

контекст. Такой подход расширяет функциональность системного ПО и делает его более персонализированным.

Интерфейсы продолжают эволюционировать. Сегодня появляются голосовые ассистенты, контекстные панели, интерфейсы дополненной реальности (AR) и виртуальной реальности (VR). Искусственный интеллект помогает предугадывать действия пользователя, предлагает команды, автоматизирует рутинные задачи.

В системном программном обеспечении это проявляется в виде интеллектуальных помощников, визуализации логов и метрик, адаптивных панелей управления. Интерфейс становится не просто оболочкой, а активным участником взаимодействия, способным обучаться и подстраиваться под пользователя.

В ходе проведенного исследования были систематизированы ключевые этапы эволюции интерфейсов системного программного обеспечения: от командной строки (CLI) к графическим интерфейсам (GUI), а затем к гибридным и интеллектуальным решениям. Установлено, что каждый этап развития интерфейсов отражает не только технологические изменения, но и трансформацию пользовательских ожиданий, уровня подготовки и задач, решаемых с помощью системного ПО.

Полученные результаты позволяют выделить следующие закономерности: CLI-интерфейсы обеспечивают высокую точность и гибкость, оставаясь актуальными в профессиональной среде; GUI-интерфейсы значительно расширили аудиторию пользователей, сделав системное ПО доступным для непрофессионалов; гибридные и интеллектуальные интерфейсы обеспечивают персонализацию, адаптацию под контекст и интеграцию с современными технологиями, включая искусственный интеллект и облачные решения.

Таким образом, интерфейс системного программного обеспечения эволюционирует от инструмента управления к интеллектуальному посреднику, способному активно участвовать в процессе взаимодействия, обучаться и подстраиваться под пользователя. Это открывает перспективы для дальнейших исследований в области проектирования адаптивных, безопасных и инклюзивных интерфейсов, ориентированных на разнообразные сценарии использования.

Литература

1. Как появился графический интерфейс пользователя: история в лицах, деталях, фактах и курсорах / Хабр. – URL: <https://habr.com/ru/companies/first/articles/784820/> (дата обращения: 07.10.2025).
2. Эволюция пользовательских интерфейсов: От кнопок к голосовому управлению / Young Senior | Дзен. – URL: <https://dzen.ru/a/ZpZ1BCKFNVqbxoM4> (дата обращения: 07.10.2025).

ВИЗУАЛИЗАЦИЯ УСЛОВИЙ КВАНТОВАНИЯ ДЛЯ ЧЕТНЫХ И НЕЧЕТНЫХ ФУНКЦИЙ: ИДЕЯ ИНТЕРАКТИВНОГО ПРИЛОЖЕНИЯ

В. И. Дубовик, В. Н. Капшай

*Гомельский государственный университет имени Франциска Скорины,
Республика Беларусь*

Предложена идея по созданию приложения на языке Python, которое наглядно покажет, как квантуются четные и нечетные функции. Оно будет отображать спектры и граничные условия для таких симметричных функций, а пользователи смогут менять параметры и смотреть, как симметрия влияет на общее условие квантования.

Ключевые слова: квантование, четные функции, нечетные функции, граничные условия, интерактивное приложение, численные методы.

VISUALIZATION OF QUANTIZATION CONDITIONS FOR EVEN AND ODD FUNCTIONS: INTERACTIVE APPLICATION**V. I. Dubovik, V. N. Kapshay***Francisk Skorina Gomel State University, Republic of Belarus*

The idea of creating a Python application is proposed, which will clearly show how even and odd functions are quantized. It will display the spectra and boundary conditions for such symmetric functions, and users will be able to change the parameters and see how the symmetry affects the overall quantization condition.

Keywords: quantization, even functions, odd functions, boundary conditions, interactive application, numerical methods.

Цель проекта заключается в создании программного обеспечения, позволяющего пользователям понять влияние симметрии и граничных условий на общие условия квантования четных и нечетных функций.

Значение проекта обусловлено растущим спросом на визуальные и интерактивные средства обучения и анализа сложных математических явлений. В условиях внедрения цифровых технологий в образование и научные исследования подобные приложения позволяют объединить теоретические знания с практическими навыками, делая абстрактные концепции доступными и наглядными.

Создание приложения, которое помогает визуализировать условия квантования четных и нечетных функций, – это не просто технический проект, а способ сделать сложную математику доступной и наглядной. Такое решение может пригодиться тем, кто изучает физику или математику, преподает эти дисциплины или занимается прикладными исследованиями. Возможность менять параметры модели и сразу видеть, как это отражается на графиках, превращает сухую теорию в понятный процесс, где абстрактные формулы обретают форму и смысл на экране.

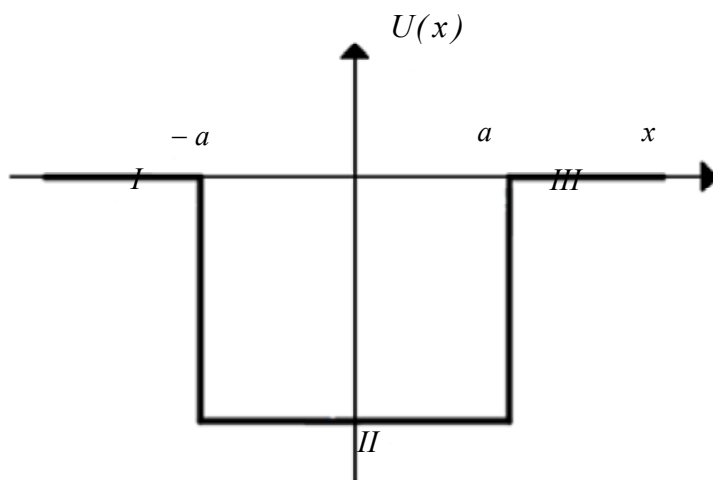


Рис. 1. Одномерная прямоугольная потенциальная яма

Общее условие квантования в задачах с симметричными функциями формируется на основе граничных условий и свойств оператора. Показать четность и нечетность волновой функции возможно при комбинировании условий сшивания в граничных точках, показанных на рис. 1 (используется свойство непрерывности

волновой функции и ее первой производной). После нормировки волновой функции можно вывести общее условие квантования:

$$(\kappa \cdot \cos(ak) - k \cdot \sin(ak))(k \cdot \cos(ak) + \kappa \cdot \sin(ak)) = 0, \quad (1)$$

которое разбивается на два альтернативных варианта [1]:

$$\kappa \cdot \cos(ak) - k \cdot \sin(ak) = 0; \quad (2)$$

$$k \cdot \cos(ak) + \kappa \cdot \sin(ak) = 0. \quad (3)$$

Чтобы построить графики условий квантования для четных и нечетных функций, нужно разобраться с уравнениями, которые описывают, при каких параметрах система «разрешает» те или иные значения, например, энергии. Эти уравнения зависят от граничных условий и симметрии функции, и чтобы найти нужные значения, приходится прибегать к численным методам.

Создание приложения на Python для визуализации условий квантования четных и нечетных функций – это способ превратить сложные математические уравнения в понятные графики, с которыми можно работать и экспериментировать. В основе лежат формулы, описывающие, при каких значениях энергии система «разрешает» существование устойчивых состояний, и чтобы найти эти значения, используются численные методы. С помощью библиотек SciPy и NumPy можно быстро вычислить корни уравнений, а Matplotlib или Plotly помогают превратить эти расчеты в наглядные графики.

С помощью Python и библиотек SciPy, NumPy и Matplotlib можно точно рассчитывать собственные значения и визуализировать поведение функций при разных граничных условиях. Это делает квантовые явления более понятными и доступными для анализа и обучения.

Литература

1. Griffiths, David J. Introduction to Quantum Mechanics / David J. Griffiths, Darrell F. Schroeter. – Cambridge : Cambridge University Press, 2018. – 508 с.

АСИНХРОННЫЙ ПОДХОД В ПРОГРАММИРОВАНИИ

Н. С. Железко, С. А. Лукашевич

*Гомельский государственный университет имени Франциска Скорины,
Республика Беларусь*

Рассмотрены преимущества и недостатки асинхронного подхода с практическим примером реализации на Python.

Ключевые слова: python, асинхронность, AsyncIO, await, корутина, поллинг.

ASYNCHRONOUS APPROACH TO PROGRAMMING

M. S. Zhelezko, S. A. Lukashevich

Francisk Skorina Gomel State University, Republic of Belarus

Discusses the advantages and disadvantages of the asynchronous approach with a practical example of implementation in Python.

Keywords: python, asynchrony, AsyncIO, await, coroutine, polling.