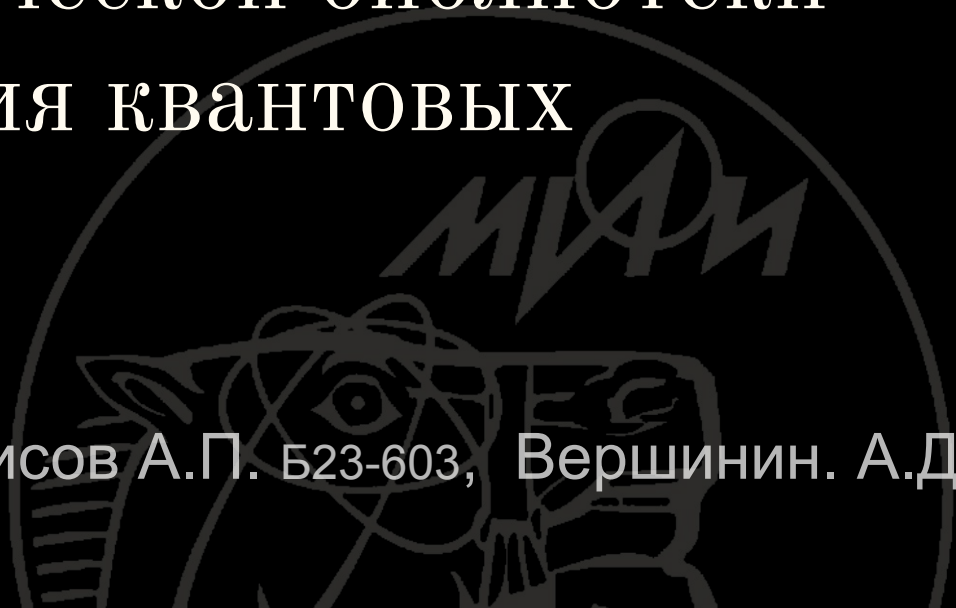


Летняя научная сессия  
СНО НИЯУ МИФИ — 2025

# Создание динамической библиотеки для моделирования квантовых систем

Подготовили:

Кузнецов А.А. Б23-222, Денисов А.П. Б23-603, Вершинин. А.Д.  
Б24-504



## Актуальность:

Мы живем в век квантовых технологий. Все чаще становится необходимо моделировать квантовые системы для разработки новых материалов, элементов квантовых вычислителей и симуляторов и так далее. Однако моделирование таких систем зачастую требует больших вычислительных мощностей. Данный проект возможным решением данной проблемы.

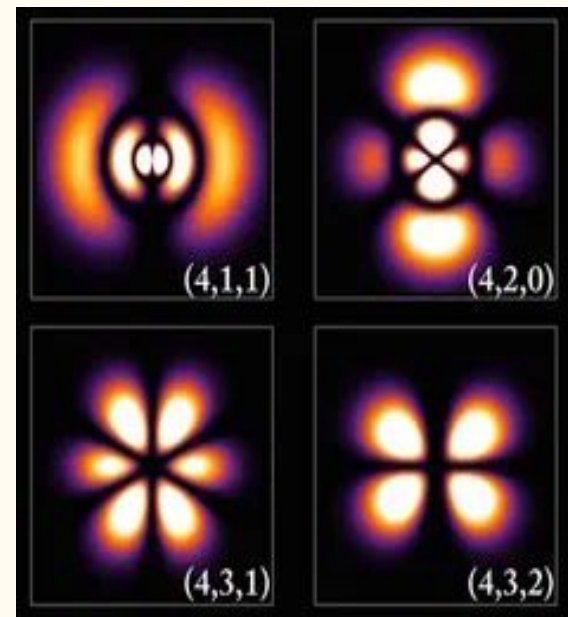


График волновой функции водорода, смоделированный на компьютере.

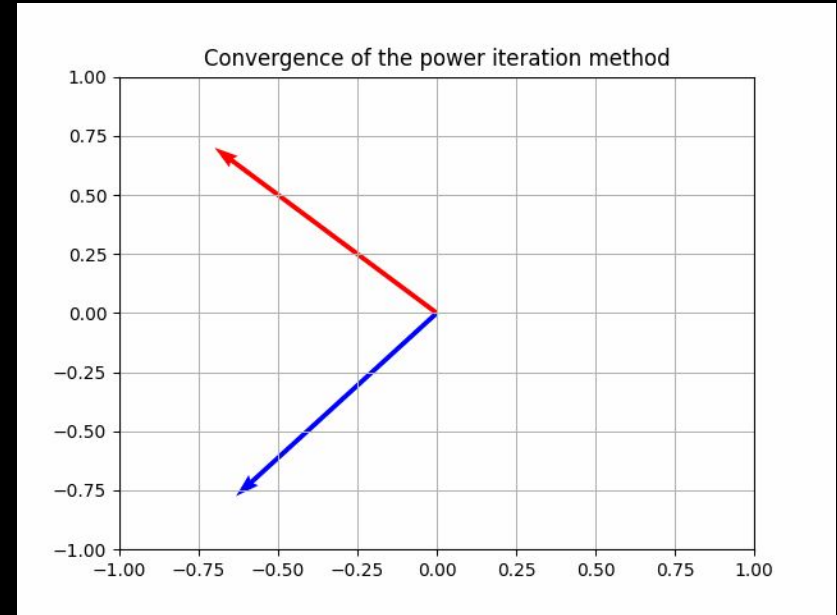
# Цели и задачи



- Изучение алгоритма составления матрицы Гамильтона, описывающей заданную квантовую систему.
  - Поиск подходящего для актуальных задач алгоритмов.
  - Разработка динамической библиотеки на языке C.
  - Сравнение эффективности.
-

# Алгоритм вычисления собственных векторов матрицы:

- 1) Создать единичного вектора.
- 2) Умножить вектор на матрицу слева.
- 3) Нормировать вектор.
- 4) Перейти к пункту один, если после пункта 3 отклонение меньше заданного.



Иллюстрация

$$\hat{H} = \frac{4e^2}{3C}(\hat{n}_1^2 + \hat{n}_1\hat{n}_2 + \hat{n}_2^2) + E_j(-\cos\hat{\varphi}_1 - \cos\hat{\varphi}_2 - \cos(\hat{\varphi}_2 - \hat{\varphi}_1))$$

$$\hat{\varphi}_1 = \hat{\varphi} \otimes \hat{I} \quad \hat{n}_1 = \hat{n} \otimes \hat{I} \quad \hat{\varphi} = 2\pi \frac{\hat{\phi}}{\phi_0}$$

$$\hat{\varphi}_2 = \hat{I} \otimes \hat{\varphi} \quad \hat{n}_2 = \hat{I} \otimes \hat{n} \quad \hat{Q} = 2e\hat{n}$$

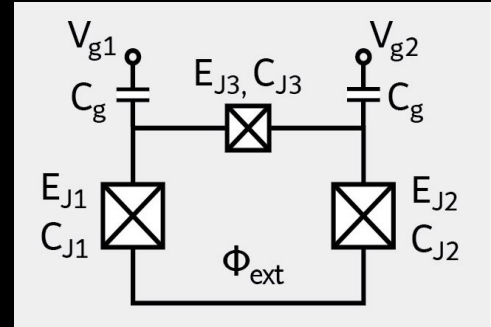
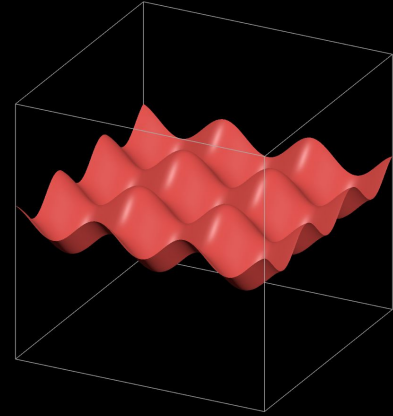
$$\hat{H} |\psi\rangle = E |\psi\rangle$$

$|\psi\rangle$  - Собственный вектор

$$\begin{pmatrix} n_0 & & & 0 \\ & n_1 & & \\ & & \ddots & \\ 0 & & & n_N \end{pmatrix} = \hat{n} \begin{pmatrix} 0 & \frac{1}{2\Delta x} & & \\ \frac{1}{2\Delta x} & 0 & \frac{1}{2\Delta x} & \\ & \frac{1}{2\Delta x} & \ddots & \ddots \\ & & \ddots & 0 & \frac{1}{2\Delta x} \\ & & & \frac{1}{2\Delta x} & 0 \end{pmatrix} = \frac{\hat{\epsilon}}{\partial x}$$

$$\cos(\hat{\varphi} \otimes \hat{I}) = \hat{I} \otimes \hat{I} - \frac{\hat{\varphi}^2 \otimes \hat{I}}{2} + \frac{\hat{\varphi}^4 \otimes \hat{I}}{24} - \dots = \cos(\hat{\varphi}) \otimes \hat{I}$$

$$\hat{Q} = -i\hbar \frac{\partial}{\partial \phi}$$



# Реализация библиотеки

На языке C



Для обеспечения высокой скорости работы и отказоустойчивости библиотеки был выбран язык C.

Далее показаны структуры и прототипы основных функций, которые могут быть применены, например, в Python.

```
typedef struct
{
    double im; //мнимая часть
    double re; //действительная часть
} complex;

typedef struct
{
    unsigned int dim_x;
    unsigned int dim_y;
    complex *arr;
} matrix;

matrix *do_matrix(unsigned int dim_x, unsigned int dim_y);
//создание матрицы. Возвращает NULL при ошибке / невозможности создать матрицу

matrix *find_s_vector(matrix* arr);
//поиск собственного вектора матрицы
```

# Основные функции библиотеки

- 1) `find_s_vector(matrix* arr)`
- 2) `com_mult(complex *a, complex *b, complex *res)`
- 3) `sum_matrix(matrix* a, matrix *b)`
- 4) `minus_matrix(matrix* a, matrix *b)`
- 5) `rand_matrix(matrix* m)`
- 6) `mul_m_v(matrix* m, matrix* v)`
- 7) `mul_v_c(matrix* v, double c)`
- 8) `do_unit_vector(unsigned int size)`
- 9) `norma_v(matrix* v)`

- 1) Поиск собственного вектора.
- 2) Умножение матриц.
- 3) Сложение матриц.
- 4) Разность матриц.
- 5) Заполнение матрицы случайными значениями.
- 6) Умножение матрицы на вектор.
- 7) Умножение матрицы на константу.
- 8) Создание единичного вектора
- 9) Поиск нормы.

# Результаты:

Созданная на основе степенного метода и реализованная на C библиотека способна осуществлять ряд основных математических операций над матрицами комплексных чисел и с их помощью осуществлять поиск собственного вектора, отвечающего наибольшему собственному числу.

Также скорость работы библиотеки была измерена при решении задач моделирования сверхпроводникового потокового кубита. Она позволила уменьшить затрачиваемое на расчеты время *в 8 раз*.





**Ключевые слова:**

**Языки программирования:** C, python.

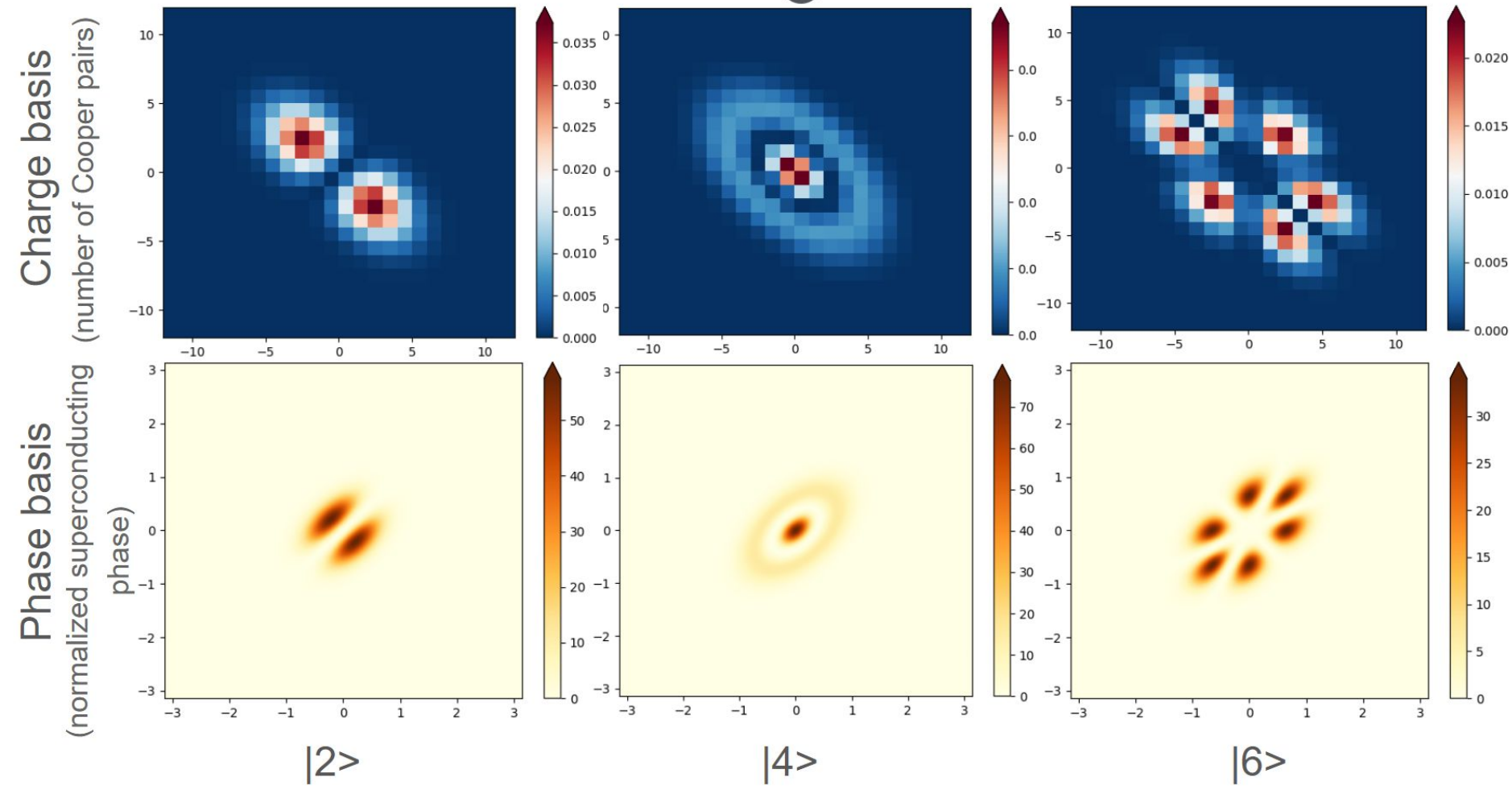
**Ключевые библиотеки:** `stdlib.h`, `ctypes`, `matplotlib`.

**Математика:** численные методы, линейная алгебра, математический анализ.

**Физика:** квантовая механика.



# Wave functions of excited states



# Список литературы

1. *Michel H. Devoret quantum fluctuations in electrical circuits (1995)*
2. Richard von Mises and H. Pollaczek-Geiringer, *Praktische Verfahren der Gleichungsauflösung*, ZAMM — Zeitschrift für Angewandte Mathematik und Mechanik 9, 152—164 (1929).
3. Брайан Керниган и Деннис Ритчи *ISO/IEC 9899:1999 (1999)*

