Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

ОТЧЁТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ

Студент ______________________________
(Фамилия, Имя, Отчество)

Группа ________

Тип практики ______________________________

Название предприятия АИС МГТУ им. Н. Э. Баумана

Студент ______ (Группа) ______________ (Подпись, дата) ______________ (И.О. Фамилия)

Руководитель практики ______________ (Подпись, дата) ______________ (И.О. Фамилия)

Оценка ______________________________

2022 г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

1. Разработка приложения поддержки исследования для дипломной работы по теме «Моделирование нестационарных линейных систем на основе метода Галёркина». Приложение представляет собой интерфейс для исследования нестационарных линейных систем, задающихся линейным дифференциальным уравнением второго порядка с переменными коэффициентами. Приложение должно включать в себя такие функции как ввод исходных данных, вывод результатов на экран, а также сохранение полученных результатов с возможностью их обратной загрузки в приложение.
2. Реализация метода Галёркина для решения краевой задачи для линейного дифференциального уравнения второго порядка с переменными коэффициентами следующего вида

$$\begin{cases} y'' + \alpha(x)y' + \beta(x)y = \gamma(x) \\ y(0) = y_0 \\ y(1) = y_1 \end{cases},$$

где аналитические вещественнозначные функции $\alpha(x)$, $\beta(x)$, $\gamma(x)$ могут быть произвольными, как и числовые значения граничных условий y_0, y_1. Данный метод должен использоваться для исследования нестационарных линейных систем в приложении.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Нестационарными линейными системами или *линейными системами с переменными параметрами* называют системы, которые описываются линейными дифференциальными уравнениями с переменными коэффициентами. Для их описания помимо дифференциальных уравнений могут быть использованы предаточные функции, переходные и весовые (импульсные переходные) функции, частотные функции и их характеристики.

В данной практической работе рассматриваются нестационарные линейные системы, задающиеся одним линейным дифференциальным уравнением второго порядка с переменными коэффициентами.

Для решения таких дифференциальных уравнений используется метод Галёркина, который является представителем класса так называемых *методов взвешенных невязок*. Данный метод позволяет сводить задачу решения дифференциального уравнения к решению системы линейных алгебраических уравнений с целью получения приближения решения. В настоящем отчете кратко описываются основы метода Галёркина (см. раздел 2).

Для удобства исследования поведения различных дифференциальных уравнений было разработано интерфейсное приложение, позволяющее исследователю задавать коэффициенты уравнения и граничные условия, получать результат решения заданного уравнения методом Галёркина в виде графика полученной функции и оценки погрешности.

Структура отчета представлена следующим образом. Сначала вкратце описывается предприятие, на котором проходила практика (раздел 1). Затем дается представление о сути метода Галёркина (раздел 2). И наконец идет описание структуры приложения, его возможностей и инструкция по его использованию (раздел 3).

Настоящий отчет полностью написан с помощью системы верстки документов LaTeX, поэтому в электронной версии все ссылки на разделы и источники, а также номера страниц в содержании кликабельны.

1 О базе практики

Скажу пару слов о предприятии, на котором проходила практика.

Московский государственный технический университет имени Н. Э. Баумана (также известный как МГТУ им. Баумана) — это российский национальный исследовательский университет, научный центр, особо ценный объект культурного наследия народов России.

Вуз активно участвует в Болонском процессе. В 2008 году он получил награду «Европейское качество» «за стремление достичь высокого качества образовательных услуг в соответствии с международными стандартами». МГТУ им. Н. Э. Баумана в течение более чем 14 лет является головным вузом Ассоциации технических университетов, в состав которой входят более 130 российских университетов. МГТУ — первый российский вуз, ставший членом ассоциации «Top Industrial Managers for Europe».

Отдел «Автоматизированных информационных систем» (АИС) — это отдел МГТУ, который поддерживает всю информационную составляющую деятельности МГТУ, в том числе корпоративные сети. Сотрудники АИС работают с высокопроизводительными системами, обеспечивающими вычисления в рамках учебной деятельности и НИР.

2 Метод Галёркина [2]

Методы Галёркина к настоящему времени были применены при решении многочисленных задач механики конструкций, динамики сооружений, гидромеханики, теории гидродинамической устойчивости, магнитной гидродинамики, теории гидродинамической устойчивости, магнитной гидродинамики, теории тепло- и массообмена, акустики, теории распространения микроволн, теории переноса нейтронов и т. п. С помощью представлений Галёркина были проведены исследования обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных и интегральных уравнений. Стационарные и нестационарные задачи, а также задачи на собственные значения оказались в равной степени поддающимися исследованию на основе подходов Галёркина. По существу, любая задача, для которой можно выписать определяющие уравнения, может быть решена с помощью одной из разновидностей метода Галёркина.

Важнейшие особенности метода Галёркина можно сформулировать в следующей очень компактной форме. Согласно предположению, некая двумерная задача описывается линейным дифференциальным уравнением

$$L(u) = 0 \tag{1}$$

в области $D(x,y)$ при граничных условиях

$$S(u) = 0 \tag{2}$$

на линии ∂D, являющейся границей области D. В методе Галёркина предполагается, что неизвестная u может быть достаточно точно представлена приближенным решением

$$u_a(x,y) = u_0(x,y) + \sum_{k=1}^{n} a_k u_k(x,y), \tag{3}$$

где u_k — это известные *аналитические функции*; функция u_0 введена, чтобы удовлетворять граничным условиям, тогда как a_k — это коэффициенты, подлежащие определению. Подстановка выражения (3) в уравнение (1) приводит к отличной от нуля невязке R, выражаемой в виде

$$R(a_0,a_1,\ldots,a_n,x,y) = L(u_a) = L(u_0) + \sum_{k=1}^{n} a_k L(u_k). \tag{4}$$

Удобно дать следующее определени скалярного произведения:

$$(f,g) = \iint_D f(x,y)g(x,y)dxdy. \tag{5}$$

При обращении к методу Галёркина неизвестные коэффициенты a_k, входящие в выражение (3), должны определяться из решения следующей системы уравнений:

$$(R,u_l) = 0, \quad l = 1,\ldots,n \tag{6}$$

Здесь R — невязка данного уравнения, а u_k — те же самые аналитические функции, которые фигурируют в (3). Предложенный здесь пример связан с решением линейного дифференциального уравнения, а поэтому уравнения (6) могут быть записаны непосредственно в форме матричного уравнения относительно коэффициентов a_k, а именно

$$\sum_{k=1}^{n} a_k (L(u_k),u_l) = -(L(u_0),u_l). \tag{7}$$

Подстановка величин a_k, определяемых путем решения уравнения (7) в формулу (3) дает искомое приближение решения u_a.

Так как реальная искомая функция почти никогда не бывает известна, для оценки погрешности используется *норма невязки*, вычисляющаяся по следующей формуле

$$\|R\| = \sqrt{(R,R)}. \tag{8}$$

Вполне естественно, что при хорошем приближении искомой функции норма невязки должна быть близка к нулю. На практике так и происходит, поэтому с помощью нормы невязки можно оценивать сверху абсолютную погрешность $\|u - u_a\|$.

3 Описание приложения

Приложение полностью написано на высокоуровневом языке Python 3.8 и тестировалось на персональном компьютере Lenovo ideapad 310-15isk с установленной операционной системой Windows 10.

Весь исходный код приложения можно загрузить со страницы на GitHub [1]

3.1 Требования для запуска

Для запуска приложения на компьютере должен быть установлен интерпретатор языка Python версии не ниже 3.5. Например, последнюю версию можно скачать по ссылке [3].

Также должны быть установлены следующие библиотеки, используемые в приложении:

- math (набор математических функций и констант, обычно встроен в интерпретатор по умолчанию)
- numpy (для общих математических и числовых операций, обычно встроен в интерпретатор по умолчанию)
- scipy (библиотека фундаментальных математических алгоритмов)
- matplotlib (для построения графиков)
- PyQt5 (реализует фреймворк для разработки кроссплатформенного программного обеспечения Qt)

Для установки недостающих библиотек нужно зайти в командной строке Windows в папку с установленным Python, зайти в директорию Scripts и выполнить команду из листинга 1. Вместо «matplotlib» может быть название другой

Листинг 1 — Установка библиотеки matplotlib

```
pip3 install matplotlib
```

недостающей библиотеки. Если переменная pip3 добавлена в настройках среды в PATH, то эту команду можно выполнить из любого места на компьютере.

3.2 Структура и запуск

Исходный код приложения состоит из следующих шести файлов и двух директорий:

- app1.py (основной файл запуска)
- function.py (вспомогательные финтифлюшки для работы с функциями)
- SODE.py (реализация метода Галёркина)
- Main.py (главное окно приложения)
- SODEW.py (основное рабочее окно приложения)
- helper.py (вспомогательные структуры данных)
- miscellaneous (директория для хранения вспомогательных файлов, например изображений)
- saves (директория для хранения сохранененных результатов)

Перед запуском приложения следует убедиться, что все эти восемь файлов присутствуют в директории приложения. Отсутствие какого-либо из них приведет к ошибкам при запуске приложения или работе с ним. Из директории miscellaneous также нельзя удалять никакие файлы, в то время как из директории saves — можно (однако саму эту папку удалять нельзя!). Добавление в директорию saves сторонних файлов извне приложения также может привести к ошибкам при работе с приложением.

Для запуска приложения следует зайти в интегрированную среду разработки и обучения на языке Python — IDLE, открыть в ней файл app1.py и запустить его. Должно открыться главное окно приложения (рис. 1). Также теоретически возможен запуск приложения из командной строки с помощью команды python app1.py, однако такой вариант не тестировался.

3.3 Инструкция по использованию

После того как мы открыли главное окно приложения (рис. 1), следует нажать на кнопку Start для запуска основного рабочего окна (рис. 2). Кнопка Exit может быть нажата в любой момент работы с приложением для полного закрытия всех окон и завершения работы приложения.

Опишу теперь по очереди компоненты рабочего окна приложения. На рис. 3 можно видеть компоненту, отображающую текущий вид краевой задачи для

Рисунок 1 — Главное окно.

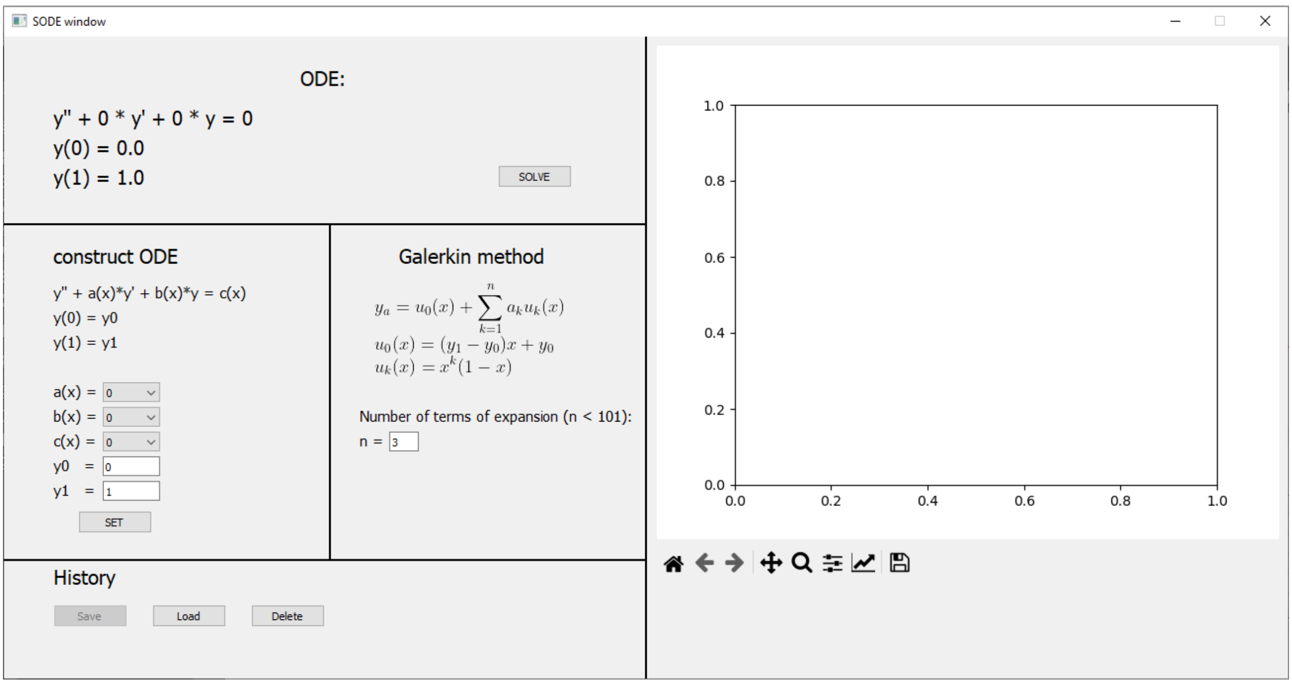


Рисунок 2 — Рабочее окно.

линейного дифференциального уравнения второго порядка. По умолчанию она

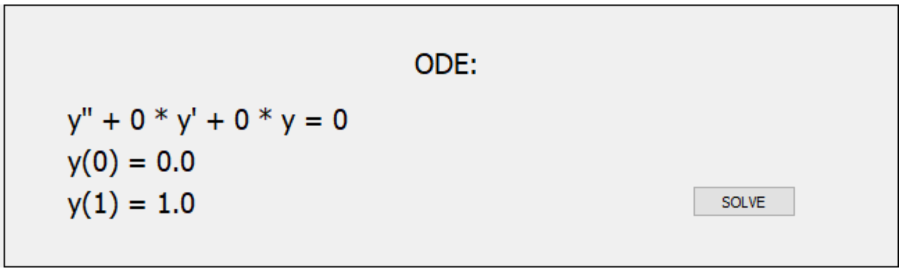


Рисунок 3 — Компонента ODE.

имеет простенький вид:

$$\begin{cases} y'' = 0 \\ y(0) = 0 \\ y(1) = 1 \end{cases}.$$

Кнопка SOLVE решает данную краевую задачу и отображает результаты в правой части рабочего окна приложения (подробнее ниже).

Следующая компонента (рис. 11) позволяет задавать треуемые коэффициенты $a(x)$, $b(x)$, $c(x)$ дифференциального уравнения и соответственно значения на границе y_0, y_1. В выпадающем списке (рис. 5 а)) можно выбрать одну из восьми

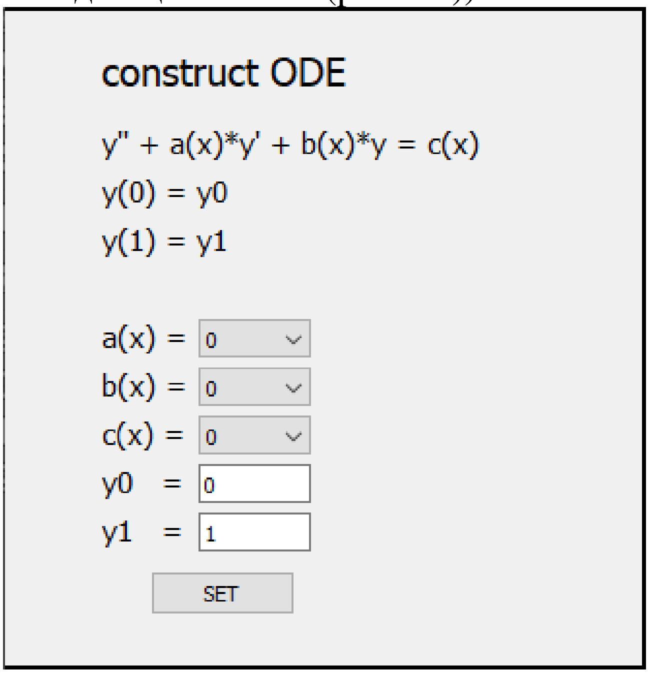
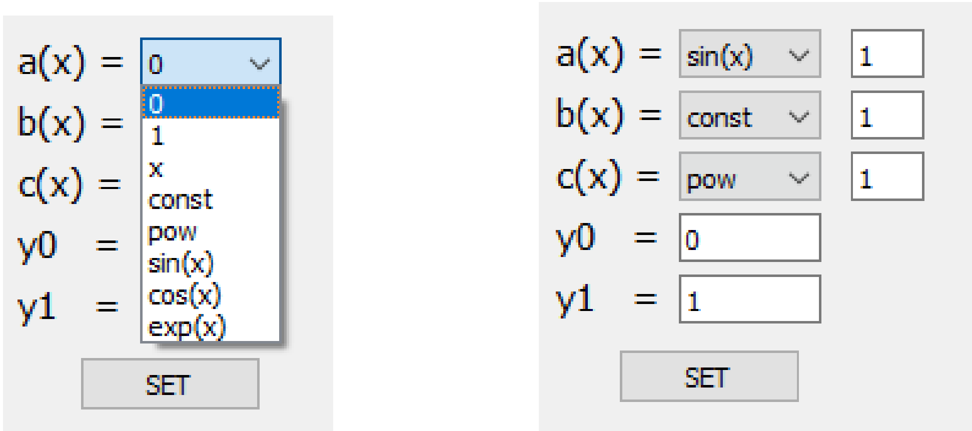


Рисунок 4 — Компонента construct ODE.

допустимых функций, пять из которых имею параметр a:

— 0 (тождественно равная нулю функция)

— 1 (тождественно равная единице функция)



а) Список допустимых функций. б) Параметры функций.

Рисунок 5 — Задание коэффициентов.

— x (тождественная функция $f(x) = x$)

— const (константная функция, принимающая всюду значение, равное a)

— pow (степенная функция, параметром обозначается степень $f(x) = x^a$)

— $\sin(\mathrm{x})$ $(f(x) = \sin(ax))$

— $\cos(\mathrm{x})$ $(f(x) = \cos(ax))$

— $\exp(\mathrm{x})$ $(f(x) = \exp(ax) = e^{ax})$

Для задания параметра a нужно ввести его в появляющейся строке справа от названия функции (рис. 5 б)). Также можно ввести желаемые значения функции на границе отрезка $[0{,}1]$. После введения коэффициентов и граничных условий нужно нажать на кнопку SET, после чего краевая задача из компоненты ODE (рис. 3) примет требуемый вид. При введении параметров функции и значений на границах следует убедиться в том, что введены именно числовые значения, а также что введенные значения не слишком длинны. При несоблюдении этих условий после нажатия на кнопку SET будут выведены соответствующие сообщения об ошибках (рис. 6).

В следующей компоненте (рис. 7) описывается, какая система функций выбрана для приближения решения поставленной задачи методом Галёркина. Также предлагается выбрать число членов разложения n. Чем больше членов разложения будет выбрано, тем точнее будет результат. Реализация допускает выбирать от 1 до 100 членов разложения, однако на практике достаточно не более 5 членов для получения достаточно точного результата. Выбирать более 20 членов вообще не рекомендуется, так как в таком случае вычисление может производиться долго.

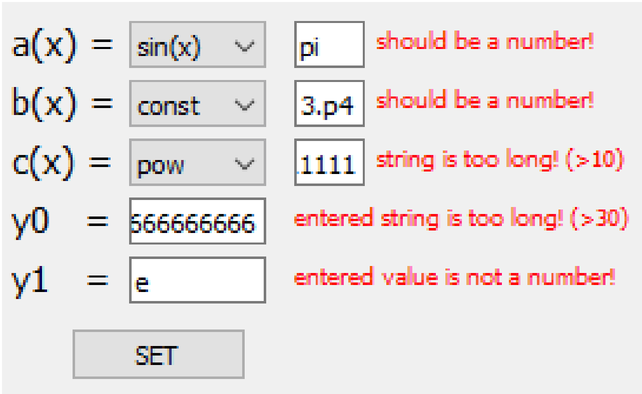


Рисунок 6 — Примеры ошибок введенных значений.

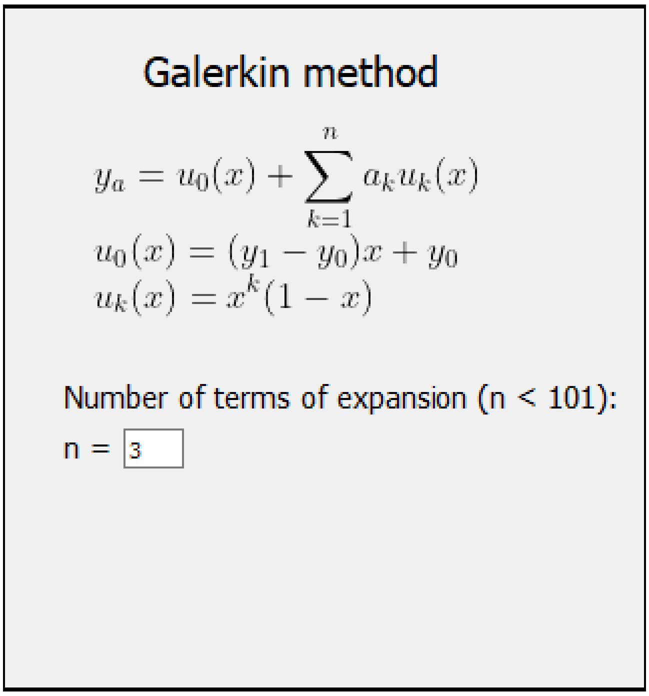


Рисунок 7 — Компонента Galerkin method.

В правой части окна (рис. 8) выводятся результаты решения краевой задачи. Они состоят из графика приближенной функции на отрезке $[0,1]$ и места под оценку погрешности (в самом низу компоненты). В качестве оценки выводится норма невязки (см. раздел 2). График строится с помощью библиотеки $\mathrm{matplotlib}$, и к нему приложен стандартный набор инструментов этой библиотеки для работы с графиком. Поэтому можно свободно увеличивать и двигать график, менять масштабы осей и другие действия, привычные при работе с графиками, построенными с помощью данной библиотеки.

В последней компоненте (рис. 9) представлен функционал для сохранения, загрузки и удаления сохраненных результатов. По умолчанию кнопка Add явля-

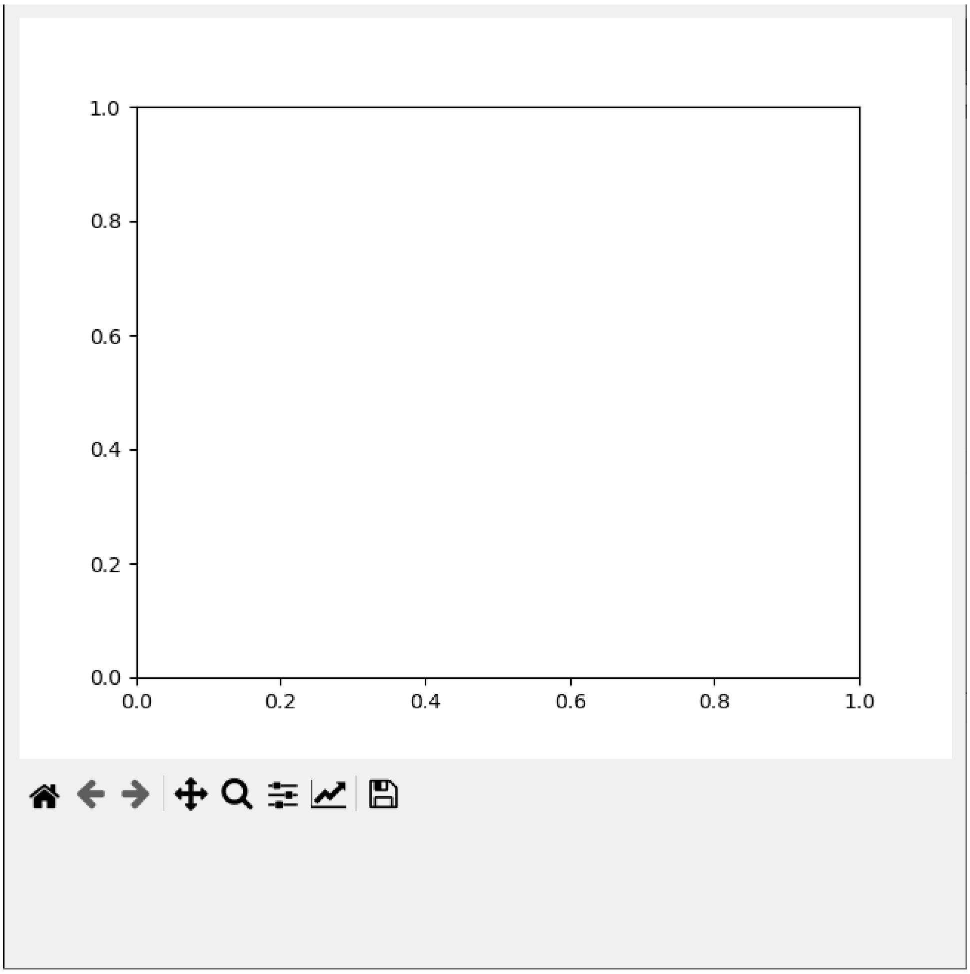


Рисунок 8 — Компонента вывода результатов.

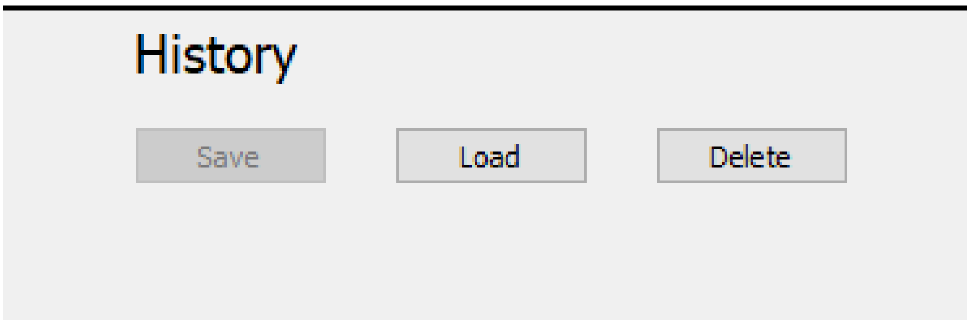


Рисунок 9 — Компонента History.

ется недоступной. Она разблокируется как только будет выполнено вычисление какой-либо задачи. При ее нажатии открывается окно (рис. 10), предлагающее ввести название фигуры в текстовое поле и сохранить текущий результат. После этого в том же окне появится сообщение об успешном сохранении, а введенное название отобразится над графиком полученной функции.

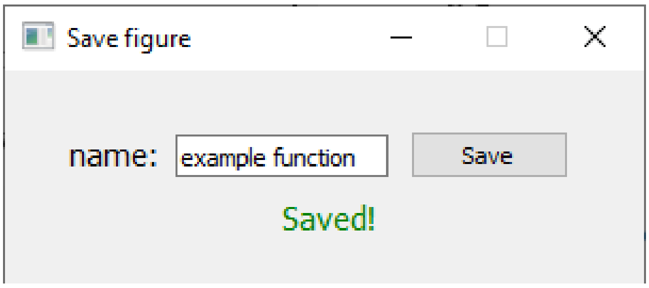


Рисунок 10 — Окно сохранения.

Кнопки Load и Delete открывают окна, в которых можно загрузить на экран сохраненный результат или же удалить его (рис. 12). Для этого нужно выбрать нужную фигуру из списка и нажать на кнопки Load или Delete соответственно. Загрузка и удаление также сопровождаются сообщениями об успехе. Причем при

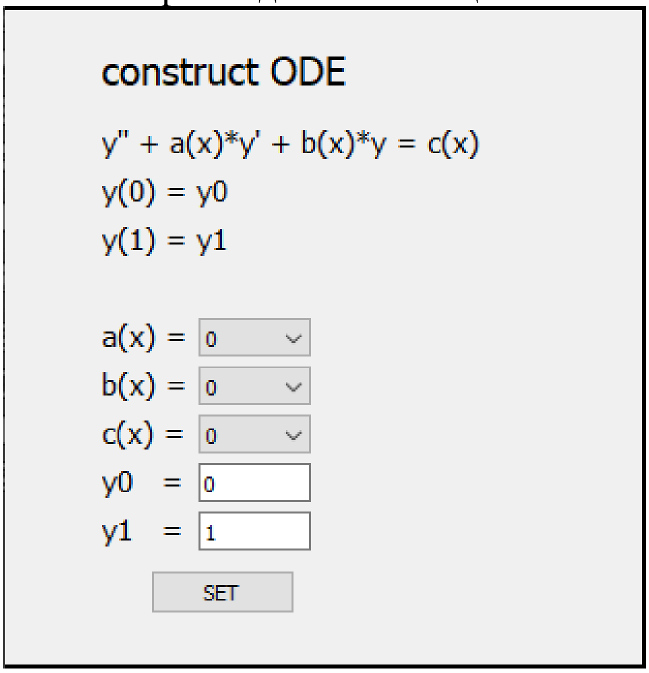
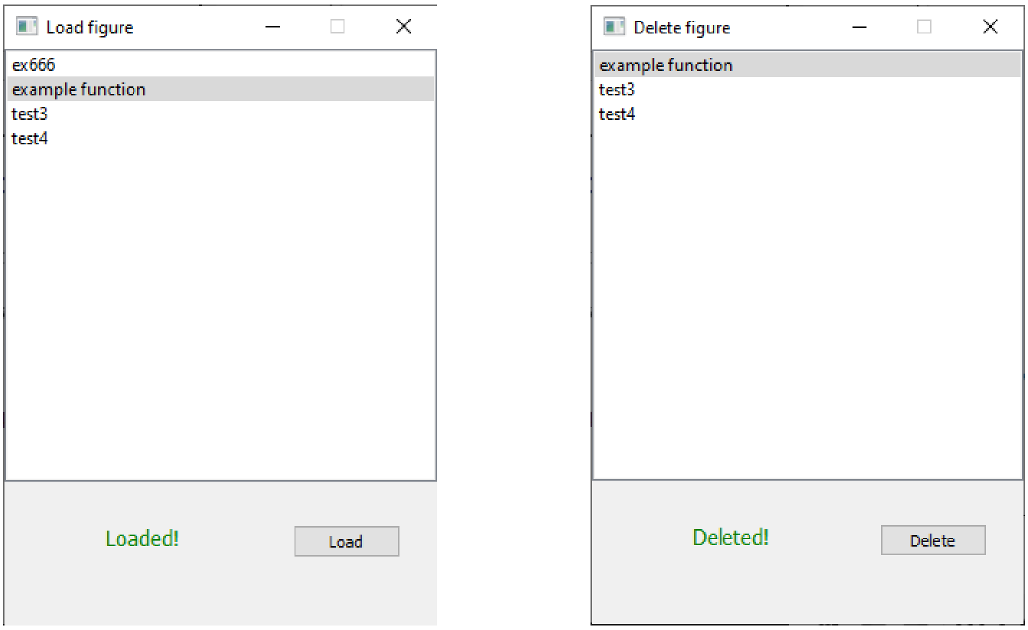


Рисунок 11 — Компонента construct ODE.

загрузке фигуры помимо вывода графика и оценки погрешности также приводится к соответствующему виду краевая задача и все заданные параметры. Стоит отметить, что информация о графике также сохраняется, поэтому никаких повторных вычислений при загрузке не производится.

В итоге при успешном введении данных и решении задачи мы получаем нужный нам результат (см. рис 13).



а) Окно загрузки. б) Окно удаления.

Рисунок 12 — Окна загрузки и удаления.

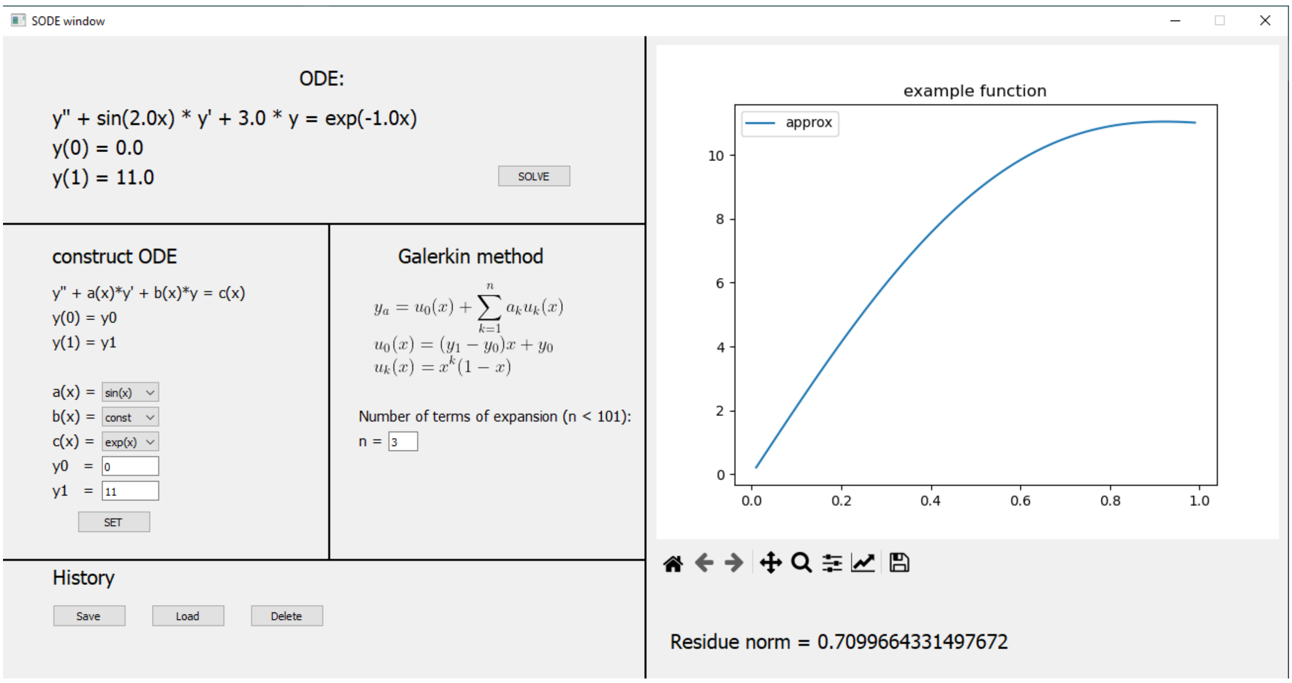


Рисунок 13 — Результат работы программы.

3.4 Неожиданные результаты

Дифференциальные уравнения — вообще говоря довольно сложный объект. Решать их в общем виде до сих пор никто не умеет. И даже в таких условно простеньких примерах, как линейное дифференциальное уравнение второго порядка с переменными коэффициентами, решение может быть безуспешным. Традиционный метод Галеркина не лишен таких проблем, поэтому некторые виды дифференциальных уравнений могут быть не решены. Особенно этим грешит использование степенных функций (в особенности отрицательных и дробных степеней). Например, попытка решить дифференциальное уравнение $y'' + \frac{y'}{x} + x^{-2.3}y = x^5$ методом, заложенным в приложении, не дает точного результата (см. рис. 14). Видно, что даже при выборе 50 членов норма невязки слишком высока. Тем не менее, было ре-

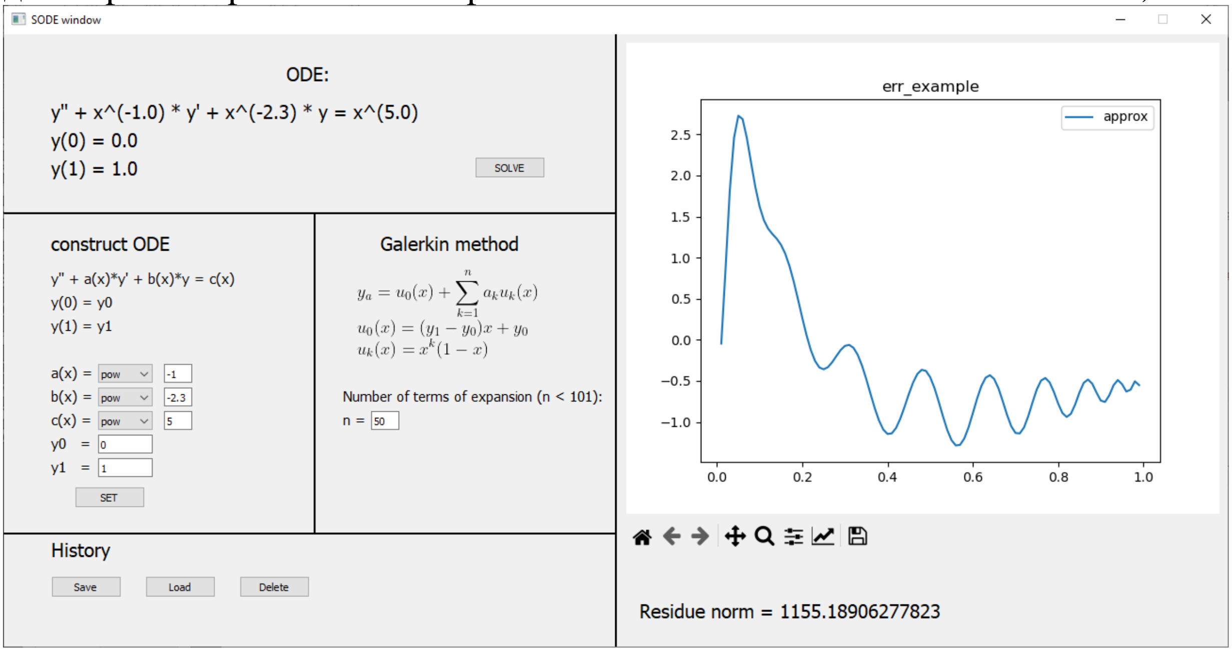


Рисунок 14 — Расхождение вычисления.

шено оставить степенную функцию в приложении, потому что для исследователя полезно понимать, какие уравнения решить данным методом не удается.

ЗАКЛЮЧЕНИЕ

Данный отчет по практике подготовлен с помощью системы верстки документов LaTeX. Его можно рассматривать как инструкцию к использованию разработанного в рамках вышеупомянутой практики приложения.

Само приложение в первую очередь представляет собой продукт поддержки исследования для моей выпускной квалификационной работы по теме «Моделирование нестационарных линейных систем на основе метода Галёркина». Оно позволяет исследовать поведение тех или иных сложных технических систем, задающихся линейным дифференциальным уравнением второго порядка с переменными коэффициентами, а также обеспечивает сохранение и загрузку результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] GitHub [Electronic source]. — Access mode: https://github.com/NotThatWay/Diploma.

[2] K. Fletcher. Computational Galerkin Methods. — New York : Springer-Verlag New York Inc., 1984.

[3] Link to download a current version of Python IDLE. — Access mode: https://www.python.org/downloads/.