

# Отчет о реализации проекта

Симуляция перехвата воздушной цели

Кузьмин, Ларионов

10 декабря 2025 г.

## Содержание

<b>1 Введение</b>	<b>2</b>
<b>2 Архитектура системы</b>	<b>2</b>
2.1 Технологический стек . . . . .	2
2.2 Логика работы . . . . .	2
<b>3 Физическая модель</b>	<b>2</b>
3.1 Основное уравнение динамики ракеты . . . . .	2
3.2 Действующие силы . . . . .	3
3.2.1 Сила тяги . . . . .	3
3.2.2 Сила тяжести . . . . .	3
3.2.3 Аэродинамическое сопротивление . . . . .	3
3.2.4 Сила Кориолиса . . . . .	3
3.3 Переменная масса . . . . .	3
3.4 Модель движения цели . . . . .	4
<b>4 Численные методы и алгоритмы</b>	<b>4</b>
4.1 Метод интегрирования . . . . .	4
4.2 Алгоритм поиска траектории . . . . .	4
<b>5 Пример работы</b>	<b>5</b>
5.1 Без изменения ветра во время полета . . . . .	5
5.2 С изменением ветра во время полета . . . . .	6
<b>6 Заключение</b>	<b>6</b>

# 1 Введение

Целью данного проекта является разработка программного комплекса для симуляции полета ракеты-перехватчика и поражения движущейся воздушной цели. Ключевой особенностью системы является реалистичная физическая модель (3 DoF — 3 степени свободы), учитывающая переменную массу, аэродинамическое сопротивление, ветер и вращение Земли.

Проект реализован в виде веб-приложения, где серверная часть выполняет сложные баллистические расчеты, а клиентская часть обеспечивает 3D-визуализацию в реальном времени.

## 2 Архитектура системы

Система построена на клиент-серверной архитектуре с использованием протокола WebSocket для обмена данными в реальном времени.

### 2.1 Технологический стек

- **Backend:** Python (FastAPI). Отвечает за физические вычисления, численное интегрирование и поиск оптимальной траектории.
- **Frontend:** JavaScript, HTML5, CSS. Использует библиотеку *Plotly.js* для построения трехмерных графиков траекторий.
- **Протокол:** WebSockets (JSON). Обеспечивает потоковую передачу состояния симуляции (координаты, скорость, ветер).

### 2.2 Логика работы

Процесс разделен на два этапа:

1. **Расчет:** Сервер получает параметры цели и ветра, после чего выполняет поиск оптимального вектора запуска ( $\vec{n}_{\text{пуска}}$ ), который обеспечит попадание.
2. **Симуляция:** Запуск ракеты с рассчитанным вектором. На этом этапе пользователь может менять параметры ветра в реальном времени, наблюдая отклонение ракеты от идеальной траектории.

## 3 Физическая модель

В основе симуляции лежит численное решение дифференциальных уравнений движения материальной точки переменной массы в трехмерном пространстве.

### 3.1 Основное уравнение динамики ракеты

Движение ракеты описывается вторым законом Ньютона:

$$\vec{a}(t) = \frac{d\vec{v}}{dt} = \frac{\sum \vec{F}}{m(t)} \quad (1)$$

где  $\sum \vec{F}$  — векторная сумма всех действующих сил:

$$\sum \vec{F} = \vec{F}_{\text{тяги}} + \vec{F}_{\text{тяж}} + \vec{F}_{\text{сопр}} + \vec{F}_{\text{Кор}} \quad (2)$$

## 3.2 Действующие силы

### 3.2.1 Сила тяги

Тяга создается реактивным двигателем. В модели принята активная стабилизация вектора тяги относительно точки старта.

$$\vec{F}_{\text{тяги}} = \begin{cases} F_{\text{дв}} \cdot \vec{n}_{\text{пуска}}, & \text{при } t < t_{\text{роп}} \text{ и } m_{\text{топл}} > 0 \\ 0, & \text{иначе} \end{cases} \quad (3)$$

где  $F_{\text{дв}}$  — номинальная сила тяги (Н),  $\vec{n}_{\text{пуска}}$  — единичный вектор направления запуска.

### 3.2.2 Сила тяжести

Действует вертикально вниз по оси  $Z$ :

$$\vec{F}_{\text{тяж}} = m(t) \cdot \vec{g} \quad (4)$$

где  $\vec{g} = (0, 0, -9.81) \text{ м/с}^2$ .

### 3.2.3 Аэродинамическое сопротивление

Сила сопротивления направлена против вектора относительной скорости. Именно через эту силу реализовано влияние ветра.

$$\vec{F}_{\text{сопр}} = -\frac{1}{2} C_d A \rho |\vec{v}_{\text{отн}}| \vec{v}_{\text{отн}} \quad (5)$$

где:

- $\vec{v}_{\text{отн}} = \vec{v}_{\text{ракеты}} - \vec{W}_{\text{ветра}}$  — скорость ракеты относительно воздуха.
- $\rho = 1.225 \text{ кг/м}^3$  — плотность воздуха.
- $C_d$  — коэффициент лобового сопротивления.
- $A$  — площадь поперечного сечения (миделя).

### 3.2.4 Сила Кориолиса

Учитывает инерциальные эффекты вращения Земли для заданной широты  $\phi$ :

$$\vec{F}_{\text{Кор}} = -2m(t)(\vec{\Omega} \times \vec{v}) \quad (6)$$

Вектор угловой скорости Земли  $\vec{\Omega}$  в локальной системе координат:

$$\vec{\Omega} = [0, \omega \cos \phi, \omega \sin \phi] \quad (7)$$

где  $\omega \approx 7.2921 \cdot 10^{-5} \text{ рад/с}$ .

В качестве широты выбрана широта Нью-Йорка

## 3.3 Переменная масса

Масса ракеты уменьшается по мере выгорания топлива:

$$m(t) = m_{\text{пуст}} + m_{\text{топл}}(t) \quad (8)$$

Скорость расхода топлива считается постоянной во время работы двигателя:

$$\frac{dm}{dt} = -\frac{m_{\text{топл\_всего}}}{t_{\text{роп}}} \quad (9)$$

### 3.4 Модель движения цели

В отличие от ракеты-перехватчика, для воздушной цели принята кинематическая модель равномерного прямолинейного движения. Учет силы Кориолиса (п. 3.2.4) и аэродинамического сопротивления (п. 3.2.3) для цели не производится. Данное упрощение обосновано двумя факторами:

1. **Имитация управляемого полета.** Цель рассматривается как летательный аппарат с активной системой автоматического управления (САУ). Предполагается, что САУ компенсирует внешние возмущения (ветер, дрейф) для удержания заданного курса и скорости.
2. **Оптимизация вычислительной сложности.** Алгоритм наведения (см. раздел 4) предполагает многократное итеративное решение системы дифференциальных уравнений для поиска оптимальной траектории. Интегрирование уравнений движения двух тел с учетом полной физической модели на каждом шаге оптимизации привело бы к критическому увеличению времени расчета. В рамках веб-приложения приоритет был отдан скорости отклика системы, в отличие от реальных комплексов ПВО, использующих высокопроизводительные вычислительные кластеры.

## 4 Численные методы и алгоритмы

### 4.1 Метод интегрирования

Для решения системы дифференциальных уравнений используется метод **Рунге-Кутты 4-го порядка**. В отличие от метода Эйлера, метод Рунге-Кутты обеспечивает высокую точность расчетов даже при относительно большом шаге по времени ( $dt$ ).

Состояние системы на шаге  $i$  описывается вектором  $S_i = \{\vec{r}, \vec{v}, t_{\text{топл}}\}$ . На каждом шаге вычисляются четыре коэффициента  $(k_1, k_2, k_3, k_4)$  для координат и скоростей, что позволяет аппроксимировать нелинейное поведение аэродинамических сил.

### 4.2 Алгоритм поиска траектории

Задача наведения сводится к поиску вектора запуска  $\vec{n}_{\text{пуска}}$  (определенного азимутом и углом места), который минимизирует дистанцию промаха ( $d_{\text{промах}}$ ) до движущейся цели.

Реализован многоступенчатый алгоритм оптимизации:

1. **Грубый поиск:** Сканирование сферической сетки углов (азимут  $0..360^\circ$ , угол места  $-20..85^\circ$ ) с большим шагом.
2. **Локальное уточнение:** Итеративное уменьшение области поиска вокруг лучшего кандидата с уменьшением шага.
3. **Координатный спуск:** Тонкая настройка углов для минимизации ошибки.
4. **Метод Нелдера-Мида:** Финальная оптимизация для нахождения глобального минимума функции промаха.

Для ускорения расчетов используется ранняя остановка симуляции, если промах заведомо велик, и кэширование результатов.

# 5 Пример работы

## 5.1 Без изменения ветра во время полета

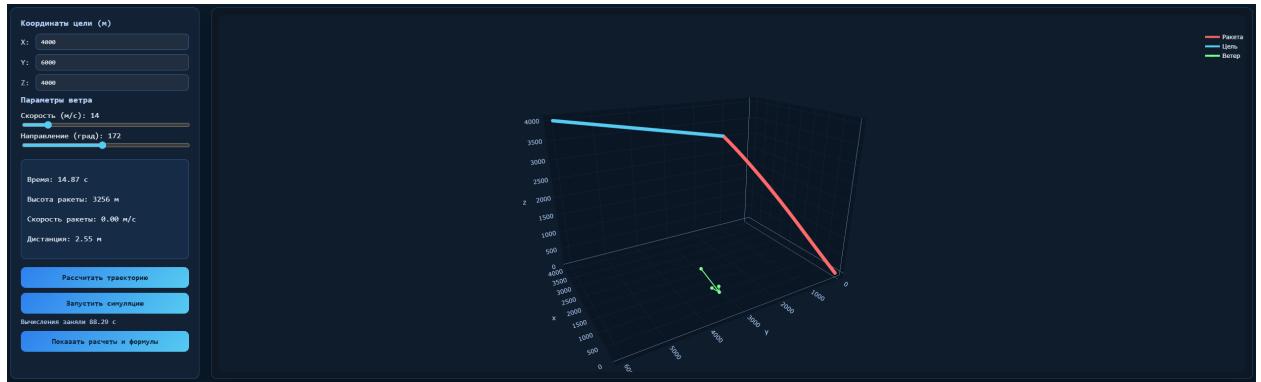


Рис. 1: Траектория полета без изменения ветра

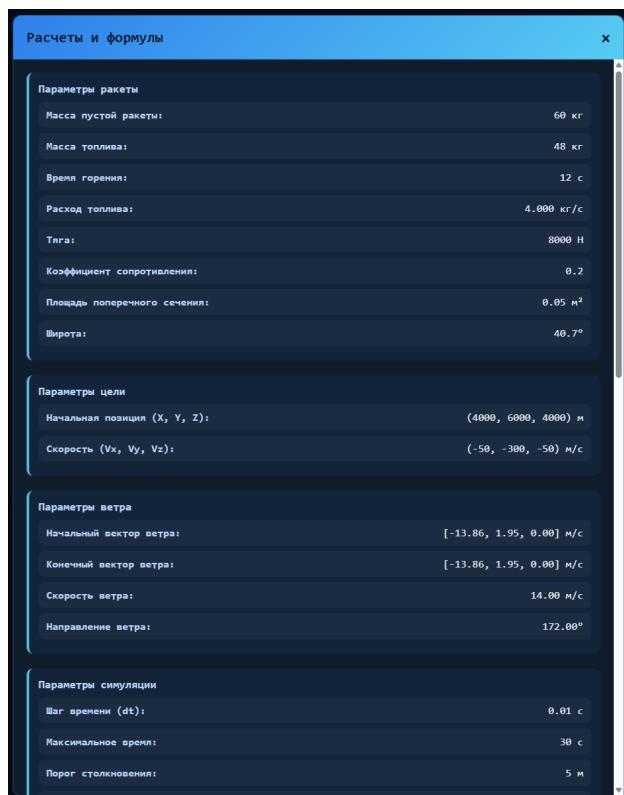


Рис. 2: Расчеты параметров без изменения ветра

## 5.2 С изменением ветра во время полета

