Санкт-Петербургский государственный политехнический университет
Институт компьютерных наук и технологий
Кафедра «Компьютерные системы и программные технологии»

КУРСОВАЯ РАБОТА

По теме: разработка программы для вычисления параметра жесткости маятника переменной длины

По дисциплине: Вычислительная математика

Выполнил
студент гр.3530901\90003 ______________ А.К. Руднев

Руководитель
доцент, к.т.н. ______________ _В.Н. Цыган

«___»____________2021 г.

Санкт-Петербург
2021

Санкт-Петербургский государственный политехнический университет

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

студенту группы 3530901/90003 Рудневу Александру Константиновичу
(номер группы) *(фамилия, имя, отчество)*

1. Тема работы: Разработка программы для вычисления параметра жесткости маятника переменной длины.
2. Срок сдачи законченного проекта (работы) 30 апреля 2021 г.

3. Исходные данные к проекту (работе): Общий вид дифференциальных уравнений, описывающих движение маятника переменной длины; положение маятника начала системы координат; масса маятника; начальные условия для решения дифференциальных уравнений; рекомендованный временной интервал исследования численного решения системы дифференциальных уравнений; перечень заданных преподавателем параметров уравнения с указанием уравнений или соотношений для их нахождения (вариант К-3-15)

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов): введение, основная часть (*анализ задания; вычисление значений заданных преподавателем параметров дифференциальных уравнений; разработка программы для вычисления параметра жесткости маятника переменной длины; анализрезультатов численного решения уравнений; оценка погрешности*
результата; исследование влияния на точность решения уравнений погрешности задания исходных данных), заключение, список использованных источников, приложение.

Дата получения задания: «01» марта 2021 г.

Руководитель__________________ В.Н. Цыган
(подпись) *(инициалы, фамилия)*

Задание принял к исполнению______________ А.К. Руднев
(подпись студента) *(инициалы, фамилия)*

____
(дата)

Содержание

Введение

На практике решения дифференциального уравнения или системы уравнений описывают динамику разнообразных явлений и процессов (например, движение совокупности взаимодействующих материальных точек, химическую кинетику, процессы в электрических цепях и т.п.). Следовательно умение решать уравнения или системы уравнений может пригодиться каждому инженеру. Задачей данной работы является закрепление знаний об изученных подпрограммах, позволяющих вычислять значение интеграла, находить корни уравнения, а также решение заданных дифференциальных уравнений. Цель работы: разработка программы, которая вычисляет параметр жесткости маятника переменной длины по экспериментальным значениям.

Рассматривается движение маятника переменной длины (рис. 1).

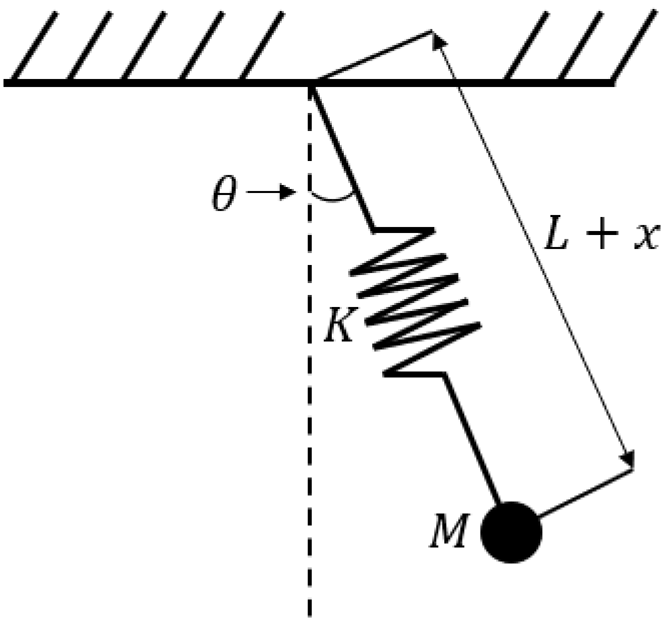

Рисунок 1 – Маятник переменной длины

На рисунке: M – масса маятника, θ – угол, L - расстояние маятника от оси, x – удлинение маятника, K – искомый параметр жесткости.

Значение L – 0.5836896 * $\int_0^1 e^x$dx (задано преподавателем)

Дифференциальные уравнения, описывающие движения маятника переменной длины имеют общий вид:

$$\begin{cases} \ddot{x} + \frac{K}{M}x + g(1 - cos\theta) - (L + x)\left(\dot{\theta}\right)^2 = 0 \\ \ddot{\theta} + \frac{g}{L + x}sin\theta + \frac{2}{L + x}\dot{x}\dot{\theta} = 0 \end{cases}$$

Где g = 9.81, M = 1.

Также даны начальные условия для дифференциальных уравнений:

$$\mathrm{x}(0) = \dot{x}(0) = \theta(0) = 0, \qquad \dot{\theta}(0) = 4$$

Исходя из экспериментальных наблюдений за изменением координаты x, следует оценить параметр жесткости K. (Экспериментальные наблюдения за изменением x приведены в таблице 1.)

Таблица 1 – Экспериментальные значения x

t	x
0	0
0.4	0.303
0.8	-0.465
1.2	0.592
1.6	-0.409
2.0	0.164
2.4	0.180

1. Выполнение задания

1.1 Вычисление параметров дифференциальных уравнений:

Для нахождения значения L необходимо вычислить интеграл:

$$L = 0.5836896 * \int_0^1 e^x \,\mathrm{dx}$$

Для этого воспользуюсь подпрограммой QUANC8():

```
//Функция e^2 для метода calcL
double funL(double x) {
    return exp(x);              //Подинтегральная функция
}

//Вычисление параметра L (начальной длины пружины)
double calcL() {
    double result;              //Переменная для результата
    double errest, flag;        //Параметры для quanc8
    int nofun;                  //Параметр для quanc8
    quanc8(funL, A: 0.0, B: 1.0, ABSERR: 1e-3, RELERR: 1e-3, &result, &errest, &nofun, &flag)
                                //Вычиление интеграла с помощью quanc8
    result *= 0.5836896;        //Вычисления результата
    return result;
}
```

Рисунок 1.1 – Код для нахождения параметра L

1.2 Решение системы дифференциальных уравнений:

$$\begin{cases} \ddot{x} + \dfrac{K}{M}x + g(1 - cos\theta) - (L + x)\left(\dot{\theta}\right)^2 = 0 \\ \ddot{\theta} + \dfrac{g}{L + x}sin\theta + \dfrac{2}{L + x}\dot{x}\dot{\theta} = 0 \end{cases}$$

Проведу следующую замену переменных:

$$x = z_1, x' = z_2, x'' = z_3, \theta = z_4, \theta' = z_5, \theta'' = z_6$$

После получу систему дифференциальных уравнений:

$$\begin{cases} z'_5 = \dfrac{-g}{L + z_1}\sin(z_4) - \dfrac{2}{L + z_1}z_2 z_5 \\ z'_4 = z_5 \\ z'_2 = -\left(\dfrac{K}{M}\right)z_1 - g(1 - \cos(z_4)) + (L + z_1)z_5^2 \\ z_1' = z_2 \end{cases}$$

1.3 Нахождение минимального значения функции:

С помощью подпрограммы Fmin буду подставлять в функцию, решающую систему дифференциальных уравнений значения в интервале [36, 46], чтобы найти минимальное отклонение от экспериментальных данных (экспериментальные данные приведены в таблице 1). Минимальное значение функции буду вычислять при помощи среднеквадратичного критерия близости F(p):

$$F(p) = \left| \sum_{i=0}^{S} [x_{\text{эксп}}^{(i)}(t_k) - x^{(i)}(t_k)]^2 \right| \to min$$

Где $x^{(i)}(t_k)$ – решение системы, полученное с помощью подпрограммы RKF45.

```
//Поиск минимума функции
typedef double(*Func)(double x);
double Minimum(double Start, double End, Func f, double Step) {
    double xMin = Start;
    double fMin = f(Start);
    while(Start < End) {
        if(f(Start) < fMin) {
            xMin = Start;
            fMin = f(Start);
        }
        Start +=Step;
    }
    return xMin;
}
```

Рисунок 1.3 – Код метода нахождения минимального F(p)

С помощью подпрограммы RKF45 на промежутке [0.0, 2.4] с шагом 0.4 и подаваемым параметром K на промежутке [36, 46] с шагом 0.01 будет решена система дифференциальных уравнений, а также будет найдено минимальное значение F(p):

```
//RKF45 для нахождения минимального значения K
double RKF(double Parameter)  {
    T = 0.0;
    Tout = 2.4;
    double sum = 0.0;
    K = Parameter;
    double X[] = {0, 0, 0, 4};
    double rightBorder = Tout;
    Tout = T;
    for (int i = 0; Tout <= rightBorder; i++) {
        RKF45(equations, n, X, &T, &Tout, &RELERR, &ABSERR, &IFLAG, WORK, IWORK);
        Tout += h;
        sum += (X[0] - actualX[i]) * (X[0] - actualX[i]);
    }
    IWORK[60];
    IFLAG = 1;
    WORK[20];
    return sum;
}
```

Рисунок 1.4 – Код решение системы дифференциальных уравнений

2. Результаты работы

2.1 Результаты работы программы:

В ходе выполнения данного курсового проекта была реализована программа на языке Си, решающая задачу, описанную во Введении. Код данной программы представлен в Приложении 1.

При решении данной задачи использовались:

Для quanc8: глобальная и локальная погрешности = 1e-14

Для rkf45: глобальная и локальная погрешности = 1e-9

Для Fmin: погрешность = 1e-3

Результаты работы программы представлены на рисунке 2.1

Tout – координата параметра x.

Calculated – вычисленные значения x при данной координате

Real – значения x, заданные условием

EPS – разность между вычисленным значением и экспериментальным.

```
Calculated parameter: L = 1.0029
Calculated parameter K = 39.482

Tout    Calculated      Real     EPS
0.0     0.000000        0.000   0.000000
0.4     0.302849        0.303   -0.000151
0.8     -0.466920       -0.465  -0.001920
1.2     0.593651        0.592   0.001651
1.6     -0.414142       -0.409  -0.005142
2.0     0.161142        0.164   -0.002858
2.4     0.187485        0.180   0.007485
```

Рисунок 2.1 – Результаты работы программы

2.2 Оценка погрешности результата:

Погрешность L хранится в переменной errest после окончания работы подпрограммы quanc8():

$$\varepsilon_C = 0.0000000000000011113$$

Верхняя граница погрешности F задается пользователем:

$$\varepsilon_f = 0.00000000000001$$

2.3 Влияние погрешности на результат измерений:

Сначала исследую устойчивость системы, путем изменения начального условия L на 1%:

```
Calculated parameter: L * 1.01 = 1.0130
Calculated parameter K = 40.562

Tout    Calculated       Real     EPS
0.0     0.000000         0.000    0.000000
0.4     0.299211         0.303    -0.003789
0.8     -0.473260        -0.465   -0.008260
1.2     0.598900         0.592    0.006900
1.6     -0.436973        -0.409   -0.027973
2.0     0.158847         0.164    -0.005153
2.4     0.200218         0.180    0.020218
```

Рисунок 2.2 – Результаты работы при L * 1.01

```
Calculated parameter: L * 0.99 = 0.9929
Calculated parameter K = 38.840

Tout    Calculated       Real     EPS
0.0     0.000000         0.000    0.000000
0.4     0.303163         0.303    0.000163
0.8     -0.459035        -0.465   0.005965
1.2     0.587846         0.592    -0.004154
1.6     -0.400492        -0.409   0.008508
2.0     0.172105         0.164    0.008105
2.4     0.161236         0.180    -0.018764
```

Рисунок 2.3 – Результаты работы при L * 0.99

Из полученных результатов можно сделать вывод, что система является устойчивой для изменения параметра L, ведь EPS = вычисленное значение – экспериментальное – мало для серьезного изменения данных. График, описывающий отсутствие значительных изменения приведен на рисунке 2.10 в разделе 2.4 – Графики.

Также была исследована зависимость от задания погрешности в подпрограмму rkf45:

```
Calculated parameter: L = 1.0029
Calculated parameter K = 39.688

Tout    Calculated        Real     EPS
0.0     0.000000          0.000   0.000000
0.4     0.301237          0.303   -0.001763
0.8     -0.465984         -0.465  -0.000984
1.2     0.592270          0.592   0.000270
1.6     -0.417729         -0.409  -0.008729
2.0     0.160395          0.164   -0.003605
2.4     0.186388          0.180   0.006388
```

Рис. 2.4 – Уменьшение локальной погрешности до 0.01

```
Calculated parameter: L = 1.0029
Calculated parameter K = 39.635

Tout    Calculated        Real     EPS
0.0     0.000000          0.000   0.000000
0.4     0.303391          0.303   0.000391
0.8     -0.468470         -0.465  -0.003470
1.2     0.595412          0.592   0.003412
1.6     -0.417788         -0.409  -0.008788
2.0     0.160383          0.164   -0.003617
2.4     0.192184          0.180   0.012184
```

Рис. 2.4 – Уменьшение локальной погрешности до 0.01

Графики, отражающие незначительное изменение параметров из-за изменения локальной и глобальной погрешности представлены на рисунке 2.11.

Также было исследовано вычисление параметра K в интервале [36, 46] с шагом 1, 0.1, 0.01, 0.001, чтобы убедиться, что параметр K можно найти с более высокой точностью, а также исходя из полученных данных можно сделать вывод, что при уменьшении шага будет и уменьшаться EPS:

```
Calculated parameter: L = 1.0029
Calculated parameter K = 39.000

Tout    Calculated        Real     EPS
0.0     0.000000          0.000   0.000000
0.4     0.306456          0.303   0.003456
0.8     -0.467749         -0.465  -0.002749
1.2     0.592923          0.592   0.000923
1.6     -0.400349         -0.409  0.008651
2.0     0.152679          0.164   -0.011321
2.4     0.191859          0.180   0.011859
```

Рис. 2.5 – K = 1

```
Calculated parameter: L = 1.0029
Calculated parameter K = 39.500

Tout    Calculated        Real     EPS
0.0     0.000000          0.000   0.000000
0.4     0.302714          0.303   -0.000286
0.8     -0.466885         -0.465  -0.001885
1.2     0.593672          0.592   0.001672
1.6     -0.414642         -0.409  -0.005642
2.0     0.161475          0.164   -0.002525
2.4     0.187268          0.180   0.007268
```

Рис. 2.6 – K = 0.1

```
Calculated parameter: L = 1.0029
Calculated parameter K = 39.480

Tout     Calculated          Real     EPS
0.0      0.000000            0.000    0.000000
0.4      0.302864            0.303    -0.000136
0.8      -0.466924           -0.465   -0.001924
1.2      0.593649            0.592    0.001649
1.6      -0.414086           -0.409   -0.005086
2.0      0.161106            0.164    -0.002894
2.4      0.187509            0.180    0.007509
```

Рис. 2.7 – K = 0.01

```
Calculated parameter: L = 1.0029
Calculated parameter K = 39.482

Tout     Calculated          Real     EPS
0.0      0.000000            0.000    0.000000
0.4      0.302849            0.303    -0.000151
0.8      -0.466920           -0.465   -0.001920
1.2      0.593651            0.592    0.001651
1.6      -0.414142           -0.409   -0.005142
2.0      0.161142            0.164    -0.002858
2.4      0.187485            0.180    0.007485
```

Рис. 2.8 – K = 0.001

2.4 Графики

На рисунке 2.9 приведен график полученных значений, при погрешности описанной в пункте 2.1, а также экспериментальных значений. Исходя из графика можно сказать, что полученные данные практически равны начальным экспериментальным данным.

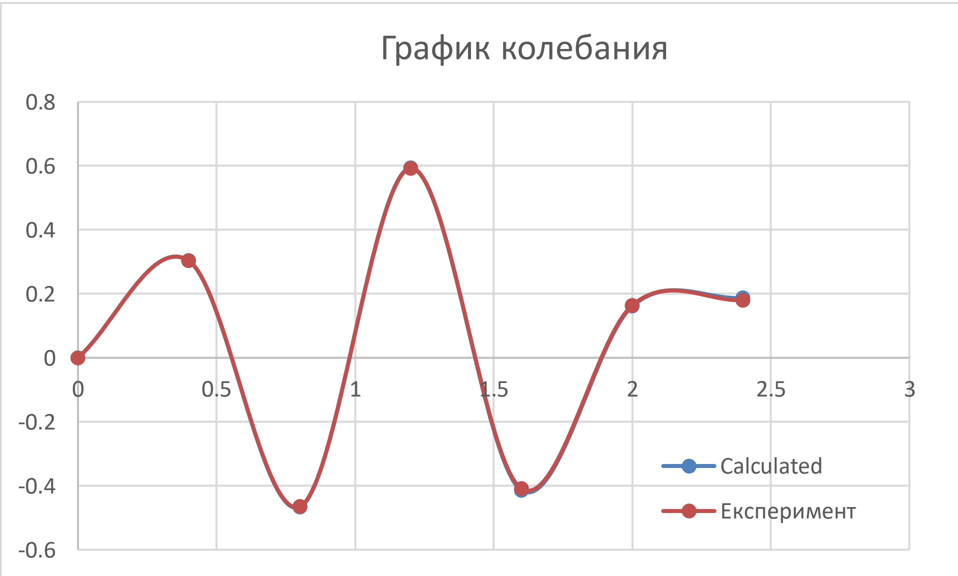

Рисунок 2.9 – График колебаний, полученный вычислением программы

На рисунке 2.10 отражена устойчивость системы от изменения начального условия на 1%. По графику можно сделать вывод, что изменение параметра L незначительно, но влияет на изменение параметра, что можно наблюдать при L * 1.01 в точке 1.6.

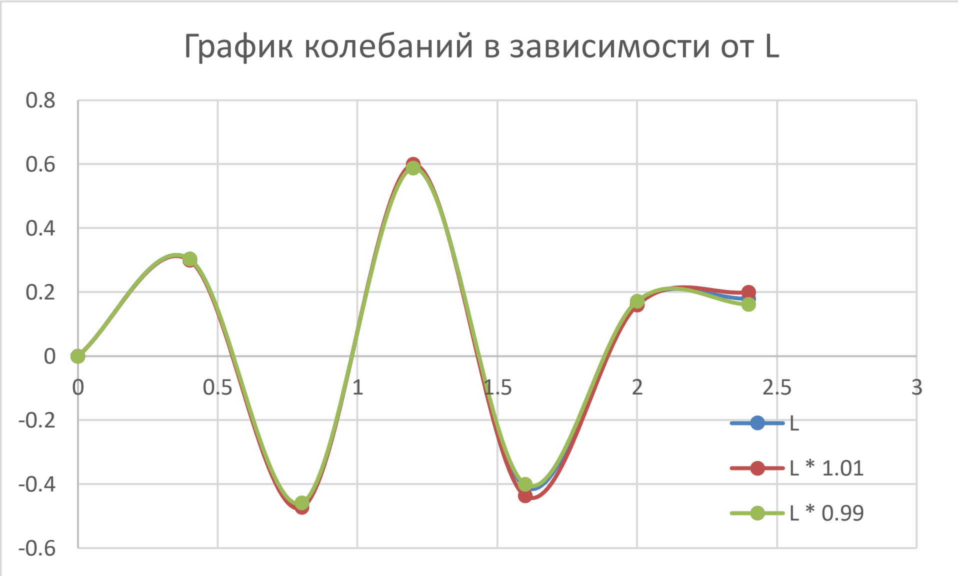

Рисунок 2.10 – График колебаний при изменении L на 1%

На рисунке 2.11 представлен график зависимости колебаний от изменения локальной и глобальной погрешности при решении подпрограммой rkf45.

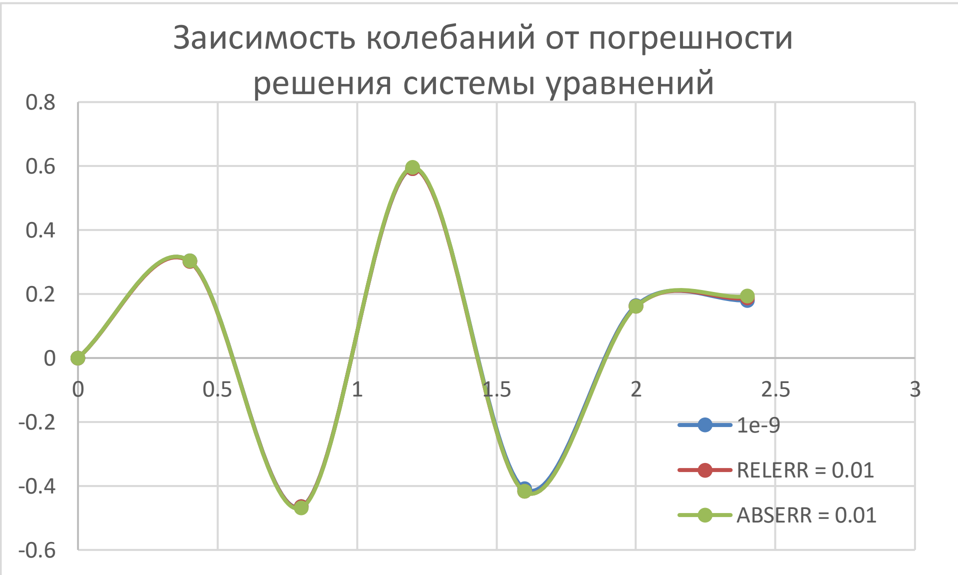

Рисунок 2.11 – Зависимость от погрешности при решении дифференциальных уравнений

Заключительным шагом является моделирование движения маятника переменной длины с разным шагом поиска F(p). Результаты приведены на рисунке 2.12:

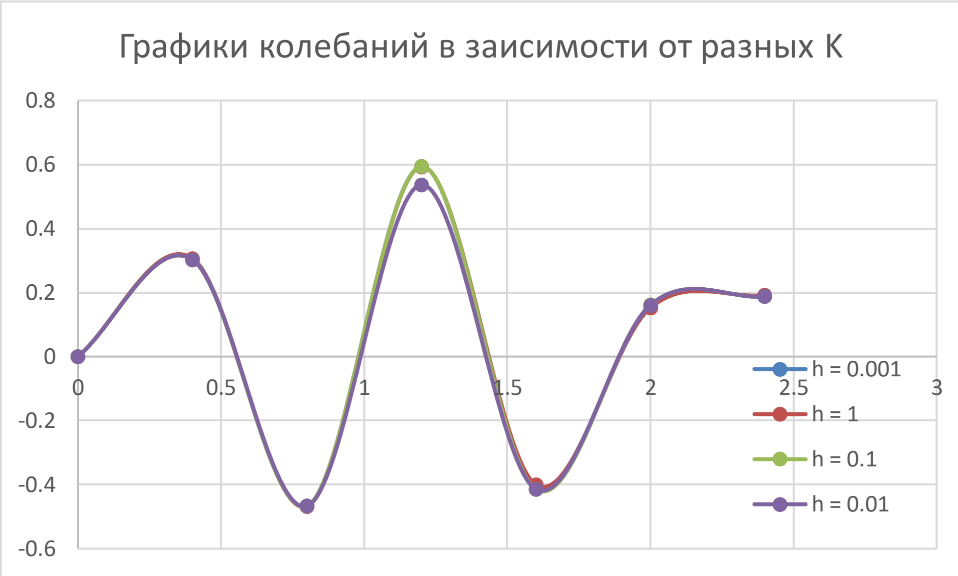

Рисунок 2.12 – Графики колебаний в зависимости от разных К

По рисунку 2.12 можно сделать вывод, что при уменьшении шага поиска минимального значения, будет и увеличиваться точность вычисления параметра жесткости и параметров x.

3. Заключение

В ходе выполнения данного курсового проекта была решена оценки значения жесткости пружины K у маятника переменной длины. Ответом является значение K = 39.482. Причем погрешность полученных значений X при сравнении с экспериментальными составляет порядка 0.001. Для проверки истинности данного результата были проведены исследования устойчивости системы дифференциальных уравнений. При изменении значения K на 1% вычисленные значения X имели погрешность 0.01. А при изменении значения L на 1% полученное значение K отличалось примерно на 2.1% от истинного. Также было произведено исследование на зависимость от глобальной и локальной погрешности подпрограммы rkf45, а также получены результаты для различных шагов поиска ссреднеквадратичного критерия близости.

В процессе выполнения работы были закреплены навыки взаимодействия с подпрограммой quanc8(), rkf45() и Fmin.

Список использованных источников

1. Устинов С.М., Зимницкий В.А. Вычислительная математика. – СПб.: БХВ-Петербург, 2009. – 126 с.: ил. – (Учебное пособие).

2. Эккель Б., Эллисон Ч. Философия C++. Практическое программирование. – СПб.: Питер, 2004. – 442 с.

3. Лафоре Р. Объектно-ориентированное программирование в C++. – СПб.: Питер, 2004. – 98 с.

Приложение 1

```
#include <iostream>
#include "quanc8.h"
#include "rkf45.h"
#include <cmath>


//Функиця e^2 для метода calcL
double funL(double x) {
    return exp(x);                  //Подинтегральная функция
}

//Вычисление параметра L (начальной длины пружины)
double calcL() {
    double result;                  //Переменная для результата
    double errest, flag;            //Параметры для quanc8
    int nofun;                      //Параметр для quanc8
    quanc8(funL, 0.0, 1.0, 1e-14, 1e-14, &result, &errest, &nofun, &flag);
                                    //Вычиление интеграла с помощью quanc8
    result *= 0.5836896;            //Вычисления результата
    return result;
}

//ПАРАМЕТРЫ:
const double g = 9.81;              //Гравитационная постоянная
const double M = 1;                 //Масса маятника
double K;                           //Исследуемый параметр K
double L = calcL();                 //Начальная длина пружины L

//Параметры для решения
const int n = 4;                    //Количество уравнений в системе
double X[] = {0.0, 0.0, 0.0, 4.0}; //массив из двух уравнений
double actualX[7] = {0.0, 0.303, -0.465, 0.592, -0.409, 0.164, 0.180};
                                    //Экспериментальные значения
double T = 0.0;                     //Начала интервала
double Tout = 2.4;                  //Конец интервала
double h = 0.4;                     //Шаг

//Параметры для решения методом RKF45
double RELERR = 0.000000001;              //относительная погрешность
double ABSERR = 0.000000001;              //абсоляютная погрешность
int IFLAG = 1;                      //
double WORK[60];                    // - Параметры для RKF45
int IWORK[20];                      //

//Система дифференциальных уравнений
void equations(double t, double *y, double *dy) {
    dy[0] = y[1];
    dy[1] = -K * y[0] / M - g * (1 - cos(y[2])) + (L + y[0]) * y[3] * y[3];
    dy[2] = y[3];
    dy[3] = -g * sin(y[2]) / (L + y[0]) - 2 * y[1] * y[3] / (L + y[0]);
}

//RKF45 для нахождения минимального значения K
double RKF(double Parameter)  {
    T = 0.0;
    Tout = 2.4;
    double sum = 0.0;
    K = Parameter;
    double X[] = {0, 0, 0, 4};
    double rightBorder = Tout;
    Tout = T;
    for (int i = 0; Tout <= rightBorder; i++) {
        RKF45(equations, n, X, &T, &Tout, &RELERR, &ABSERR, &IFLAG, WORK,
```

```
IWORK);
        Tout += h;
        sum += (X[0] - actualX[i]) * (X[0] - actualX[i]);
    }
    IWORK[60];
    IFLAG = 1;
    WORK[20];
    return sum;
}

//RKF45 для вычисления точек и разницы между экспериментальными значениями
double RKF_end(double Parameter)  {
    T = 0.0;
    Tout = 2.4;
    double sum = 0.0;
    K = Parameter;
    double X[] = {0, 0, 0, 4};
    double rightBorder = Tout;
    Tout = T;
    printf("\nTout\tCalculated\t Real\t  EPS\n");
    for (int i = 0; Tout <= rightBorder; i++) {
        RKF45(equations, n, X, &T, &Tout, &RELERR, &ABSERR, &IFLAG, WORK,
IWORK);
            printf("%.1f\t%.6f\t%.3f\t%.6f\n", Tout, X[0], actualX[i],
fabs(X[0] - actualX[i]));
        Tout += h;
        sum += (X[0] - actualX[i]) * (X[0] - actualX[i]);
    }
    IWORK[60];
    IFLAG = 1;
    WORK[20];
    return sum;
}

//Поиск минимума функции
typedef double(*Func)(double x);
double Minimum(double Start, double End, Func f, double Step) {
    double xMin = Start;
    double fMin = f(Start);
    while(Start < End) {
        if(f(Start) < fMin) {
            xMin = Start;
            fMin = f(Start);
        }
        Start +=Step;
    }
    return xMin;
}

int main() {
    printf("Calculated parameter: L = %.4f\n", L);
    K = Minimum(36, 46, RKF, 0.001);
    printf("Calculated parameter K = %.3f\n", K);
    RKF_end(K);
    return 0;
}
```

Рисунок 1 – Код программы