

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО

Кафедра дискретной математики и
информационных технологий

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ БИБЛИОТЕКИ MRMATN НА ЯЗЫКЕ
PYTHON ДЛЯ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ РОЖДЕНИЯ ЭЛЕКТРОН-
ПОЗИТРОННОЙ ПЛАЗМЫ В СИЛЬНЫХ ЭЛЕКТРИЧЕСКИХ ПОЛЯХ И
ИССЛЕДОВАНИЕ ДОСТИЖИМОЙ ТОЧНОСТИ МОДЕЛИРОВАНИЯ НА
ПРИМЕРЕ АНАЛИТИЧЕСКОГО РЕШЕНИЯ ДЛЯ ИМПУЛЬСА ЗАУТЕРА

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 421 группы
направления 09.03.01 – Информатика и вычислительная техника
факультета КНиИТ
Войтенко Сергея Георгиевича

Научный руководитель
доцент, к.ф.-м.н., доцент

А.Д. Панферов

Заведующий кафедрой
доцент, к.ф.-м.н., доцент

Л.Б. Тяпаев

Саратов 2017

ВВЕДЕНИЕ

Для современных научных исследований информационные технологии и вычислительные системы являются столь же важным инструментом, как и сложные лабораторные установки и эксперименты. Важным этапом проверки новых научных теорий, гипотез, предположений является построение соответствующих математических моделей и исследование их поведения на мощных вычислительных системах. В современной науке успешно моделируются процессы самых разных пространственных и временных масштабов от процесса эволюции всей нашей Вселенной и до отдельных почти мгновенных актов взаимодействия микрочастиц. Принципиально важным является вопрос о точности и достоверности такого моделирования. Очевидно, самый совершенный компьютер не может полностью воспроизвести все аспекты и элементы даже относительно простого процесса.

Вопрос об адекватности и точности моделирования включает два составляющих его подвопроса: адекватность и точность модели и точность численного воспроизведения её характеристик. Вопросы построения моделей находятся в предметной области соответствующего научного направления, а вот точность симуляции её поведения определяется используемыми численными технологиями, программными решениями и аппаратными возможностями.

В предлагаемой работе я исследую эти аспекты моделирования на примере модели поведения физического вакуума в сильном электрическом поле. Модель основана на теории кинетических уравнений применительно к законам квантовой электродинамики (КЭД) и считается на настоящее время достаточно точной и адекватной поскольку не содержит каких либо упрощающих предположений или приближений. Вариантов же её численного анализа достаточно много. В том числе в настоящее время исследователи для построения программных решений часто выбирают в качестве языка

реализации Python. Это новая тенденция, связанная с удобством и все более широким распространением этого относительно молодого языка.

Передо мной была поставлена задача реализовать численную модель процесса языковыми средствами Python с использованием общедоступных математических библиотек для него и провести верификацию работы этой модели для специального типа электрического поля с зависимостью от времени в форме одиночного импульса Заутера. Выбор этого специального типа поля продиктован наличием для него точного аналитического решения, значения которого должны воспроизводиться по результатам моделирования. Доступные для моделируемого процесса значения параметров имеют очень большой диапазон значений от 1.0 и до сколь угодно малых. Мне необходимо было оценить уровень значений параметров числовой модели, при которой (в рамках заданных программных инструментов) её результаты правильно воспроизводят исследуемый физический процесс.

Для решения этой задачи мне потребуется:

- рассмотреть возможности языка Python
- рассмотреть инструментарий для численного решения систем дифференциальных уравнений библиотеки Mpmath;
- рассмотреть возможности библиотеки Matplotlib для построения графиков;
- изучить инструментарий для создания пользовательского интерфейса библиотеки Tkinter;
- создать приложение для моделирования физического процесса.

Научная новизна работы: были исследованы возможности библиотеки Mpmath высокоуровневого языка программирования Python. В результате данного исследования была разработана программа для моделирования процессов рождения электрон-позитронной плазмы в сильных электрических полях.

Разработанная программа предназначена для моделирования на примере

аналитического решения для импульса Заутера, которое на данный момент нельзя смоделировать физически, а также отвечает всем современным требованиям быстродействия, простоты взаимодействия с пользователем.

1 Основное содержание работы

1 Постановка задачи

Основное уравнение описывающие физический процесс может быть представлено в виде системы ОДУ, которая и будет решаться в данной работе:

$$\begin{cases} f' = \frac{1}{2} \lambda u, \\ u' = \lambda(1 - 2f) - 2\omega v, \\ v' = 2\omega u. \end{cases}$$

В данной системе функция f имеет тот же смысл, что и в исходном интегро-дифференциальном уравнение, а u и v представляют собой вспомогательные функции. Данная система уравнений справедлива при любой зависимости электрического поля от времени, но она получена при условии, что поле не меняет своего направления в течении времени и однородно в пространстве.

Исследуемое переменное электрическое поле задается следующей зависимостью напряженности от времени:

$$E(t) = E_0 * \cosh^{-2} \left(\frac{t}{\tau} \right)$$

где τ – характерное время процесса, E_0 – амплитуда напряженности электрического поля. Начальное значение векторного потенциала поля вычисляется по формуле:

$$A(t) = -\tau * E_0 * \tanh^{-2} \left(\frac{t_{min}}{\tau} \right)$$

На рисунке 1 представлен график зависимость поля от времени:

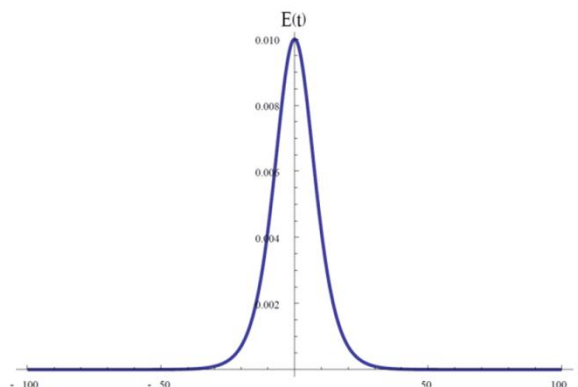


Рисунок 1 – График зависимости поля от времени.

2 Понятие ОДУ. Методы решения ОДУ

Обыкновенным дифференциальным уравнением n -го порядка называется уравнение вида

$$F(x, y(x), y'(x), y''(x), \dots, y^{(n)}(x)) = 0$$

где F — известная функция $(n + 2)$ -х переменных, x — независимая переменная из интервала (a, b) , $y(x)$ — неизвестная функция. Число n называется порядком уравнения.

Задача решения обыкновенных дифференциальных уравнений сложнее задачи вычисления однократных интегралов, и доля задач, интегрируемых в явном виде, здесь существенно меньше.

Когда говорят об интегрируемости, имеют в виду, что решение может быть вычислено при помощи конечного числа «элементарных» операций: сложения, вычитания, умножения, деления, возведения в степень, логарифмирования, потенцирования, вычисления синуса и косинуса и т. п. Уже в период, предшествовавший появлению ЭВМ, понятия «элементарной» операции претерпели изменение. Решения некоторых частных задач настолько часто встречаются в приложениях, что пришлось составить таблицы их значений, в частности таблицы интегралов Френеля, функций Бесселя и ряда других, так называемых специальных функций. При наличии таких таблиц исчезает принципиальная разница между вычислением тригонометрических функций и специальных функций. В том и другом случаях можно вычислять значения этих функций при помощи таблицы, и те и другие функции можно вычислять, приближая их многочленами, рациональными дробями и т.д. Таким образом, в класс задач, интегрируемых в явном виде, включались задачи, решения которых выражаются через специальные функции. Однако и этот, более широкий, класс составляет относительно малую долю задач, предъявляемых к решению. Существенное расширение класса реально решаемых дифференциальных уравнений, а следовательно, и расширение

сферы применения математики произошло с разработкой численных методов и активным повсеместным использованием ЭВМ.

Наиболее известные алгоритмы решения систем ОДУ:

- Решение задачи Коши с помощью формулы Тейлора;
- Схемы Эйлера для решения ОДУ;
- Метод Розенброка;
- Модификации методов Рунге-Кутты;

3 Обзор программных инструментов для решения задачи

Python – это простой в освоении язык программирования. Он имеет структуры данных высокого уровня, а также эффективный подход к объектно-ориентированному программированию. Удобный синтаксис и динамическая типизация Python вместе с его интерпретируемой природой делают его идеальным языком для написания сценариев и быстрой разработки приложений во многих областях на большинстве платформ.

Интерпретатор Python и стандартные библиотеки находятся в свободном доступе, в исходной или машинной форме, для всех основных платформ на веб-сайте Python. Этот же сайт содержит также дистрибутивы и указатели на многие свободно распространяемые сторонние модули, программы и инструменты Python, а также дополнительную документацию.

Mpmath – это свободная библиотека Python, для арифметических вычислений с плавающей запятой с произвольной точностью. Она была разработана Фредриком Йоханссоном в 2007 году, при участии других разработчиков. С помощью Mpmath почти любой расчет для действительных и комплексных чисел может быть выполнен с одинаковой точностью независимо от количества знаков после запятой (будь то 10 или 1000).

NumPy – это библиотека с открытым исходным кодом для научных вычислений на Python. Библиотека была разработана в 1995 году Джимом Хаганином.

Цель библиотеки Matplotlib заключается в том, чтобы помочь вам визуализировать ваши данные как можно проще. Она позволяет строить графики, гистограммы, спектры мощности, гистограммы, диаграммы ошибок, диаграммы рассеяния и т.д.

Графический объект matplotlib представляет собой иерархическую структуру. Главной единицей является рисунок (Figure). Рисунок является объектом самого верхнего уровня, на котором располагаются одна или несколько областей рисования (Axes), элементы рисунка Artists (заголовки, легенда и т.д.) и основа-холст (Canvas).

Tkinter (от англ. tk interface) - это графическая библиотека, позволяющая создавать программы с оконным интерфейсом. Эта библиотека является интерфейсом к популярному языку программирования и инструменту создания графических приложений tcl/tk. Tkinter, как и tcl/tk, является кроссплатформенной библиотекой и может быть использована в большинстве распространённых операционных систем (Windows, Linux, Mac OS X и др.).

4 Разработка приложения

В соответствии с поставленными задачами была произведена разработка программы моделирования вакуумного зарождения частиц в электромагнитных полях. При разработке использовались: язык программирования Python, интерпретатор версии 3.6.1, для расчетов библиотеки Mpmath, NumPy, для построения графиков - библиотека Matplotlib. Интерфейс построен с помощью инструментария предоставляемого графической библиотекой Tkinter.

Сначала были проведены исследования зависимости результатов решения от значения параметра точности tol. Из исследования можно сделать вывод, что для небольших исследований, расчетов и построения функции хватает точности $tol = 0.01 - 0.001$, как это написано в документации библиотеки.

Далее было разработана программа для моделирования физического процесса. Внешний вид получившейся программы представлен на рисунке 2.

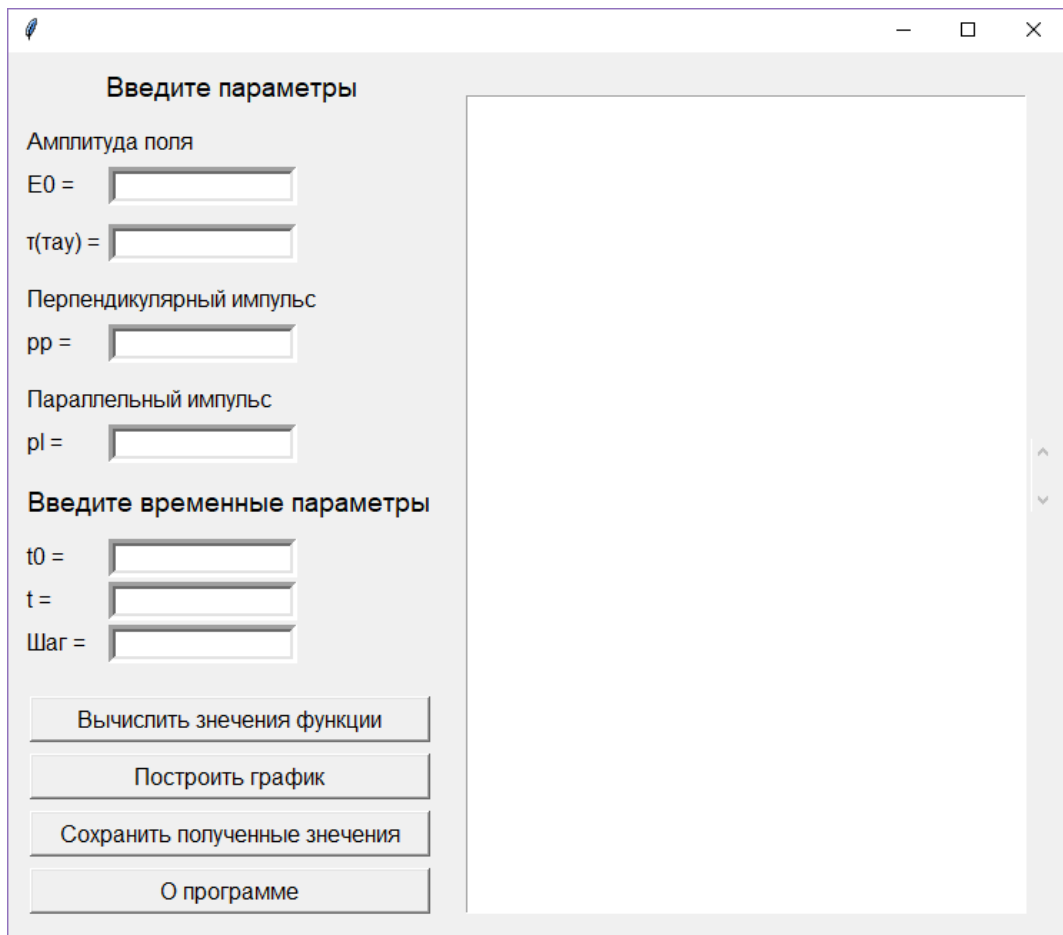


Рисунок 2 – Главное окно приложения.

Перед запуском вычислений значений функции программа проверяет все ли параметры были введены и правильность введенных данных. В случае выявления ошибок, программа укажет их в белом диалоговом поле.

Если все параметры заполнены верно, после нажатия «Вычислить значения функций» запускается процесс вычислений. Данный процесс займет некоторое время, чем больше интервал времени – тем дольше производятся расчеты. По завершению, в белом диалоговом поле отобразятся значения функции в соответствующий момент времени.

После получение на экране успешных вычислений, по полученным значениям можно построить график нажав кнопку «Построить график». После нажатие кнопки откроется окно с графиком. Изображения всех графиков интерактивны. Можно изменять масштаб, двигаться вдоль осей, отменять действия и возвращаться к начальному виду графика. График может быть

сохранен в формате .png или .pdf. Также значения можно сохранить в текстовом файле нажав кнопку «Сохранить полученные значения». Значения сохраняются в файле programm_result.txt, который будет создан в директории с программой.

Значение функции вероятности может достигать очень маленьких значений, что показано на рисунке 3, но не всегда точность позволяет достигнуть столь малых значений. На рисунке 4 можно пронаблюдать, какие результаты получается в результате использования не высокой точности при решении. С помощью используемой библиотеки можно задавать большую точность и приблизится к возможным значениям.

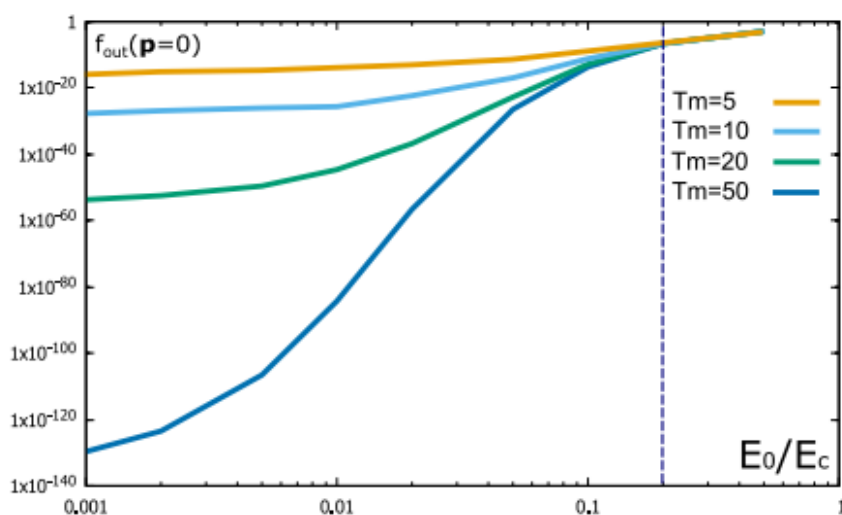


Рисунок 3 – Диапазон возможных значений функции.

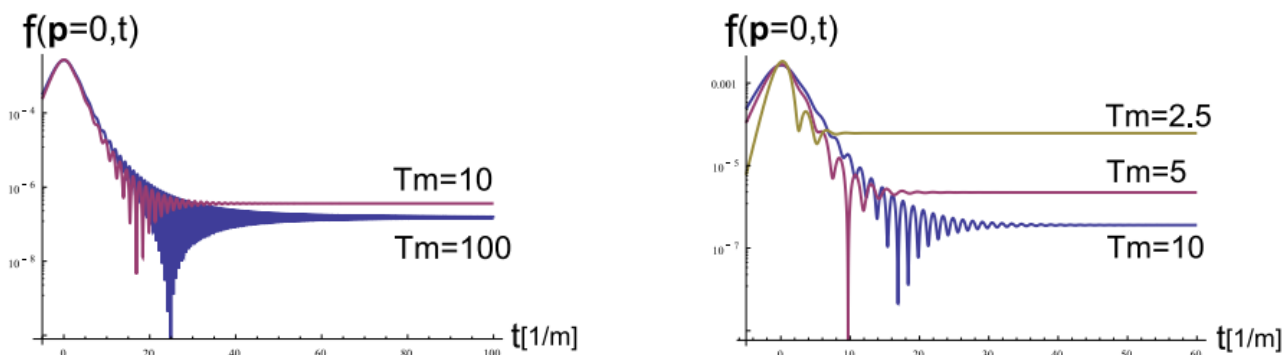


Рисунок 4 – Графики поведения функции.

Для проверки насколько близко будут значения, получаемые в результате работы программы, были запущены вычисления с параметрами, показанными

на рисунке 5. На рисунке 6 показан график построенный по полученным значениям. Точность $\text{tol} = 0.0000001$.

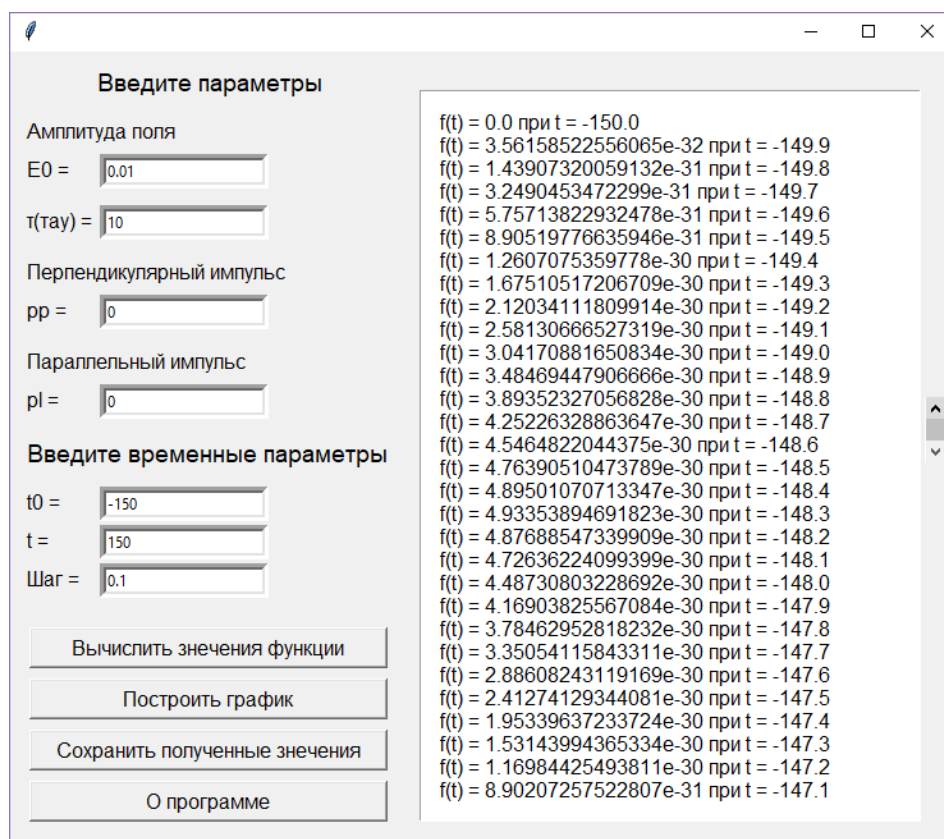


Рисунок 5 – Результат работы программы.

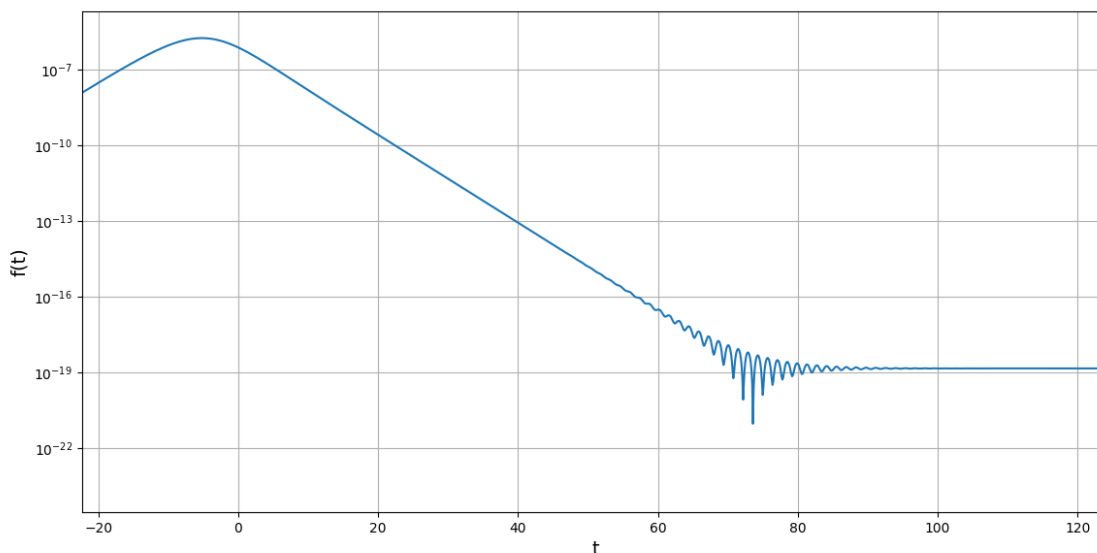


Рисунок 6 – График по полученным расчетам.

Как можно увидеть, увеличение точности позволило приблизиться к возможным значениям функции. В начальный момент времени значения достигли минимальных значений 10^{-32} .

ЗАКЛЮЧЕНИЕ

Поставленная передо мной задача по реализации численной модели физического процесса языковыми средствами Python была успешно решена. Для этого мне потребовалось познакомиться с основными методами численного решения систем обыкновенных дифференциальных уравнений их возможностями и ограничениями, их реализацией в доступных математических библиотеках для выбранного языка программирования.

Программа для моделирования реализована в форме приложения с графическим интерфейсом. Результаты моделирования можно выводить в числовой форме для детального анализа или в графической для качественной оценки результатов.

Предметом исследования в практической части работы была оценка достижимой точности моделирования в области малых значений параметров модели. Верификация полученных результатов осуществлялась по точному аналитическому решению для электрического поля с заданной зависимостью от времени. Было продемонстрировано, что возможности библиотеки Mpmath не позволяют продвинуться в область решений со значениями менее 10^{-20} . Кроме того, пока эта библиотека реализует только один метод решения ОДУ, что тоже существенно сказывается на области допустимых параметров модели.

Проделанная работа продемонстрировала эффективность и удобство использования Python для построения численных моделей достаточно сложных физических процессов. С другой стороны, показана актуальность исследования более широкого круга математических библиотек для эффективного моделирования рассмотренного процесса в широком диапазоне параметров.