МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования «Казанский национальный исследовательский

технический университет им. А. Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

(наименование института (факультета), филиала)

Кафедра Прикладной математики и информатики

(наименование кафедры)

(09.03.04) Программная инженерия

(шифр и наименование направления подготовки (специальности))

Отчет по лабораторным работам по теме «Аналитическое моделирование»

Дисциплина: «Компьютерное моделирование процессов и систем»

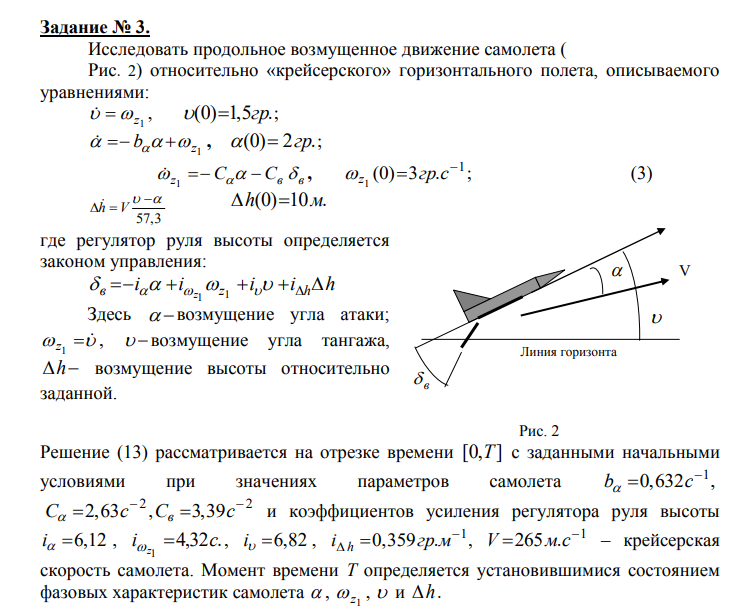
Лабораторная работа №3

Вариант 7

Выполнил: студент группы 4411

Хасаншин Ш.Р

Казань, 2024



Подставим в уравнения известные константы и заменим переменные на более удобные:

,

,

,

Для решения системы мы воспользуемся методом Рунге-Кутта 2-го порядка в модификации Богацки и Шампина

Программная реализация:   
import numpy as np

import matplotlib.pyplot as plt

# Описание модели динамики системы

def model\_dynamics(t, state):

pos, angle, pitch\_rate, altitude\_diff = state

dpos\_dt = pitch\_rate

dangle\_dt = -0.632 \* angle + pitch\_rate

dpitch\_rate\_dt = 18.1168 \* angle - 14.6448 \* pitch\_rate - 23.1198 \* pos - 1.12701 \* altitude\_diff

daltitude\_diff\_dt = 4.6248 \* (pos - angle)

return np.array([dpos\_dt, dangle\_dt, dpitch\_rate\_dt, daltitude\_diff\_dt])

# Реализация метода Рунге-Кутта 2-го порядка

def rk2\_solver(system, initial\_state, time\_interval, step\_size):

time\_points = np.arange(time\_interval[0], time\_interval[1], step\_size)

solution = np.zeros((len(time\_points), len(initial\_state)))

solution[0] = initial\_state

i = 1 # Инициализируем счётчик для цикла while

while i < len(time\_points):

current\_time = time\_points[i - 1]

current\_state = solution[i - 1]

k1 = system(current\_time, current\_state)

k2 = system(current\_time + 2/3 \* step\_size, current\_state + 2/3 \* step\_size \* k1)

solution[i] = current\_state + step\_size \* (3/4 \* k1 + 1/4 \* k2)

i += 1 # Увеличиваем счётчик на каждом шаге

return time\_points, solution

# Задаём начальные параметры

initial\_conditions = [1.5, 2, 3, 10] # Начальные состояния для переменных: pos(0), angle(0), pitch\_rate(0), altitude\_diff(0)

time\_span = [0, 10] # Промежуток времени для симуляции

time\_step = 0.1 # Шаг интегрирования

# Получаем решение задачи

time\_results, state\_results = rk2\_solver(model\_dynamics, initial\_conditions, time\_span, time\_step)

# Вывод результатов: таблица

print("Результаты численного моделирования:")

print(f"{'Время':>8} | {'Положение':>12} | {'Угол атаки':>12} | {'Скорость тангажа':>18} | {'Разница высот':>15}")

print("-" \* 67)

for time, state in zip(time\_results, state\_results):

print(f"{time:8.2f} | {state[0]:12.4f} | {state[1]:12.4f} | {state[2]:18.4f} | {state[3]:15.4f}")

# Визуализация результатов: график

plt.figure(figsize=(10, 6))

plt.plot(time\_results, state\_results[:, 0], label='Позиция', linestyle='-', color='dodgerblue', linewidth=2)

plt.plot(time\_results, state\_results[:, 1], label='Угол атаки', linestyle='--', color='orange', linewidth=2)

plt.plot(time\_results, state\_results[:, 2], label='Скорость тангажа', linestyle=':', color='green', linewidth=2)

plt.plot(time\_results, state\_results[:, 3], label='Разница высот', linestyle='-.', color='red', linewidth=2)

plt.title('Динамика системы с использованием метода Рунге-Кутта 2-го порядка', fontsize=14)

plt.xlabel('Время (с)', fontsize=12)

plt.ylabel('Значения переменных', fontsize=12)

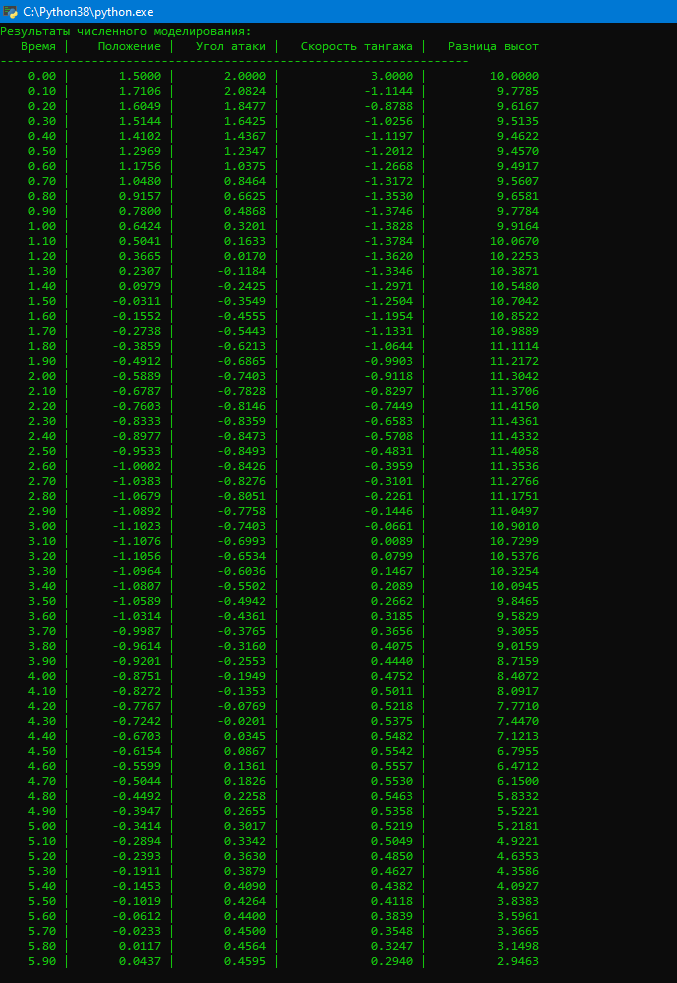
plt.legend(loc='best', fontsize=10)

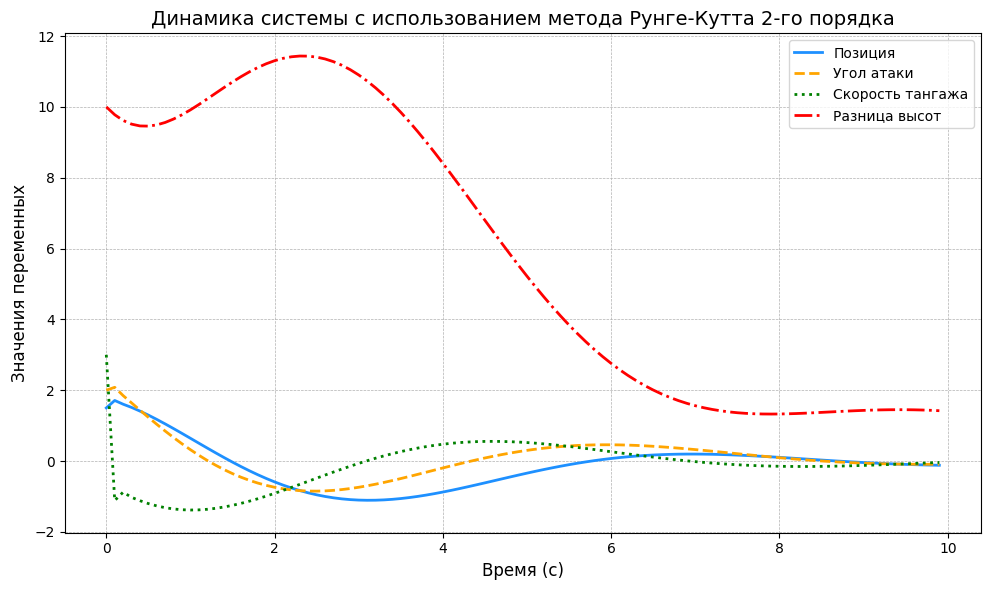
plt.grid(True, which='both', linestyle='--', linewidth=0.5)

plt.tight\_layout()

# Отображаем график

plt.show()

Вывод таблицы:  


Вывод графика: 

**Вывод**

В этой лабораторной работе мы провели моделирование системы с помощью метода Рунге-Кутта второго порядка. В результате получены графики изменения положения, угла атаки, скорости тангажа и разницы высот во времени. Эти графики показывают, как изменяются основные параметры системы при заданных начальных условиях.

Метод Рунге-Кутта позволил получить точные результаты, наглядно показывая динамику системы.