

## МОДЕЛИРОВАНИЕ КВАНТОВЫХ СИСТЕМ С ПОМОЩЬЮ GNU OCTAVE И PYTHON

Тырышкин С.Ю.

ФГБОУ ВО «Алтайский государственный технический университет имени И.И. Ползунова»,  
Барнаул, e-mail: service.vip-spe@yandex.ru

Статья посвящена изучению особенностей моделирования квантовых систем с использованием системы компьютерной алгебры GNU Octave и языка программирования Python, а также рассмотрению различных численных алгоритмов и инструментов для эффективного моделирования. Исследование основано на применении подходов и инструментов, таких как GNU Octave, позволяющий проводить численные вычисления и оперировать матричными структурами, и специальных пакетов, таких как Quantum Package for Octave (QPO) и MATLAB Quantum Computing Toolbox. Рассматривались также симуляторы квантовых вычислений QuEST и ProjectQ, написанные на Python и C++. Примеры схем подготовки квантово-запутанных состояний, таких как состояние Белла и состояние GHZ, иллюстрируют принципы моделирования. В результате исследования установлено, что GNU Octave обладает достаточной гибкостью и переносимостью для моделирования до 10 кубитов, а также поддерживает работу со смешанными состояниями (матрицами плотности). Применение специальных пакетов и библиотек позволило реализовать сложные квантовые операции, такие как подготовка чистых состояний и управление квантовыми воротами. Анализ симуляторов QuEST и ProjectQ показал различия в потреблении памяти и производительности, что важно учитывать при выборе инструмента для конкретных задач моделирования. Несмотря на высокую точность моделирования, существующие симуляторы имеют ограничения, связанные с большими объемами потребляемой памяти и сложностью масштабируемости. Проведенное исследование демонстрирует эффективность моделирования квантовых систем на классических компьютерах с использованием GNU Octave и Python для изучения квантовых алгоритмов и проверки гипотез в области квантовых технологий. Дальнейшее развитие методов и инструментов моделирования позволит улучшить производительность и снизить требования к ресурсам, что станет важным шагом на пути к созданию полноценных квантовых компьютеров.

**Ключевые слова:** моделирование, квантовые системы, квантовые вычисления, квантовая цепь

## MODELING QUANTUM SYSTEMS USING GNU OCTAVE AND PYTHON

Tyryshkin S.Yu.

Polzunov Altai State Technical University, Barnaul, e-mail: service.vip-spe@yandex.ru

The article is devoted to the study of the features of modeling quantum systems using the GNU Octave computer algebra system and the Python programming language, as well as the consideration of various numerical algorithms and tools for effective modeling. The research is based on the application of approaches and tools such as GNU Octave, which allows performing numerical calculations and operating with matrix structures, and special packages such as the Quantum Package for Octave (QPO) and the MATLAB Quantum Computing Toolbox. Simulators for quantum computing, QuEST and ProjectQ, written in Python and C++, were also considered. Examples of schemes for preparing entangled quantum states, such as Bell state and GHZ state, illustrate the principles of modeling. As a result of the study, it was established that GNU Octave has sufficient flexibility and portability for simulating up to 10 qubits, and also supports working with mixed states (density matrices). The use of special packages and libraries made it possible to implement complex quantum operations, such as preparation of pure states and control of quantum gates. Analysis of QuEST and ProjectQ simulators showed differences in memory consumption and performance, which are important to consider when choosing a tool for specific modeling tasks. Despite the high accuracy of simulation, existing simulators have limitations associated with large amounts of consumed memory and scalability complexity. The research conducted demonstrates the effectiveness of modeling quantum systems on classical computers using GNU Octave and Python for studying quantum algorithms and testing hypotheses in the field of quantum technologies. Further development of modeling methods and tools will improve performance and reduce resource requirements, which will be an important step towards creating full-fledged quantum computers.

**Keywords:** modeling, quantum systems, quantum computing, quantum circuit

### Введение

За последние несколько лет появилась и получила широкое распространение революционная парадигма вычислений, основу которой, в отличие от классических подходов, таких как модель фон Неймана, составляет квантовая механика, а не классическая физика [1]. Как известно, квантовые компьютеры дают возможность находить решения ряда задач, которые были до этого неразрешимы, например факторизация боль-

ших чисел, что является центральной проблемой безопасного шифрования данных. В то же время, несмотря на то, что развитие технологий, позволяющих создавать квантовые устройства, все еще находится в самом начале, сегодня уже можно констатировать определенный прогресс в проектировании и управлении квантовыми системами. К примеру, в начале 2024 г. компания IBM объявила [2] о том, что ей удалось создать рабочую квантовую схему, которая смогла

успешно разложить 15 на 3 и 5 благодаря применению технологии ядерно-магнитного резонанса.

В контексте вышеизложенного следует отметить, что моделирование квантовых компьютеров представляет собой очень важную задачу в квантовой информатике. Следует принять во внимание тот факт, что полномасштабный квантовый компьютер пока еще не воплощен в промышленных масштабах, поэтому моделирование квантовых компьютеров в настоящее время происходит на классических вычислительных машинах. В настоящее время моделирование квантовых систем – это едва ли не единственный метод, который позволяет протестировать имеющиеся квантовые алгоритмы и фиксировать квантовые эффекты, связанные с передачей и обработкой квантовой информации. Однако, хотя такой способ доступен, у него есть один существенный недостаток: для проведения моделирования на классическом оборудовании необходимы огромные вычислительные ресурсы, что в результате делает широкий спектр задач моделирования просто невыполнимым.

Ряд наметившихся последних тенденций, связанных с автоматизацией электронного проектирования, позволяет значительным образом расширить диапазон автоматизированного моделирования, а именно до:

(i) символического моделирования высокоуровневых программ на Verilog или даже C++;

(ii) решателей полевых уравнений.

В данном контексте, по мнению автора, особого внимания заслуживает альтернатива квантовых симуляторов с открытым исходным кодом, которая может быть реализована с использованием инструментов GNU Octave и Python. Таким образом, исследования в данном направлении представляются актуальными, теоретически и практически значимыми, что и предопределило выбор темы данной статьи.

В то же время, несмотря на то, что сегодня существует ряд обзоров, в которых рассматриваются теоретические и/или экспериментальные аспекты квантовых вычислений, вопрос обсуждения методов моделирования идеальных и физических моделей квантовых компьютеров остается открытым. Также отдельного внимания заслуживают задачи сравнения различных симуляторов с точки зрения их производительности. В дальнейшем развитии нуждаются методы улучшения асимптотического времени и сложности памяти квантового моделирования в случаях, когда задействованные квантовые операторы демонстрируют значительную структуру.

Таким образом, **цель исследования** заключается в рассмотрении особенностей моделирования квантовых систем с помощью GNU Octave и Python.

### Материалы и методы исследования

Основными методами исследования в этой работе являются: методы интеллектуального анализа; методы объектно-ориентированного и функционального анализа; методы численного моделирования; сравнительный анализ.

### Результаты исследования и их обсуждение

GNU Octave представляет собой систему компьютерной алгебры (CAS), которая используется для проведения численных, а не символьных вычислений. Главная структура данных в Octave – это матрица, но система также позволяет оперировать C-подобными структурами. Синтаксис языка Octave имеет много схожих черт с языком MatLab. GNU Octave позволяет наблюдать унитарную эволюцию квантовых систем [3]. Этот пакет для моделирования был выбран как инструмент реализации простых вычислений на чистых и смешанных квантовых состояниях, а GNU Octave определена в качестве целевой CAS из-за ее гибкости и переносимости.

С помощью GNU Octave можно моделировать до 10 кубитов. Достоинством GNU Octave является то, что данная система позволяет включить формализм смешанных состояний (матриц плотности), что дает возможность пользователю оперировать на разных уровнях абстракции. Функции нижнего уровня обеспечивают процессы сбора сложных квантовых ворот, в том числе ворот с произвольными управляемыми и целевыми кубитами, также они позволяют готовить чистые состояния и произвольные смеси чистых состояний.

Примерами команд нижнего уровня являются:

– Ket – производит чистое квантовое состояние,

– State – выдает матрицу плотности для данного чистого состояния,

– MixStates – готовит смесь матриц плотности,

– ProductGate – дает возможность построить множество квантовых ворот,

– ControlledGate – позволяет построить управляемые ворота с множеством управляющих и целевых кубитов,

– CNot = ControlledGate.

Последнее равенство указывает на то, что CNot – это сокращение от ControlledGate. Функция принимает в качестве аргументов

ControlledGate. Первый аргумент определяет размер выходного гейта, а второй – элементарный гейт, которым нужно управлять. В качестве второго аргумента функции ControlledGate можно использовать любые однокубитовые ворота. Третий и четвертый аргументы этой функции представляют управляемый и целевой кубиты соответственно, при этом функция не будет работать правильно, если они перекрываются [4].

Для моделирования квантовых вычислений в GNU Octave можно использовать различные специализированные пакеты и библиотеки. Ниже приведен список библиотек для GNU Octave, применяемых для моделирования квантовых вычислений, с краткими описаниями.

1. Quantum Package for Octave (QPO). Это специализированный пакет для моделирования и симуляции квантовых систем в Octave. Он содержит множество функций для создания квантовых состояний, операторов, проведения измерений и анализа результатов.

2. QuTiP (Quantum Toolbox in Python). Хотя QuTiP изначально разрабатывался для Python, он также может использоваться в Octave благодаря совместимости синтаксиса и возможности вызова Python-скриптов из Octave.

3. MATLAB Quantum Computing Toolbox. Несмотря на то, что этот инструмент

предназначен для MATLAB, многие его функции могут быть перенесены в Octave с минимальными изменениями.

Рассмотрим конкретные примеры практической реализации моделирования квантовых систем с помощью GNU Octave.

#### Схема подготовки состояния Белла

Схема подготовки состояния Белла (EPR) применяется для целей генерации двух максимально запутанных квантовых состояний (EPR-пары). Цель использования данной схемы заключается в проверке неравенств Белла в квантовой механике [5, 6]. Она состоит из ворот CNOT и ворот Хадмарда. Математическое представление этих ворот имеет следующий вид:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

На рис. 1 показана схема подготовки состояния Белла.

---

#### Алгоритм 1 Алгоритм генерации пар EPR

---

- 1: **начало**
  - 2: ввод  $|a\rangle$
  - 3: ввод  $|b\rangle$
  - 4: применить H к
  - 5: вычислить тензорное произведение  $|a\rangle$  и  $|b\rangle$
  - 6: применить CNOT к произведению
  - 7: **конец**
- 

Рис. 1. Алгоритм генерации EPR

---

#### Алгоритм 2 Алгоритм подготовки спутанности 3 кубитов

---

- 1: **старт**
  - 2: ввод  $|a\rangle$
  - 3: ввод  $|b\rangle$
  - 4: ввод  $|c\rangle$
  - 5: применить H к  $|a\rangle$
  - 6: вычислить тензорное произведение  $|a\rangle$  и  $|b\rangle$
  - 7: применить CNOT к произведению
  - 8: вычислить тензорное произведение результата и  $|c\rangle$
  - 9: применить CNOT к  $|b\rangle$  и  $|c\rangle$
  - 10: **конец**
- 

Рис. 2. Алгоритм подготовки спутанности 3 кубитов

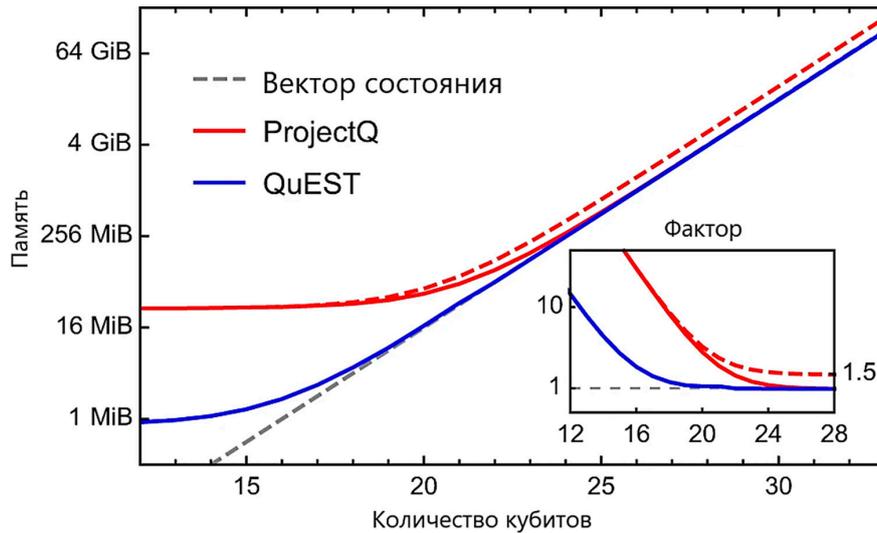


Рис. 3. Потребление памяти процессами QuEST и ProjectQ на Python, о котором сообщает Linux's/proc/self/status во время моделирования случайных цепей на одном вычислительном узле ARCUS с памятью 256 ГБ

### Схема подготовки состояний GHZ

Схема GHZ (названная в честь ее создателей Гринбергера, Хорна и Цайлингера) используется для генерации трех или более максимально запутанных квантовых состояний. В ней также применяются ворота Хадамарда и CNOT [7–9]. На рис. 2 показана схема, которая используется для генерации трех максимально запутанных кубитов и подготовки трехквантового состояния GHZ.

Также отдельного внимания заслуживают симуляторы квантовых вычислений, которые могут найти свое эффективное применение для оптимизации производительности. Например, QuEST и ProjectQ — они представляют собой квантовую вычислительную среду с открытым исходным кодом, включающие компилятор, ориентированный на квантовое оборудование, и квантовый компьютерный симулятор C++ с интерфейсом Python. Оба симулятора хранят чистое состояние в  $2^n$  комплексных числах с плавающей запятой для системы из  $n$  кубитов, с двойной точностью в каждой мнимой и вещественной компоненте. В случае необходимости QuEST может быть настроен на применение четверной или одинарной точности. QuEST и ProjectQ хранят состояние в примитивах C/C++, и поэтому (по умолчанию) потребляют  $16 \times 2^n$  B32 только в векторе состояния. Однако для ProjectQ могут быть характерны накладные расходы памяти  $\times 1,5$  когда осуществляется распределение состояния [10]. Традиционные затраты памяти, характерные для обоих симуляторов в рамках одного

потока представлены на рис. 3, как видно, они незначительно отличаются от затрат в многопоточном режиме.

В то время как QuEST позволяет напрямую читать и записывать вектор состояния, в ProjectQ выборка одной амплитуды имеет накладные расходы на Python, а запись поддерживается только в пакетном режиме, что требует больших затрат памяти, поскольку объекты Python занимают больше памяти, чем сопоставимый примитив C — до  $3 \times 33,34$ . Таким образом, итерация вектора состояния в ProjectQ либо очень медленная, либо требует значительных затрат памяти, что может ограничить ее полезность в некоторых приложениях моделирования.

Стоит отметить, что наряду с рассмотренными программными пакетами и языками программирования существует большое множество других инструментов моделирования квантовых вычислительных процессов. Многие современные компании занимаются перспективными разработками в данном направлении.

### Заключение

Квантовые вычисления — это новая парадигма вычислений, использующая принципы квантовой механики. Ее теоретическая основа допускает вычислительную мощность, экспоненциально превышающую классические компьютеры. Однако, учитывая зарождающуюся стадию данной парадигмы и технические трудности в поддержании квантовых битов (кубитов), практические квантовые компьютеры все еще находятся в экспериментальной разработке.

В данном контексте особое значение приобретает моделирование квантовых систем.

В статье рассмотрены особенности моделирования квантовых схем с использованием GNU Octave и Python. Эти инструменты для квантового точного моделирования позволяют проводить эффективные симуляции квантовых систем, а также реализовывать вычисления со смешанными состояниями.

#### Список литературы

1. Пантелеев П.А. Тензорные сети для эффективной симуляции квантовых вычислений // Интеллектуальные системы. Теория и приложения. 2023. Т. 27, № 1. С. 152.
2. Eun Oh, Xuanying Lai Distributed Quantum Computing with Photons and Atomic Memories // Advanced Quantum Technologies. 2023. Vol. 6, Is. 6. P. 87–94.
3. Якубова Н.С., Абдурасулова Г.Э. Исследование нечетких регуляторов в системах интеллектуального управления на основе квантовых вычислений // Universum: технические науки. 2023. № 3–1 (108). С. 22–25.
4. Huynh T.T. Tran, Hieu T. Nguyen, Long Thanh Vu, Samuel T. Ojetola. Solving differential-algebraic equations in power system dynamic analysis with quantum computing // Energy Conversion and Economics. 2024. Vol. 5, Is. 1. P. 120–127.
5. Нильсен М.А., Чанг И.Л. Квантовая информация и квантовые вычисления. М.: Мир, 2006. 824 с.
6. Ефимов А.И. Основы квантовых вычислений. М.: Физматлит, 2020. 320 с.
7. Гринбергер Д., Хорн М., Цайлингер А. Bell's Theorem without Inequalities // Physical Review Letters. 1990. Vol. 65, Is. 13. P. 33–40.
8. Гринштейн Дж. Квантовые вычисления для программистов / Пер. с англ. А.А. Слинкина. СПб.: Питер, 2019. 352 с.
9. Нильсен М., Чанг И. Квантовая информация и квантовые вычисления / Пер. с англ. Е.Н. Петровой. Долгопрудный: Интеллект, 2006. 824 с.
10. Vaishali Sood, Rishi Pal Chauhan. Progress and prospects of quantum computing in sustainable development: An analytical review // Expert Systems. 2023. Vol. 41, Is. 7. P. 26–34.