

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

---

**Разработка информационной системы для автоматизации**

---

**выполнения лабораторных работ**

---

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 247 группы

направление 09.04.03 — Прикладная информатика

---

механико-математического факультета

---

Лисюткина Михаила Юрьевича

---

Научный руководитель  
доцент, к.т.н., доцент

И. А. Панкратов

---

Зав. кафедрой  
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

---

Саратов 2025

**Введение.** Цифровизация образования в России и за рубежом формирует новые направления использования технологий. В настоящее время существует множество информационных систем и технологий, которые позволяют людям эффективно удовлетворять свои потребности и упрощают их деятельность. Перед современными исследователями стоит задача внедрения различных информационных систем в образовательный процесс с целью увеличения качества и доступности образования. В настоящий момент в мире активно ведутся исследования и разработки в области автоматизации учебных и научных экспериментов.

Целью магистерской работы является разработка информационной системы для автоматизации выполнения лабораторных работ по курсу «Математические модели управления движением».

**Основная часть.** Магистерская работа состоит из семи разделов. В первом разделе описывается актуальность работы. Во втором разделе представлен обзор литературы. С третий по пятый раздел дается описание программы, ее функций, используемой базы данных, стек технологий и архитектура разработанного приложения. В шестом разделе приведены примеры использования программы на реальных задачах курса «Методы математического управления движением». В седьмом разделе проводиться анализ разработанной системы, выявляются ее сильные и слабые стороны, а также дается описание дальнейшего развития программы.

Специфика лабораторных работ по техническим дисциплинам заключается в следующем:

1. применение вычислительной техники;
2. работа с точными данными;
3. выполнение сложных математических расчётов;
4. применение современных средств автоматизации и использования специализированного программного обеспечения.

Сама система подразумевает формирования множества отчетов после выполнения учебных задач. Большое количество формул и расчетов полученных в результате работ необходимо хранить и обрабатывать по мере необходимости.

## **Установка.**

Используется следующий список зависимостей:

1. python v3.12
2. dill v0.3.9
3. scipy v1.15.2
4. matplotlib v3.10.0
5. pylatex v1.4.2
6. sympy v1.13.3

Опишем каждый из них:

Matplotlib — это одна из самых популярных библиотек визуализации данных на Python. Она была создана Джоном Хантером в 2003 году как инструмент для построения графиков и диаграмм с интерфейсом, напоминающим MATLAB. Благодаря своей кроссплатформенности и способности выводить изображения в различных форматах (PNG, PDF, SVG, EPS и другие), Matplotlib стала стандартом для визуализации в научной среде. В настоящее время поддерживается сообществом разработчиков и активно развивается.

PyLaTeX — это библиотека, предназначенная для создания и манипуляции документами LaTeX средствами Python. PyLaTeX позволяет программно генерировать сложные документы, включая таблицы, изображения, формулы и списки, не прибегая к ручному редактированию .tex-файлов. Библиотека была разработана в начале 2010-х годов как удобный инструмент для автоматизации создания отчётов и научных публикаций.

Dill — это библиотека сериализации объектов Python, которая расширяет возможности стандартного модуля pickle. Dill позволяет сохранять практически любые объекты Python в файл или строку и восстанавливать их позже, даже в другой сессии. Это особенно ценно при работе с большими или сложными объектами, такими как функции, классы, лямбда-выражения и даже интерпретаторы. В отличие от pickle, Dill поддерживает более широкий спектр типов данных и может сериализовать интерактивные сессии. Благодаря этому Dill часто используется в параллельных и распределённых вычислениях, а также в машинном обучении для сохранения состояния моделей.

SciPy — это фундаментальный пакет для научных и технических вычислений на Python. Библиотека базируется на NumPy и предоставляет широкий набор функций для численного анализа, линейной алгебры, оптимизации, ин-

терполяции, интегрирования, обработки сигналов и изображений и многое другое. SciPy является одним из ключевых элементов научного стека Python и используется во множестве областей: от физики и химии до экономики и биоинформатики.

SymPy — это библиотека символьных вычислений на Python. Она позволяет выполнять алгебраические преобразования, решать уравнения, дифференцировать и интегрировать функции, работать с матрицами, пределами, рядами и другими математическими конструкциями. Библиотека широко применяется в образовании, исследованиях и инженерных расчётах, где требуется точное математическое выражение и возможность его аналитической обработки.

Для установки всех требуемых пакетов в терминале выполняется следующая команда:

```
1 pip install -r requirements.txt
```

Сборка проекта осуществляется с помощью утилиты pyinstaller. PyInstaller — это утилита для упаковки Python-приложений в автономные исполняемые файлы. Она позволяет создавать программы, которые можно запускать на устройствах без установленного Python или зависимостей. Для сборки программы из исходных файлов в исполняемый файл выполняется следующая команда:

```
1 pyinstaller --onefile --windowed --name "MyLab" main.py
```

Во время первого запуска программы создается директория с названием программы, которая содержит следующие каталоги и файлы: «saves», «output», «functionList.db».

1. Директория «saves» содержит информацию о сохраненных пользователем системах.
2. В директория «output» содержится сгенерированный отчет об выполнении программы.
3. Файл «functionList.db» содержит информацию о базе данных, которая используется во время работы приложения.

**Описание программы.** При проектировании информационной системы предполагалось, что пользователю будут доступны следующие варианты использования программы:

1. Выбор лабораторной работы
2. Задание граничных условий задачи
3. Задание системы дифференциальных уравнений
4. Задание дополнительных параметров в зависимости от выбранной задачи
5. Составление отчета с результатами решения

**Описание функциональных возможностей программы.** Программа позволяет пользователю искать решения систем дифференциальных уравнения в контексте оптимального управления.

Диаграмма активностей в соответствии с описанием функциональных возможностей представлена в соответствии с рисунком 1.



Рисунок 1 — Диаграмма активностей программы.

## База данных программы.

Для поддержки работы программы была разработана база данных на основе СУБД SQLite со следующими таблицами:

1. *BORDERDB* - таблица со значениями граничных условий задач,
2. *TASKINFO* - таблица с информацией о задаче,
3. *TASKNAME* - таблица с названием задач,
4. *TASKS* - таблица реализации связи «многие ко многим» между задачами и пользователями,
5. *USERS* - таблица пользователей системы,
6. *SOLVEINFO* - таблица с настройками решателя,
7. *GROUPS* - таблица с номерами допусимых групп.

Названия задач хранятся в таблице *TASKNAME*. Структура таблицы *TASKNAME* представлена в соответствии с таблицей 1.

Таблица 1 – Структура таблицы *TASKNAME*

Название поля	Тип данных	Значение
<i>id</i>	integer	первичный ключ
<i>value</i>	text	название задачи

Так как один студент может решать множество задач, была создана таблица *TASKS*, реализующая связь «многие ко многим». Структура таблицы *TASKS* представлена в соответствии с таблицей 2.

Таблица 2 – Структура таблицы *TASKS*

Название поля	Тип данных	Значение
<i>id</i>	integer	первичный ключ
<i>taskid</i>	int	номер задачи
<i>userid</i>	int	номер пользователя

Так как один студент может решать множество задач, была создана таблица *USERS*, реализующая связь «многие ко многим». Структура таблицы *USERS* представлена в соответствии с таблицей 3.

Таблица 3 – Структура таблицы *USERS*

Название поля	Тип данных	Значение
id	integer	первичный ключ
F	text	фамилия студента
I	text	имя студента
O	text	отчество студента
groupid	integer	индекс группы студента

Таблица *SOLVEINFO* содержит следующие поля: Для хранения информации о настройках решателя была реализована таблица *SOLVEINFO*. Структура таблицы *SOLVEINFO* представлена в соответствии с таблицей 4.

Таблица 4 – Структура таблицы *SOLVEINFO*

Название поля	Тип данных	Значение
id	integer	первичный ключ
varepsilon	int	точность найденного решения
num	int	степень разбиения при поиске ответа
diffmethod	text	метод интегрирования
method	text	метод решения нелинейных систем

Таблица *GROUPS* содержит следующие поля: Для хранения информации о допустимых группах студентов была реализована таблица *GROUPS*. Структура таблицы *GROUPS* представлена в соответствии с таблицей 5.

Таблица 5 – Структура таблицы *GROUPS*

Название поля	Тип данных	Значение
id	integer	первичный ключ
value	int	номер группы студента

Таблица *BORDERDB* содержит следующие поля: Для хранения информации о введеных студентом граничных условий была реализована таблица *BORDERDB*. Структура таблицы *BORDERDB* представлена в соответствии с таблицей 6.

Таблица 6 – Структура таблицы *BORDERDB*

Название поля	Тип данных	Значение
id	integer	первичный ключ
value	text	набор граничных значений

Таблица *TASKINFO* содержит информацию о решаемой задаче и содержит такие поля: Для хранения информации о дополнительных параметров задачи была реализована таблица *TASKINFO*. Структура таблицы *TASKINFO* представлена в соответствии с таблицей 7.

Таблица 7 – Структура таблицы *TASKINFO*

Название поля	Тип данных	Значение
id	integer	первичный ключ
borderid	integer	индекс граничных условий
solveid	integer	индекс настроек решателя
taskNameId	integer	индекс названия решаемой задачи
n	integer	размерность задачи
findBalancePoint	bool	поиска точек равновесия в задаче
isbkfun	bool	способ задания граничных условий

Все взаимосвязи созданных таблиц в базе данных представлены в соответствии с рисунком 2.

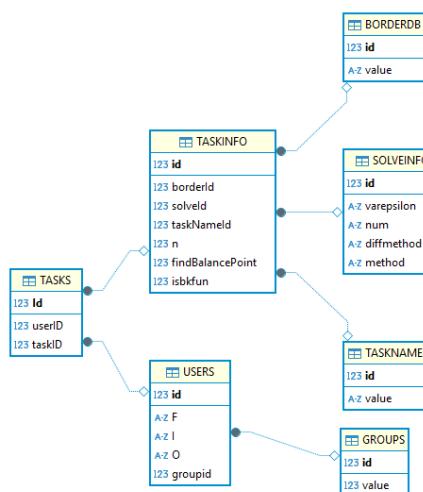


Рисунок 2 – Диаграмма классов базы данных программы

**Пример работы программы.** Приведем пример работы пользователя в созданной информационной системе.

Пусть система имеет следующий вид:

$$\begin{cases} \frac{dx}{dt} = v(t) \\ \frac{dv}{dt} = \frac{F_1(t)}{m} + \frac{F_2(x, v)}{m}. \end{cases} \quad (1)$$

Границные условия имеют вид:

$$\begin{cases} x(0) = x_0 \\ v(0) = v_0 \\ \phi(x(t_k), v(t_k)) = 0 \end{cases}$$

Функционал задачи имеет следующий вид:

$$J = \int_0^{t_k} u^2 dt \quad (2)$$

Пусть функция  $\phi(x, v)$  имеет следующий вид:

$$\phi(x, v) = \frac{a}{x_1} + bx_2,$$

а функция  $F_2(x, v)$  равна:

$$F_2(x, v) = -kx^3$$

Так как функционал 2 не соответствует функционалу на наискорейшее или энергетически оптимальное движение, то пользователь выбирает вариант задачи «Линейные системы с квадратичным критерием качества».

Затем пользователь в соответствии с рисунком 3 вводит уравнения 1, начальные и граничные условия.

На втором экране пользователь может создать несколько проводимых численных экспериментов. В каждом эксперименте пользователь может выбрать начальные значения сопряженной системы  $\psi$  и вариант управления в задаче: ограниченный или неограниченный. Пусть пользователь поставит

MyLab

Задача Настройки О программе

Линейные системы с квадратичным критерием качества

Выбор аргументов      Поиск решения

$\frac{d}{dt}x_1(t) =$

$\frac{d}{dt}x_2(t) =$

$$J = \int_{t_0}^{t_k} (\alpha_1 + \alpha_2 u^2) dt$$

Начальные и граничные условия

$\xi_{10} = 1$

$\xi_{20} = 1$

$\phi(t) =$

Задать уравнением

Начальный момент времени

$t_0 = 0$

Дополнительные параметры

$\alpha_1 = 0$

$\alpha_2 = 1$

Назад      Далее

Не готово      Compile pdf

Рисунок 3 — Выбор варианта решаемой задачи

два эксперимента и для каждого  $\psi = [2, 1]$  и  $t_k = 3$ . Первый эксперимент не имеет ограничение на управление. А второй эксперимент имеет ограничение вида:  $|u(t)| \leq 1$ . После интегрирования пользователь получит следующие результаты в соответствии с рисунком 4а для первого эксперимента и в соответствии с рисунком 4б для второго эксперимента. Затем студент производит интегрирование получившейся системы уравнений.

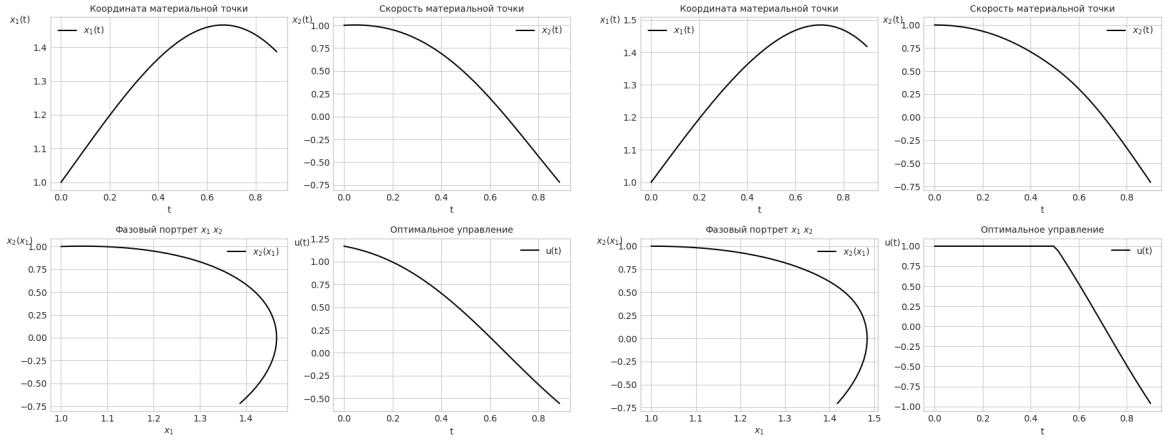


Рисунок 4 — Результаты экспериментов

После того как пользователь получил решение системы, с помощью *Compile pdf* производится генерация отчета о работе программы.

**Заключение.** В работе представлен пример реализации учебно-методической системы, предназначеннной для выполнения лабораторных работ студентов. Были описаны основные модули системы, процесс установки. Была описана база данных информационной системы и взаимосвязи внутри нее. В магистерской работе был проведен анализ системы, выявления сильных и слабых сторон, предложены варианты ее дальнейшего развития.