

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РЕАЛИЗАЦИЯ ЭТАПОВ АНАЛИЗА И МОДИФИКАЦИИ  
РАСЧЁТНОЙ СЕТКИ ДЛЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 421 группы  
направления 09.03.01 — Информатика и вычислительная техника  
факультета КНиИТ  
Андреева Максима Артемовича

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

\_\_\_\_\_

А. Д. Панфёров

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

\_\_\_\_\_

Л. Б. Тяпаев

Саратов 2024

## ВВЕДЕНИЕ

Для дальнейшего развития современной электронике очень нужны новые материалы с улучшенными свойствами и характеристиками. Перспективной областью поиска материалов с необычными свойствами являются двумерные моно-молекулярные и моно-атомные слои. Одним из таких перспективных материалов является графен. Внимание к нему обусловлено особенностями его отклика на действие внешних электрических полей. Наряду с большим количеством экспериментальных работ по изучению свойств новых материалов много усилий сосредоточено на разработке теоретических моделей и исследовании их поведения.

Описание и прогнозирование свойств любых материалов является чрезвычайно сложной физической и математической задачей в силу огромного количества частиц, формирующих их. Как правило, результатом компромисса является некоторое приближение или модель, правильность которой проверяется сравнением с экспериментом. Сами модели доступны для анализа только в форме вычислительного моделирования на высокопроизводительных вычислительных системах. В СГУ ведутся работы над одной из версий такой модели и передо мной была поставлена задача реализовать новый алгоритм анализа промежуточных результатов. Такой анализ необходим при итерационном построении сетки состояний модели в зависимости от набора физических параметров моделируемого процесса. После успешной разработки модуля, реализующего указанный алгоритм, была разработана системы интерактивного управления всей процедурой построения сетки состояний. Она обеспечила последовательный запуск расчетных модулей, модулей анализа и модификации сетки с выводом ключевых параметров процесса для его контроля.

Для достижения поставленной цели было необходимо решить ряд задач:

1. Изучить спецификацию системы моделирования;
2. Изучить структуру файловой базы данных, в которой хранятся промежуточные и конечные результаты вычислений;
3. Изучить инструменты Wolfram Mathematica для работы с файлами и данными;
4. Разработать модуль анализа по заданному алгоритму;
5. Разработать систему управления процедурой построения сетки для мо-

делирования;

6. Провести их тестирование на реальных данных;
7. Оформить и представить полученные результаты.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе кратко представлена рассматриваемая задача формирования расчетной сетки. Это один из этапов численного моделирования поведения физической системы, представляющей собой двумерный материал во внешнем электрическом поле. Система моделирования вычисляет с использованием квантового кинетического уравнения функцию распределения  $f(p_1, p_2, t)$  – вероятность присутствия носителей заряда в любом доступном в этом материале состоянии  $(p_1, p_2)$  и наблюдаемые проявления протекающих процессов.

В силу периодичности кристаллической решетки моделируемой системы, периодична и функция распределения в пространстве  $(p_1, p_2)$ . Для полного воспроизведения поведения модели достаточно воспроизвести поведение функции распределения в пределах выбранной области с набором уникальных её значений. В качестве такой области используется т.н. примитивная ячейка обратной решетки.

Зная  $f(p_1, p_2, t)$  во всей выбранной области, можно вычислять доступные для наблюдения и измерения величины, определяющие свойства протекающих процессов. Основная проблема в использовании такого подхода заключается в том, что кинетическое уравнение (точнее, это система из трех связанных обыкновенных дифференциальных уравнений первого порядка) может решаться только численно и в каждой точке пространства  $(p_1, p_2)$  отдельно, поскольку решения ведут себя независимо. Возникает задача построения сетки состояний в интересующей нас области пространства  $(p_1, p_2)$ , которая, с одной стороны, была бы достаточно плотная и представительная для корректного вычисления наблюдаемых величин, а с другой, включала в себя минимальное количество рассматриваемых состояний [1].

Такие сетки можно строить по результатам анализа максимальных значений функции  $f(p_1, p_2, t)$ , достигаемых за время моделирования поведения каждого из состояний –  $f_{max}(p_1, p_2)$ .

Сетка формируется поэтапно, с постепенным увеличением плотности покрытия и локализацией в областях, где наблюдаются высокие значения функции распределения. На каждом этапе выполняется добавление новых узлов и для них вычисляются значения  $f_{max}(p_1, p_2)$ . После этого полученные результаты анализируются и определяются области, где необходима дальней-

шая детализация сетки [2].

**Во второй главе** рассмотрена среда Wolfram Mathematica, изучив ее основные функциональные возможности и некоторые принципиальные отличия, а также применения в различных областях. Mathematica является как системой компьютерной алгебры, так и высокоуровневым языком программирования. Как язык программирования, язык Wolfram имеет типичные атрибуты объектно-ориентированных языков, являясь процедурно-функциональным языком. В то время как функциональные языки программирования могут показаться непривычными для тех, кто привык работать с процедурными языками, они предоставляют более краткие и естественные способы программирования для пользователей с небольшим опытом.

Хотя функциональные программы в целом работают медленнее, чем процедурные программы, время разработки и, что еще более важно, время внесения изменений в программы существенно меньше для функциональных программ. Это особенно важно для исследовательской работы. Например, сравнение времени выполнения вычислений с использованием массивов, списков и тензоров в пакетах Matlab, Maple и Mathematica показало преимущество последнего. Также стоит отметить, что Mathematica использует механизм упаковки массивов, что не только ускоряет работу с ними, но и значительно уменьшает использование оперативной и внешней памяти [3].

Mathematica в основном реализована на своем собственном встроенном языке программирования Wolfram, однако для оптимизации часть ее ядра и многие важные функции написаны на языке программирования C. Таким образом, реализация Mathematica основана от части на языке программирования C, который доказал свою высокую эффективность как язык системного программирования.

В общем, именно своим широким функционалом и возможностями эта среда привлекает как студентов, так и опытных исследователей, предоставляя им удобное средство для проведения вычислений, анализа данных и обработки информации в различных областях науки [4].

**В третьей главе** исследован процесс реализации этапов анализа расчетной сетки для дальнейшего повышения эффективности вычислений. С целью оптимизации процесса, было принято решение создать алгоритм помечения точек расчетной сетки к разбиению, что позволит более точно и

эффективно анализировать полученные результаты.

Процедура анализа является ключевой для построения многомасштабной сетки. Анализ выполняется только для узлов сетки со статусом «лист». Опишем алгоритм анализа:

1. Проверяем не исчерпан ли лимит количества шагов масштабирования;
2. Проверяем не осталось ли не посчитанных узлов сетки;
3. Выбираются все узлы со статусом «лист», значения которых больше значения МЗФР. Если таких узлов нет, то процедура прерывается и выводится соответствующее предупреждение;
4. Проводится проверка на количество выбранных узлов, чтобы убедиться, что можно воспользоваться вторым алгоритмом анализа. Если количество узлов проходит проверку, то узлы подвергаются проверки точности аппроксимации.

Блок-схема алгоритма анализа для выбора точек к разбиению показана на рисунке 1.

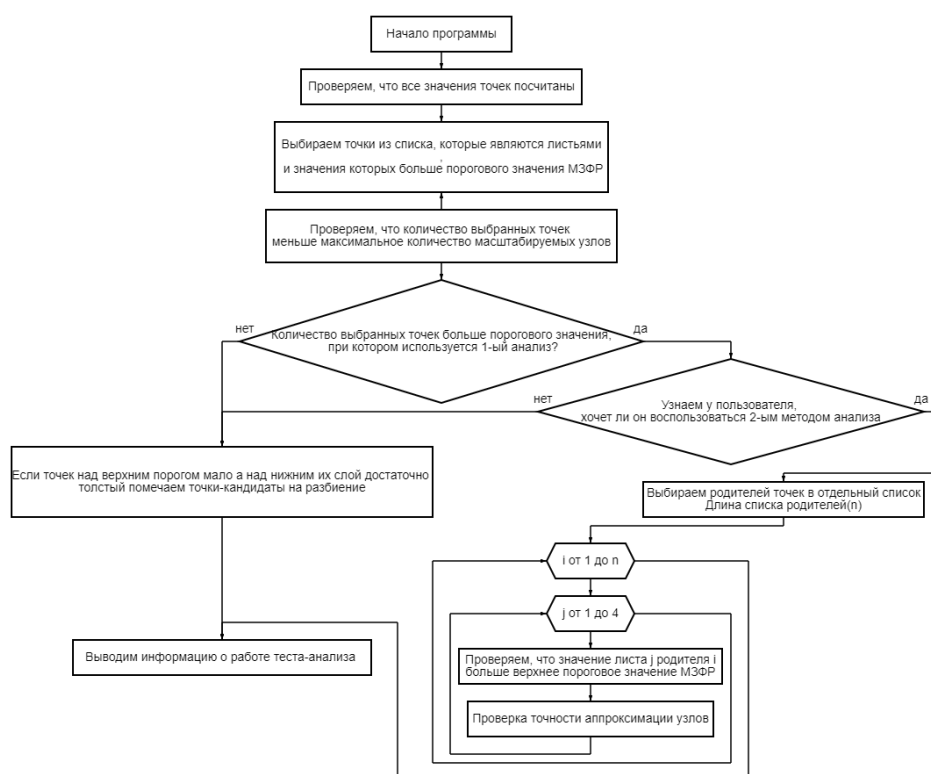


Рисунок 1 – Блок-схема алгоритм анализа для выбора точек к разбиению

Процедура проверки точности аппроксимации выполняется с использованием уже посчитанных значений МЗФР для пяти точек одной семьи. Введем обозначения: родительская точка  $f_0$  и её дети  $f_1, f_2, f_3, f_4$ . Эти точки располагаются на двух диагоналях:  $(f_1, f_0, f_3)$  и  $(f_2, f_0, f_4)$ .

Для оценки результатов используется значение относительной точности аппроксимации на этапе адаптации(а). Анализ выполняется для каждой из четырех точек ( $i=1,2,3,4$ ) последовательно по следующему алгоритму:

1. Если  $f_i = 0$  и  $f_0 = 0$ , то  $f_i$  не помечается для дальнейшего разбиения;
2. Если  $f_i = 0$  и  $f_0 \neq 0$ , то  $f_i$  не помечается для дальнейшего разбиения;
3. Если  $f_i \neq 0$  и  $f_0 = 0$ , то  $f_i$  помечается для дальнейшего разбиения;
4. Если  $f_i \neq 0$  и  $f_0 \neq 0$  и  $\left(\frac{f_i+f_{i+2}}{2} - f_0\right) / f_0 < a$ , то  $f_i$  не помечается для дальнейшего разбиения;
5. Если  $f_i \neq 0$  и  $f_0 \neq 0$  и  $\left(\frac{f_i+f_{i+2}}{2} - f_0\right) / f_0 \geq a$ , то  $f_i$  помечается для дальнейшего разбиения.

Под  $f_{i+2}$  понимается третий элемент на рассматриваемой диагонали.

Далее рассматривается программная реализация алгоритма анализа.

**В четвертой главе** описаны функции необходимые для работы диалогового окна, показан пример его работы и сравнение алгоритмов анализа. Блок-схема работы диалогового окна показана на рисунке 2.

Начало работы программы начинается с вызова функции CheckFiles, если файлы с данными уже существуют, то пользователю в диалоговом окне показывается выбор действий – продолжение масштабирования уже созданной сетки или начало подсчета новой. Если файлов с данными нет, запускается функция Start для создания изначальной сетки.

В случае выбора пользователем опции удаления файлов будет запущена функция DeleteFiles для удаления информации о предыдущей сетке, после чего будет вызвана функция Start. После расчета начальной сетки функция Start отобразит информацию о начальной сетке в диалоговом окне.

Далее вызывается функция ContinueStep[] и выводится диалоговое окно, в котором пользователь может указать необходимое количество шагов масштабирования.

После ввода количества шагов, которые необходимо выполнить, вызывается функция Step[n\_]. В ней происходит анализ, добавление и подсчет новых точек на расчетной сетке, после чего информация по сетке выводится

В ДИАЛОГОВОМ ОКНЕ.

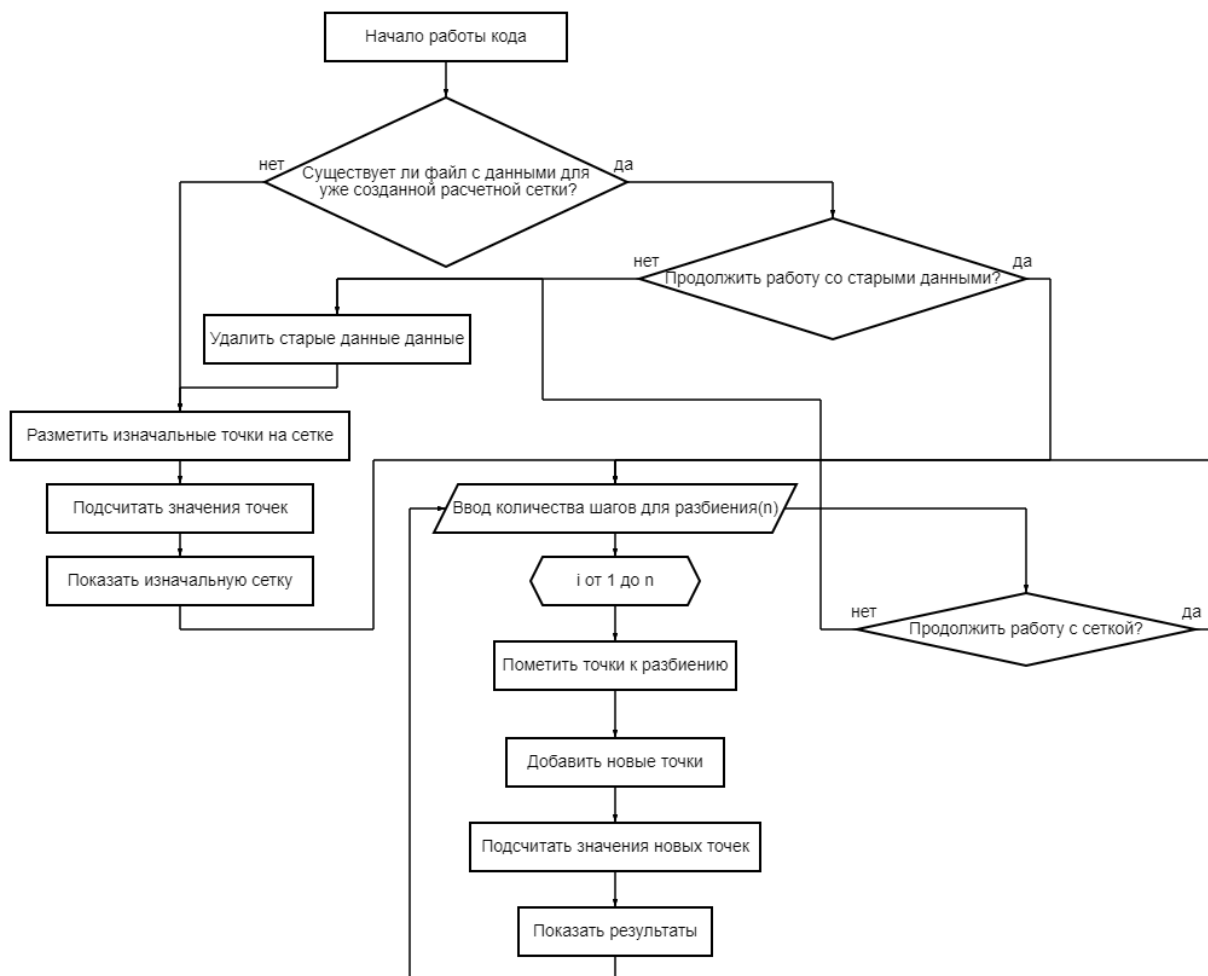


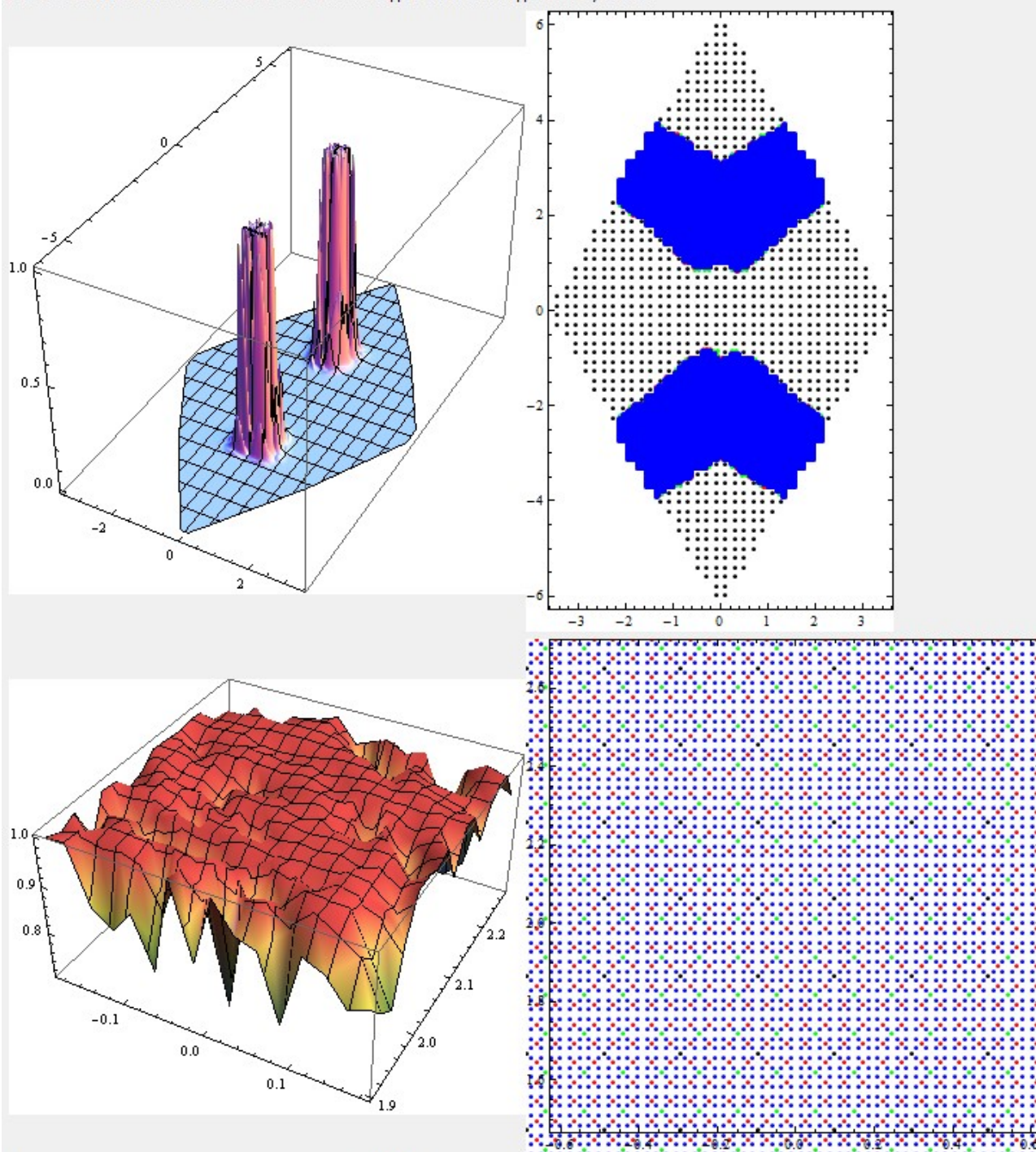
Рисунок 2 – Блок-схема работы диалогового окна

Для одной и той же расчетной сетки было сделано 2 шага масштабирования, на которых выбирался 2-ой алгоритм анализа, и 2 шага масштабирования, на которых выбирался только 1-ый алгоритм анализа. На рисунках 3 и 4 показаны этапы масштабирования сетки разными алгоритмами.

Из полученных данных можно увидеть, что второй алгоритм анализа требует больше времени из-за необходимости выборки значений родителей и их листьев, а также их последующего сравнения. Несмотря на то, что работа второго алгоритма требует больше времени, он отмечает намного меньше точек для разбиения, что благоприятно сказывается на последующих этапах масштабирования.



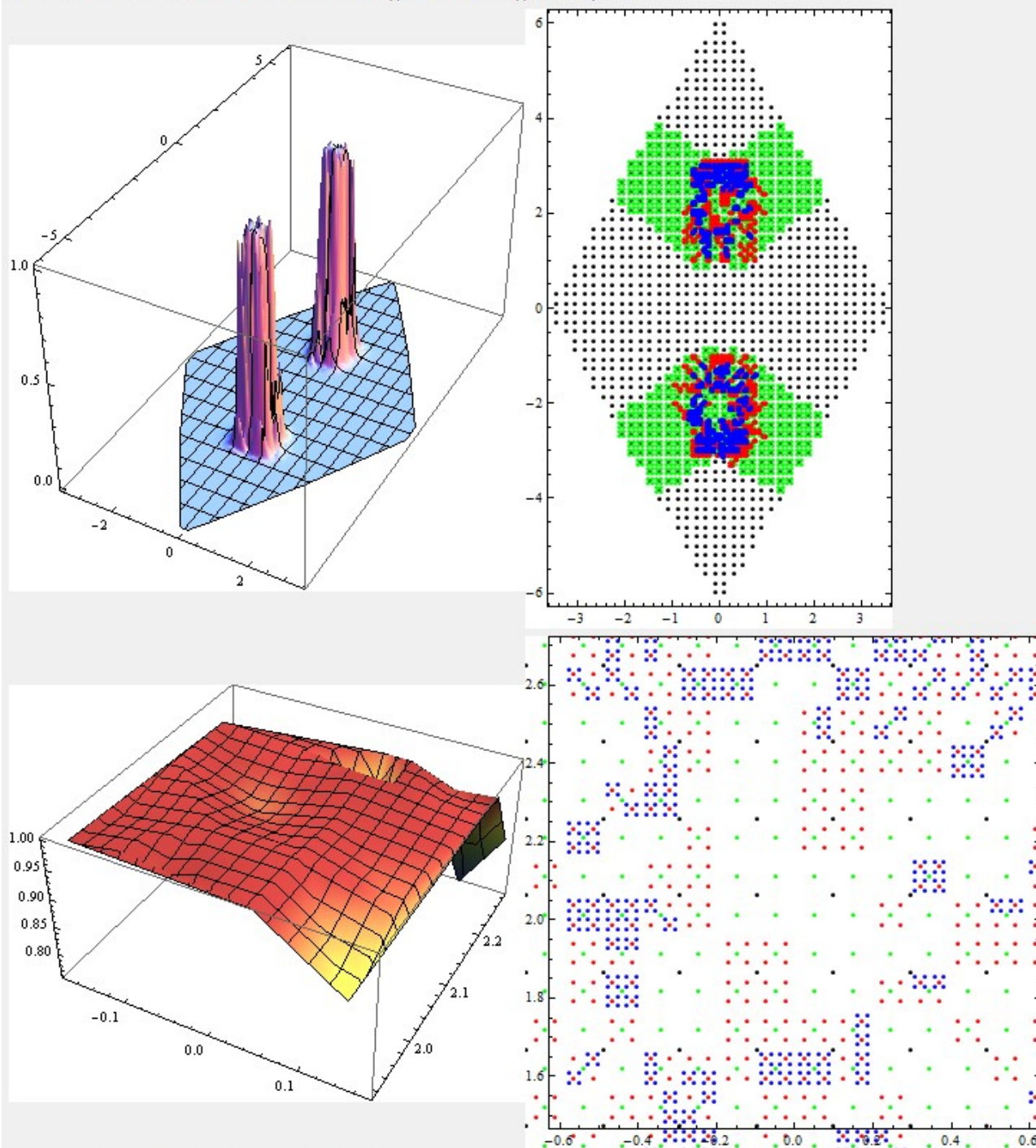
Все шаги были выполнены. Новые точки были добавлены и подсчитаны, шаг 3:



Минимальное расстояние между точками = 0.19635  
Grid\_test. Test choice = 1 Points to split = 7019 Time = 6.7860132  
Number of new points = 28076 Time = 10.4219133  
Calculate step: Density on step =  $2.54133 \cdot 10^{14}$  Time = 682.8932724

Рисунок 3 – 3-ий шаг масштабирования с использованием 1-ого алгоритма анализа

Все шаги были выполнены. Новые точки были добавлены и подсчитаны, шаг 3:



Минимальное расстояние между точками = 0.19635  
Grid\_test. Test choice = 2 Points to split = 442 Time = 22.8611663  
Number of new points = 1768 Time = 1.5928756  
Calculate step: Density on step =  $2.55789 \cdot 10^{14}$  Time = 42.3486756

Рисунок 4 – 3-ий шаг масштабирования с использованием 2-ого алгоритма анализа

## ЗАКЛЮЧЕНИЕ

Цель работы была успешно достигнута. Ставившиеся задачи решены в полном объёме.

Разработанный программный модуль анализа поведения максимальных значений функции распределения обеспечил возможность использовать альтернативный алгоритм построения расчетной сетки, с существенно меньшим количеством узлов в областях с высокими вероятностями появления возбужденных состояний.

Разработанная и протестированная системы интерактивного управления процедурой построения сетки состояний обеспечила последовательный запуск расчетных модулей, модулей анализа и модификации сетки с выводом ключевых параметров процесса для его контроля. Такой режим работы позволяет облегчить и ускорить адаптацию расчетной модели под параметры конкретного физического процесса с заданным набором параметров. Исследователь, проводящий численный эксперимент, получает возможность в режиме реального времени корректировать процесс и добиваться оптимальных результатов.

Полученные результаты обеспечат существенную экономию вычислительных ресурсов при исследовании свойств новых перспективных материалов.

### **Основные источники информации:**

- 1 Моделирование поведения графена во внешних электрических полях / А. Д. Панферов, Н. А. Новиков, А. А. Трунов, 2021. – Программные системы: теория и приложения – 19 с.
- 2 Моделирование поведения двухуровневой квантовой системы с использованием масштабируемых регулярных сеток / А. Д. Панферов, Н. В. Поснова, А. А. Ульянова, 2023 – Программные системы: теория и приложения – 21 с.
- 3 Избранные системные задачи в программной среде Mathematica / В.З. Аладьев, В.А. Ваганов, Д.С. Гринь, 2013 – 556 с.
- 4 The Student's Introduction to Mathematica and the Wolfram Language, 3rd Edition / Bruce Torrence, Eve A. Torrence, 2019 – 544 с. (Дата обращения: 27.03.2024) – Яз. англ.