

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Курсовая работа

по теме: «Анализ колебаний маятника переменной длины.»

Выполнил
студент гр. в3530904/00030

В.С. Баганов

Руководитель
профессор, к.т.н.

С.М. Устинов

«_____» _____ 202__ г.

Санкт-Петербург
2023

Содержание

1. Постановка задачи	3
2. План расчетной части курсовой работы	5
3. DECOMP и SOLVE	6
4. QUANC8	7
5. ZEROIN	8
6. Значения L, M, K и A, B, C, D	10
7. RKF45. Код программы C++	11
8. Результат программы RKF45	14
9. Результат программы с возмущением по параметру L на 1%	16
10. Сравнение возмущения $L = 1.01 \%$ (на 1 %)	18
11. Сравнение возмущения $L = 0.99 \%$ (на 1 %)	20
12. Оценка точности и влияния на точность погрешности исходных данных	22
13. Выводы	22

1. Постановка задачи

Анализ колебаний маятника переменной длины.

Маятник массой M , роль стержня которого выполняет пружина с жесткостью K , совершает сложные колебания относительно положения равновесия.

L - начальная длина пружины,

K - жесткость пружины,

x - удлинение пружины относительно положения равновесия, θ

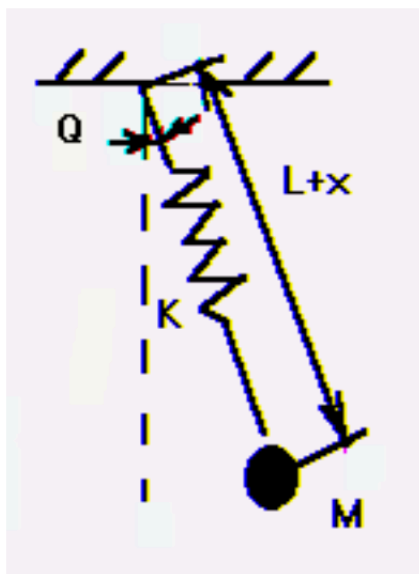


Рисунок 1.1. Схема маятника

Дифференциальное уравнение движения имеет вид:

$$\ddot{x} + \frac{K}{M}x + g(1 - \cos\theta) - (L+x)\dot{\theta}^2 = 0;$$

$$\ddot{\theta} + \frac{g}{L+x}\sin\theta + \frac{2}{L+x}\dot{x}\dot{\theta} = 0;$$

$g=9,81$

Начальные условия: $X(0)=A$; $\dot{x}(0)=B$; $\dot{\theta}(0)=C$; $\theta(0) = D$;

Построить графики изменения x и θ

$t \in [0, 1]$

и оценить погрешность результата и влияние на точность погрешности исходных данных.

Значения L , M , K , а также A , B , C , D задаются преподавателем.

Вариант N 14В.

Значения L, M, K являются решением системы уравнений:

$$\begin{cases} 46L - 24M - 42K = -1626.08 \\ -24L + 16M + 18K = 698.32 \\ -42L + 18M + 49K = 1898.76 \end{cases}$$

$$A = \left(\int_{0.5}^{0.8} \sqrt{1+x^3} dx - 0.34031875 \right)^4;$$

$$D = (x^* + 0.53727448)^4; \text{ где } x^* - \text{отрицательный корень уравнения: } e^{-x} + x^2 = 2.$$

$$C=4, B=0.$$

2. План расчетной части курсовой работы

Для решения дифференциальных уравнений движения маятника, необходимо получить коэффициенты L , M , K , а также A , B , C , D с помощью программ DECOMP и SOLVE (решим систему уравнений), QUANC8 (посчитаем определенный интеграл заданной функции), ZEROIN (найдем отрицательный корень уравнения). Далее мы должны преобразовать дифференциальные уравнения движения второго порядка к 4 уравнениям, выразив значения производных, и в программе RK45 получить значение X и ϑ и их производных на промежутке времени от 0 до 4 используя начальные значения A , B , C , D .

Вторая часть курсовой работы будет посвящена анализу изменения результатов при возмущении параметра длины L на 1% (увеличение на 0,01 и уменьшение на 0,99). Для удобства анализа данных, представим таблицы сравнения разности значений по модулю.

Прежде чем начать расчет, оценим как будет двигаться маятник. K - жесткость пружины, $L+x$ - длина маятника, причем величина x будет принимать как положительные, так и отрицательные значения (пружина будет сжиматься и растягиваться). M - масса груза, который будет качаться в разные стороны, отклоняясь в разные стороны, относительно центральной оси на угол ϑ Тетта, следовательно ϑ Тетта будет принимать отрицательные и положительные значения. С течением времени, значения ϑ Тетта и x (растяжение) должны затухать.

3. DECOMP и SOLVE

$$\begin{cases} 46L - 24M - 42K = -1626.08 \\ -24L + 16M + 18K = 698.32 \\ -42L + 18M + 49K = 1898.76 \end{cases}$$

```

1 use Environment
2 implicit none
3 character(*), parameter :: input_file = "../data/input.txt",
   ↪ output_file = "output.txt"
4 integer :: In = 0, Out = 0, N = 0
5 integer, allocatable :: IPVT(:)
6 real :: COND = 0
7 real(R_), allocatable :: WORK(:), A(:, :), B(:)
8
9 open (file=input_file, newunit=In)
10 read (In, *) N
11 allocate (A(N,N))
12 read (In, *) A
13 close (In)
14
15 open (file=output_file, encoding=E_, newunit=Out)
16 write (Out, "(a)") "Source matrix"
17 write (Out, "("//N//"f10.3)") A
18 close (Out)
19
20 allocate (IPVT(N), WORK(N), B(N))
21
22 call DECOMP(N, N, A, COND, IPVT, WORK)
23 B = [-1626.08, 698.32, 1898.76]
24 call SOLVE(N, N, A, B, IPVT)
25
26 open (file=output_file, encoding=E_, newunit=Out, position='append')
27 write (Out, *)
28 write (Out, "("//N//"f10.4)") B
29 close (Out)

```

Листинг 1: Код и результат программы DECOMP и SOLVE

```
Press ENTER or type command to continue
cd ./bin; ./app;
cat bin/output.txt
Source matrix
    46.000    -24.000    -42.000
   -24.000     16.000     18.000
   -42.000     18.000     49.000

    1.0000     1.0000    39.2400

Press ENTER or type command to continue
```

4. QUANC8

$$A = \left(\int_{0.5}^{0.8} \sqrt{1+x^3} dx - 0.34031875 \right)^4$$

```
1 use Environment
2 implicit none
3 character(*), parameter :: output_file = "output.txt"
4 real(R_) :: a = 0.5, b = 0.8, relerr = 0.001, abserr
5 integer :: Out = 0, nofun
6
7 call QUANC8(fun, a, b, abserr, relerr, result, errest, nofun, flag)
8 A = (result-0.34031875)**4
9
10 open (file=output_file, encoding=E_, newunit=Out)
11 write(Out, '(a, f19.15)') «A= ", A
12 close (Out)
13
14 contains
15 real function fun(x)
16 real x
17 intent(in) x
18 fun = SQRT(1 + x * x * x)
19 end function fun
```

Листинг 2: Код и результат программы QUANC8

```
Press ENTER or type command to continue
cd ./bin; ./app;
cat bin/output.txt
A = 0.00000000000000000000
Press ENTER or type command to continue
```

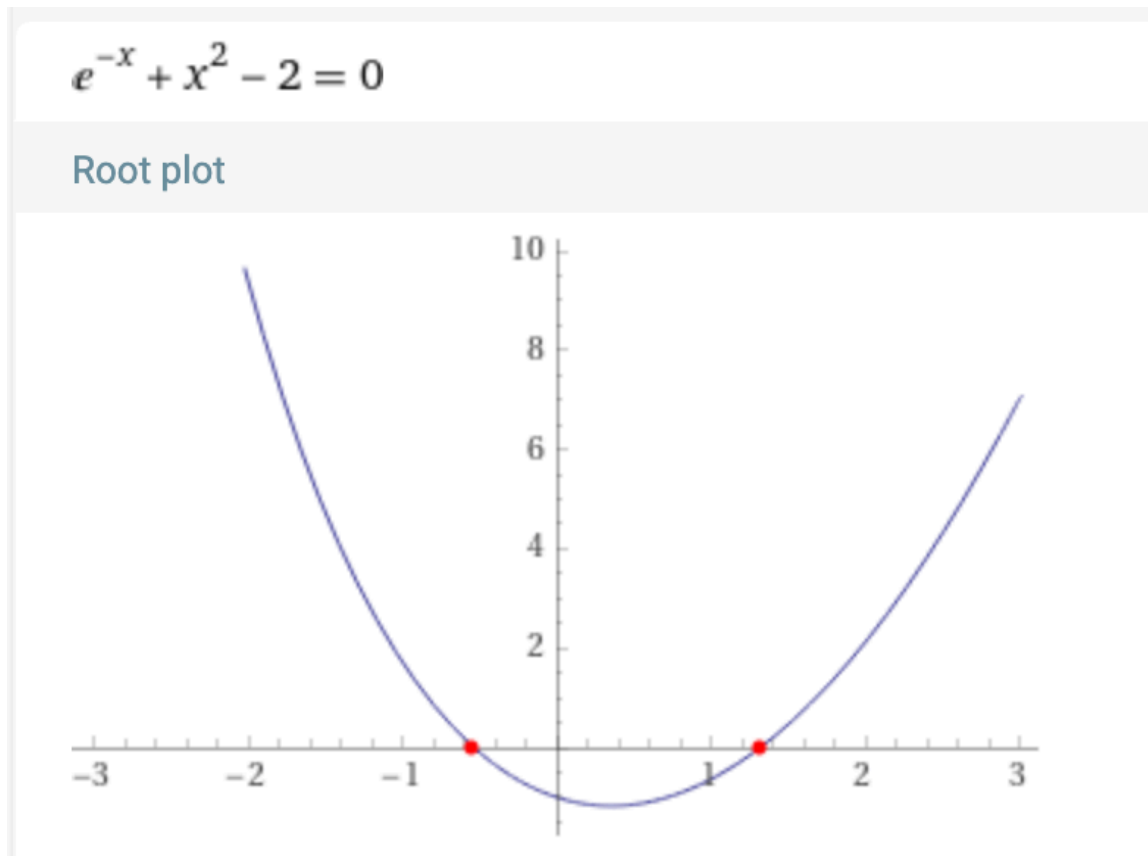
5. ZEROIN

$$D = (x^* + 0.53727448)^4$$

где x^* - отрицательный корень уравнения: $e^{-x} + x^2 - 2 = 0$

Построив график функции, делаем вывод, что нас интересует промежуток от -1 до 0.

В программе задаем данный промежуток $AX = -1.0$, $BX = 0.0$.




```

1  use Environment
2
3  implicit none
4  character(*), parameter      :: output_file = "output.txt"
5  real(R_)                    :: AX = -1.0, BX = 0.0, Z, TOL = 1.0E-7,
   ↪ D
6  integer                     :: Out = 0
7  real, external              :: ZEROIN
8
9  Z = ZEROIN(AX, BX, fun, TOL)
10 D = (Z + 0.53727448)**4
11
12 open (file=output_file, encoding=E_, newunit=Out)
13   write(Out, '(a, f16.13)') "D = ", D
14 close (Out)
15
16 contains
17   real function fun(x)
18     real x
19     intent(in) x
20     fun = exp(-x)+(x*x) - 2
21   end function fun

```

Листинг 3: Код и результат программы ZEROIN

```

Press ENTER or type command to continue
cd ./bin; ./app;
cat bin/output.txt
D =  0.0000000000000000
Press ENTER or type command to continue

```

6. Значения L, M, K и A, B, C, D

Значения $L = 1$, $M = 1$, $K = 39,24$ получены с помощью DECOMP и SOLVE.

Начальные условия:

$X(0) = A = 0$; получен с помощью QUANC8.

$\dot{x}(0) = B = 0$; дано по условию.

$\dot{\theta}(0) = C = 4$; дано по условию.

$\theta(0) = D = 0$; получен с помощью ZEROIN.

7. RKF45. Код программы C++

```
1  #include "rkf45.h"
2  #include <string.h>
3  #include <fstream>
4
5  using namespace std;
6
7  std::ofstream out("out.txt");
8
9  char* cmathmsg(int routine, int flag)
10 {
11     static char s[64];
12     switch (routine)
13     {
14         case RKFINIT_C:
15             switch (flag)
16             {
17                 case 0: strcpy(s, "rkfinit() : normal return");
18                     break;
19                 case 1: strcpy(s,
20                     ↪ "rkfinit() : could not allocate workspace");
21                     break;
22                 case 2: strcpy(s, "rkfinit() : illegal value for n");
23                     break;
24                 default: strcpy(s, "rkfinit() : no such error");
25             };
26             break;
27         case RKF45_C:
28             switch (flag)
29             {
30                 case -2: strcpy(s, "rkf45() : normal return");
31                     break;
32                 case 2: strcpy(s, "rkf45() : normal return");
33                     break;
34                 case 3: strcpy(s, "rkf45() : relerr too small");
35                     break;
36                 case 4: strcpy(s, "rkf45() : too many steps");
37                     break;
38                 case 5: strcpy(s,
39                     ↪ "rkf45() : abserr needs to be nonzero");
40                     break;
41                 case 6: strcpy(s,
42                     ↪ "rkf45() : stepsize has become too small");
43                     break;
44                 case 7: strcpy(s, "rkf45() : rkf45 is inefficient");
45                     break;
46                 case 8: strcpy(s, "rkf45() : invalid user input");
47                     break;
48                 default: strcpy(s, "rkf45() : no such error");
49             };
50             break;
51         default: strcpy(s, "CMATH : no such routine");
52     }
53     return (s);
54 }
```

```

54 double L = 1.00;
55 //double L = 1.01;
56 double M = 1;
57 double K = 39.24;
58 double g = 9.81;
59
60 int f(int n, double t, double x[3], double dxdt[4])
61 {
62
63
64
65     //x[0] - x
66     //x[1] - тетта
67     //x[2] - dx/dt
68     //x[3] - dТетта/dt
69
70     dxdt[0]=x[2]; //dx[0]/dt=x[2] 1 -производная X-переменная
71     dxdt[1]=x[3]; //dx[1]/dt=x[3] 1 -производная Тетта-переменная
72     dxdt[2] = - (K / M) * x[0] - g * (1- cos(x[1])) + (L + x[0]) * (x[3]
73     ↪ * x[3]); // x''
74     dxdt[3] = -(g * sin(x[1]) + 2 * x[2] * x[3]) / (L + x[0]);
75     ↪ // ?
76     return (0);
77 }
78
79 int main()
80 {
81
82     out <<
83     ↪ ("\\n=====RK45:=====\\n\\n");
84     double h, relerr, abserr, t1, t2, L, K, M, g;
85     int n, flag, nfe, maxfe, fail, step;
86     double x[4], yp[4];
87     n = 4;
88     flag = 1;
89     maxfe = 5000;
90     relerr = 1.0e-4;
91     abserr = 1.0e-4;
92     rkfinit(n, &fail);
93
94     out << ("%s\\n\\n", cmathmsg(RKFINIT_C, fail));
95
96     if (fail == 0)
97     {
98         x[0] = 0.0; // - x
99         x[1] = 0.0; // - ? тетта
100        x[2] = 0.0; // - dx/dt
101        x[3] = 4.0; // - dТетта/dt ?
102
103
104        out << ("\\nstep      t      x[0]  [1]  x'[2]  [3]\\n")
105        << ("-----\\n");
106
107        for (step = 0; step ≤ 160; ++step)
108        {
109            t2 = step * 0.025; //t+h

```

```

110     t1 = t2 - 0.025;
111     rkf45(f, n, x, yp, &t1, t2, &relerr, abserr,
112           &h, &nfe, maxfe, &flag);
113     out << std::fixed << step << "\t\t" << t1 << "\t" << x[0] <<
114         ↪ "\t" << x[1] &
115                                         << "\t" << x[2]
116                                         ↪ << "\t" << x[3]
117                                         ↪ << "\n";
118
119     if (flag ≠ 2)
120     {
121         out << ("%s\n", cmathmsg(RKF45_C, flag));
122         break;
123     }
124
125     rkfend();
126     out << ("\n%s\n", cmathmsg(RKF45_C, flag)) << "\n"
127         << "nfe: " << nfe << "\n"
128         << "step size: " << h << "\n";
129
130     return 0;
131 }

```

Листинг 4: RKF45. Код программы C++

8. Результат программы RKF45

```

1 =====RKF45=====
2
3 rkfinit() : normal return
4 step t x[0] θ[1] x'[2] θ'[3]
5 -----
6 0 0.000000 0.004970 0.099568 0.395174 3.948401
7 1 0.025000 0.019523 0.196620 0.762219 3.801181
8 2 0.050000 0.042642 0.288996 1.076814 3.578466
9 3 0.075000 0.072775 0.375134 1.320999 3.306671
10 4 0.100000 0.108013 0.454139 1.483970 3.011826
11 5 0.125000 0.146260 0.525711 1.561435 2.715075
12 6 0.150000 0.185379 0.590000 1.554272 2.430997
13 7 0.175000 0.223306 0.647438 1.467099 2.168004
14 8 0.200000 0.258126 0.698607 1.307104 1.929717
15 9 0.225000 0.288128 0.744134 1.083190 1.716490
16 10 0.250000 0.311837 0.784627 0.805404 1.526680
17 11 0.275000 0.328039 0.820640 0.484543 1.357543
18 12 0.300000 0.335799 0.852649 0.131876 1.205800
19 13 0.325000 0.334464 0.881046 -0.241082 1.067937
20 14 0.350000 0.323672 0.906131 -0.622779 0.940328
21 15 0.375000 0.303347 0.928116 -1.001812 0.819226
22 16 0.400000 0.273696 0.947114 -1.367120 0.700668
23 17 0.425000 0.235194 0.963135 -1.708160 0.580292
24 18 0.450000 0.188572 0.976072 -2.015058 0.453094
25 19 0.475000 0.134801 0.985683 -2.278737 0.313087
26 20 0.500000 0.075064 0.991559 -2.490985 0.152889
27 21 0.525000 0.010742 0.993083 -2.644450 -0.036779
28 22 0.550000 -0.056613 0.989378 -2.732530 -0.267614
29 23 0.575000 -0.125289 0.979239 -2.749123 -0.554092
30 24 0.600000 -0.193423 0.961068 -2.688212 -0.913210
31 25 0.625000 -0.259000 0.932825 -2.543313 -1.362619
32 26 0.650000 -0.319827 0.892072 -2.306956 -1.915304
33 27 0.675000 -0.373515 0.836215 -1.970769 -2.568219
34 28 0.700000 -0.417470 0.763124 -1.527266 -3.283751
35 29 0.725000 -0.448967 0.672248 -0.974863 -3.970874
36 30 0.750000 -0.465404 0.565989 -0.326105 -4.488831
37 31 0.775000 -0.464743 0.450389 0.385619 -4.699993
38 32 0.800000 -0.446023 0.334009 1.109320 -4.553255
39 33 0.825000 -0.409628 0.225139 1.791370 -4.119362
40 34 0.850000 -0.357146 0.129266 2.391014 -3.537885
41 35 0.875000 -0.290958 0.048393 2.885457 -2.936770
42 36 0.900000 -0.213827 -0.018062 3.265683 -2.392778
43 37 0.925000 -0.128638 -0.071975 3.530218 -1.935373
44 38 0.950000 -0.038265 -0.115565 3.680932 -1.565714
45 39 0.975000 0.054488 -0.150900 3.721214 -1.272661
46 40 1.000000 0.146912 -0.179717 3.655508 -1.041898
47 41 1.025000 0.236426 -0.203401 3.489396 -0.859992
48 42 1.050000 0.320602 -0.223029 3.229796 -0.715721
49 43 1.075000 0.397206 -0.239426 2.885098 -0.600195
50 44 1.100000 0.464230 -0.253221 2.465197 -0.506535
51 45 1.125000 0.519933 -0.264891 1.981425 -0.429454
52 46 1.150000 0.562874 -0.274797 1.446393 -0.364868
53 47 1.175000 0.591940 -0.283211 0.873772 -0.309576
54 48 1.200000 0.606370 -0.290332 0.278013 -0.260999
55 49 1.225000 0.605772 -0.296299 -0.325962 -0.216979
56 50 1.250000 0.590129 -0.301203 -0.923098 -0.175603
57 51 1.275000 0.559798 -0.305086 -1.498546 -0.135057

```

```

58 52 1.300000 0.515502 -0.307947 -2.038010 -0.093476
59 53 1.325000 0.458309 -0.309734 -2.528080 -0.048782
60 54 1.350000 0.389611 -0.310340 -2.956533 0.001504
61 55 1.375000 0.311086 -0.309587 -3.312610 0.060512
62 56 1.400000 0.224660 -0.307208 -3.587239 0.132344
63 57 1.425000 0.132464 -0.302817 -3.773200 0.222522
64 58 1.450000 0.036784 -0.295867 -3.865214 0.338545
65 59 1.475000 -0.059986 -0.285591 -3.859941 0.490576
66 60 1.500000 -0.155389 -0.270927 -3.755849 0.692067
67 61 1.525000 -0.246955 -0.250436 -3.552965 0.959846
68 62 1.550000 -0.332224 -0.222227 -3.252497 1.312409
69 63 1.575000 -0.408783 -0.183986 -2.856506 1.763858
70 64 1.600000 -0.474278 -0.133237 -2.368041 2.309641
71 65 1.625000 -0.526459 -0.068093 -1.792681 2.902454
72 66 1.650000 -0.563285 0.011360 -1.142413 3.430087
73 67 1.675000 -0.583149 0.101518 -0.440838 3.731517
74 68 1.700000 -0.585201 0.194965 0.275647 3.681301
75 69 1.725000 -0.569586 0.282672 0.966041 3.286596
76 70 1.750000 -0.537393 0.357552 1.597641 2.683745
77 71 1.775000 -0.490350 0.416516 2.152153 2.037577
78 72 1.800000 -0.430490 0.460009 2.622344 1.458348
79 73 1.825000 -0.359955 0.490359 3.005915 0.988802
80 74 1.850000 -0.280926 0.510365 3.301672 0.628644
81 75 1.875000 -0.195613 0.522552 3.508371 0.359709
82 76 1.900000 -0.106258 0.528931 3.624938 0.160482
83 77 1.925000 -0.015121 0.530998 3.651046 0.012041
84 78 1.950000 0.075547 0.529831 3.587635 -0.100453
85 79 1.975000 0.163536 0.526182 3.437254 -0.188069
86 80 2.000000 0.246721 0.520567 3.204237 -0.258906
87 81 2.025000 0.323110 0.513327 2.894751 -0.318897
88 82 2.050000 0.390887 0.504675 2.516737 -0.372466
89 83 2.075000 0.448455 0.494729 2.079780 -0.423027
90 84 2.100000 0.494476 0.483526 1.594914 -0.473355
91 85 2.125000 0.527904 0.471043 1.074390 -0.525863
92 86 2.150000 0.548009 0.457196 0.531403 -0.582816
93 87 2.175000 0.554402 0.441846 -0.020192 -0.646518
94 88 2.200000 0.547046 0.424793 -0.566187 -0.719474
95 89 2.225000 0.526258 0.405771 -1.092311 -0.804551
96 90 2.250000 0.492713 0.384436 -1.584520 -0.905149
97 91 2.275000 0.447428 0.360350 -2.029249 -1.025374
98 92 2.300000 0.391753 0.332963 -2.413633 -1.170218
99 93 2.325000 0.327350 0.301584 -2.725672 -1.345688
100 94 2.350000 0.256165 0.265365 -2.954344 -1.558814
101 95 2.375000 0.180411 0.223265 -3.089673 -1.817347
102 96 2.400000 0.102533 0.174056 -3.122799 -2.128809
103 97 2.425000 0.025186 0.116339 -3.046169 -2.498335
104 98 2.450000 -0.048812 0.048663 -2.854100 -2.924584
105 99 2.475000 -0.116535 -0.030249 -2.544111 -3.393214
106 100 2.500000 -0.175060 -0.121065 -2.119489 -3.868979
107 101 2.525000 -0.221658 -0.223256 -1.593149 -4.291050
108 102 2.550000 -0.254085 -0.334509 -0.991463 -4.580298
109 103 2.550000 -0.254085 -0.334509 -0.991463 -4.580298
110 rkf45() : rkf45 is inefficientrkf45() : rkf45 is
    ↪ inefficient
111 nfe: 619

```

Листинг 5: Результат программы

9. Результат программы с возмущением по параметру L на 1%

```

1  =====RKF45 с возмущением по L = 1.01 =====
2
3  rkfinit() : normal return
4  step      t      x[0]      θ[1]      x'[2]      θ'[3]
5  -----
6  0      0.000000      0.005019      0.099569      0.399138      3.948521
7  1      0.025000      0.019720      0.196627      0.769935      3.801637
8  2      0.050000      0.043074      0.289021      1.087879      3.579423
9  3      0.075000      0.073520      0.375191      1.334860      3.308226
10 4      0.100000      0.109133      0.454242      1.499967      3.014021
11 5      0.125000      0.147798      0.525878      1.578847      2.717919
12 6      0.150000      0.187363      0.590245      1.572354      2.434485
13 7      0.175000      0.225743      0.647779      1.485115      2.172133
14 8      0.200000      0.261005      0.699059      1.324348      1.934497
15 9      0.225000      0.291422      0.744713      1.099006      1.721949
16 10     0.250000      0.315501      0.785352      0.819196      1.532868
17 11     0.275000      0.332018      0.821530      0.495789      1.364539
18 12     0.300000      0.340022      0.853725      0.140136      1.213712
19 13     0.325000      0.338853      0.882332      -0.236164      1.076908
20 14     0.350000      0.328139      0.907658      -0.621469      0.950545
21 15     0.375000      0.307800      0.929916      -1.004286      0.830931
22 16     0.400000      0.278038      0.949228      -1.373468      0.714169
23 17     0.425000      0.239329      0.965613      -1.718388      0.595989
24 18     0.450000      0.192404      0.978975      -2.029105      0.471499
25 19     0.475000      0.138235      0.989087      -2.296484      0.334861
26 20     0.500000      0.078010      0.995558      -2.512281      0.178883
27 21     0.525000      0.013113      0.997795      -2.669142      -0.005479
28 22     0.550000      -0.054901      0.994953      -2.760505      -0.229643
29 23     0.575000      -0.124317      0.985864      -2.780366      -0.507797
30 24     0.600000      -0.193275      0.968973      -2.722875      -0.856736
31 25     0.625000      -0.259764      0.942288      -2.581785      -1.294224
32 26     0.650000      -0.321608      0.903405      -2.349914      -1.834125
33 27     0.675000      -0.376435      0.849726      -2.019103      -2.475757
34 28     0.700000      -0.421674      0.779030      -1.581750      -3.186006
35 29     0.725000      -0.454611      0.690546      -1.035417      -3.879855
36 30     0.750000      -0.472620      0.586311      -0.390817      -4.420509
37 31     0.775000      -0.473586      0.471995      0.320970      -4.666907
38 32     0.800000      -0.456423      0.355975      1.050409      -4.556651
39 33     0.825000      -0.421373      0.246662      1.743502      -4.149209
40 34     0.850000      -0.369910      0.149857      2.357796      -3.580424
41 35     0.875000      -0.304349      0.067878      2.868639      -2.981337
42 36     0.900000      -0.227428      0.000347      3.265685      -2.433694
43 37     0.925000      -0.142033      -0.054520      3.546667      -1.970714
44 38     0.950000      -0.051051      -0.098924      3.712987      -1.595562
45 39     0.975000      0.042686      -0.134945      3.767711      -1.297892
46 40     1.000000      0.136439      -0.164346      3.715016      -1.063563
47 41     1.025000      0.227586      -0.188538      3.560243      -0.879059
48 42     1.050000      0.313656      -0.208619      3.310094      -0.733005
49 43     1.075000      0.392364      -0.225433      2.972766      -0.616361
50 44     1.100000      0.461649      -0.239623      2.558005      -0.522130
51 45     1.125000      0.519712      -0.251681      2.077033      -0.444941
52 46     1.150000      0.565053      -0.261978      1.542406      -0.380658
53 47     1.175000      0.596499      -0.270794      0.967789      -0.326059
54 48     1.200000      0.613230      -0.278340      0.367683      -0.278572
55 49     1.225000      0.614796      -0.284764      -0.242885      -0.236071

```


56	50	1.250000	0.601125	-0.290169	-0.848705	-0.196707
57	51	1.275000	0.572526	-0.294611	-1.434727	-0.158762
58	52	1.300000	0.529675	-0.298105	-1.986413	-0.120511
59	53	1.325000	0.473606	-0.300619	-2.490077	-0.080074
60	54	1.350000	0.405676	-0.302072	-2.933200	-0.035242
61	55	1.375000	0.327543	-0.302322	-3.304713	0.016736
62	56	1.400000	0.241116	-0.301146	-3.595243	0.079449
63	57	1.425000	0.148519	-0.298220	-3.797290	0.157711
64	58	1.450000	0.052035	-0.293077	-3.905348	0.258077
65	59	1.475000	-0.045936	-0.285058	-3.915932	0.389485
66	60	1.500000	-0.142936	-0.273245	-3.827495	0.563917
67	61	1.525000	-0.236487	-0.256377	-3.640226	0.796737
68	62	1.550000	-0.324136	-0.232773	-3.355715	1.105751
69	63	1.575000	-0.403484	-0.200318	-2.976588	1.506932
70	64	1.600000	-0.472206	-0.156624	-2.506397	2.003346
71	65	1.625000	-0.528088	-0.099595	-1.950482	2.564418
72	66	1.650000	-0.569096	-0.028582	-1.318771	3.101755
73	67	1.675000	-0.593551	0.054082	-0.630393	3.469736
74	68	1.700000	-0.600401	0.142292	0.083524	3.527953
75	69	1.725000	-0.589483	0.227563	0.784635	3.241249
76	70	1.750000	-0.561565	0.302309	1.438741	2.711181
77	71	1.775000	-0.518124	0.362438	2.023884	2.097582
78	72	1.800000	-0.461035	0.407547	2.529496	1.524648
79	73	1.825000	-0.392350	0.439504	2.951057	1.050090
80	74	1.850000	-0.314204	0.460949	3.285931	0.682554
81	75	1.875000	-0.228795	0.474404	3.531782	0.407583
82	76	1.900000	-0.138374	0.481928	3.686557	0.204580
83	77	1.925000	-0.045237	0.485073	3.748998	0.054458
84	78	1.950000	0.048306	0.484963	3.719146	-0.058063
85	79	1.975000	0.139965	0.482387	3.598714	-0.144451
86	80	2.000000	0.227516	0.477888	3.391278	-0.213087
87	81	2.025000	0.308849	0.471829	3.102342	-0.270069
88	82	2.050000	0.382015	0.464444	2.739303	-0.319887
89	83	2.075000	0.445273	0.455866	2.311325	-0.365950
90	84	2.100000	0.497129	0.446155	1.829159	-0.410965
91	85	2.125000	0.536379	0.435307	1.304905	-0.457224
92	86	2.150000	0.562134	0.423266	0.751749	-0.506824
93	87	2.175000	0.573842	0.409921	0.183663	-0.561846
94	88	2.200000	0.571313	0.395110	-0.384904	-0.624513
95	89	2.225000	0.554715	0.378611	-0.939340	-0.697347
96	90	2.250000	0.524584	0.360133	-1.465177	-0.783337
97	91	2.275000	0.481812	0.339304	-1.948375	-0.886113
98	92	2.300000	0.427632	0.315650	-2.375569	-1.010154
99	93	2.325000	0.363604	0.288574	-2.734275	-1.160984
100	94	2.350000	0.291585	0.257322	-3.013028	-1.345334
101	95	2.375000	0.213706	0.220962	-3.201456	-1.571151
102	96	2.400000	0.132342	0.178346	-3.290312	-1.847210
103	97	2.425000	0.050088	0.128112	-3.271503	-2.181905
104	98	2.450000	-0.030279	0.068716	-3.138289	-2.580460
105	99	2.475000	-0.105833	-0.001421	-2.885956	-3.039638
106	100	2.500000	-0.173573	-0.083612	-2.513483	-3.539451
107	101	2.525000	-0.230551	-0.178357	-2.026716	-4.033819
108	102	2.550000	-0.274096	-0.284633	-1.442892	-4.447217
109	103	2.575000	-0.302153	-0.399281	-0.794413	-4.688847
110	104	2.575000	-0.302153	-0.399281	-0.794413	-4.688847
111	rkf45() : rkf45 is inefficientrkf45() : rkf45 is inefficient					
112	nfe:	625				
113	step size:	0.053288				

10. Сравнение возмущения $L = 1.01 \%$ (на 1%)

====RKF45 сравнение возмущения 1.01 % (на 1 %) =====					
step	t	$\Delta x[0]$	$\Delta \theta[1]$	$\Delta x'[2]$	$\Delta \theta'[3]$
0	0.000000	0,000049	0,00000100	0,003964	0,00012
1	0.025000	0,000197	0,00000700	0,007716	0,000456
2	0.050000	0,000432	0,000024999	0,011065	0,000957
3	0.075000	0,000745	0,00005699	0,013861	0,001555
4	0.100000	0,001120	0,000103	0,015997	0,002195
5	0.125000	0,001538	0,000167	0,017412	0,002844
6	0.150000	0,001984	0,000245	0,018082	0,003488
7	0.175000	0,002437	0,000341	0,018016	0,004129
8	0.200000	0,002879	0,000452	0,017244	0,00478
9	0.225000	0,003294	0,000579	0,015816	0,005459
10	0.250000	0,003664	0,000725	0,013792	0,006188
11	0.275000	0,003979	0,000890	0,011246	0,006996
12	0.300000	0,004223	0,001076	0,00826	0,007912
13	0.325000	0,004389	0,001286	0,004918	0,008971
14	0.350000	0,004467	0,001527	0,00131	0,010217
15	0.375000	0,004453	0,001800	0,002474	0,011705
16	0.400000	0,004342	0,002114	0,006348	0,013501
17	0.425000	0,004135	0,002478	0,010228	0,015697
18	0.450000	0,003832	0,002903	0,014047	0,018405
19	0.475000	0,003434	0,003404	0,017747	0,021774
20	0.500000	0,002946	0,003999	0,021296	0,025994
21	0.525000	0,002371	0,004712	0,024692	0,0313
22	0.550000	0,001712	0,005575	0,027975	0,037971
23	0.575000	0,000972	0,006625	0,031243	0,046295
24	0.600000	0,000148	0,007905	0,034663	0,056474
25	0.625000	0,000764	0,009463	0,038472	0,068395
26	0.650000	0,001781	0,011333	0,042958	0,081179
27	0.675000	0,002920	0,013511	0,048334	0,092462
28	0.700000	0,004204	0,015906	0,054484	0,097745
29	0.725000	0,005644	0,018298	0,060554	0,091019
30	0.750000	0,007216	0,020322	0,064712	0,068322
31	0.775000	0,008843	0,021606	0,064649	0,033086
32	0.800000	0,010400	0,021966	0,058911	0,003396
33	0.825000	0,011745	0,021523	0,047868	0,029847
34	0.850000	0,012764	0,020591	0,033218	0,042539
35	0.875000	0,013391	0,019485	0,016818	0,044567
36	0.900000	0,013601	0,018409	2E06	0,040916
37	0.925000	0,013395	0,017455	0,016449	0,035341
38	0.950000	0,012786	0,016641	0,032055	0,029848
39	0.975000	0,011802	0,015955	0,046497	0,025231
40	1.000000	0,010473	0,015371	0,059508	0,021665
41	1.025000	0,008840	0,014863	0,070847	0,019067
42	1.050000	0,006946	0,014410	0,080298	0,017284
43	1.075000	0,004842	0,013993	0,087668	0,016166
44	1.100000	0,002581	0,013598	0,092808	0,015595
45	1.125000	0,000221	0,013210	0,095608	0,015487
46	1.150000	0,002179	0,012819	0,096013	0,01579
47	1.175000	0,004559	0,012417	0,094017	0,016483
48	1.200000	0,006860	0,011992	0,089670	0,017573
49	1.225000	0,009024	0,011535	0,083077	0,019092
50	1.250000	0,010996	0,011034	0,074393	0,021104
51	1.275000	0,012728	0,010475	0,063819	0,023705
52	1.300000	0,014173	0,009842	0,051597	0,027035
53	1.325000	0,015297	0,009115	0,038003	0,031292

58	54	1.350000	0,016065	0,008268	0,023333	0,036746
59	55	1.375000	0,016457	0,007265	0,007897	0,043776
60	56	1.400000	0,016456	0,006062	0,008004	0,052895
61	57	1.425000	0,016055	0,004597	0,024090	0,064811
62	58	1.450000	0,015251	0,002790	0,040134	0,080468
63	59	1.475000	0,014050	0,000533	0,055991	0,101091
64	60	1.500000	0,012453	0,002318	0,071646	0,12815
65	61	1.525000	0,010468	0,005941	0,087261	0,163109
66	62	1.550000	0,008088	0,010546	0,103218	0,206658
67	63	1.575000	0,005299	0,016332	0,120082	0,256926
68	64	1.600000	0,002072	0,023387	0,138356	0,306295
69	65	1.625000	0,001629	0,031502	0,157801	0,338036
70	66	1.650000	0,005811	0,039942	0,176358	0,328332
71	67	1.675000	0,010402	0,047436	0,189555	0,261781
72	68	1.700000	0,015200	0,052673	0,192123	0,153348
73	69	1.725000	0,019897	0,055109	0,181406	0,045347
74	70	1.750000	0,024172	0,055243	0,158900	0,027436
75	71	1.775000	0,027774	0,054078	0,128269	0,060005
76	72	1.800000	0,030545	0,052462	0,092848	0,0663
77	73	1.825000	0,032395	0,050855	0,054858	0,061288
78	74	1.850000	0,033278	0,049416	0,015741	0,05391
79	75	1.875000	0,033182	0,048148	0,023411	0,047874
80	76	1.900000	0,032116	0,047003	0,061619	0,044098
81	77	1.925000	0,030116	0,045925	0,097952	0,042417
82	78	1.950000	0,027241	0,044868	0,131511	0,04239
83	79	1.975000	0,023571	0,043795	0,161460	0,043618
84	80	2.000000	0,019205	0,042679	0,187041	0,045819
85	81	2.025000	0,014261	0,041498	0,207591	0,048828
86	82	2.050000	0,008872	0,040231	0,222566	0,052579
87	83	2.075000	0,003182	0,038863	0,231545	0,057077
88	84	2.100000	0,002653	0,037371	0,234245	0,06239
89	85	2.125000	0,008475	0,035736	0,230515	0,068639
90	86	2.150000	0,014125	0,033930	0,220346	0,075992
91	87	2.175000	0,019440	0,031925	0,203855	0,084672
92	88	2.200000	0,024267	0,029683	0,181283	0,094961
93	89	2.225000	0,028457	0,027160	0,152971	0,107204
94	90	2.250000	0,031871	0,024303	0,119343	0,121812
95	91	2.275000	0,034384	0,021046	0,080874	0,139261
96	92	2.300000	0,035879	0,017313	0,038064	0,160064
97	93	2.325000	0,036254	0,013010	0,008603	0,184704
98	94	2.350000	0,035420	0,008043	0,058684	0,21348
99	95	2.375000	0,033295	0,002303	0,111783	0,246196
100	96	2.400000	0,029809	0,004290	0,167513	0,281599
101	97	2.425000	0,024902	0,011773	0,225334	0,31643
102	98	2.450000	0,018533	0,020053	0,284189	0,344124
103	99	2.475000	0,010702	0,028828	0,341845	0,353576
104	100	2.500000	0,001487	0,037453	0,393994	0,329528
105	101	2.525000	0,008893	0,044899	0,433567	0,257231
106	102	2.550000	0,020011	0,049876	0,451429	0,133081
107	103	2.575000	0,048068	0,064772	0,197050	0,108549
108	-----					
109						
110	Среднее значение	0,01259375	0,01999576	0,09369419	0,0810815	

Листинг 6: Сравнение возмущения $L = 1.01 \%$ (на 1%)

11. Сравнение возмущения $L = 0.99\%$ (на 1%)

=====RKF45 сравнение возмущения 0.99 % (на 1 %) =====					
step	t	$\Delta x[0]$	$\Delta \theta[1]$	$\Delta x'[2]$	$\Delta \theta'[3]$
0	0.000000	0,000049	0,000001	0,003963	0,000122
1	0.025000	0,000196	0,0000007	0,007715	0,000466
2	0.050000	0,000432	0,0000259	0,011065	0,000976
3	0.075000	0,000745	0,0000580	0,01386	0,001586
4	0.100000	0,001120	0,000105	0,015996	0,00224
5	0.125000	0,001539	0,000169	0,01741	0,002902
6	0.150000	0,001984	0,000250	0,01808	0,003559
7	0.175000	0,002437	0,000347	0,018012	0,004212
8	0.200000	0,002879	0,000461	0,017238	0,004877
9	0.225000	0,003293	0,000592	0,015805	0,005571
10	0.250000	0,003664	0,000740	0,013778	0,006318
11	0.275000	0,003978	0,000908	0,011228	0,007145
12	0.300000	0,004222	0,001098	0,008235	0,008082
13	0.325000	0,004387	0,001314	0,004886	0,009166
14	0.350000	0,004464	0,001558	0,001269	0,010443
15	0.375000	0,004448	0,001837	0,002526	0,011966
16	0.400000	0,004337	0,002159	0,006412	0,013809
17	0.425000	0,004128	0,002532	0,010310	0,01606
18	0.450000	0,003821	0,002966	0,014148	0,018838
19	0.475000	0,003421	0,003479	0,017874	0,022294
20	0.500000	0,002929	0,004089	0,021455	0,026621
21	0.525000	0,002350	0,004820	0,024894	0,03206
22	0.550000	0,001685	0,005703	0,028234	0,038889
23	0.575000	0,000938	0,006778	0,031577	0,047389
24	0.600000	0,000106	0,008088	0,035094	0,057738
25	0.625000	0,000820	0,009679	0,039029	0,069762
26	0.650000	0,001852	0,011582	0,043657	0,082448
27	0.675000	0,003010	0,013786	0,049166	0,093221
28	0.700000	0,004317	0,016188	0,055368	0,097343
29	0.725000	0,005777	0,018546	0,061297	0,088786
30	0.750000	0,007363	0,020491	0,065016	0,064154
31	0.775000	0,008990	0,021655	0,064235	0,027903
32	0.800000	0,010527	0,021888	0,057692	0,008107
33	0.825000	0,011831	0,021344	0,045974	0,0331
34	0.850000	0,012797	0,020350	0,030894	0,044272
35	0.875000	0,013363	0,019216	0,014285	0,045235
36	0.900000	0,013509	0,018131	0,002579	0,041004
37	0.925000	0,013238	0,017178	0,018956	0,03518
38	0.950000	0,012569	0,016370	0,034409	0,029616
39	0.975000	0,011528	0,015689	0,048631	0,02501
40	1.000000	0,010150	0,015110	0,061368	0,021488
41	1.025000	0,008474	0,014607	0,072385	0,018947
42	1.050000	0,006547	0,014157	0,081473	0,017223
43	1.075000	0,004418	0,013741	0,088451	0,016165
44	1.100000	0,002143	0,013343	0,093176	0,015654
45	1.125000	0,000221	0,012954	0,095551	0,015609
46	1.150000	0,002615	0,012560	0,095526	0,015981
47	1.175000	0,004978	0,012152	0,093108	0,016753
48	1.200000	0,007251	0,011719	0,088357	0,017933
49	1.225000	0,009377	0,011251	0,081384	0,019562
50	1.250000	0,011303	0,010736	0,072352	0,021708
51	1.275000	0,012979	0,010161	0,061467	0,024476
52	1.300000	0,014363	0,009506	0,048979	0,028019
53	1.325000	0,015417	0,008752	0,035162	0,032546

58	54	1.350000	0,016112	0,007868	0,020313	0,038353
59	55	1.375000	0,016426	0,006819	0,004736	0,04584
60	56	1.400000	0,016345	0,005557	0,011278	0,055562
61	57	1.425000	0,015861	0,004017	0,027467	0,068264
62	58	1.450000	0,014973	0,002111	0,043627	0,08494
63	59	1.475000	0,013681	0,000274	0,059656	0,106844
64	60	1.500000	0,011991	0,003286	0,075589	0,135407
65	61	1.525000	0,009900	0,007110	0,091658	0,171865
66	62	1.550000	0,007403	0,011946	0,108307	0,216246
67	63	1.575000	0,004475	0,017962	0,126106	0,265183
68	64	1.600000	0,001085	0,025162	0,145370	0,308444
69	65	1.625000	0,002799	0,033175	0,165305	0,326439
70	66	1.650000	0,007162	0,041074	0,182902	0,295859
71	67	1.675000	0,011882	0,047493	0,192924	0,209526
72	68	1.700000	0,016703	0,051305	0,190481	0,094479
73	69	1.725000	0,021291	0,052362	0,174473	0,003953
74	70	1.750000	0,025337	0,051472	0,147799	0,05981
75	71	1.775000	0,028626	0,049693	0,114517	0,077504
76	72	1.800000	0,031034	0,047770	0,077699	0,074169
77	73	1.825000	0,032498	0,046041	0,039209	0,063849
78	74	1.850000	0,032991	0,044575	0,000238	0,05376
79	75	1.875000	0,032513	0,043330	0,038265	0,046379
80	76	1.900000	0,031088	0,042231	0,075406	0,041956
81	77	1.925000	0,028762	0,041211	0,110307	0,039989
82	78	1.950000	0,025600	0,040216	0,142122	0,039885
83	79	1.975000	0,021688	0,039206	0,170058	0,041166
84	80	2.000000	0,017134	0,038149	0,193404	0,043511
85	81	2.025000	0,012061	0,037022	0,211550	0,046736
86	82	2.050000	0,006604	0,035806	0,224002	0,050763
87	83	2.075000	0,000912	0,034477	0,230395	0,055594
88	84	2.100000	0,004862	0,033018	0,230494	0,061298
89	85	2.125000	0,010560	0,031404	0,224204	0,067996
90	86	2.150000	0,016020	0,029608	0,211560	0,075866
91	87	2.175000	0,021086	0,027599	0,192724	0,085136
92	88	2.200000	0,025607	0,025338	0,167967	0,09609
93	89	2.225000	0,029438	0,022778	0,137659	0,109069
94	90	2.250000	0,032447	0,019865	0,102236	0,124461
95	91	2.275000	0,034511	0,016532	0,062184	0,142687
96	92	2.300000	0,035521	0,012703	0,017998	0,164138
97	93	2.325000	0,035381	0,008296	0,029832	0,18906
98	94	2.350000	0,034004	0,003222	0,080837	0,217314
99	95	2.375000	0,031317	0,002590	0,134538	0,247953
100	96	2.400000	0,027259	0,009176	0,190324	0,278512
101	97	2.425000	0,021791	0,016475	0,247186	0,304018
102	98	2.450000	0,014907	0,024265	0,303220	0,31596
103	99	2.475000	0,006666	0,032059	0,354915	0,302215
104	100	2.500000	0,002756	0,039051	0,396498	0,249789
105	101	2.525000	0,013010	0,044167	0,420169	0,152123
106	102	2.550000	0,023549	0,046354	0,418153	0,018757
107	103	2.575000	0,016815	0,161088	1,023011	0,03717
108	-----					
109						
110	Среднее значение	0,0120935	0,01997143	0,100114067	0,075215587	

Листинг 7: Сравнение возмущения $L = 0.99 \%$ (на 1%)

12. Оценка точности и влияния на точность погрешности исходных данных

Благодаря программе RKF45, а именно ее возможности автоматически менять шаг интегрирования, мы получили данные с высокой точностью. Мы провели расчет с возмущением по параметру L (длина маятника) на 1 % увеличения ($L = 1.01$) и уменьшения ($L = 0.99$). Получив и сравнив данные, среднее значение разности по модулю :

при $L = 1$ и $L = 1.01$ получили:

Среднее значение

$$\Delta x [0] = 0,01259375$$

$$\Delta \vartheta [1] = 0,01999576$$

$$\Delta x' [2] = 0,09369419$$

$$\Delta \vartheta' [3] = 0,081081$$

при $L = 1$ и $L = 0.99$ получили:

Среднее значение

$$\Delta x [0] = 0,0120935$$

$$\Delta \vartheta [1] = 0,01997143$$

$$\Delta x' [2] = 0,100114067$$

$$\Delta \vartheta' [3] = 0,075215587$$

13. Выводы

Можно сделать вывод, при возмущении параметров на 1%, среднее отклонение по результатам $x [0]$ и $\vartheta [1]$ будет около 1%, благодаря тому, что мы использовали программу RKF45, которая снижает накапливаемую погрешность