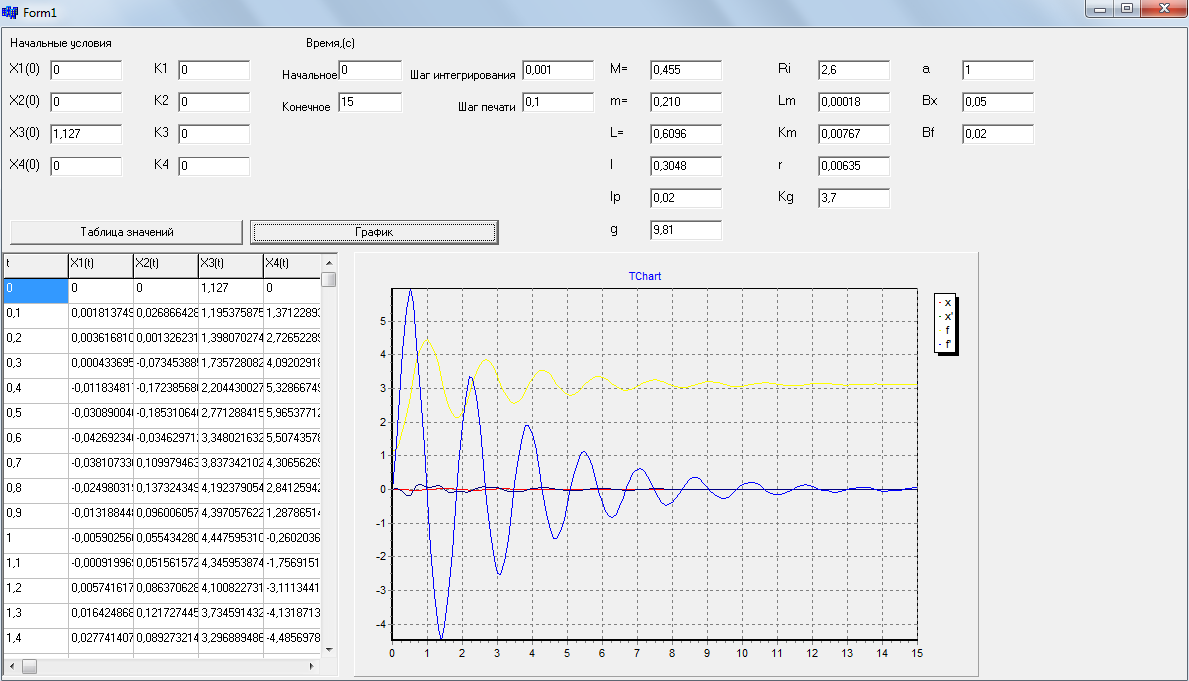


Рисунок 1. Интерфейс программы с исходными данными и результатами счета

# Сравнение результатов счета с использованием пакета MathCad

1. И с х о д н ы е д а н н ы е























2. Н а ч а л ь н ы е у с л о в и я





3. Д и ф ф е р е н ц и р о в а н и е с и с т е м ы о д н о р о д н ы х у р -н и й



4. П о с т р о е н и е т а б л и ц ы з н а ч е н и й



5. П о с т р о е н и е т р а е к т о р и и д в и ж е н и я м а я н и к а н а т е л е ж к е

