

**Руководство разработчика:
Программный комплекс
для моделирования
аэродинамических течений**

Щетина С. А.

Руководство разработчика: Программный комплекс для моделирования аэродинамических течений

Щетина С. А.

Версия 1.0

Авторские права © 2025 Все права защищены

Содержание

Навигация	vi
1. Введение	1
2. Системные требования	2
3. Установка и запуск	3
4. Описание интерфейса	4
.....	v
5. Обзор архитектуры	6
6. Структура компонентов	7
7. Классовая структура	8
Модуль ввода данных	8
Модуль визуализации	8
8. Интеграция с внешними системами	9
Gmsh	9
Matplotlib и PySide6	9
9. Форматы обмена данными	10
Экспорт из модуля ввода данных	10
Импорт в модуль визуализации	10
10. Расширяемость	11
Поддержка нового формата экспорта результатов	11
Интеграция с новым вычислительным модулем	11

Список иллюстраций

4.1. Главное окно модуля ввода данных	4
---	---

Список таблиц

1. Пример файла с результатами вычислений, полученного из вычислительного модуля	v
---	---

Навигация

Перейти к: Руководство пользователя (PDF) [user.pdf]

[google.com](https://www.google.com)

Глава 1. Введение

Данный программный комплекс предназначен для моделирования ламинарных и турбулентных аэродинамических течений. Он объединяет три компонента: модуль ввода данных, вычислительный модуль и модуль визуализации результатов.

Комплекс ориентирован на решение типовых задач, таких как обтекание профиля крыла или трубы, и обеспечивает сквозной цикл от постановки задачи до анализа результатов.

Глава 2. Системные требования

- Операционная система: Windows, Linux или macOS
- Python 3.13 или выше
- Зависимости: PySide6, Matplotlib, Pandas, Gmsh
- Вычислительный модуль (например, COMSOL Multiphysics) для выполнения расчётов

Глава 3. Установка и запуск

1. Установите пакетный менеджер uv.

2. Запустите модуль ввода данных:

```
uv run modules/data/main.py
```

3. Для визуализации результатов запустите:

```
uv run modules/visualization/main.py
```

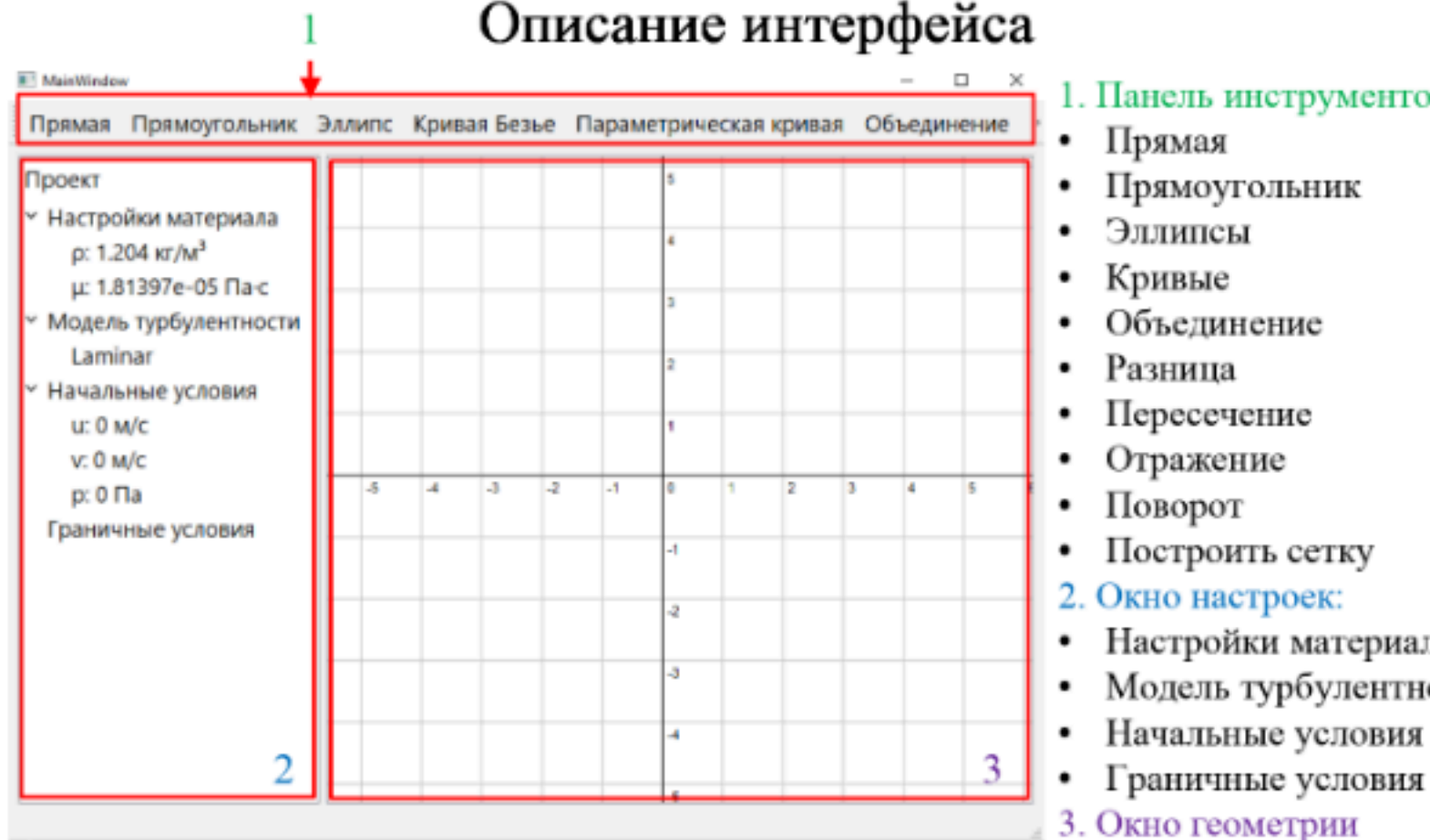
Глава 4. Описание интерфейса

Программный комплекс состоит из двух основных приложений:

- *Модуль ввода данных* — содержит (см. Рисунок 4.1, «Главное окно модуля ввода данных»):
 - Дерево проекта (настройки материала, модель турбулентности, начальные и граничные условия)
 - Рабочую зону для создания геометрии (примитивы, булевы операции)
 - Панель инструментов (построение сетки, экспорт конфигурации)

Рисунок 4.1. Главное окно модуля ввода данных

Описание интерфейса



- *Модуль визуализации* — содержит:
 - Панель настроек (выбор файла с результатами, тип графика, параметры отображения)
 - Область вывода графика (контурные карты, линии тока, одномерные зависимости)

Таблица 1. Пример файла с результатами вычислений, полученного из вычислительного модуля

x	y	U (скорость)	u (пр. U на ось x)	v (пр. U на ось y)	p (давление)
0	0	50	10	15	15
0	1	49	11	15	14
0	1	49	11	15	14

Глава 5. Обзор архитектуры

Программный комплекс реализован на языке Python и построен по модульному принципу. Он состоит из трёх независимых, но взаимодействующих модулей: *модуль ввода данных*, *вычислительный модуль* (внешний) и *модуль визуализации*. Такая архитектура обеспечивает гибкость, упрощает тестирование и позволяет легко заменять или обновлять отдельные компоненты.

Модули обмениваются данными через файлы в стандартизированных форматах: геометрия и сетка — в формате `.msh` (Gmsh), параметры задачи — в `.json`, результаты расчётов — в `.csv` или `.txt`.

Глава 6. Структура компонентов

- *Модуль ввода данных* — реализован на PySide6, содержит:
 - Дерево проекта для настройки материала, модели турбулентности, начальных и граничных условий
 - Рабочую зону для создания геометрии (примитивы, булевы операции)
 - Интеграцию с библиотекой Gmsh для генерации конечно-элементной сетки
- *Вычислительный модуль* — внешняя система (например, COMSOL Multiphysics). Программный комплекс обеспечивает только экспорт данных для неё; интеграция на уровне API не реализована из-за ограничений стороннего ПО.
- *Модуль визуализации* — реализован на PySide6 с интеграцией Matplotlib для построения контурных карт, линий тока и одномерных графиков.
- *Вспомогательные утилиты* — парсинг CSV/JSON, работа с файловой системой.

Глава 7. Классовая структура

Модуль ввода данных

- `MainWindow` — основное окно приложения, управляет компоновкой и взаимодействием виджетов.
- `ProjectTreeWidget` — виджет дерева проекта. Хранит и отображает иерархию настроек: материал, модель турбулентности, начальные и граничные условия.
- `GeometryScene` и `GeometryView` — наследники `QGraphicsScene` и `QGraphicsView` для визуального редактирования геометрии.
- `MeshGenerator` — обёртка над Gmsh API для генерации сетки на основе построенной геометрии и привязанных физических групп (граничных условий).

Модуль визуализации

- `VisualizationWindow` — основное окно, содержит панель настроек и область графика.
- `DataParser` — использует библиотеку `pandas` для загрузки и структурирования данных из CSV/TXT файлов.
- `Plotter` — инкапсулирует логику построения графиков с помощью `Matplotlib` и их встраивания в интерфейс PySide6 через `FigureCanvasQTAgg`.

Глава 8. Интеграция с внешними системами

Gmsh

Библиотека Gmsh используется для генерации двумерной треугольной сетки. Интеграция осуществляется через официальный Python API. Перед использованием необходимо инициализировать Gmsh с помощью `gmsh.initialize()`.

Геометрические примитивы из рабочей зоны преобразуются в объекты Gmsh, а граничные условия привязываются к ним как физические группы.

Matplotlib и PySide6

Для встраивания графиков Matplotlib в GUI используется виджет `FigureCanvasQTAgg`. Это позволяет создавать интерактивные и статические графики, которые полностью интегрированы в оконную систему Qt.

Глава 9. Форматы обмена данными

Экспорт из модуля ввода данных

- `geometry.msh` — файл сетки в формате Gmsh, содержащий геометрию, сетку и физические группы (метки для граничных условий).
- `parameters.json` — файл с параметрами задачи. Пример структуры:

```
{
  "rho": 1.204,
  "mu": 1.81e-5,
  "turb_model": "SST",
  "init_conditions": {"u": 0, "v": 0, "p": 0, "k": 4.2e-7, "omega": 2.78}
}
```

Импорт в модуль визуализации

Модуль ожидает текстовый файл с разделителями (CSV/TXT), содержащий столбцы: `x`, `y`, `u`, `v`, `p`, `U` (координаты, компоненты скорости, давление, модуль скорости). Используется библиотека `pandas` для парсинга.

Глава 10. Расширяемость

Поддержка нового формата экспорта результатов

1. Расширьте класс `DataParser`, добавив метод для чтения нового формата.
2. Добавьте опцию выбора формата в интерфейс модуля визуализации.

Интеграция с новым вычислительным модулем

Для интеграции с новым вычислителем необходимо обеспечить совместимость форматов обмена данными. В идеале, новый модуль должен уметь импортировать `.msh` и `.json`. В противном случае потребуется реализовать конвертер.