Одеський національний університет імені І.І.Мечникова
(повне найменування вищого навчального закладу)
Факультет математики, фізики та інформаційних технологій
(повне найменування інституту/факультету)
Кафедра теоретичної механіки
(повна назва кафедри)

Дипломна робота

на здобуття ступеня вищої освіти «магістр»
на тему: «**Дослідження руху тіла з порожниною**»
«Investigation motion of body with a cavity»

Виконав: студентка денної форми навчання
спеціальності 113 Прикладна математика
Рачинська Дар'я Олександрівна

Керівник д.ф.-м.н., проф. Асланов С.К.
(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент к.ф.-м.н., доцент Косой М.Б.
(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рекомендовано до захисту:
Протокол засідання кафедри
№ ____ від _____ 2018 р.

Захищено на засіданні ЕК №
протокол №____ від _____ 2018 р.
Оцінка _____ / _____ / _____
(за національною шкалою, шкалою ECTS, бали)

Завідувач кафедри
_____ Асланов С.К.
(підпис)

Голова ЕК
_____ Лещенко Д.Д.
(підпис)

Одеса – 2018

ЗМІСТ

ВСТУП.....4

РОЗДІЛ 1. РУХ ТВЕРДОГО ТІЛА З ПОРОЖНИНАМИ.....6

1.1 Основні рівняння.....6

1.2 Аналіз гідродинамічної задачі.....11

1.3 Про рішення лінійної стаціонарної крайової задачі.....15

1.4 Деякі приватні форми порожнин.....19

1.5 Рівняння динаміки тіла з рідиною.....29

РОЗДІЛ 2. ПЛОСКИЙ РУХ МАЯТНИКА З В'ЯЗКОЮ РІДИНОЮ.....35

2.1 Математичне моделювання плоского руху маятника зі сферичною порожниною.....35

2.2 Розрахунок системи диференціальних рівнянь методом Рунге-Кутта.....36

2.3 Чисельне дослідження плоского руху маятника зі сферичною порожниною.....39

РОЗДІЛ 3. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ НА МОВІ C#.....44

3.1 Визуалізація руху у тривимірному просторі з використанням бібліотеки **DirectX**.....44

3.2 Компонент **ZedGraphControl** для побудови графіків.....49

3.3 Створення власних бібліотек на мові C#.....52

РОЗДІЛ 4. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПЛОСКОГО РУХУ МАЯТНИКА ЗІ СФЕРИЧНОЮ ПОРОЖНИНОЮ.....54

4.1 Створення бібліотеки для чисельного розрахунку.....54

4.2 Проектування головної форми додатку **GForm**.....56

4.3 Проектування допоміжної форми параметрів задачі **FormParam**.58

4.4 Проектування допоміжної форми побудови графіків **Grafics**……..59

ВИСНОВОК……..60

ЛІТЕРАТУРА……...62

ДОДАТОК 1. Класи всласної бібліотеки чисельного розрахунку spherica__cavity

ДОДАТОК 2. Головна форма додатку……...64

ДОДАТОК 3. Форма параметрів задачі……73

ДОДАТОК 4. Форма графіків……74

ДОДАТОК 5. Головне вікно додатку……76

ДОДАТОК 6. Допоміжна форма графіків……..77

ДОДАТОК 7. Вікно параметрів моделі……..78

ДОДАТОК 8. Графік зміни кута відхилення маятника……79

ВСТУП

Задачі динаміки тіл з порожнинами, частково або повністю заповненими рідиною ось уже більше ста років привертають увагу дослідників, представляючи як практичний, так і теоретичний інтерес. Будучи класичними завданнями механіки і перебуваючи на стику таких дисциплін, як теоретична механіка і гідродинаміка, завдання руху тіл з рідиною в порожнинах мають величезне практичне значення. Практичне застосування даних завдань пов'язано в першу чергу з динамікою літальних апаратів, що мають на борту запас рідкого палива, теорією руху кораблів і підводних човнів. Останнім часом багато уваги приділяється розробці ефективних демпферів коливань різних висотних конструкцій, які являють собою судини, частково заповнені рідиною з частотою першої моди, узгоджується з власною частотою коливання конструкції.

За більш ніж сто років досліджень завдання руху тіл з порожнинами, частково або повністю заповненими рідиною, отримали рішення для деяких окремих випадків моделей рідкого наповнювача і моделей руху тіла. Так, завдання руху тіла з порожниною, повністю заповненою ідеальною рідиною, що робить потенційне рух, була зведена до вирішення завдання Неймана для конкретної форми порожнини. Завдання руху тіл, що містять сильно в'язку рідину, була зведена до вирішення системи звичайних диференціальних рівнянь, а для випадку руху тіл з порожнинами, заповненими слабо в'язкою рідиною, були отримані інтегро-диференціальні рівняння, що описують рух в лінійній постановці. Рішення задач руху тіл з порожнинами, частина заповненими рідиною, вдалося звести до вирішення завдання визначення потенціалу хвильового руху і завдання руху тіла з порожниною, повністю заповненою рідиною. У завданнях розробки гасителів коливань основна увага приділяється експериментального дослідження та чисельного дослідження наближених рівнянь динаміки хвильового руху рідини (рівняння мілкої води). Однак в даний час майже не досліджено рух тіл з порожнинами, частково або повністю заповненими в'язкою рідиною, в повній

нелінійній постановці, і мало даних за течією рідини, що виникає в рухомих порожнинах.

В дипломній роботі проводится комп'ютерне моделювання руху маятника з порожниною. Порожнина має сферичну форму та цілком заповнена рідиною великої вязкості.

Математична модель складається з диференціального рівняння другого порядку з одним параметром. Проведено повне дослідження впливу параметру та початкових умов на рух супутника. За допомогою тривимірної графіки змодельовано рух згідно отриманих чисельних розрахунків.

ВИСНОВОК

Для візуалізації руху маятника з порожниною близько нерухомої осі підвісу було розроблено **Windows**-додаток на мові програмування **C#**. Додаток імітує власну віртуальну лабораторію і надає користувачеві можливість спостерігати анимированное рух системи в реальному часі при заданих параметрах завдання.

Для реалізації поставленого завдання використовується бібліотека **DirectX9**.0. За допомогою бібліотеки створюються сцена, освітлення і **Mesh** - об'єкти, з яких складається система. Всі фігури, що зображуються на сцені за допомогою **Direct3D**, представляють собою сукупність точок - вершин. Інформація про кожну вершині містить такі параметри, як положення в просторі, колір вершини, інформацію про використану текстурі і т.д. **Mesh**-об'єкти можуть використовуватися для збереження будь-якого зображення будь-якого типу графічних даних, але головним чином призначені для формування складових моделей.

У розробленому додатку система моделюється за допомогою наступних **Mesh** - об'єктів: стіна, кріплення, маятник, що складається з осі і сфери з порожниною. Для реальності зображення на сцені використовувалися текстури цегляної стіни і рідини.

Головне вікно програми має вигляд, представлений в додатку 5. У верхній частині вона розташована інформаційна панель для відображення інформації про параметри розрахунку моделі маятника, рух якого здійснюється на сцені головного додатки. У лівій частині вікна розташовується функціональна панель з кнопками: **Start** - новий розрахунок завдання, починаючи з моменту часу; **Play** - продовження анімації руху з поточного моменту часу $\tau = 0$; **Pause** - зупинка анімації руху маятника; **Plus** і **Minus** - для наближення або видалення об'єктів системи на сцені; а так само кнопки побудови графіків основних характеристик руху і зміни параметрів розрахунку завдання.

У нижній частині головного вікна (див. додаток 5) є інформативна область з назвою **«Новый расчет»**. При анімації руху системи в поточний момент часу виводяться значення: кута відхилення маятника від нижнього положення рівноваги; кутова швидкість маятника; кінетична, потенційна і повна енергія системи.

У поточний момент часу користувач може побудувати графіки основних характеристик руху системи за допомогою допоміжного вікна, представленого у додатку 6.

На формі користувач має функціональну можливість вибору фізичної величини, графік якої буде відображатися у вікні. Написи в вікнах вибору зроблені різними кольорами, які відповідають кольору кривої графіка.

У додатку 6 представлений графік зміни кута відхилення маятника від положення рівноваги при малій в'язкості рідини в порожнині $\alpha = 0.01$. Видно, що має місце згасання коливань на невелику величину.

Для зміни параметрів моделі використовується вікно представлене у додатку 7.

Змінимо безрозмірний коефіцієнт сил в'язкої рідини (2.1.7) $\alpha = 0.1$, початкова швидкість $\omega_0 = 0$, а кут відхилення $\varphi_0 = 145^0$. Для побудованої моделі графік зміни кута відхилення маятника від положення рівноваги має вигляд представлений у додатку 8. Видно, що загасання коливань стає істотним.

У даній роботі розроблено Windows-додаток, що моделює обертальний рух маятника з порожниною, цілком заповненою в'язкою рідиною. Windows-додаток має інтуїтивно зрозумілий і функціонально зручний інтерфейс, який дозволяє дослідник рух маятника для різних параметрів моделі і проводити візуалізацію руху в реальному часі.

ЛІТЕРАТУРА

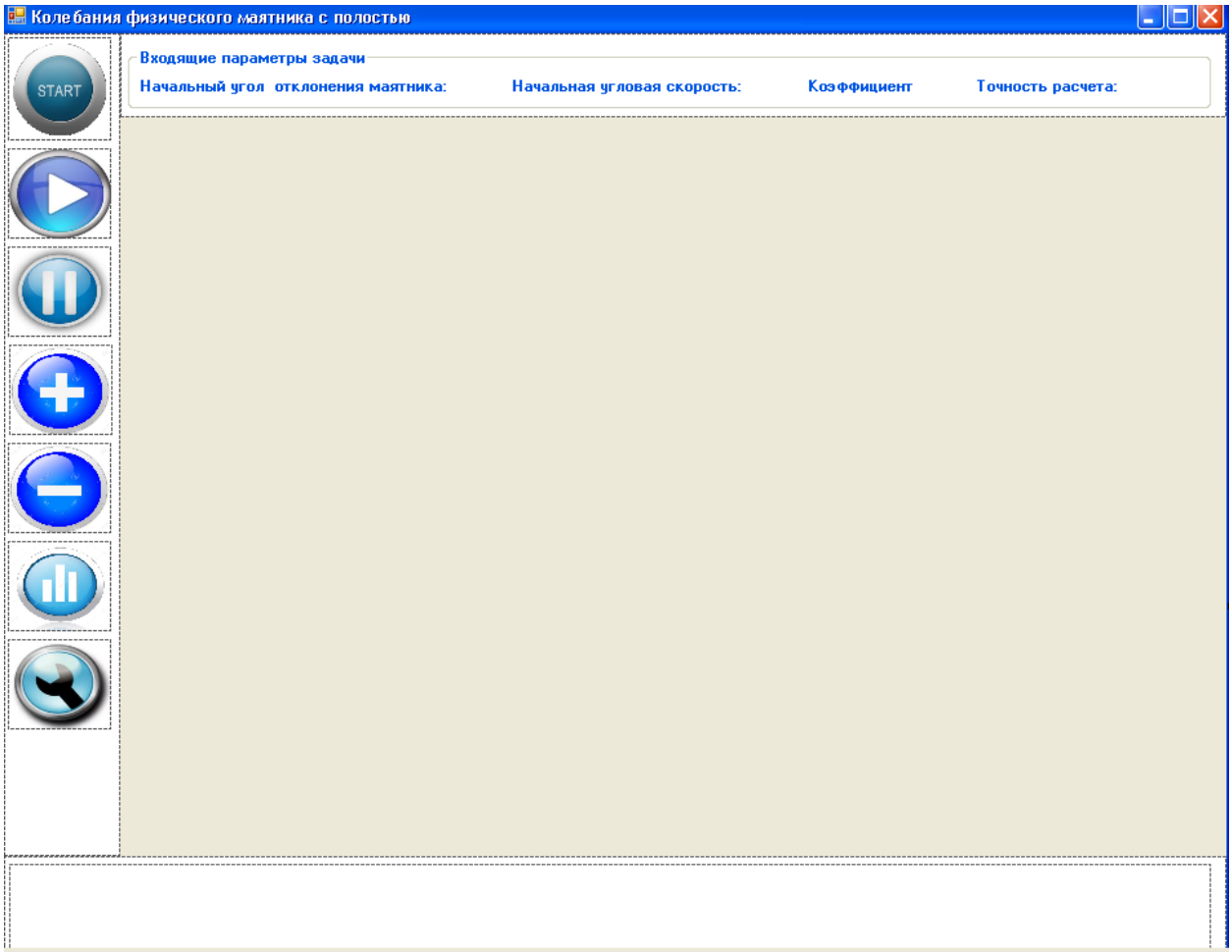
1. Ладыженская О.А. Математические вопросы динамики вязкой несжимаемой жидкости / Ольга Александровна Ладыженская. – М.: Физматгиз, 1981. – 288 с.
2. Вишик М.И. Регулярное вырождении и пограничный слой для линейных дифференциальных уравнений с малым параметром / М.И. Вишик, Л.А. Люстерник // УМН. – 1957. – Т.12, вып. 5. – С. 3-122.
3. Кочин Н.Е. Векторное исчисление и начала тензорного числения / Николай Евграфович Кочин. – М.: Изд-во АН СССР, 1961. – 427 с.
4. Ватсон Г.Н. Теория бесселевых функций / Джордж Невил Ватсон. – М.:Изд-во иностр. лит-ры. – 1949. – 799 с.
5. Янке Е., Эмде Ф. Таблицы функций с формулами и кривымию. – М.:Физматгиз. – 1959. – 344 с.
6. Боголюбов Н.Н., Митропольский Ю.А. Асимптотические методы в теории нелинейных колебаний / Николай Николаевич Боголюбов, Юрий Алексеевич Митропольский. – М.: Физматгиз. – 1963. – 503 с.
7. Волосов В.М. Усреднение в системах обыкновенных дифференциальных уравнений // УМН. – 1962. – Т.17, вып. 6. – С. 3-126.
8. Черноусько Ф.Л. Движение твердого тела с полостями, содержащими вязкую жидкость / Ф.Л. Черноусько // М.: Изд-во ВЦ АН СССР, 1968. – 230 с.
9. Жуковский Н.Е. О движении твердого тела, имеющего полости, наполненные однородною капельною жидкостью / Н.Е. Жуковский // Избранные сочинения. Т.I. – М. – Л.: Гостехиздат, 1948. – С.31-152.
10. Эндрю Троелсен. C# и платформа .NET. / Библиотека программиста/ Москва., Санкт-Петербург: Издательство «Питер». – 2004. – 785 с.
11. Том Миллер. DirectX 9 с управляемым кодом. Программирование игр и графика./ М.: Издательский дом «КомБук», 2005. – 395 с.

12. Калиткин Н.Н. Численные методы./ Математика/ - М. : Наука. 1978. – 512с.
13. Купчак А., Рачинская Д. Анимация движения физического маятника с полостью/ А. Купчак, Д. Рачинская // Матеріали Всеукраїнської науково-практичної інтернет-конференції «ВІТЧИЗНЯНА НАУКА НА ЗЛАМІ ЕПОХ: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ» Вип. 43, 2018. – С. 212 - 217.

ДОДАТОК 1.
Класи всласної бібліотеки чисельного розрахунку spherica cavity

```
public class Motion_calculation
{
    double _alfa;

    public double alfa
    {
        set
        {
            _alfa = value;
        }
        get
        {
            return _alfa;
        }
    }
    public double f1(double x, double[] y, int n)
    {
        return y[1];
    }

    public double f2(double x, double[] y, int n)
    {
        return -Math.Sin(y[0]) - alfa * Math.Cos(y[0]) * y[1];
    }
}
public class Runge
{
     public delegate double myF(double x, double[] y, int n);

    public double[] RungeR(double x0, double[] y, myF[] f, int n, double h)
    {
        int j, i;
        double[] y_new = new double[n];
        double[] k1 = new double[n];
        double[] k2 = new double[n];
        double[] k3 = new double[n];
        double[] k4 = new double[n];
        double x = x0;
        for (j = 0; j < n; j++)
            k1[j] = h * f[j](x, y, n);
        for (i = 0; i < n; i++)
            y_new[i] = y[i] + 0.5 * k1[i];
        for (j = 0; j < n; j++)
            k2[j] = h * f[j](x + 0.5 * h, y_new, n);
        for (i = 0; i < n; i++)
            y_new[i] = y[i] + 0.5 * k2[i];
        for (j = 0; j < n; j++)
            k3[j] = h * f[j](x + 0.5 * h, y_new, n);
        for (i = 0; i < n; i++)
            y_new[i] = y[i] + k3[i];
        for (j = 0; j < n; j++)
            k4[j] = h * f[j](x + h, y_new, n);
        for (i = 0; i < n; i++)
            y_new[i] = y[i] + (k1[i] + 2 * k2[i] + 2 * k3[i] + k4[i]) / 6;
        return y_new;
    }

}
```

ДОДАТОК 2.
Головна форма додатку




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Microsoft.DirectX;
using Microsoft.DirectX.Direct3D;
using System.IO;
using spherica__cavity;
using ZedGraph;
using System.Threading;



namespace WindowsFormsApplication1
{
    public partial class GForm : Form
    {
        //графика
        Device d3d;
        Mesh podshipnik, krep, stenka;
        Mesh telo1, telo2,telo3;

        float r1 = 0.04f;
        float r2 = 1.0f;
        float l1 = 5.0f;
```

```
        float l_krep = 1.5f;

        float r_p = 0.1f;
        float ykrep = 2.78f;
        float sdvig_z = 20.0f,sdvig_x=0.6f,fi;
        float delta = 0.01f;
        float a = 20f;

        int k = 0;
        Material kolesoMaterial, verevkaMaterial, boxMaterial;
        Texture m1, m2, m3;


        //расчет
        Runge rashet;
        Runge.myF[] masF;
        Motion_calculation mc;

        static public double eps=0.01;
        static public float fi0,alfa,om0;
        double[] y;


       public List<double> fir,tr,omr,H,T,P;
        double t = 0;

        //расчетные данные на листик
        Graphics g;
        System.Drawing.Drawing2D.HatchBrush hb;

        public GForm()
        {
            InitializeComponent();

            SetStyle(ControlStyles.Opaque, true);
            SetStyle(ControlStyles.UserPaint, true);
            SetStyle(ControlStyles.AllPaintingInWmPaint, true);
            fi0 = Convert.ToSingle(Math.PI / 4);
            alfa = 0.01f;
            om0 = 0;

            hb = new
System.Drawing.Drawing2D.HatchBrush(System.Drawing.Drawing2D.HatchStyle.Cross,
Color.CadetBlue, Color.White);
        }

        public void Metki()
        {
            label_eps.Text = "Точность расчета: " + String.Format("{0:F3}",eps);
            label_alfa.Text = "Коэффициент " + Convert.ToString('\u03b1') + ": " +
Convert.ToString(alfa);
            label_v0.Text = "Начальная угловая скорость: " +
Convert.ToString(om0);
            label_fi0.Text = "Начальный угол отклонения маятника: "  +
String.Format("{0:F2}",fi);
        }


        private void myGraph(Graphics g)
        {
            System.Drawing.Text.InstalledFontCollection fonts = new
System.Drawing.Text.InstalledFontCollection();
            g.FillRectangle(hb,new Rectangle(new Point(0,0),pictureBox6.Size));
            g.DrawString("Новый расчет", new
System.Drawing.Font(fonts.Families[42].Name, 30, FontStyle.Italic), new
SolidBrush(Color.Blue), new PointF(pictureBox6.Size.Width / 2-50, 10));
            if (fir == null || fir.Count==0) return;
```

```
            g.DrawString("в момент времени t=", new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(30, 70));
            int n = fir.Count;
            g.DrawString(String.Format("{0:F3}",tr[n-1]), new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(250, 70));
            g.DrawString("угол отклонения маятника
"+Convert.ToString('\u03C6')+"=", new System.Drawing.Font(fonts.Families[42].Name,
20, FontStyle.Italic), new SolidBrush(Color.Black), new PointF(350, 70));
            g.DrawString(String.Format("{0:F3}", fir[n - 1]), new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(670, 70));
            g.DrawString("угловая скорость маятника " + Convert.ToString('\u03C9')
+ "=", new System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(30, 130));
            g.DrawString(String.Format("{0:F3}", omr[n - 1]), new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(355, 130));
            g.DrawString("кинетическая энергия маятника T=", new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(450, 130));
            g.DrawString(String.Format("{0:F3}", T[n - 1]), new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(820, 130));
            g.DrawString("потенциальная энергия маятника H=", new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(30, 190));
            g.DrawString(String.Format("{0:F3}", P[n - 1]), new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(410, 190));
            g.DrawString("полная энергия маятника E=", new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(480, 190));
            g.DrawString(String.Format("{0:F3}", H[n - 1]), new
System.Drawing.Font(fonts.Families[42].Name, 20, FontStyle.Italic), new
SolidBrush(Color.Black), new PointF(780, 190));



        }

        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                PresentParameters d3dpp = new PresentParameters();
                d3dpp.BackBufferCount = 1;
                d3dpp.SwapEffect = SwapEffect.Discard;
                d3dpp.Windowed = true; // Выводим графику в окно
                d3dpp.MultiSample = MultiSampleType.None; // Выключаем
антиалиасинг
                d3dpp.EnableAutoDepthStencil = true; // Разрешаем создание z-
буфера
                d3dpp.AutoDepthStencilFormat = DepthFormat.D16; // Z-буфер в 16
бит
                d3d = new Device(0, // D3D_ADAPTER_DEFAULT - видеоадаптер по
                    // умолчанию
                      DeviceType.Hardware, // Тип устройства - аппаратный
ускоритель
                      this, // Окно для вывода графики
                      CreateFlags.SoftwareVertexProcessing,
                      d3dpp);
            }
            catch (Exception exc)
            {
                MessageBox.Show(this, exc.Message, "Ошибка инициализации");
```

```
                Close(); // Закрываем окно

            }

            stenka = Mesh.Box(d3d, a, a, a);
            podshipnik = Mesh.Sphere(d3d, r_p, 10, 10);
            krep = Mesh.Cylinder(d3d, r1, r1, l_krep, 50, 10);

            telo1 = Mesh.Cylinder(d3d, r1, 2.5f * r1, l1, 50, 10);
            telo2 = Mesh.Sphere(d3d, r2, 25, 25);
            telo3 = Mesh.Sphere(d3d, r2 + delta, 25, 25);


            kolesoMaterial = new Material();
            kolesoMaterial.Diffuse = Color.DarkMagenta;
            kolesoMaterial.Specular = Color.White;

            verevkaMaterial = new Material();
            verevkaMaterial.Diffuse = Color.DarkGray;
            verevkaMaterial.Specular = Color.White;

            boxMaterial = new Material();
            boxMaterial.Diffuse = Color.BlueViolet;
            boxMaterial.Specular = Color.White;

            using (FileStream textureFile = new FileStream(@"..\..\Images\b.jpg",
FileMode.Open))
            {
                m1 = Texture.FromStream(d3d, textureFile, 0, Pool.Managed);
                textureFile.Close();
            }
            using (FileStream textureFile = new
FileStream(@"..\..\Images\bel.jpg", FileMode.Open))
            {
                m2 = Texture.FromStream(d3d, textureFile, 0, Pool.Managed);
                textureFile.Close();
            }
            using (FileStream textureFile = new
FileStream(@"..\..\Images\water.jpg", FileMode.Open))
            {
                m3 = Texture.FromStream(d3d, textureFile, 0, Pool.Managed);
                textureFile.Close();
            }
            fi = Convert.ToSingle(Math.PI / 4);
            Metki();
            g = pictureBox6.CreateGraphics();
            myGraph(g);
            fir = new List<double>();
            tr = new List<double>();
            omr = new List<double>();
            H = new List<double>();
            T = new List<double>();
            P = new List<double>();
        }
        void Picture()
        {
            d3d.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.LightBlue,
1.0f, 0);
            SetupProekcii();
            d3d.BeginScene();
            d3d.RenderState.FillMode = FillMode.Solid;
            d3d.TextureState[0].TextureCoordinateIndex
=(int)TextureCoordinateIndex.CameraSpacePosition;

            // Преобразуем текстурные координаты по заданным формулам
            d3d.Transform.Texture0 = Matrix.Scaling(0.5f, -0.5f, 1.0f) *
Matrix.Translation(0.5f, 0.5f, 0.0f);
```

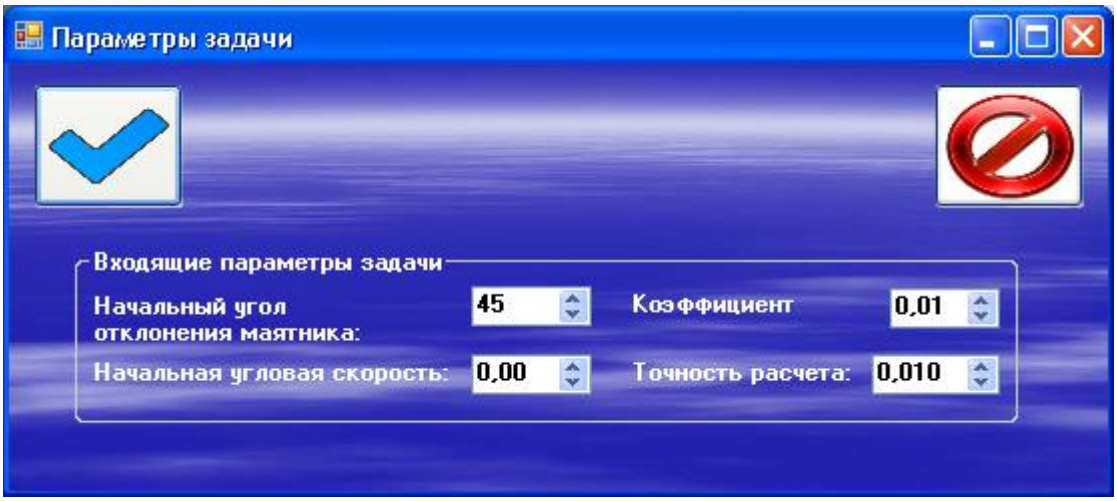
```
            // Передаем блоку растеризации две текстурные координаты
            d3d.TextureState[0].TextureTransform = TextureTransform.Count2;

            // Выключаем расчет освещения
            d3d.RenderState.Lighting = false;
            d3d.SetTexture(0, m2);

            d3d.RenderState.CullMode = Cull.None;

            //стенка
            d3d.Material = verevkaMaterial;
            d3d.Transform.World = Matrix.Translation(0, 0, a / 2) *
Matrix.RotationY((float)((hScrollBar1.Value - 90) * Math.PI / 180)) *
Matrix.Translation(sdvig_x, 0, sdvig_z);
            stenka.DrawSubset(0);

            // В качестве текстурных координат берем вектор нормали
            d3d.TextureState[0].TextureCoordinateIndex =
                    (int)TextureCoordinateIndex.CameraSpaceNormal;
            d3d.SetTexture(0, m1);
            //подшипник
            d3d.Material = boxMaterial;
            d3d.Transform.World = Matrix.Translation(0, 0, -l_krep) *
Matrix.RotationY((float)((hScrollBar1.Value - 90) * Math.PI / 180)) *
Matrix.Translation(sdvig_x, 0 + ykrep - 2 * delta - r_p, sdvig_z);
            podshipnik.DrawSubset(0);

            //kreplenie
            d3d.Transform.World = Matrix.Translation(0, ykrep - 2 * delta - r_p, -
l_krep / 2) * Matrix.RotationY((float)((hScrollBar1.Value - 90) * Math.PI / 180))
* Matrix.Translation(sdvig_x, 0, sdvig_z);
            krep.DrawSubset(0);

            //тело
            d3d.Transform.World = Matrix.RotationX((float)Math.PI / 2) *
Matrix.Translation(0, -l1 / 2, 0 - l_krep) * Matrix.RotationZ(fi) *
Matrix.RotationY((float)((hScrollBar1.Value - 90) * Math.PI / 180)) *
Matrix.Translation(sdvig_x, 0 + ykrep - 2 * delta - r_p, sdvig_z);
            telo1.DrawSubset(0);

            d3d.TextureState[0].TextureCoordinateIndex
=(int)TextureCoordinateIndex.CameraSpaceNormal;
            d3d.Transform.Texture0 = Matrix.Scaling(0.5f, -0.5f, 1.0f)
*Matrix.Translation(0.5f, 0.5f, 0.0f);
            d3d.TextureState[0].TextureTransform = TextureTransform.Count2;
            d3d.SetTexture(0, m3);
            d3d.SamplerState[0].MinFilter = TextureFilter.Linear;
            d3d.SamplerState[0].MagFilter = TextureFilter.Linear;
            d3d.Transform.World = Matrix.Translation(0, -0.75f * l1, 0 - l_krep) *
Matrix.RotationZ(fi) * Matrix.RotationY((float)((hScrollBar1.Value - 90) * Math.PI
/ 180)) * Matrix.Translation(sdvig_x, 0 + ykrep - 2 * delta - r_p, sdvig_z);
            telo2.DrawSubset(0);
            d3d.SetTexture(0, m1);
            d3d.RenderState.FillMode = FillMode.WireFrame;
            d3d.Transform.World = Matrix.RotationX((float)Math.PI / 2) *
Matrix.Translation(0, -0.75f * l1, 0 - l_krep) * Matrix.RotationZ(fi) *
Matrix.RotationY((float)((hScrollBar1.Value - 90) * Math.PI / 180)) *
Matrix.Translation(sdvig_x, 0 + ykrep - 2 * delta - r_p, sdvig_z);
            telo3.DrawSubset(0);
            d3d.EndScene();
            d3d.Present();
        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Picture();
        }
```

```
        private void SetupProekcii()
        {
            d3d.Lights[0].Enabled = true;
            d3d.Lights[0].Diffuse = Color.White;
            d3d.Lights[0].Position = new Vector3(0, 0, 0);
            d3d.Transform.Projection = Matrix.PerspectiveFovLH((float)Math.PI / 4,
this.Width / this.Height, 1.0f, 50.0f);
        }

        private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
        {
            this.Invalidate();
        }

        void Parametrs()
        {
            t = 0;
            rashet = new Runge();
            mc = new Motion_calculation();
            mc.alfa = alfa;
            masF = new Runge.myF[2];
            masF[0] = mc.f1;
            masF[1] = mc.f2;
            y = new double[2];
            fir = new List<double>();
            tr = new List<double>();
            omr = new List<double>();
            H = new List<double>();
            T = new List<double>();
            P = new List<double>();
            y[0] = fi0;
            y[1] = om0;
            fi = fi0;
            fir.Add(y[0]);
            tr.Add(0);
            omr.Add(y[1]);
            Func_H(0);
            Func_T(0);
            Func_P(0);

        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            Parametrs();
            timer1.Enabled = true;
            SolidBrush sb = new SolidBrush(Color.White);
            g.FillRectangle(sb, new Rectangle(new Point(0, 0), pictureBox6.Size));
            myGraph(g);
        }

        private void pictureBox3_Click(object sender, EventArgs e)
        {
            timer1.Enabled = false;
        }

        private void pictureBox2_Click(object sender, EventArgs e)
        {
            sdvig_z -= 0.5f;
            Picture();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            k++;
            try
```

```
            {
                fi = Convert.ToSingle(fir[fir.Count - 1]);
                y = rashet.RungeR(t, y, masF, 2, eps);
                t = t + eps;

                {
                    fir.Add(y[0]);
                    tr.Add(t);
                    omr.Add(y[1]);
                    Func_H(fir.Count - 1);
                    Func_P(fir.Count - 1);
                    Func_T(fir.Count - 1);
                }
                Picture();

                    myGraph(g);
            }
            catch(NullReferenceException)
            {
                return;
            }
        }

        private void pictureBox4_Click(object sender, EventArgs e)
        {
            sdvig_z += 0.5f;
            Picture();
        }
        private void pictureBox5_Click(object sender, EventArgs e)
        {
            timer1.Enabled = true;
        }
        private void numFi0_ValueChanged(object sender, EventArgs e)
        {
            Parametrs();
        }
        private void pictureBox6_Paint(object sender, PaintEventArgs e)
        {
            myGraph(e.Graphics);
        }

        private void pictureBox7_Click(object sender, EventArgs e)
        {
            SolidBrush sb = new SolidBrush(Color.White);
            g.FillRectangle(sb, new Rectangle(new Point(0, 0), pictureBox6.Size));
            timer1.Enabled = false;
            t = 0;
            fir.Clear();
            tr.Clear();
            omr.Clear();
            H.Clear();
            T.Clear();
            P.Clear();
            myGraph(g);
            FormParams fp = new FormParams();
            fp.ShowDialog();
            Parametrs();
            Metki();
        }

        private void pictureBox8_Click(object sender, EventArgs e)
        {
            Grafics fp = new Grafics();
            fp.Owner = this;
            fp.ShowDialog();
        }
```

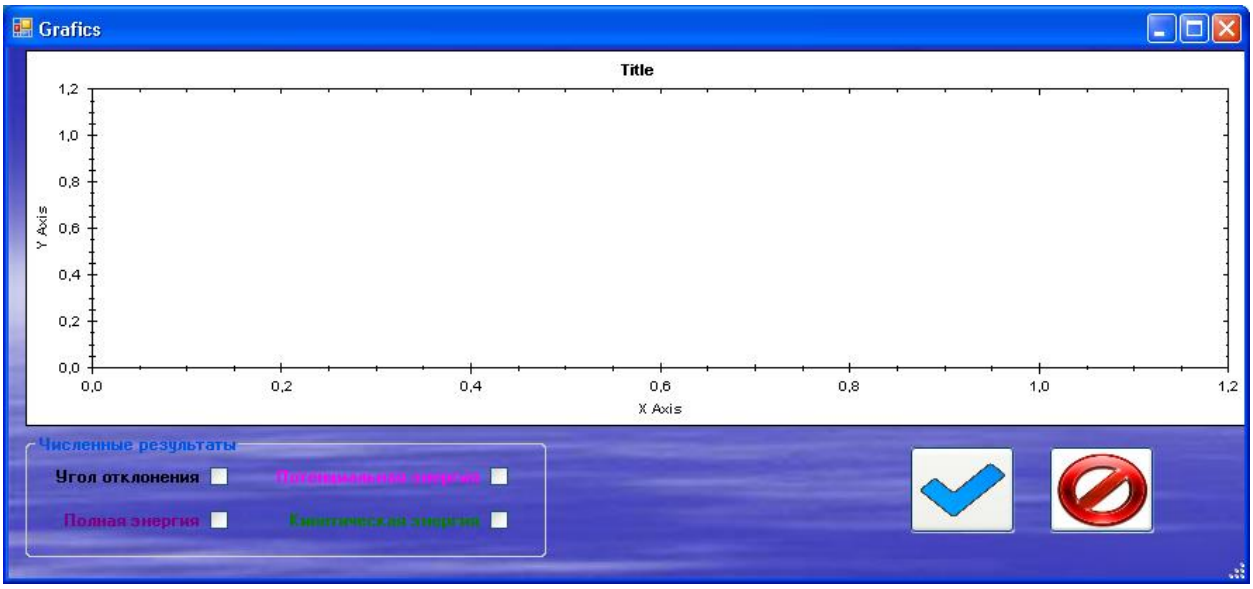
```
        void Func_H(int i)
        {


                if (alfa != 0)

                    H.Add(0.5 * omr[i] * omr[i] * alfa - (1 + Math.Cos(fir[i])));
                else
                    H.Add(0.5 * omr[i] * omr[i] - (1 + Math.Cos(fir[i])));

        }
        void Func_T(int i)
        {


                if (alfa != 0)
                    T.Add(0.5 * omr[i] * omr[i] / alfa);
                else
                    T.Add(0.5 * omr[i] * omr[i]);

        }
        void Func_P(int i)
        {

                P.Add(-(1 + Math.Cos(fir[i])));

        }



    }
}
```

ДОДАТОК 3.
Форма параметрів задачі




```
namespace WindowsFormsApplication1
{
    public partial class FormParams : Form
    {
        public FormParams()
        {
            InitializeComponent();
        }

        private void FormParams_Shown(object sender, EventArgs e)
        {
            numFi0.Value=Convert.ToDecimal(GForm.fi0*180/Math.PI);
            numOmega0.Value = Convert.ToDecimal(GForm.om0);
            numAlfa.Value = Convert.ToDecimal(GForm.alfa);
            numEps.Value = Convert.ToDecimal(GForm.eps);
        }

        private void OK_Click(object sender, EventArgs e)
        {
            GForm.fi0 =
            Convert.ToSingle(Convert.ToSingle(numFi0.Value)*Math.PI/180);
            GForm.om0 = Convert.ToSingle(numOmega0.Value);
            GForm.alfa = Convert.ToSingle(numAlfa.Value);
            GForm.eps = Convert.ToDouble(numEps.Value);
        }


    }
}
```

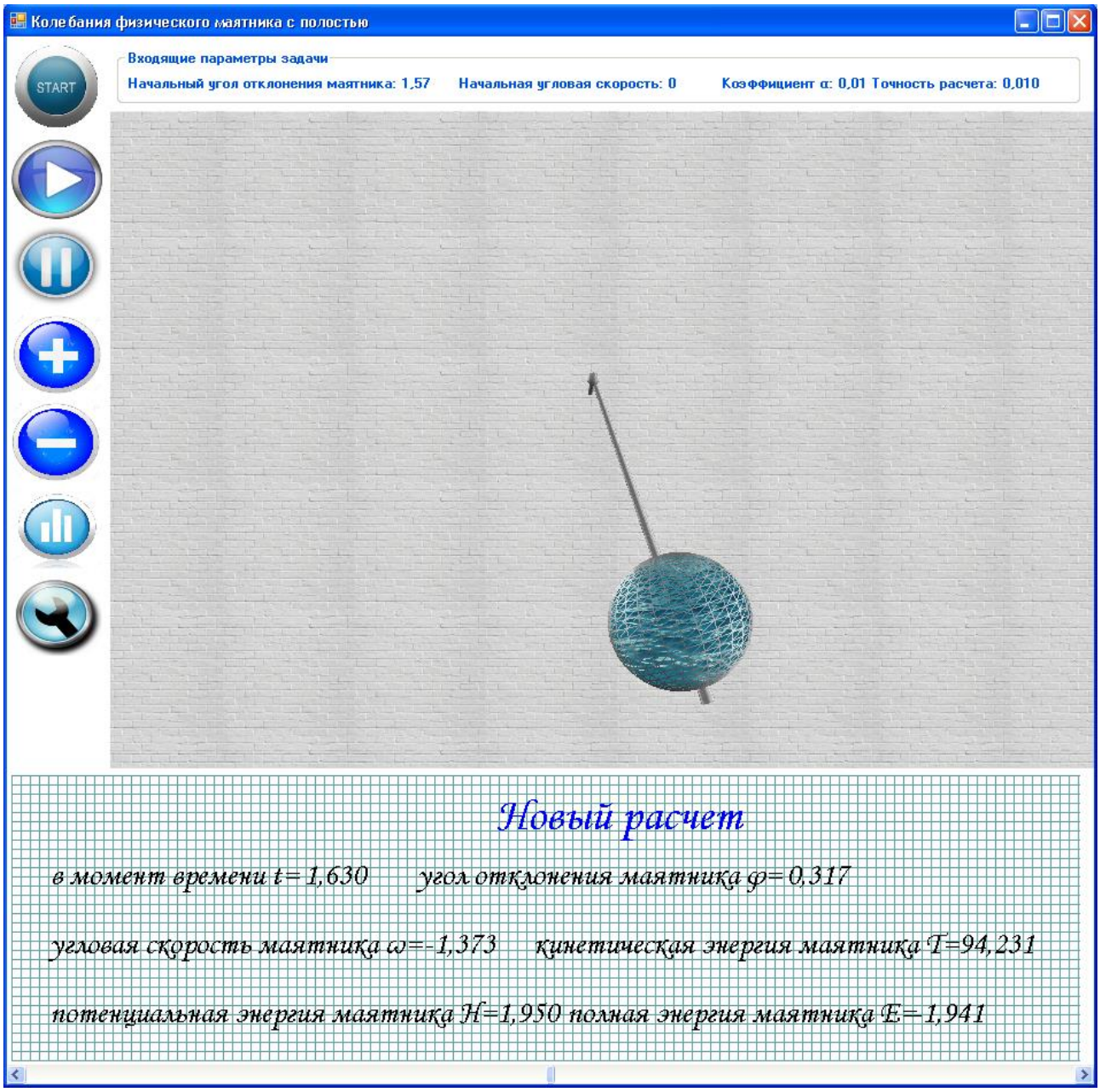
ДОДАТОК 4.
Форма графіків



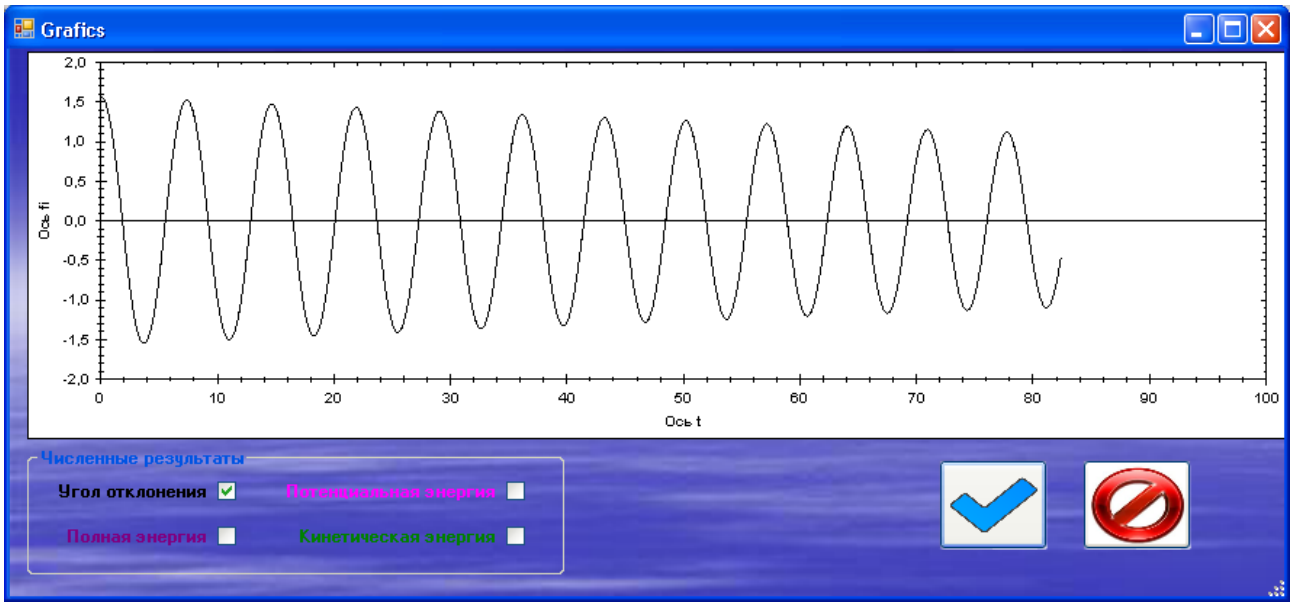

```
namespace WindowsFormsApplication1
{
    public partial class Grafics : Form
    {
        GraphPane myPane = null;
        PointPairList list, list1, list2, list3;

        public Grafics()
        {
            InitializeComponent();
            // Создадим список точек угол
            list = new PointPairList();

            // Создадим список точек полная энергия
            list1 = new PointPairList();

            // Создадим список точек кинетическая энергия
            list2 = new PointPairList();

            // Создадим список точек потенциальная энергия
            list3 = new PointPairList();
        }
       public void Grafic()
        {
            GForm fG=Owner as GForm;
            // Задаем название графика и осей
            myPane.Title.Text = "";
            myPane.XAxis.Title.Text = "Ось t";
            myPane.YAxis.Title.Text = "Ось fi";

            int n = fG.fir.Count;
            list.Clear();
            list1.Clear();
            list2.Clear();
            list3.Clear();

            LineItem myCurve1;
            if (chFi.Checked)
            {
                for (int i = 0; i < n; i++)
```

```
                list.Add(fG.tr[i], fG.fir[i]);
            myCurve1 = myPane.AddCurve("", list, Color.Black);
            myCurve1.Symbol.IsVisible = false;
        }
        LineItem myCurve2;
        if (chP.Checked)
        {
            for (int i = 0; i < n; i++)
                list1.Add(fG.tr[i], fG.H[i]);
            myCurve2 = myPane.AddCurve("", list1, Color.Purple);
            myCurve2.Symbol.IsVisible = false;

        }
        LineItem myCurve3;
        if (chT.Checked)
        {

            for (int i = 0; i < n; i++)
                list2.Add(fG.tr[i], fG.T[i]);
            myCurve3 = myPane.AddCurve("", list2, Color.Green);
            myCurve3.Symbol.IsVisible = false;

        }
        LineItem myCurve4;
        if (chH.Checked)
        {

            for (int i = 0; i < n; i++)
                list3.Add(fG.tr[i], fG.P[i]);
            myCurve4 = myPane.AddCurve("", list3, Color.Fuchsia);
            myCurve4.Symbol.IsVisible = false;

        }



        zedGraphControl1.AxisChange();
                    zedGraphControl1.Invalidate();
    }

    private void Grafics_Load(object sender, EventArgs e)
    {
        myPane = zedGraphControl1.GraphPane;
    }

    private void OK_Click(object sender, EventArgs e)
    {
        Grafic();
    }
  }
}
```

ДОДАТОК 5.
Головне вікно додатку

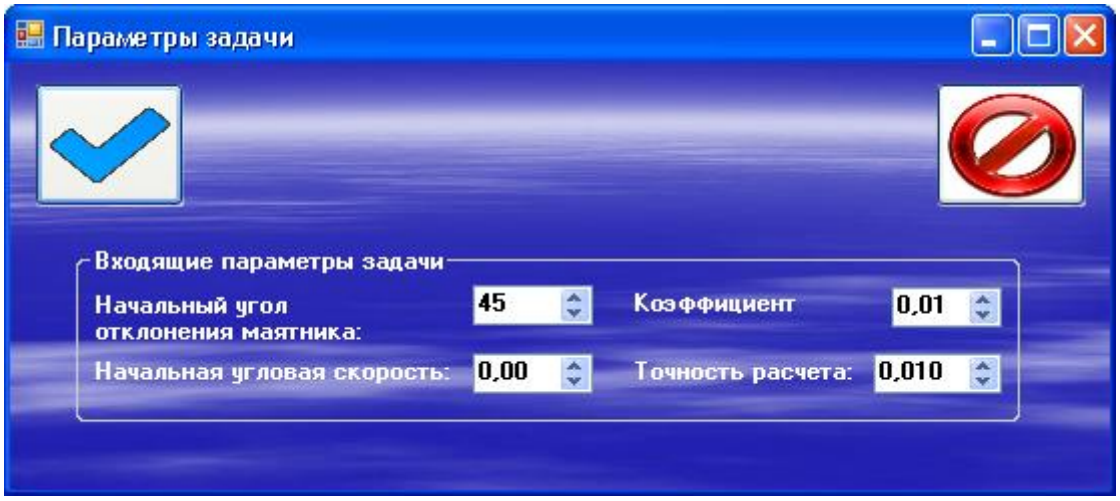


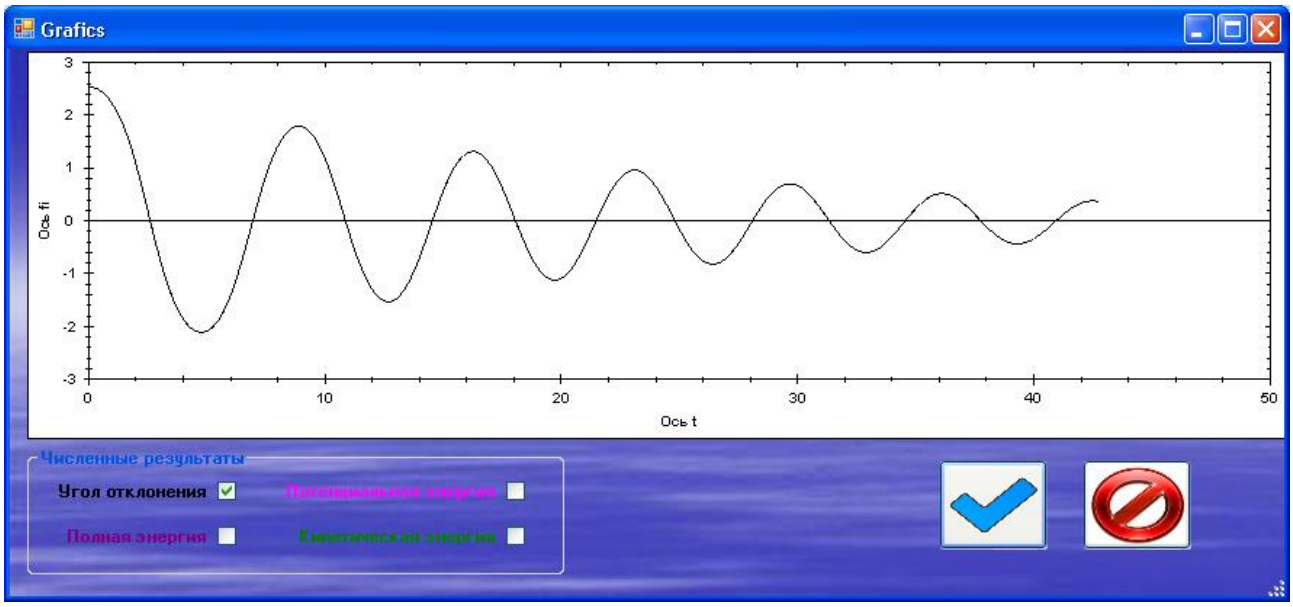
ДОДАТОК 6.
Допоміжна форма графіків



ДОДАТОК 7.

Вікно параметрів моделі



ДОДАТОК 8.

Графік зміни кута відхилення маятника