

УДК 519.6

АВТОМАТИЗАЦИЯ РЕШЕНИЯ ЗАДАЧ, ОПИСЫВАЕМЫХ ДИФФЕРЕНЦИАЛЬНЫМИ УРАВНЕНИЯМИ ВТОРОГО ПОРЯДКА

Ильичев В.Ю., Назаров Е.С.

*Калужский филиал ФГБОУ ВО «Московский государственный технический университет
имени Н.Э. Баумана (национальный исследовательский университет)»,
Калуга, e-mail: patrol8@yandex.ru*

При решении задач практически во всех отраслях науки и техники используются дифференциальные уравнения второго порядка. Как правило, они описывают изменение функции с течением времени или по какой-либо оси координат. Специализированные математические программы дают возможность решать такие уравнения, однако не во всех случаях. Кроме того, данные программные продукты дороги и отличаются низкой скоростью работы. К настоящему времени всё большей популярностью пользуются универсальные языки программирования, свободно распространяемые в сети Интернет, для которых существует огромное количество библиотек функций, позволяющих решать различные задачи. Среди таких языков программирования наиболее известен Python, для которого есть подключаемые модули, с помощью которых можно расширить функциональность данного языка для решения проблем практически любого типа численными методами. Кроме перечисленных достоинств, по языку Python существует большое количество документации, а также форумов, на которых можно найти ответ на возникающие у программистов вопросы. В качестве основной библиотеки для научных задач, в том числе решения дифференциальных уравнений, часто используется модуль Scipy, выбранный и для данной работы. В качестве дополнительных используются модуль создания и заполнения массивов Numpy и библиотека вывода результатов расчётов в виде качественных графиков Matplotlib. В результате исследований созданы методика, алгоритм и код программы, позволяющие решать практически любые дифференциальные уравнения второго порядка. Описаны результаты апробации данной методики на примере решения дифференциального уравнения Шредингера для волновой функции частицы (электрона), находящейся в прямоугольной потенциальной яме. Также рассчитана полная энергия электрона на разных энергетических уровнях. В конце работы приведены рекомендации по дальнейшему использованию созданной методики и программы, а также в целом по применению функций языка Python.

Ключевые слова: дифференциальное уравнение, уравнение Шредингера, потенциальная яма, язык Python, библиотека Scipy, модуль Numpy, модуль Matplotlib

AUTOMATION OF SOLVING PROBLEMS DESCRIBED BY DIFFERENTIAL EQUATIONS OF SECOND ORDER

Ilichev V.Yu., Nazarov E.S.

Kaluga branch of Bauman Moscow State Technical University, Kaluga, e-mail: patrol8@yandex.ru

In solving problems, differential equations of second order are used in almost all branches of science and technology. They describe change in function over time or along coordinate axis. Specialized mathematical programs make possible to solve such equations, but not in all cases. These software products are expensive and are characterized by low speed. To date, universal programming languages are increasingly popular, freely distributed on Internet, for which there are huge number of function libraries that allow to solve various problems. Among such programming languages, Python is best known, for which there are plugins with which you can expand functionality of this language to solve problems of almost any type by numerical methods. In addition to these advantages, Python has large documentation and forums where you can find answers to questions. As main library for scientific problems, Scipy module, chosen for this work, is used. As additional, module for creating and filling arrays Numpy and library for displaying calculation results in form of quality graphs Matplotlib are used. As result of research, technique, algorithm and program code were created that allows to solve almost any differential equations of second order. Results of this technique are described on example of solution of Schrödinger differential equation for wave function of particle (electron) located in rectangular potential pit. Total energy of electron at different energy levels is calculated. At the end of work, recommendations are made on further use of methodology and program and of use Python language functions in general.

Keywords: differential equation, Schrödinger equation, potential pit, Python language, Scipy library, Numpy module, Matplotlib module

Дифференциальные уравнения второго порядка [1] используются для описания множества физических, технических, биологических, химических, социологических, экономических и прочих видов процессов, явлений или характеристик систем, например движения объектов под действием внешних сил [2], изменения параметров электрических величин, динамики сгорания газообразного топлива, изменения чис-

ленности популяций или энергии микро-частиц. Представление функции процесса или системы в виде дифференциального уравнения означает, что она изменяется в зависимости от изменения какого-либо параметра или нескольких параметров (чаще всего времени и координат).

Основным из типов решения таких дифференциальных уравнений является так называемая задача Коши [3], состоящая

в нахождении основной функции путём интегрирования при некотором заданном наборе начальных условий (чаще всего в момент времени $t = 0$). Также, как правило, на значения искомой функции накладываются некоторые ограничения по границам рассматриваемой области её определения (задаются так называемые граничные условия) [4].

Самые простые из дифференциальных уравнений второго порядка позволяют найти решение точными математическими методами, однако далеко не все. К счастью, к настоящему времени разработано множество программных продуктов, позволяющих решать такие задачи численными методами, при этом не вдаваясь в подробности функционирования данных методов. Наиболее известными программами, позволяющими решать дифференциальные уравнения, являются MathCAD [5] и MatLab [6], однако они отличаются высокой стоимостью лицензий, а также малым быстродействием.

Альтернативным (и всё более популярным) путём решения дифференциальных уравнений является применение специальных научных библиотек для различных языков программирования, в частности для широко известного свободно распространяемого высокоуровневого языка программирования Python [7]. Данный язык отличается от прочих вычислительных средств простотой освоения и постоянным развитием, появлением всё более совершенных версий библиотек математических функций.

Целью описываемого исследования является разработка методики численного решения дифференциальных уравнений второго порядка, создание на её основе алгоритма и кода программы на языке программирования Python. Результаты решения должны выводиться в наглядной графической форме. Для реализации всех описанных функций программы необходимо использовать специально подобранные для этого библиотеки функций (модули) Python. Далее, написанный код программы необходимо апробировать с помощью рассмотрения конкретного физического явления, описываемого дифференциальным уравнением второго порядка.

В качестве такого явления выбрано изучение одной из задач квантовой механики – движения элементарной частицы, находящейся в так называемой одномерной прямоугольной потенциальной яме [8], имеющей бесконечную глубину (с бесконечно высокими непроницаемыми стенками) (рис. 1).

При движении частицы вдоль оси x в области $x = 0 \dots a$ потенциальная энергия частицы $U(x)$ равна нулю, а при $x < 0$ и $x > a$ энергия $U(x) = \infty$.

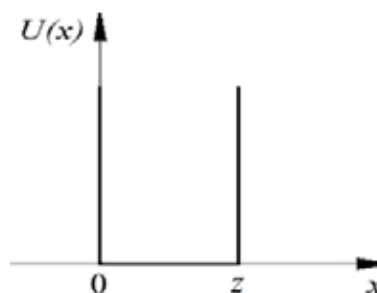


Рис. 1. Прямоугольная потенциальная яма

В результате необходимо получить и изучить так называемую волновую функцию [9] частицы, описывающую чистое состояние (положение) частицы, путём построения графиков волновых функций на разных энергетических уровнях [10]. По графикам можно узнать частоты колебаний частицы (например, электрона) на различных энергетических уровнях. После этого, найдя квадрат значения волновой функции, получим графики, характеризующие плотность вероятности нахождения частицы в точке с заданными координатами в данный момент времени для каждого энергетического уровня.

В ходе решения задачи предполагается также найти полную энергию частицы на разных энергетических уровнях (которая постоянна в пределах $x = 0 \dots a$) и построить графики значений этой энергии. При этом можно выявить особенности движения частицы в прямоугольной потенциальной яме, такие, например, как квантование энергии, вырождение энергетических уровней и т.д.

Материалы и методы исследования

Таким образом, конечным материалом описываемого исследования является разработка кода программы на языке программирования Python, позволяющей решать дифференциальные уравнения второго порядка (содержащие вторую производную от какого-либо параметра). Основным выбранным методом является численное решение на основе библиотеки научных расчётов Scipy для Python, а именно интегрирующей функции Odeint [11] из подмодуля этой библиотеки Scipy.integrate.

Алгоритм разработанной на основе указанных принципов программы состоит из следующих блоков:

1. Загрузка следующих библиотек для языка Python: вышеупомянутого подмодуля Scipy.integrate, модуля Numpy [12] для создания и заполнения массивов данных (результатов расчётов для построения графиков), модуля Math (для использова-

ния некоторых математических функций и констант, не содержащихся в языке Python по умолчанию – например, для вызова числа π), подмодуля Matplotlib.pyplot для представления результатов расчётов в виде двухмерных графических зависимостей.

2. Задание исходных данных, представляющих из себя коэффициенты дифференциального уравнения или уравнений, а также начальные и граничные условия.

3. В случае необходимости – определение формул или условий для нахождения дополнительных функций, используемых в конкретных дифференциальных уравнениях.

4. Создание функции, описывающей левые части дифференциальных уравнений первого порядка, полученных после преобразования исходного дифференциального уравнения второго порядка (методика такого преобразования и состав левых и правых частей уравнений будут продемонстрированы далее на примере).

5. Создание главной функции программы, в рамках которой происходит определение двух массивов, которые далее при выполнении циклических вычислений будут заполнены значениями переменной величины и соответствующей ей функции. Команда определения функции Integrate.odeint включает в качестве аргументов левые части дифференциальных уравнений (п. 4), правые части тех же уравнений (начальные условия) и массив, содержащий значения переменной, при которых следует вычислять функцию.

Далее функция, вычисленная при всех значениях переменной, выводится с использованием графической библиотеки на экран.

6. Вычисление некоторых вспомогательных величин и главной функции, описанной в п. 5 алгоритма.

Далее рассмотрим применение разработанной программы для решения примера из квантовой физики – движение частицы вдоль оси x в пределах изображённой на рис. 1 прямоугольной потенциальной ямы. Данный процесс описывается одномерным дифференциальным уравнением Шредингера:

$$\frac{d^2\psi(x)}{dx^2} + \frac{2m}{\hbar^2} [E - U(x)] \psi(x) = 0,$$

где $\psi(x)$ – волновая функция движения частицы;

x – координата положения частицы;

m – масса частицы (так как в качестве примера элементарной частицы мы рассматриваем электрон, то $m = 9.10938356 \cdot 10^{-31}$ кг);

\hbar – постоянная Планка, равная $1.054571817 \cdot 10^{-34}$ Дж·с;

E – полная энергия частицы, Дж;

$U(x)$ – потенциальная энергия частицы, Дж.

Как видим, представленное дифференциальное уравнение движения элементарной частицы имеет вторую производную по координате x , а следовательно, относится к классу исследуемых в данной работе зависимостей. Однако для применения в программе его необходимо преобразовать – представить в виде системы двух дифференциальных уравнений первого порядка, что делается при помощи простой замены переменных:

$$b(x) = \frac{da(x)}{dx}.$$

Тогда уравнение Шредингера может быть представлено следующим образом (при переносе первого его слагаемого в правую часть):

$$-\frac{2m}{\hbar^2} [E - U(x)] \psi(x) = \frac{db(x)}{dx}.$$

Для решения системы из двух представленных дифференциальных уравнений необходимо задаться начальными и граничными условиями задачи.

Начальными условиями являются следующие: при $b(x) = \frac{da(x)}{dx} = 1$ (можно взять

любое постоянное число – единица выбрана для определённости, и она является правой частью первого уравнения системы) $\frac{db(x)}{dx} = 0$ (правая часть второго уравнения,

проистекающая из рассмотрения уравнения Шредингера).

Граничные условия ясны из рассмотрения рис. 1 и являются начальной и конечной координатами стенок потенциальной ямы: $x = 0$ и $x = z$, где z – любое натуральное число. Для определённости в программе использовано значение $z = 1000$ при шаге вычисления всех параметров системы, равном 1, что вполне достаточно для построения по результатам расчётов плавных графических зависимостей.

Результаты исследования и их обсуждение

Продemonстрируем работу разработанной программы на описанном выше примере – рассмотрении движения элементарной частицы (в качестве которой выбран электрон) в одномерной прямоугольной потенциальной яме.

Для формирования полного набора исходных данных в программу необходи-

мо добавить формулу для полной энергии частицы:

$$E = \frac{\pi^2 \hbar^2}{2mz^2} n^2,$$

где n – целое число, называемое квантовым уровнем энергии частицы. Значение остальных констант, входящих в формулу, приведено выше.

В квантовой теории доказывается, что решение уравнения Шредингера для прямоугольной потенциальной ямы (волновая функция) равно нулю только при дискретных целых значениях n .

Вычисляя с помощью программы E для электрона при $n = 1, 2, 3, 4, 5$ и визуализируя полученные значения с помощью Microsoft Excel, получим следующую картину (рис. 2).

Представленная диаграмма наглядно демонстрирует тот факт, что с увеличением квантового уровня n на единицу энергия частицы возрастает всё более стремительно.

На следующем графике (рис. 3) покажем выводимые программой графики волновых функций для двух значений $n = 2$ и $n = 5$, а также квадратов волновых функций для тех же n . При вычислениях используются значения E , представленные на рис. 2 и соответствующие текущему n (программа показывает, что при других значениях E волновая функция во всём диапазоне x от 0 до z равна нулю).

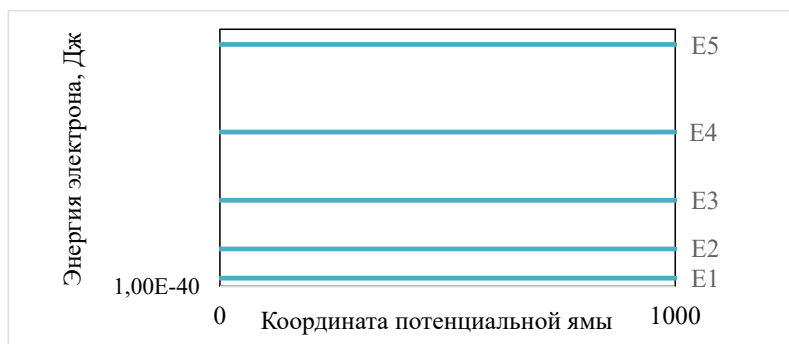


Рис. 2. Вычисленные квантовые уровни энергии электрона

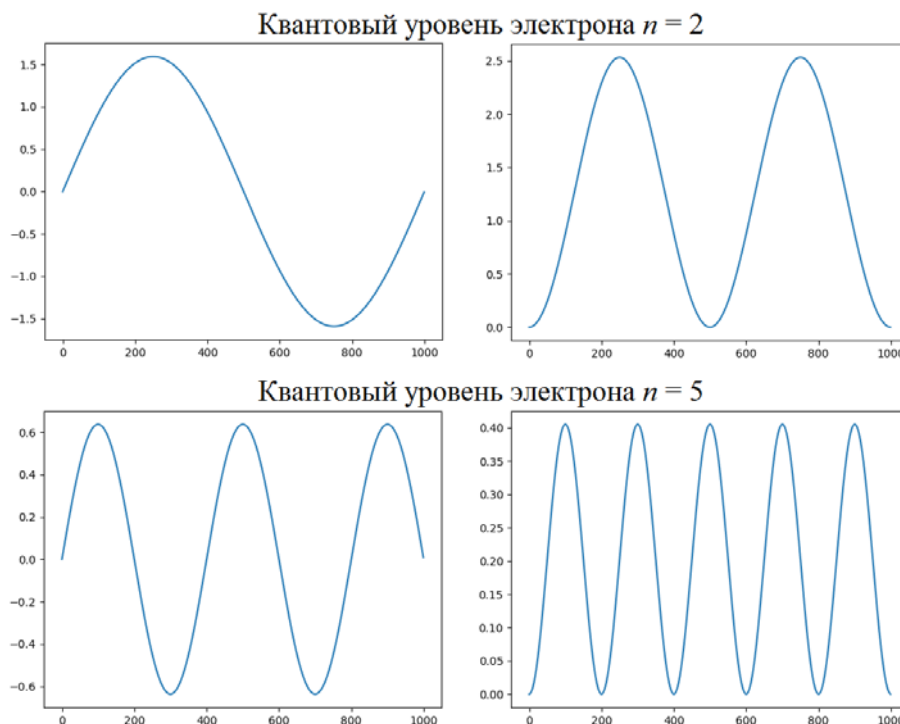


Рис. 3. Графики волновых функций и их квадратов для электрона

Таким образом, по осям абсцисс отложено ширина потенциальной ямы в условных координатах для двух значений квантовых уровней. Правые графики, изображающие изменение квадрата волновой функции ψ^2 , позволяют судить о вероятности нахождения частицы в разных местах по координате x (по относительному значению по оси ординат). В этом состоит основной физический смысл построения таких функций. В данном случае наглядно видно, что количество координат x , соответствующих максимальной вероятности нахождения электрона, равно номеру рассматриваемого квантового уровня n .

Заключение

Полученные в ходе проведения работы методика и программа решения дифференциальных уравнений второго порядка могут быть использованы в любых подобных рассмотренному случаях с минимальными доработками. Для разных дифференциальных уравнений второго порядка в программе необходимо только прописать свои выражения для правых и левых их частей, а также изменить начальные и граничные условия.

Выводы по результатам решения уравнений можно сделать на основе выводимых программой наглядных графических зависимостей искомой функции от аргумента. Описанная в статье апробация разработанной методики позволяет рекомендовать использовать язык Python совместно с описанными библиотеками функций (главной из которых является модуль Scipy) для решения рассмотренного круга задач благодаря следующим доказанным достоинствам данного подхода:

- 1) простота кода программы и его модификации под разные задачи;
- 2) высокая скорость вычислений;
- 3) наглядная визуализация результатов;
- 4) бесплатность программных продуктов и их доступность для всех желающих;
- 5) высокая степень вовлечённости исследователя в процесс формулирования и решения задач, что особенно важно для достижения понимания физико-матема-

тических связей в исследуемых проблемах, особенно для начинающих.

С помощью языка Python возможно (и запланировано) дальнейшее совершенствование разработанного программного продукта для решения более сложных классов уравнений, например систем дифференциальных уравнений в частных производных.

Список литературы

1. Океанов Е.Н. Метод решения линейного дифференциального уравнения второго порядка // Актуальные проблемы современной науки. 2019. № 3 (106). С. 96–101.
2. Свирский М.С., Свирская Л.М. Второй закон Ньютона в квантовой механике // Современные наукоемкие технологии. 2008. № 4. С. 104.
3. Абрамова В.В. Численное решение задачи Коши для обыкновенного дифференциального уравнения с параметром при производной // Евразийский союз ученых. 2020. № 2–6. С. 35–39.
4. Белубекян М.В., Терзян С.А. Влияние граничных условий на условия появления локализованной неустойчивости прямоугольной пластинки // Актуальные проблемы механики сплошной среды: материалы VI международной конференции. 2019. С. 60–63.
5. Павлова Т.А., Уварова М.Н. Использование программы MathCAD при решении прикладных задач // Ученые записки Орловского государственного университета. 2018. № 3 (80). С. 322–325.
6. Бебихов Ю.В., Семёнов А.С., Семёнова М.Н., Якушев И.А. Анализ методов моделирования технических систем в среде MatLab // Моделирование, оптимизация и информационные технологии. 2019. Т. 7. № 3 (26). С. 12.
7. Ильичев В.Ю. Использование алгоритма дифференциальной эволюции для решения оптимизационных задач // Системный администратор. 2021. № 4 (221). С. 80–83.
8. Комаров Д.А., Морев С.П., Дармаев А.Н., Ряднов А.Ю. Численное решение уравнения Шредингера для электрона в потенциальной яме произвольной формы // Радиотехника и электроника. 2014. Т. 59. № 8. С. 799.
9. Пупышев В.В. Двумерное кулоновское рассеяние квантовой частицы: волновые функции и функции Грина // Теоретическая и математическая физика. 2016. Т. 186. № 2. С. 252–271.
10. Коваль О.А., Коваль Е.А. Численный анализ энергетических уровней квантовой частицы в поле двумерного диполя // Вестник Московского государственного областного университета. Серия: Физика – математика. 2020. № 3. С. 23–37.
11. Ильичев В.Ю., Чухраев И.В. Использование библиотеки Scipy для языка Python с целью изучения параметров затухающего гармонического осциллятора // E-Scio. 2021. № 7 (58). С. 301–310.
12. Ilchev V.Yu. Development of program for determination of fractal dimensions of images. International Research Journal. 2021. no. 4–1 (106). P. 6–10.