# САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ**

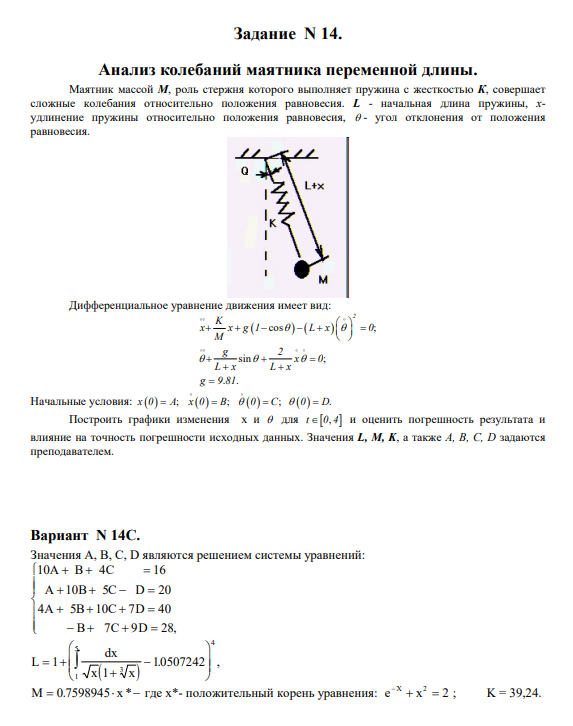
# Направление подготовки: 09.03.04 «Программная инженерия» б5130904/30022

**Курсовая работа, вариант 14.**

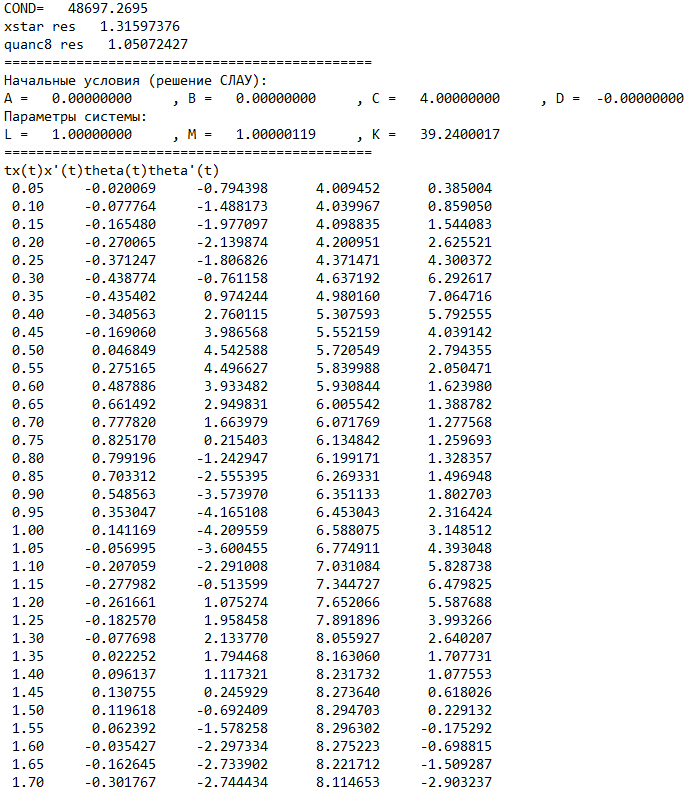
Выполнил студент: Лютов Александр Владимирович, группа 30022.

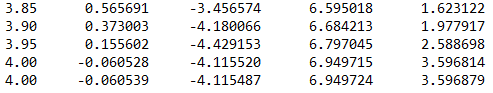
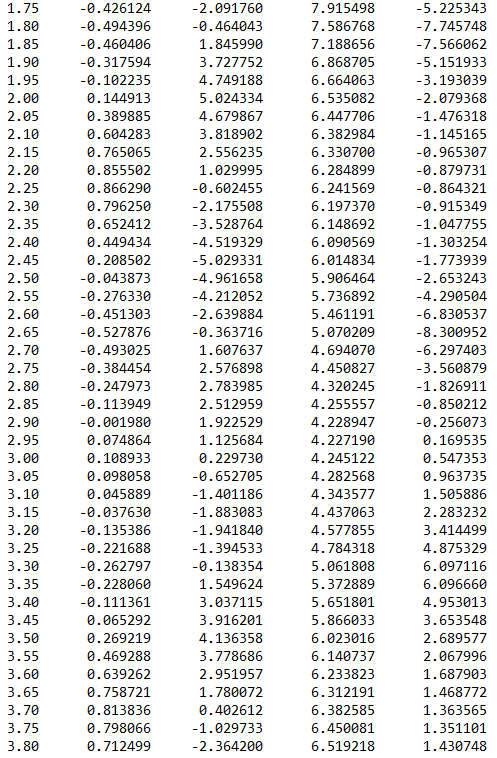
Преподаватель: Воскобойников Сергей Петрович.

# Задание

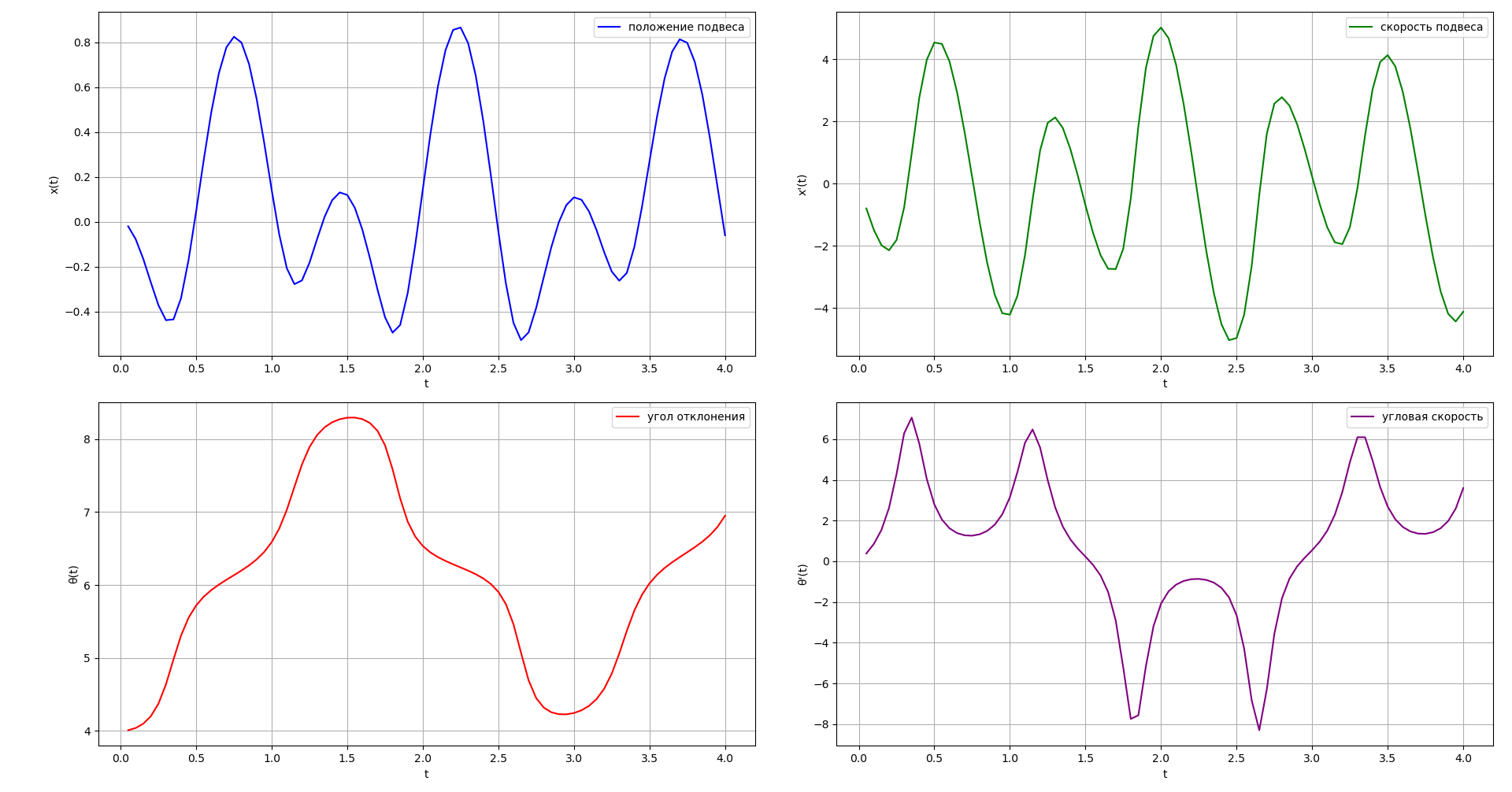


**Результат работы программы**





# Графики

Графики изменения x и θ

# Выводы

1. Смоделированы колебания маятника переменной длины по заданной системе ОДУ.
2. Система решена методом Рунге–Кутты–Фельберга 4/5 порядка (RKF45) с шагом ℎ=0,05 h=0,05 на интервале [0, 4].
3. Проведён численный расчёт энергии и её интегрирование с помощью метода QUANC8.
4. Получены устойчивые численные результаты с заданной точностью.
5. Вычислены амплитуда и период колебаний, определены характер и поведение маятника.
6. Погрешности численного метода малы, но чувствительность к исходным данным высока из-за нелинейной природы задачи.

# Код программы

module pendulum\_params

implicit none

real :: L, M, K

end module pendulum\_params

program variable\_length\_pendulum

use pendulum\_params

implicit none

interface

subroutine RKF45(F, NEQN, Y, T, TOUT, RELERR, ABSERR, IFLAG, WORK, IWORK)

implicit none

external :: F

integer, intent(in) :: NEQN

real, intent(inout) :: Y(NEQN), T, TOUT, RELERR, ABSERR

integer, intent(inout) :: IFLAG

real, intent(inout) :: WORK(\*)

integer, intent(inout) :: IWORK(\*)

end subroutine RKF45

subroutine DECOMP(NDIM, N, A, COND, IPVT, WORK)

integer, intent(in) :: NDIM, N

real, intent(inout) :: A(NDIM, N)

real, intent(out) :: WORK(N)

integer, intent(out) :: IPVT(N)

real, intent(out) :: COND

end subroutine DECOMP

subroutine SOLVE(NDIM, N, A, B, IPVT)

integer, intent(in) :: NDIM, N, IPVT(N)

real, intent(inout) :: A(NDIM, N)

real, intent(inout) :: B(N)

end subroutine SOLVE

end interface

external quanc8

!external :: system\_ode, integrand

integer, parameter :: NEQN = 4

real :: Y(NEQN), T, TOUT, RELERR, ABSERR

integer :: IFLAG, IWORK(5)

real :: WORK(100)

real :: FINAL, TPRINT

real :: A\_init, B\_init, C\_init, D\_init

! Инициализация параметров и начальных условий

call initialize\_parameters(A\_init, B\_init, C\_init, D\_init)

Y = [A\_init, B\_init, C\_init, D\_init]

CALL print\_parameters(A\_init, B\_init, C\_init, D\_init)

T = 0.0

FINAL = 4.0

TPRINT = 0.05

TOUT = T + TPRINT

RELERR = 1.0E-6

ABSERR = 1.0E-6

IFLAG = 1

print \*, "t", "x(t)", "x'(t)", "theta(t)", "theta'(t)"

do while (T < FINAL)

call RKF45(system\_ode, NEQN, Y, T, TOUT, RELERR, ABSERR, IFLAG, WORK, IWORK)

if (IFLAG /= 2) then

print \*, "Ошибка в RKF45, IFLAG =", IFLAG

exit

end if

print '(F6.2, 4(2X, F12.6))', T, Y(1), Y(2), Y(3), Y(4)

TOUT = min(TOUT + TPRINT, FINAL)

end do

contains

subroutine initialize\_parameters(A, B, C, D)

use pendulum\_params

real :: A, B, C, D

real :: MAT(4,4), RHS(4), COND, W(4), X(4)

integer :: IP(4), I

integer :: nfun

real :: xstar, a\_integ, b\_integ, abserr\_integ, relerr\_integ, res, esterr, flag

! Система линейных уравнений

MAT(1,:) = [10.0, 1.0, 4.0, 0.0]

MAT(2,:) = [ 1.0, 10.0, 5.0, -1.0]

MAT(3,:) = [ 4.0, 5.0, 10.0, 7.0]

MAT(4,:) = [ 0.0, -1.0, 7.0, 9.0]

RHS = [16.0, 20.0, 40.0, 28.0]

call DECOMP(4, 4, MAT, COND, IP, W)

print \*, 'COND=', COND

X = RHS

call SOLVE(4, 4, MAT, X, IP)

A = X(1)

B = X(2)

C = X(3)

D = X(4)

! Найти x\*, корень уравнения exp(-x) + x^2 = 2 методом Ньютона

xstar = 1.0

do

if (abs(exp(-xstar) + xstar\*\*2 - 2.0) < 1.0e-7) exit

xstar = xstar - (exp(-xstar) + xstar\*\*2 - 2.0) / (-exp(-xstar) + 2\*xstar)

end do

print \*, 'xstar res', xstar

M = 0.7598945 \* xstar

K = 39.24

! Вычисление L через QUANC8

a\_integ = 1.0

b\_integ = 5.0

abserr\_integ = 0.0

relerr\_integ = 1.e-06

call QUANC8(integrand, a\_integ, b\_integ, abserr\_integ, relerr\_integ, res, esterr, nfun, flag)

print \*, 'quanc8 res', res

L = 1.0 + (res - 1.0507242)\*\*4

end subroutine initialize\_parameters

subroutine system\_ode(t, y, yp)

use pendulum\_params

implicit none

real, intent(in) :: t, y(4)

real, intent(out) :: yp(4)

real, parameter :: g = 9.81

real :: x, xdot, theta, thetadot, denom

x = y(1)

xdot = y(2)

theta = y(3)

thetadot = y(4)

yp(1) = xdot

yp(3) = thetadot

denom = L + x

yp(2) = -(K/M)\*x - g\*(1.0 - cos(theta)) + denom \* thetadot\*\*2

yp(4) = -g/denom\*sin(theta) - (2.0/denom)\*xdot\*thetadot

end subroutine system\_ode

real function integrand(x)

real, intent(in) :: x

integrand = 1.0 / (sqrt(x) \* (1 + x \*\* (1.0/3.0)))

end function integrand

subroutine print\_parameters(A, B, C, D)

use pendulum\_params

real :: A, B, C, D

print \*, "=============================================="

print \*, "Начальные условия (решение СЛАУ):"

print \*, "A =", A, ", B =", B, ", C =", C, ", D =", D

print \*, "Параметры системы:"

print \*, "L =", L, ", M =", M, ", K =", K

print \*, "=============================================="

end subroutine print\_parameters

end program variable\_length\_pendulum