ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«ДИНАМИКА МЕХАНИЧЕСКОЙ СИСТЕМЫ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ»**

**ВАРИАНТ ЗАДАНИЯ № 1**

Выполнил(а) студент группы М8О-203Б-23

Арусланов К.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

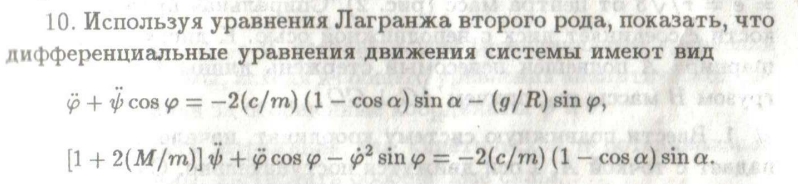
Москва, 2024

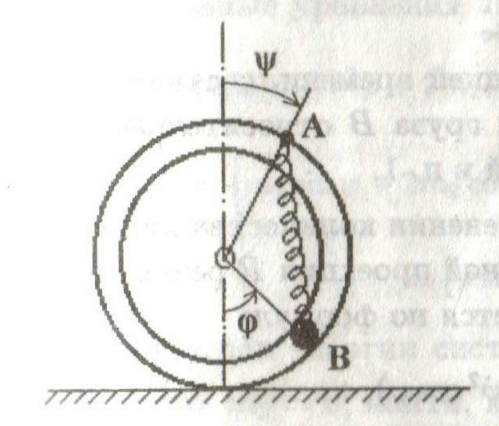
**Вариант № 1**

**Задание:**

Проинтегрировать систему дифференциальных уравнений движения системы с двумя степенями свободы с помощью средств Python. Построить анимацию движения системы, а также графики законов движения системы и указанных в задании реакций для разных случаев системы.

**Механическая система:**





**Текст программы**

Ниже представлен код, дополняющий вторую лабораторную работу

import numpy as np

from matplotlib.animation import FuncAnimation

from scipy.integrate import odeint

# --- Параметры системы ---

R = 2.0 # Радиус колеса (м)

c = 5.0 # Жесткость пружины (Н/м)

m = 1.0 # Масса шарика (кг)

M = 5.0 # Масса колеса

g = 9.81 # Ускорение свободного падения (м/с^2)

phi0 = np.pi/2 # Начальное отклонение угла фи (рад)

psi0 = 0 # Начальный угол пси (рад)

phi\_dot0 = 5.0 # Начальная угловая скорость фи (рад/с)

psi\_dot0 = 0.0 # Начальная угловая скорость пси (рад/с)

X0 = 0.0

ball\_radius = 0.2 # Радиус шарика (м)

# Параметры времени

Steps = 1001

t\_fin = 20

t = np.linspace(0, t\_fin, Steps)

dt = t[1]-t[0]

def odesys(y, t, M, m, c, g, R):

# y = [phi, psi, phi\_dot, psi\_dot]

phi = y[0]

psi = y[1]

phi\_dot = y[2]

psi\_dot = y[3]

alpha = (phi + psi)/2.0

# A \* [ddphi; ddpsi] = B

A11 = 1.0

A12 = np.cos(phi)

A21 = np.cos(phi)

A22 = 1.0 + 2.0\*(M/m)

det = A11\*A22 - A12\*A21

# Правая часть

# B1 и B2 из уравнений Лагранжа:

# phi'' + psi'' cos phi = -[2c/m(1−cos alpha) sin alpha + (g/R) sin phi]

# [1+2(M/m)] psi'' + phi'' cos phi - phi\_dot² sin phi = -2c/m(1−cos alpha) sin alpha

B1 = - (2\*c/m)\*(1 - np.cos(alpha))\*np.sin(alpha) - (g/R)\*np.sin(phi)

B2 = - (2\*c/m)\*(1 - np.cos(alpha))\*np.sin(alpha) + (phi\_dot\*\*2)\*np.sin(phi)

ddphi = (B1\*A22 - B2\*A12) / det

ddpsi = (A11\*B2 - A21\*B1) / det

return np.array([phi\_dot, psi\_dot, ddphi, ddpsi])

# Начальные условия

y0 = [phi0, psi0, phi\_dot0, psi\_dot0]

# Численное интегрирование

Y = odeint(odesys, y0, t, args=(M, m, c, g, R))

phi = Y[:, 0]

psi = Y[:, 1]

X = X0 + R \* Y[:, 1]

phi\_dot = Y[:, 2]

psi\_dot = Y[:, 3]

ddphi = np.zeros\_like(phi)

ddpsi = np.zeros\_like(phi)

for i in range(len(t)):

dydt = odesys(Y[i, :], t[i], M, m, c, g, R)

ddphi[i] = dydt[2]

ddpsi[i] = dydt[3]

# Теперь вычисляем силу N

alpha = (phi + psi)/2.0

N = (m\*(g\*np.cos(phi) + R\*(phi\_dot\*\*2 - ddpsi\*np.sin(phi))) +

2\*R\*c\*(1 - np.cos(alpha))\*np.cos(alpha))

# Построение x(t), phi(t), N(t)

fig2, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(8, 10))

ax1.plot(t, X\_O, label='x(t)')

ax1.set\_ylabel('x (м)')

ax1.set\_title('x(t)')

ax1.grid(True)

ax1.legend()

ax2.plot(t, phi, label='phi(t)', color='r')

ax2.set\_ylabel('phi (рад)')

ax2.set\_title('phi(t)')

ax2.grid(True)

ax2.legend()

ax3.plot(t, N, label='N(t)', color='g')

ax3.set\_ylabel('N (H)')

ax3.set\_title('N(t)')

ax3.grid(True)

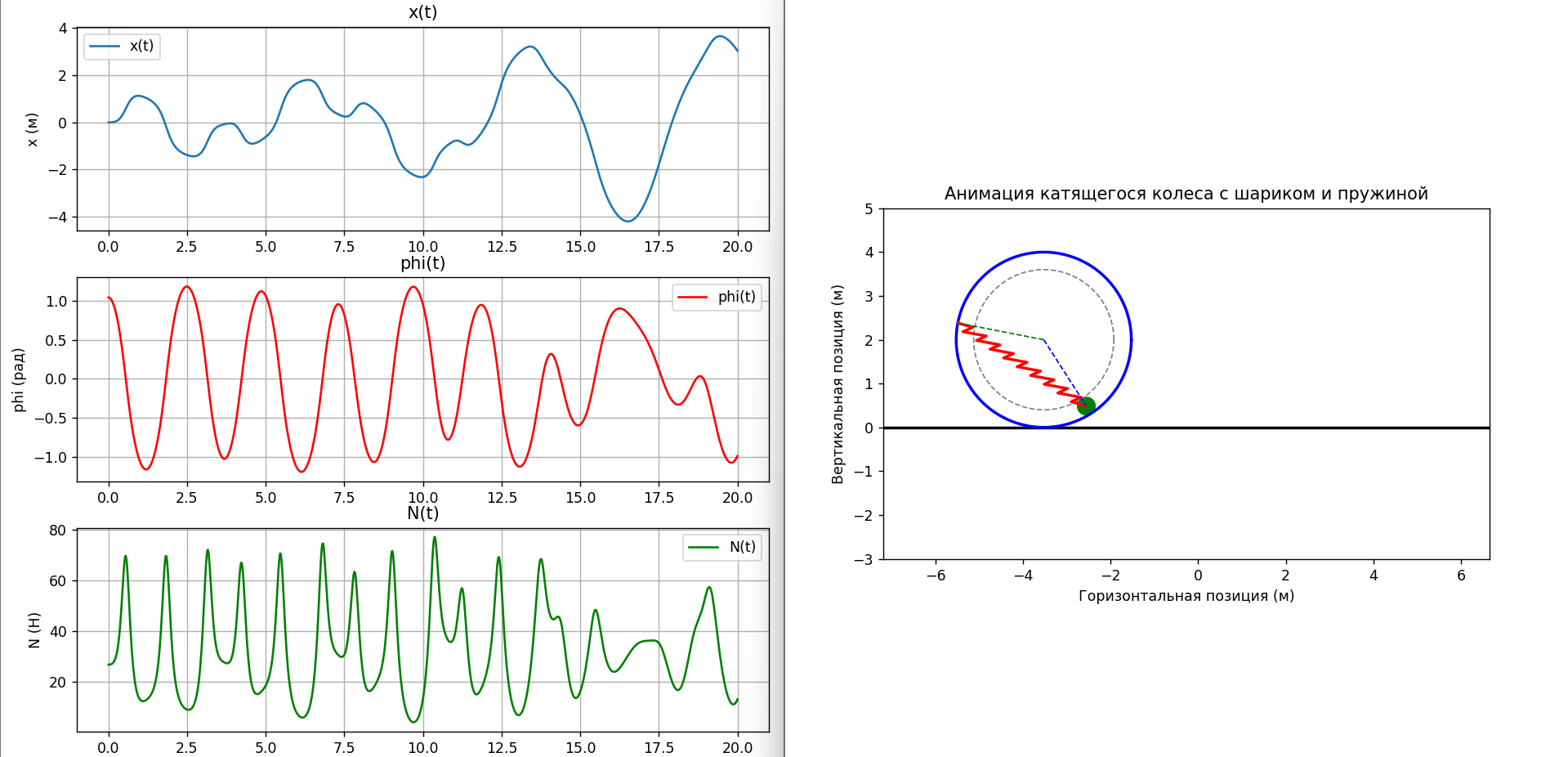
ax3.legend()

plt.tight\_layout()

plt.show()

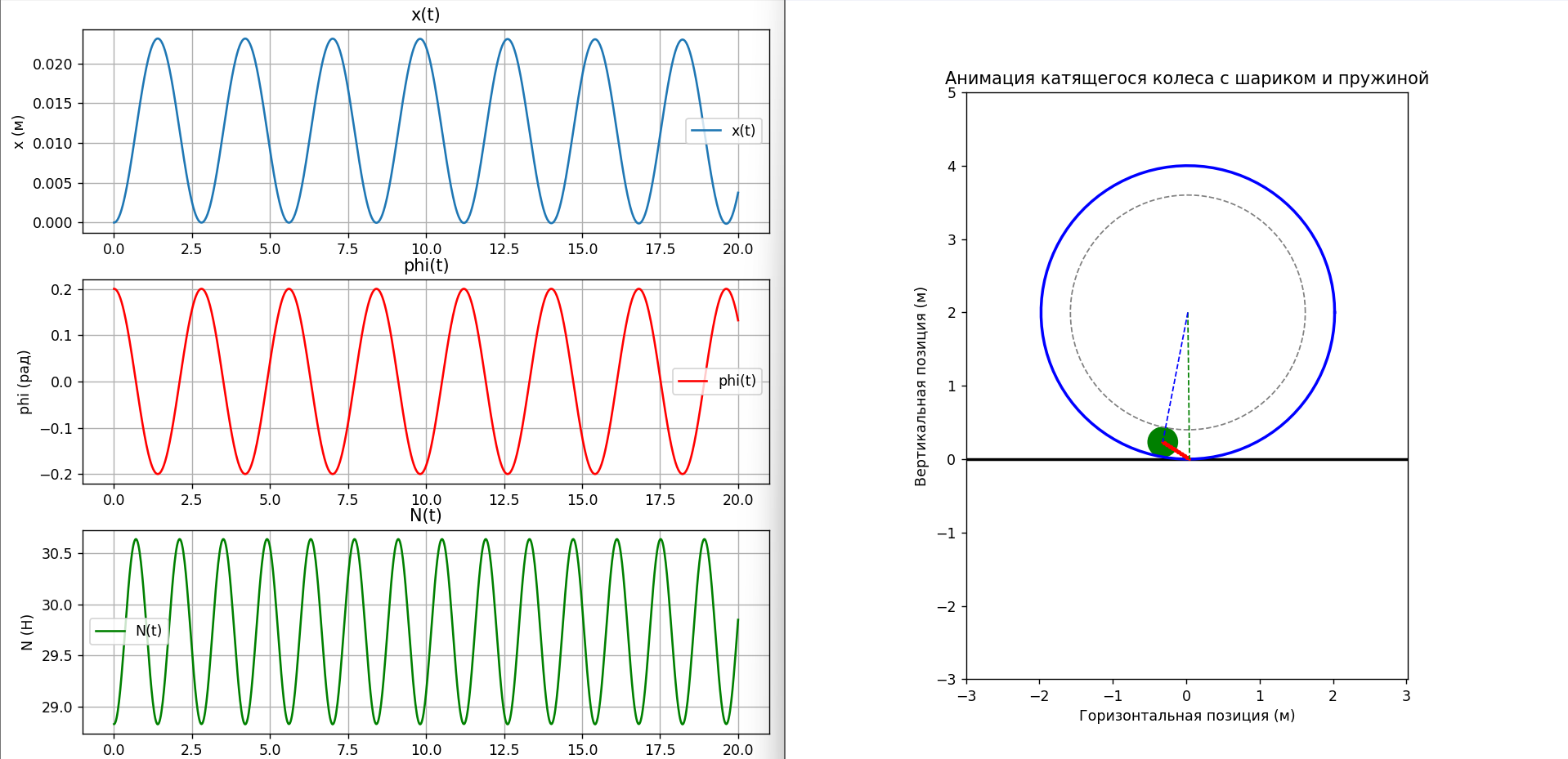
**Результат работы программы:**

1. m = 2; M = 1; R = 2; c = 40; g = 9.81; ; – масса шарика больше массы колеса, довольно жесткая пружина, умеренное начальное отклонение:

****

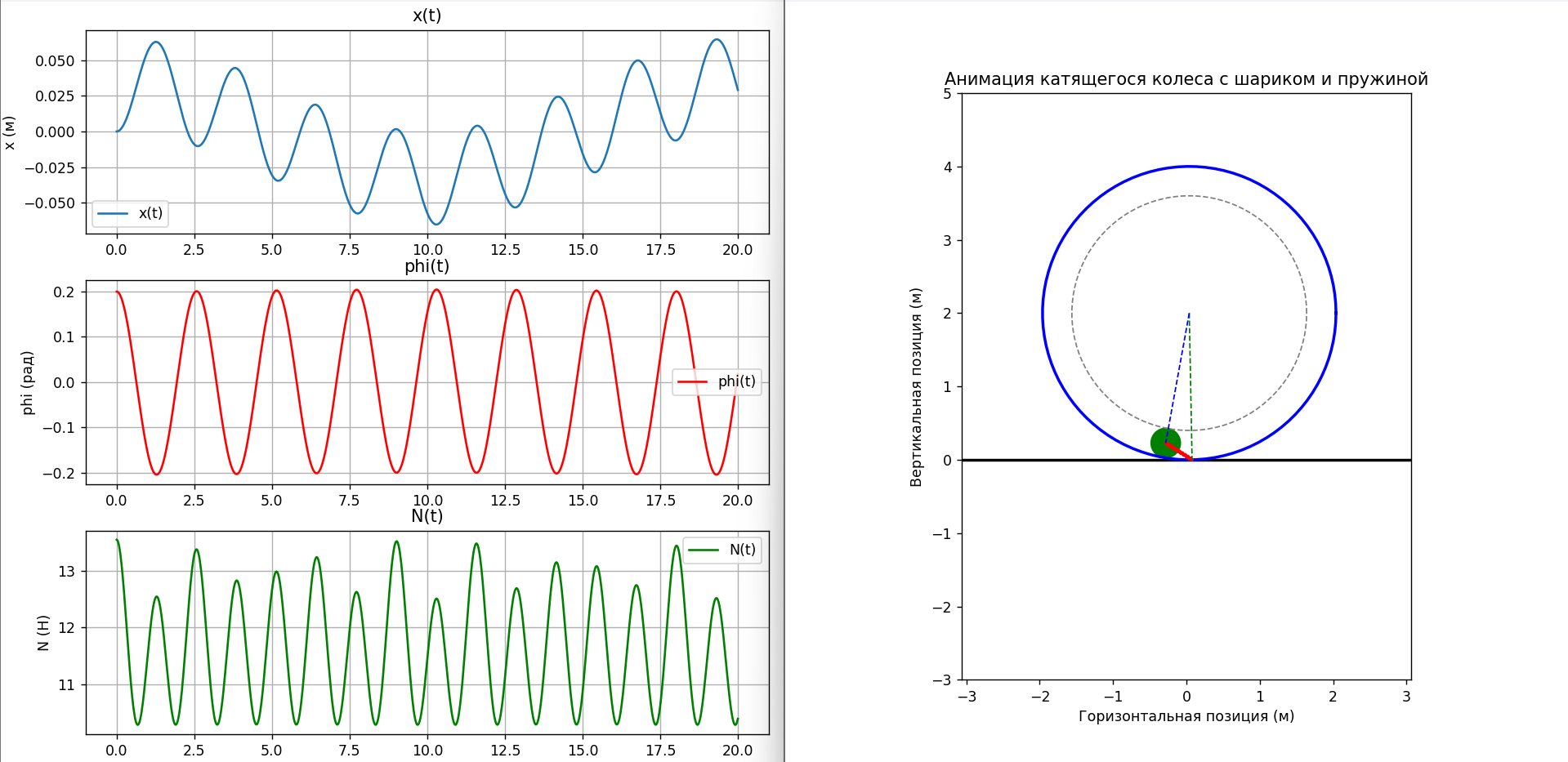
Результат: шарик с пружиной образуют подобие раскачивающихся качелей, колесо совершает колебательные движения, амплитуда возрастает.

1. m = 1; M = 50; R = 2; c = 1; g = 9.81; ; – Слабая пружина, большая масса колеса и небольшое начальное отклонение:

****

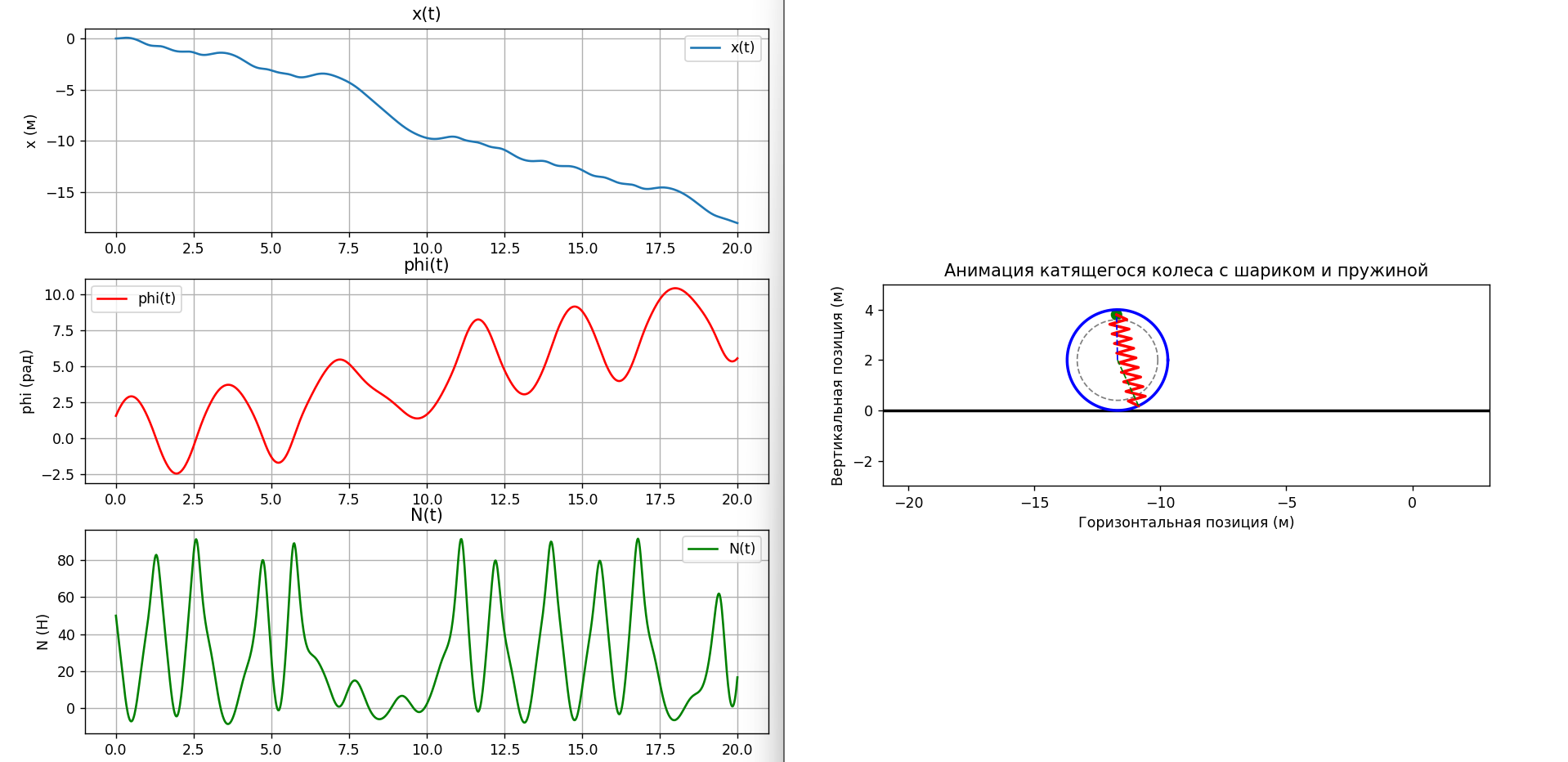
Результат: шарик совершает почти незаметные колебания около равновесия внутри колеса, колесо движется еле заметно.

1. m = 1; M = 5; R = 2; c = 200; g = 9.81; ; – сильная пружина, небольшая масса колеса и небольшое начальное отклонение:

****

Результат: шарик выполняет малые колебания вокруг стабильного положения, движение колеса еле заметно.

1. m = 1; M = 5; R = 2; c = 5; g = 9.81; ; – слабая пружина, небольшая масса колеса и сильное начальное возмущение:

****

Результат: шарик может облететь вокруг трубки, так как упругая сила не достаточно сильна, чтобы вернуть его в окрестность равновесия; колесо катится влево.

**Вывод:**

Были добавлены уравнения движения, описанные через уравнения Лагранжа второго рода, что позволило численно моделировать поведение системы. Проведённые эксперименты с разными начальными условиями и параметрами показали разнообразие поведения системы. Мои навыки программирования на Python улучшились.