

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра математического анализа и моделирования
Направление подготовки 01.04.02 – Прикладная математика и информатика
Направленность (профиль) образовательной программы «Математическое и программное обеспечение информационных систем»

ДОПУСТИТЬ К ЗАЩИТЕ
И.о. зав. кафедрой
_____ Н.Н. Максимова
« ____ » _____ 2024 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Алгоритм и программное обеспечение для задачи моделирования
бактериального роста на питательных средах

Исполнитель
студент группы 2101-ом _____ И.А. Шевкун
(подпись, дата)

Руководитель
профессор, д-р физ.-мат. наук _____ А.Г. Масловская
(подпись, дата)

Руководитель научного
содержания программы
магистратуры
профессор, д-р физ.-мат. наук _____ А.Г. Масловская
(подпись, дата)

Нормоконтроль
старший преподаватель _____ А.Н. Дудин
(подпись, дата)

Рецензент
профессор, д-р физ.-мат. наук _____ А.Ю. Чеботарев
(подпись, дата)

Благовещенск 2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра математического анализа и моделирования

УТВЕРЖДАЮ
И.о. зав. кафедрой
_____ Н.Н. Максимова
« ____ » _____ 2024 г.

З А Д А Н И Е

К магистерской диссертации студента Шевкуна Ивана Александровича

1. Тема магистерской диссертации: «Алгоритм и программное обеспечение для задачи моделирования бактериального роста на питательных средах» (утверждена приказом от 17.01.2024 № 64-уч).

2. Срок сдачи студентом законченной работы: 20.06.2024.

3. Исходные данные к магистерской диссертации: отчет по преддипломной практике, отчеты по научно-исследовательской работе, ППП Matlab.

4. Содержание магистерской диссертации (перечень подлежащих разработке вопросов): математические модели бактериального роста, решение нелинейных систем уравнений в частных производных, метод конечных разностей, фрактальный анализ растровых изображений, вычислительные эксперименты, определение управляющих параметров.

5. Перечень материалов приложения: листинг программы для численного моделирования роста бактериальной колонии в ППП Matlab.

6. Консультанты по магистерской диссертации: рецензент – профессор, д-р физ.-мат. наук Чеботарев А.Ю.; нормоконтроль – Дудин А.Н., старший преподаватель.

7. Дата выдачи задания: 29.02.2024.

Руководитель магистерской диссертации: Масловская Анна Геннадьевна, профессор, д-р физ.-мат. наук.

Задание принял к исполнению (29.02.2024): _____ Шевкун И.А.

РЕФЕРАТ

Магистерская диссертация содержит 82 с., 12 рисунков, 3 приложения, 35 источников.

БАКТЕРИАЛЬНАЯ КОЛОНИЯ, ДЕНДРИТОПОДОБНАЯ СТРУКТУРА, БАКТЕРИАЛЬНЫЙ РОСТ, МОДЕЛЬ «РЕАКЦИЯ-ДИФФУЗИЯ», НЕЯВНАЯ КОНЕЧНО-РАЗНОСТНАЯ СХЕМА, ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Математическая биология и *in silico*-исследования являются сравнительно новыми, перспективными и интенсивно развивающимися направлениями современного научного знания. Важную часть данной отрасли наук занимают исследования объектов микромира (и бактериальных популяций в частности) с применением средств и методов математического моделирования, а также технологии вычислительного эксперимента.

Магистерская диссертация направлена на реализацию системы моделирования бактериального роста на питательных средах и на выявления управляющих параметров исследуемой модели, которые влияют на формирование дендритоподобной геометрии бактериальных колоний. Предметом исследования является реакционно-диффузионная модель бактериального роста и алгоритмы ее реализации. Объектом исследования являются модели жизненного цикла бактериальных популяций, эволюционирующих на питательных средах.

Цель работы – разработать комплекс программ для имитации процессов бактериального роста на основе численной (сеточной) реализации двумерной дифференциальной эволюционной модели.

В работе впервые была выявлены условия формирования дендритоподобной геометрии в зависимости от начальных параметров, а также дан количественный анализ получаемых форм посредством фрактального анализа границы колонии.

Полученные результаты вносят вклад в развитие моделей бактериального роста, которые находят свое применение в таких областях, как медицина (противодействие размножению болезнетворных бактерий) и пищевая промышленность (как с точки зрения обеспечения безопасности продуктов, так и с точки зрения производства продуктов, при котором используются бактериальные культуры, такие как бифидобактерии).

Результаты научно-исследовательской работы были представлены в виде докладов на семи научных конференциях регионального, всероссийского и международного уровня. За время работы над темой научного исследования было опубликовано восемь научных работ и получено одно свидетельство о государственной регистрации программы для ЭВМ. Работа поддержана грантом АмГУ на выполнение научно-исследовательского проекта в 2023-2024 учебном году.

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации, проект № 122082400001-8.

СОДЕРЖАНИЕ

Введение	7
1 Реакционно-диффузионные модели математической биологии в приложении к исследованию микробиологических объектов	12
1.1 Эволюция моделей реакционно-диффузионных процессов и их приложения	12
1.2 Специфика объекта исследования: выращивание бактериальных культур на питательных средах	15
1.3 Модели популяционной динамики в математической биологии	18
1.4 Модели бактериального роста: обзор современных исследований	23
1.5 Выводы по главе	26
2 Численные методы решения нестационарных задач математической физики	27
2.1 Обзор аналитических и численных методов	27
2.2 Метод конечных разностей для решения нестационарных задач диффузионного типа	29
2.3 Программное обеспечение для решения задач математической физики	41
2.4 Фрактальный анализ растровых изображений	43
3 Система компьютерного моделирования процессов бактериального роста на питательных средах	45
3.1 Концептуальная постановка задачи моделирования	45
3.2 Математическая постановка задачи моделирования	47
3.3 Построение вычислительной схемы	49
3.4 Вычислительные эксперименты и анализ результатов	54
Заключение	59
Библиографический список	61
Приложение А. Интерфейс программы	65

Приложение Б. Листинг программы	63
Приложение В. Свидетельство о государственной регистрации программы для ЭВМ	82

ВВЕДЕНИЕ

По словам нобелевских лауреатов Франсуа Жакоба и Жана Моно, любая живая система является «системой систем», то есть сложной и трудноформализуемой системой. В настоящее время представители микромира становятся важными объектами междисциплинарных исследований в математической биологии и биоинформатике. Междисциплинарный подход в исследовании микробиологических объектов оказывается во многих случаях очень эффективным, поскольку позволяет прогнозировать состояние систем и их реакцию на внешние воздействия, привлекая инструменты прикладной математики и информатики.

В числе прочих микробных сообществ, особенно выделяются бактерии, поскольку их функционирование является важным фактором, влияющим на многие аспекты жизнедеятельности человека. Одними из самых важных и широко изучаемых процессов, связанных с бактериями, являются процессы бактериального роста.

В процессе роста бактериальные колонии могут образовывать различные геометрические формы: круг, концентрические кольца, кластерную конфигурацию, дендритоподобную разветвленную структуру и др. В настоящее время единый теоретический подход для моделирования всех геометрий не выработан, для каждой формы используется свои модели. В этой работе основное внимание уделено моделированию дендритоподобных структур.

Стремительное развитие вычислительных мощностей позволило этой области разрастаться и диверсифицироваться во все более сложные специализированные формы: например, подробные модели молекулярного уровня для описания сложных сигнальных путей или агентные модели, представляющие каждую особь. Как результат, модели можно наделить весьма специфическими свойствами, способствуя по-настоящему междисциплинарным исследованиям, сочетающим эксперимент и теорию.

Объект исследования: модели жизненного цикла бактериальных популяций, эволюционирующих на питательных средах.

Предмет исследования: реакционно-диффузионная модель бактериального роста и алгоритмы ее реализации.

Цель работы – разработать комплекс программ для имитации процессов бактериального роста на основе численной (сеточной) реализации двумерной дифференциальной эволюционной модели с добавлением стохастической процедуры.

Для достижения поставленной цели были сформулированные следующие задачи.

1. Построение вычислительной схемы для решения нелинейной системы уравнений в частных производных диффузионного типа.

2. Интеграция стохастического элемента в детерминированную модель для учета случайных факторов, характерных для биологических систем.

3. Разработка программы, реализующей построенную вычислительную схему для имитации процессов бактериального роста и потребления питательных веществ.

4. Проверка правильности и адекватности реализованной модели путем сравнения с экспериментальными данными.

5. Проведение серии вычислительных экспериментов для определения управляющих параметров модели, влияющих на образование дендритоподобной геометрии бактериальных колоний.

Исследуемая в настоящей работе модель формализуется системой из двух уравнений в частных производных диффузионного типа, одно из которых является нелинейным. Для решения системы использовалась конечно-разностная вычислительная схема с применением метода дробных шагов Яненко.

В работе впервые была выявлены условия формирования дендритоподобной геометрии в зависимости от начальных параметров при решении исследуемой модели бактериального роста типа «реакция-диффузия», а также дан коли-

чественный анализ получаемых форм посредством фрактального анализа границы колонии.

Полученные результаты вносят вклад в развитие моделей бактериального роста, которые находят свое применение в таких областях, как медицина (противодействие размножению болезнетворных бактерий) и пищевая промышленность (как с точки зрения обеспечения безопасности продуктов, так и с точки зрения производства продуктов, при котором используются бактериальные культуры, такие как бифидобактерии).

Результаты, полученные в рамках данного исследования, были апробированы на научных конференциях регионального, всероссийского и международного уровня:

1. XXXII научная конференция АмГУ «День науки», г. Благовещенск, 20 апреля 2023 г.

2. XXIV региональная научно-практическая конференция «Молодежь XXI века: шаг в будущее», г. Благовещенск, 18 мая 2023 г.

3. II Международный семинар «Вычислительные технологии и прикладная математика – 2023», г. Благовещенск, 12–16 июня 2023 г.

4. VII Международная научно-практическая конференция «Информационные технологии и высокопроизводительные вычисления», г. Хабаровск, 11–13 сентября 2023 г.

5. Всероссийская конференция «Моделирование и экспериментальные исследования динамики сложных систем», г. Нижний Новгород, 16–18 ноября 2023 г.

6. XI Международная научная конференция «Математическое и компьютерное моделирование», г. Омск, 15 марта 2024 г.

7. XXXIII научная конференция АмГУ «День науки», г. Благовещенск, 18 апреля 2024 г.

Результаты, полученные в рамках данного исследования, были представлены в шести публикациях (4 материалов конференций и 2 статьи) [9,12-16]. Ещё две статьи в настоящий момент находятся на рецензировании в редакциях

журналов (Шевкун И.А., Масловская А.Г. «Гибридный подход к моделированию и оценке структурных особенностей паттернов культивируемых бактерий» в журнале Математическое моделирование и Maslovskaya A., Kuttler Ch., Shevkun I., Chebotarev A., Kovtanyuk A. «Quorum sensing model for nutrient-dependent evolution of cultured bacteria: theoretical framework and in silico study» в журнале Nonlinear Dynamics)

Кроме того, для программной реализации вычислительного алгоритма в пакете прикладных программ было получено свидетельство о государственной регистрации программы для ЭВМ «Система моделирования и оценки геометрических характеристик дендритных структур при культивации бактерий» (Свидетельство № 2024662124 от 23.05.2024, заявка № 2024660621 от 07.05.2024).

Научно-исследовательская работа в 2023-2024 гг. была поддержана грантом АмГУ для выполнения научно-исследовательского проекта «Система моделирования бактериального роста на питательных средах: гибридный подход».

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации, проект № 122082400001-8.

Работа состоит из трех глав. В первой главе приводятся сведения о реакционно-диффузионных моделях математической биологии и их применении к исследованию микробиологических объектов, рассматривается эволюция моделей реакционно-диффузионных процессов. А также приводится краткая характеристика объекта исследования: выращивание бактериальных культур на питательных средах.

Во второй главе приводится обзор аналитических и численных методов решения нестационарных задач математической физики, обзор программного обеспечения, применяемого для решения таких задач, а также краткие сведения о фрактальном анализе изображений.

В третьей главе описывается построение вычислительной схемы исследуемой задачи на основе нелинейной системы двух уравнений в частных производных.

водных, приводятся результаты вычислительных экспериментов, включая визуализацию геометрических форм получаемых решений.

1 РЕАКЦИОННО-ДИФФУЗИОННЫЕ МОДЕЛИ МАТЕМАТИЧЕСКОЙ БИОЛОГИИ В ПРИЛОЖЕНИИ К ИССЛЕДОВАНИЮ МИКРОБИОЛОГИЧЕСКИХ ОБЪЕКТОВ

1.1 Эволюция моделей реакционно-диффузионных процессов и их приложения

В строгом смысле этого слова реакционно-диффузионные системы – это системы, в которых компоненты локально превращаются друг в друга в результате химических реакций и переносятся в пространстве путем диффузии [15]. Они возникают совершенно естественно в химии и химической технологии, но также служат отправной точкой для изучения широкого спектра явлений, выходящих за рамки строгой области химической науки, такой как науки об окружающей среде и жизни.

Реакционно-диффузионные системы в закрытом сосуде и в отсутствие внешних сил в конечном итоге развиваются до состояния химического равновесия, при котором участвующие компоненты распределяются равномерно в пространстве и каждому элементарному реакционному шагу противодействует его обратный. Давно стало понятно, что подход к равновесию может быть как в форме простого экспоненциального затухания, так и в форме более сложного переходного поведения, связанного с затухающими колебаниями или нетривиальными пространственными зависимостями, включая волновые закономерности. В то время как первые сразу же нашли ряд важных применений, таких как экспериментальное измерение констант скорости химических реакций, вторые первоначально рассматривались как диковинки. Развитие необратимой термодинамики в 1950-х годах и позже позволило объяснить происхождение этих двух типов поведения, связав их с временной эволюцией, начинающейся близко к равновесию и вдали от него соответственно. Эксперименты и моделирование реактивных систем лабораторного масштаба, таких как реакция Белоусова-Жаботинского, подтвердили эту точку зрения. Однако пока реакционно-диффузионные процессы осуществлялись в закрытом термостатируемом реакторе, не было возможности систематически анализировать происходящее, по-

сколькx в силу второго закона термодинамики система должна была рано или поздно прийти в состояние равновесия.

Важным достижением, открывшим новые горизонты в экспериментальном изучении реакционно-диффузионных систем и одновременно стимулировавшим важные теоретические разработки, стало систематическое использование открытых реакторов, посредством которых система поддерживается в неравновесном состоянии сколь угодно долго за счет перекачки свежих реагентов (скорость которой определяет расстояние от равновесия) и отток использованных продуктов. Когда пространственная однородность поддерживается внутри реактора за счет перемешивания в CSTR (проточный химический реактор с мешалкой), при изменении ключевых параметров обнаруживается богатая феноменология, в дополнение к установившемуся состоянию, экстраполирующему знакомое равновесное поведение: простые периодические, многопериодические и хаотические колебания; мультистабильность, т. е. сосуществование более чем одного одновременно стабильных состояний; и возбудимость, при которой однажды возмущенная система совершает расширенное движение, прежде чем вернуться в исходное стабильное состояние [15]. Самый захватывающий набор вариантов поведения относится к пространственной структуре, которая становится возможной, когда перемешивание не происходит внутри реактора, как это имеет место во многих реальных ситуациях в химии, технике и биологии.

Математически системы типа «реакция-диффузия» принимают форму полулинейных параболических уравнений в частных производных. Их можно представить в общем виде [3]:

$$\partial_t q = D \nabla^2 q + R(q), \quad (1.1)$$

где $q(x,t)$ – неизвестная вектор-функция, D – диагональная матрица диффузионных коэффициентов, R – член, отвечающий за локальные реакции.

Решение реакционно-диффузионных задач приводит к разнообразным конфигурациям, включая самоорганизующиеся структуры, волновые процессы, полосы, многоугольники, а также более сложные структуры, которые называют «паттернами Тьюринга». Все функции, удовлетворяющие дифференциальным диффузионным уравнениям по сути являются концентрацией.

В простейшем одномерном по пространству случае уравнение (1.1) принимает вид:

$$\partial_t u = D \partial_x^2 u + R(u). \quad (1.2)$$

Это уравнение известно как уравнение Колмогорова-Петровского-Пискунова. При отсутствии $R(u)$ уравнение (1.2) описывает чистый диффузионный процесс и превращается в уравнение Фика. При $R(u) = u(1 - u)$ уравнение (1.2) представляет собой уравнение Фишера, описывающее распространение биологических популяций. При $R(u) = u(1 - u^2)$ уравнение описывает конвекцию Рэлея – Бенара. Более общий вид $R(u) = u(1 - u)e^{-\beta(1-u)}$ приводит к уравнению Зельдовича–Франка-Каменецкого, которое находит применение в теории горения.

Двумерные по пространству системы допускают гораздо больший диапазон возможных явлений, чем их одномерные аналоги. Важная идея, впервые предложенная Аланом Тьюрингом, заключается в том, что состояние, устойчивое в локальной системе, может стать нестабильным в присутствии диффузии. Также были предложены уравнения реакции-диффузии для более чем двух пространственных измерений, например реакция Белоусова-Жаботинского, реакция свертывания крови, волны деления или плоские газоразрядные системы.

В последнее время реакционно-диффузионные системы вызвали большой интерес как прототип модели формирования структур. Вышеупомянутые паттерны (фронты, спирали, мишени, шестиугольники, полосы и диссипативные солитоны) обнаруживаются в различных типах реакционно-диффузионных систем, несмотря на большие различия в их природе. Также предполагается, что реакционно-диффузионные процессы являются важной основой процессов, свя-

занных с морфогенезом в биологии, и могут даже быть связаны с шерстью животных и пигментацией кожи. Другие применения уравнений реакции-диффузии включают экологические инвазии, распространение эпидемий, рост опухолей, динамику волн деления, заживление ран и зрительные галлюцинации. Другая причина интереса к реакционно-диффузионным системам заключается в том, что, хотя они представляют собой нелинейные уравнения в частных производных, часто существуют возможности для их аналитического рассмотрения.

1.2 Специфика объекта исследования: выращивание бактериальных культур на питательных средах

Питательная среда для культивирования является твердой, жидкой или полутвердой основой, разработанной для обеспечения роста популяций микроорганизмов или клеток через процесс их деления и размножения. Она также подходит для культивирования небольших растений. Для различных типов клеток применяются разные виды питательных сред, специально адаптированные к их потребностям [22].

Наиболее распространенными средами роста микроорганизмов являются питательные бульоны и чашки с агаром; для роста микроорганизмов и клеточных культур иногда требуются специализированные среды. Некоторым организмам, называемым «привередливыми», требуются специализированные условия из-за сложных потребностей в питании. Например, вирусы являются внутриклеточными паразитами и требуют питательной среды, содержащей живые клетки.

В жидкие питательные растворы часто добавляют агар, который после добавления среды в чашку Петри, приводит к застыванию раствора и образованию твердой подложки для выращивания различных культур. Практически никакие бактерии не могут разрушить агар, поэтому такая конфигурация обеспечивает стабильную твердую основу. Эти агаровые чашки предоставляют стабильную твердую среду, поскольку лишь немногие бактерии способны разру-

шать агар. В жидких средах микроорганизмы часто образуют коллоидные системы.

Существует различие между питательными средами для клеточных и микробиологических культур. Клетки, полученные из многоклеточных организмов, обычно требуют добавления гормонов или факторов роста, которые присутствуют в организме *in vivo*, для их успешного роста в культуре. Для культивирования животных клеток часто добавляют сыворотку крови или ее синтетические аналоги. В отличие от этого, микроорганизмы, будучи одноклеточными, не нуждаются в таких добавках. Кроме того, клетки животных зачастую выращивают на плоской поверхности, в которую заливается жидкая питательная среда, покрывающая их. Бактерии же могут расти как на твердой, так и в жидкой среде.

Питательные среды делятся на химически определенные и неопределенные. Для определенных сред известно количество всех компонентов, необходимых для роста микроорганизмов, включая микроэлементы, витамины, а также источники углерода и азота, такие как глюкоза или глицерин и аммонийные соли или нитраты. Неопределенные среды включают сложные компоненты, такие как дрожжевой экстракт или гидролизат казеина, состоящие из множества химических веществ в неизвестных пропорциях. Выбор неопределенной среды может быть обусловлен её стоимостью или необходимостью, так как некоторые микроорганизмы не растут на определенных средах.

Примером питательной среды является сусло, используемое в пивоварении. Оно содержит все необходимые питательные вещества для роста дрожжей и позволяет в анаэробных условиях производить спирт.

Питательные среды можно классифицировать следующим образом [21]:

Культурные среды: эти среды включают все необходимые компоненты для роста большинства бактерий. Они используются для общего выращивания и поддержания лабораторных коллекций культур. Культурные среды могут быть неопределенными (с неизвестным точным составом) или определенными. Типичным представителем является питательный агар.

Минимальные среды: содержат только те компоненты, которые необходимы для поддержания роста микроорганизмов. Количество добавляемых ингредиентов варьируется в зависимости от вида бактерий. Обычно минимальные среды включают источник углерода (например, глюкозу или сахар), разнообразные соли для обеспечения магнием, азотом, фосфором и серой, которые необходимы для синтеза белков и нуклеиновых кислот, а также воду.

Селективные среды: эти среды предназначены для выращивания только определенных организмов. К примеру, если бактерия устойчива к конкретному антибиотику, то его можно добавить в раствор, чтобы подавить рост других клеток. Селективные среды также используются в клеточной культуре для обеспечения выживания или пролиферации клеток с определенными характеристиками, такими как устойчивость к антибиотикам или способность вырабатывать определенные метаболиты.

Индикаторные (дифференциальные): эти среды помогают различать различные типы организмов, растущих на одной и той же среде, на основании их биохимических характеристик. В такие среды добавляют питательные вещества или индикаторы (например, нейтральный красный, феноловый красный, эозин или метиленовый синий), которые позволяют визуально идентифицировать определенные характеристики бактерий. Дифференциальные среды широко используются в молекулярной биологии для обнаружения микроорганизмов и штаммов. Примером такой среды является кровяной агар на основе крови сердца быка, которая становится прозрачной в присутствии таких бактерий, как стрептококк или стафилококк.

Транспортные среды: эти среды используются для временного хранения транспортируемых образцов для последующего их выращивания. Эти среды должны поддерживать жизнеспособность всех микроорганизмов в образце без изменения их количества. Транспортные среды не должны содержать углерод, азот и другие факторы роста, чтобы избежать размножения микробов. Транспортные среды для анаэробных микроорганизмов не должны содержать кислород.

1.3 Модели популяционной динамики в математической биологии

Все живые системы, включая бактерии, имеют свои уникальные характеристики, которые необходимо учитывать при создании моделей их роста [8]:

1) Биологические системы представляют собой сложные многокомпонентные структуры с индивидуальными элементами, обладающими пространственной организацией. При их моделировании возможны два подхода. Первый – агрегированный или феноменологический, который выделяет ключевые атрибуты системы и исследует их качественное поведение во времени (устойчивость стационарных состояний, наличие колебаний, пространственную неоднородность). Этот подход имеет исторические корни и широко применяется в нестационарной теории популяций. Вторым подходом является детальное изучение элементов системы и их взаимодействий, создание имитационных моделей с физически и биологически значимыми параметрами. Хотя такие модели не предусматривают аналитического анализа, они могут обеспечить количественные прогнозы поведения системы при различных внешних воздействиях при хорошем экспериментальном изучении ее частей.

2) Системы с репродукцией, они характеризуются способностью перерабатывать различные химические вещества для синтеза биологических молекул, клеток и организмов. Большую роль в развитии сложных пространственно-временных режимов играют процессы взаимодействия компонентов (биохимические реакции) и процессы переноса, включая как хаотическую диффузию, так и направленные внешними силами, такими как гравитация и электромагнитные поля, или адаптивные функции живых организмов, такие как хемотаксис.

3) Биологические системы – это открытые системы без термодинамического равновесия, поэтому при их моделировании необходимо использовать нелинейные уравнения.

4) Сложные системы регуляции биологических организмов приводят к возникновению колебательных и квазистохастических режимов.

5) Живые системы имеют сложную пространственную организацию, поэтому их внутренняя среда не является однородной.

б) Имитационные модели активно применяются для описания конкретных сложных живых систем, максимально учитывая доступную информацию об объекте. Эти модели используются для исследования объектов различных уровней организации живой материи, начиная от макромолекул и заканчивая биогеоценозами.

Модели биологических систем можно классифицировать следующим образом [8]:

1) Качественные модели – это сравнительно несложные модели, предназначенные для описания основных динамических характеристик системы.

2) Регрессионные модели описывают связь между различными характеристиками системы, никаким образом не учитывая физический или биологический смысл изучаемых процессов. В качестве исходных данных используют статистические данные наблюдений.

3) Имитационные модели. Основной принцип этих моделей заключается в изучении сложных математических представлений системы через вычислительные эксперименты и анализ их результатов. При этом максимально учитывается вся доступная информация об объекте моделирования, как количественная, так и качественная. Имитационное моделирование является сложным и во многом интуитивным процессом, который, по словам Р. Шеннона, «в равной степени является искусством и наукой» [12].

Ученые с давних времен пытались найти числовые закономерности роста живых организмов. Ещё в начале 12 века Фибоначчи в «Трактате о счете» привел задачу о росте числа кроликов, размножающихся в пространстве, обнесенном стеной. Ответом к задаче стали числа, названные в честь ученого (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, ...) [2].

Следующим важным этапом стала модель Мальтуса, предложенная в 1798 году, которая описывает рост популяции со скоростью, пропорциональной ее численности. В дискретной форме этот закон выражается в виде геометриче-

ской прогрессии, а в форме дифференциального уравнения представляет собой модель экспоненциального роста популяции. Эта модель эффективно описывает рост клеточных популяций при отсутствии ограничений:

$$\frac{dx}{dt} = rx,$$

где r – параметр скорости роста колонии, отражающий генетический потенциал [8].

В 1845 году бельгийский математик Пьер Ферхюльст предложил логистическое уравнение для описания популяционной динамики. Это уравнение основано на двух основных предположениях:

– скорость размножения пропорциональна текущему количеству особей, при равенстве остальных условий;

– скорость размножения пропорциональна объему ресурсов, при равенстве остальных условий. В этом случае второй член уравнения характеризует конкуренцию за ресурсы, ограничивающую рост колонии.

Таким образом, мы получаем уравнение Ферхюльста:

$$\frac{dP}{dt} = rP \left(1 - \frac{P}{K} \right),$$

где P – размер популяции, t – время, r – параметр скорости роста (размножения), K – ёмкость среды (максимально возможную численность популяции).

В первой половине 20 века французский биохимик Жак Моно предложил непрерывную модель роста микроорганизмов с лимитированием по субстрату (говоря простым языком, при недостатке питательных веществ). Из микробиологических экспериментов известно, что при ограниченной концентрации питательного вещества (нутриента) скорость роста увеличивается пропорционально его концентрации. При избытке нутриента скорость роста достигает постоянного значения, определяемого генетическими возможностями популяции. В течение определенного периода численность популяции растет экспоненциально,

пока скорость роста не начинает ограничиваться другими факторами. В таком случае зависимость будет следующей:

$$\frac{dX}{dt} = \frac{\mu_m S}{K_s + S} X,$$

где x – концентрация клеток, S – концентрация субстрата, μ_m – максимальная скорость роста микроорганизмов, K_s – константа, численно равная концентрации субстрата, при которой скорость роста культуры равна половине максимальной.

Вид этой зависимости совпадает с видом уравнения Михаэлиса-Ментен – наиболее известной модели ферментативной кинетики:

$$\frac{dS}{dt} = -\frac{\mu_0 S}{K_m + S},$$

где μ_0 – максимальная скорость, K_m – константа Михаэлиса (одна из важнейших для ферментативных реакций величина).

Закон Михаэлиса-Ментен основан на уравнениях химической кинетики и отображает более глубокие закономерности кинетики ферментативных реакций. Эти закономерности, в свою очередь, определяют жизнедеятельность и рост микроорганизмов, описываемые эмпирической формулой Монод. Именно это определяет их сходство.

В общем виде модель, которую некоторые исследователи называют моделями «типа Моно» [1], можно записать следующим образом:

$$\begin{aligned} \frac{dX}{dt} &= \mu X, \\ \mu &= \mu(S), \\ \frac{dS}{dt} &= -\frac{1}{Y} \cdot \frac{\partial X}{\partial t}, \end{aligned}$$

где X – концентрация биомассы, S – концентрация субстрата, Y – экономический коэффициент, показывающий выход биомассы с единицы субстрата, μ – удельная скорость роста биомассы.

В 40-х годах 20 века появился новый класс моделей – агент-ориентированные модели, которые позднее в связи с развитием вычислительной техники получили широкое распространение. Агентное моделирование представляет собой метод имитационного моделирования, который исследует поведение децентрализованных агентов и влияние этого поведения на функционирование всей системы. В отличие от системной динамики, где акцент делается на поведении системы в целом, в агентном моделировании изучается поведение агентов на индивидуальном уровне, а общее поведение системы возникает как результат взаимодействия множества агентов (это подход «снизу вверх»).

Этот подход был представлен в 1940-х годах Джоном фон Нейманом с целью описания процессов самовоспроизведения в биологических системах. Клеточный автомат представляет собой набор простых (конечных) автоматов, каждый из которых изменяет свои состояния в соответствии с текущими состояниями его соседей. Определение клеточному автомату можно дать следующим образом:

- Клеточный автомат представляет собой систему, в которой имеется регулярно упорядоченный набор простых однотипных объектов, известных как клетки, который может быть как конечным, так и бесконечным.

- Каждая клетка имеет свое внутреннее состояние, которое может принимать дискретные и конечные значения.

- Обновление состояний клеток происходит одновременно (синхронно) в дискретные моменты времени.

- Правила обновления состояний клеток являются однородными, то есть не зависят ни от времени, ни от пространственного расположения клеток.

- Новое состояние каждой клетки определяется исключительно значениями состояний ее соседних клеток.

Одним из самых известных клеточных автоматов является Игра «Жизнь», его изобрел английский математик Джон Конвей в 1970 году. Автомат моделирует изменение популяции на размеченной на клетки плоскости по следующим правилам:

– если пустая (мёртвая) клетка окружена тремя живыми клетками, то она оживает;

– если у живой клетки ровно две или три живых соседа, то она продолжает жить; иначе (меньше двух живых соседей или больше трёх) клетка умирает (от «одинокства» или «перенаселения»).

Игра «Жизнь» оказала огромное влияние как на многие науки, в том числе на биологию и бактериология.

1.4 Модели бактериального роста: обзор современных исследований

Биологические процессы в реальном мире разворачиваются в пространстве и времени, характеризуясь неоднородностью. Для учета этой неоднородности можно разделить пространство на ячейки, создав пространственную сетку, и описать систему с помощью обширного, но ограниченного числа параметров в каждой ячейке. Такой подход применяется в методе конечных автоматов.

Другой подход состоит в использовании моделей, в которых состояние системы описывается функцией точки в пространстве. Эти модели, известные как распределенные в пространстве, представляют собой эволюцию системы во времени с помощью уравнений в частных производных, в отличие от предыдущих моделей, которые описывали состояние системы с помощью ограниченного числа переменных и систем обыкновенных дифференциальных уравнений.

Биологические системы проявляют различные нетривиальные виды поведения, которые определяют особенности живых организмов. Например, это может быть асимметричное формирование структур, распространение возбудителей в виде волн и автоволновые процессы. Исследования базовых моделей активных сред показали, что все эти виды поведения могут быть описаны с использованием систем нелинейных дифференциальных уравнений в частных производных, которые обычно записываются следующим образом [8]:

$$\frac{\partial x_i}{\partial t} = \frac{\partial}{\partial r} \left(\sum_{j=1}^n D_{ij} \frac{\partial x_j}{\partial r} \right) + f_i(x_1, x_2, \dots, x_n), \quad i = 1, 2, \dots, n,$$

где D_i и D_{ij} ($i \neq j$) – коэффициенты диффузии и взаимной диффузии, f_i – нелинейные функции, описывающие взаимодействие компонентов, r – вектор пространственных координат.

Эти уравнения называются уравнениями типа «реакция–диффузия». В зависимости от вида функций f_i и коэффициентов диффузии D_{ij} в таких системах могут возникать различные нетривиальные типы поведения, такие как распространение бегущего импульса, стоячие волны, синхронные автоколебания и др.

Приведем обзор некоторых научных работ, авторы которых исследуют математические модели с уравнениями типа «реакция–диффузия».

В работе Rida et al. [28] исследуются численные решения реакционно-диффузионной хемотаксической модели дробного порядка для роста бактерий. В этой модели производная по времени имеет дробный порядок, присутствуют члены, описывающие процесс бактериального хемотаксиса и скорость роста бактерий:

$$\begin{aligned} \frac{\partial^\alpha b}{\partial t^\alpha} &= \nabla \cdot (D_b(b, s) \nabla b) - \nabla \cdot (b \chi(s) \nabla s) + g(s)b, \\ \frac{\partial^\alpha s}{\partial t^\alpha} &= D_s \Delta s - g(s)b, \end{aligned}$$

где b – плотность активных бактерий, s – концентрация питательного вещества, $D_b(b, s)$ – коэффициент нелинейной диффузии бактериальных клеток, D_s – постоянный коэффициент диффузии питательного вещества, $g(s)$ – функция, описывающая удельную скорость роста бактерий, $\chi(s)$ – реакция процесса хемотаксиса (модель для случая нелинейной диффузии), $0 < \alpha \leq 1$ – порядок производной.

Для решения этих уравнений в работе использовался метод декомпозиции Адомяна. В результате компьютерной симуляции авторы получили бактериальные колонии в форме концентрических колец, что хорошо согласуется с экспериментальными данными. Авторы пришли к выводу, что рост бактерий зависит от коэффициента хемотаксиса и коэффициента линейной диффузии для бактерий, а также непрерывно зависит от порядка дробной производной. Этот же подход авторы используют и в других своих работах, например в [18].

В работе И. Бажлекова и Е. Бажлековой [14] исследуется моделирование сложных процессов на примере биологических процессов и связанное с ними распределение питательных веществ, бактерий и биопродукта в многофазных флюидных системах. Исследуемая математическая модель включает конвекцию, диффузию, биореакцию и граничные условия на границах раздела: межфазный массообмен и адсорбцию.

В этой работе используются производные дробного порядка по времени, и авторы особое внимание уделили правильной «фракционализации», то есть замены уравнений в частных производных целого порядка на уравнения с производными дробного порядка. В работе говорится о важности этого процесса и утверждается, что если для математического моделирования отдельного процесса возможна прямая замена производной целого порядка на производную дробного порядка, то в случае, когда задействовано несколько подпроцессов, каждый со своей спецификой развития, такой подход представляется нецелесообразным.

Основной вывод этого исследования заключается в том, простая замена в моделях с уравнениями типа «реакция-диффузия» производной целого порядка на производную дробного порядка может привести к некорректной с физической точки зрения модели. Это происходит потому, что большинство эволюционных моделей (например, физических, биологических и т. д.) включают закон сохранения в сочетании с определяющим уравнением, описывающим процесс. Производная по времени целого порядка в классической модели получается из закона сохранения. Прямая замена этой производной дробного порядка может

привести к нарушению соответствующего закона сохранения. Для устранения этой проблемы авторы предлагают вводить определяющее уравнение.

Герман Эберл имеет целый ряд работ, в соавторстве с другими исследователями, по моделированию различных биологических систем с использованием уравнений типа «реакция-диффузия». Так в [16] моделируется система, включающая биопленки сложных пространственных структур с пустотами и каналами с присутствием антибиотика и питательного вещества. В этой работе проводится численное моделирование, сравнение с классическими одномерными моделями, а также с моделями на основе клеточных автоматов.

В других работах Эберла исследуются модели сложного роста биопленок за пределы занятого участка [20], комплексные модели нелинейной диффузии, содержащие одновременно обыкновенные дифференциальные уравнения и уравнения в частных производных [25], система из двух видов бактерий, аэробного и анаэробного вида (статья в [24]) и многие другие вопросы. Это говорит о том, что уравнения типа «реакция-диффузия» имеют широкое разнообразие и могут применяться в различных ситуациях. И в то же время, по словам Эберла, в этом направлении остается еще много нерешенных проблем [25].

1.5 Выводы по главе

В первой главе мы рассмотрели эволюцию моделей реакционно-диффузионных процессов и увидели, что эти модели, зародившись в химии, нашли также приложения и в широком спектре других областей, в том числе и в микробиологии в моделях бактериального роста. Привели обзор различных эффектов, возникающих при решении уравнений типа «реакция-диффузия».

Мы рассмотрели различные аспекты выращивания организмов на питательных средах, привели обзор различных видов питательных сред: культурные среды, минимальные среды, селективные среды, индикаторные среды, транспортные среды.

В завершении главы мы рассмотрели этапы развития моделей популяционной динамики и провели обзор современных исследований моделей бактериального роста.

Относительно сказанного в первой главе, можно сказать следующее. В настоящей работе исследуются процессы бактериального роста на питательных средах, а именно на твердом агаре в чашках Петри. Исследуемая модель формализуется системой из двух уравнений в частных производных типа «реакция-диффузия», одно из которых является нелинейным. Уравнения имеют две пространственные координаты.

2 ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ НЕСТАЦИОНАРНЫХ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ

2.1 Обзор аналитических и численных методов

Уравнения типа реакция-диффузия являются уравнениями математической физики (уравнениями в частных производных). Для решения этих типов уравнений существуют следующие способы решения:

1) Метод разделения переменных применяется для уравнений с частными производными, содержащими несколько независимых переменных, позволяя разбить их на несколько обыкновенных дифференциальных уравнений.

2) Метод интегральных преобразований сокращает уравнение с частными производными с n независимыми переменными до уравнения с $(n - 1)$ переменными, упрощая его до обыкновенного дифференциального уравнения.

3) Метод преобразования координат сводит исходное уравнение с частными производными к более простой форме с помощью соответствующего изменения координат (поворот, смещение и т.д.).

4) Преобразование зависимой переменной позволяет изменить уравнение с частными производными так, чтобы его решение стало более доступным путем решения упрощенного уравнения.

5) Метод теории возмущений разбивает исходную нелинейную задачу на последовательность линейных задач, облегчая ее аппроксимацию и решение.

6) Метод функций Грина заменяет начальные и граничные условия системой простых источников, облегчая решение задачи путем суммирования решений для каждого источника.

7) Метод интегральных уравнений преобразует уравнение с частными производными в интегральное уравнение, для решения которого существует множество методов.

8) Вариационные методы заменяют уравнения с частными производными задачей минимизации, что позволяет найти решение исходного уравнения.

9) Метод разложения по собственным функциям ищет решение уравнения с частными производными в виде ряда, используя собственные функции, найденные решением соответствующей задачи на собственные значения.

10) Метод конечных разностей преобразует уравнение с частными производными в систему разностных уравнений для численного решения [4, 5, 10, 13].

11) Метод конечных элементов разбивает область, в которой решаются дифференциальные уравнения, на конечное число элементов для численного решения [7, 9].

При решении реальных практических задач практически никогда нельзя или крайне затруднительно найти точное решение дифференциальных уравнений модели. В этом случае приходится использовать численные методы нахождения приближенного решения. Далее рассмотрим эти методы.

2.2 Метод конечных разностей для решения нестационарных задач диффузионного типа

Методы численного решения дифференциальных уравнений основаны на том, что дифференциальная задача преобразуется в задачу разностной аппроксимации. Прежде чем перейти к обсуждению аппроксимации дифференциальных уравнений, полезно рассмотреть аппроксимацию элементарных дифференциальных операторов первого и второго порядков [4].

Рассмотрим функцию одной переменной $u = u(x)$, для которой задан интервал изменения $x \in [a, b]$. Разобьём интервал $[a, b]$ на n равных частей (рисунок 2.1).

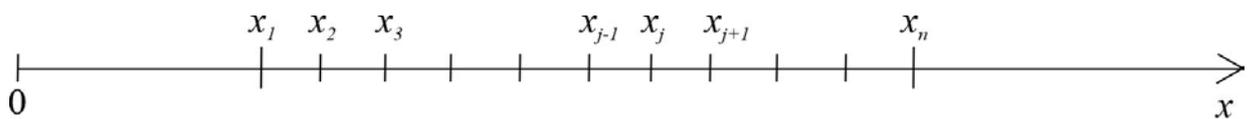


Рисунок 2.1 – Разбиение интервала изменения переменной x

Введём следующие обозначения:

j – порядковый номер точки деления;

$u(x_j) = u_j$ – значение функции $u(x)$ в точке x_j ;

$x_{j+1} - x_j = \Delta x = h$ – величина интервала между точками.

Рассмотрим производную функции $u(x)$ в точке x_j . Ее аппроксимацию можно ввести с помощью следующих разностных операторов:

– с помощью правой конечной разности

$$\lambda_x^+ u = \frac{u_{j+1} - u_j}{h};$$

– с помощью левой конечной разности

$$\lambda_x^- u = \frac{u_j - u_{j-1}}{h};$$

– с помощью центральной конечной разности

$$\lambda_x^0 u = \frac{u_{j+1} - u_{j-1}}{2h}.$$

Кроме того, разностную аппроксимацию производной первого порядка можно задать в виде линейной комбинации левой и правой разностей:

$$\lambda_x^\sigma u = \sigma \frac{u_{j+1} - u_j}{h} + (1 - \sigma) \frac{u_j - u_{j-1}}{h},$$

где $0 \leq \sigma \leq 1$.

Легко понять, что при $\sigma = 0$ мы получаем левую конечную разность, при $\sigma = 1$ получим правую конечную разность, а при $\sigma = 1/2$ получим центральную конечную разность.

Один из ключевых аспектов разностных операторов – это их порядок аппроксимации. Понятие «порядок аппроксимации» определяет точность, с которой разностный оператор приближает производную функции $u(x)$ в точке x_j : чем выше порядок аппроксимации, тем более точное приближение и, следовательно, меньшая ошибка. Порядок аппроксимации оценивается относительно

h – шага разбиения. Известно, что правая и левая разности имеют порядок аппроксимации $O(h)$, в то время как центральная разность – $O(h^2)$.

Теперь построим разностную аппроксимацию производной второго порядка для функции $u(x)$ в точке x_j . Очевидно, что первая производная $u(x)$ сама является функцией $w(x)$ от той же самой переменной x , что и исходная функция $u(x)$. Поэтому производную второго порядка функции $u(x)$ можно представить как производную первого порядка функции $w(x)$: $u''(x) = w'(x)$. Аппроксимируя производную первого порядка функции $w(x)$ правой разностью, а производную первого порядка функции $u(x)$ – левой разностью, получим формулу конечно-разностного приближения производной второго порядка функции $u(x)$:

$$\lambda_{xx}u = \frac{\frac{u_{j+1} - u_j}{h} - \frac{u_j - u_{j-1}}{h}}{h} = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2}.$$

Этот разностный оператор имеет второй порядок аппроксимации, то есть $O(h^2)$.

Для нахождения численного решения дифференциальных уравнений дифференциальные операторы заменяются на конечно-разностные, после чего находятся приближенные значения искомой функции в узлах разбиения.

2.2.1 Метод конечных разностей для одномерных моделей

Рассмотрим одномерное дифференциальное уравнение параболического типа [12]:

$$\frac{\partial u}{\partial t} = \sigma \frac{\partial^2 u}{\partial x^2} + f(u, t, x).$$

Здесь $u = u(x, y)$ – функция двух независимых переменных, для которых задан интервал их изменения: $t \in [0, t_k]$, $x \in [a, b]$.

Введем прямоугольную систему координат, где на оси абсцисс откладывается независимая переменная x , а на оси ординат – независимая переменная t . Отметим на осях заданные интервалы изменения этих переменных. Разделим интервал $[a, b]$ на несколько равных частей и проведем из каждой точки деле-

ния прямые, перпендикулярные оси x . Аналогично поступим с интервалом изменения переменной t . Построенные таким образом прямые образуют разностную сетку (рисунок 2.2). Точки пересечения этих прямых будем называть узлами разностной сетки, при этом каждому узлу будут соответствовать определенные значения независимых переменных x и t из заданных интервалов.

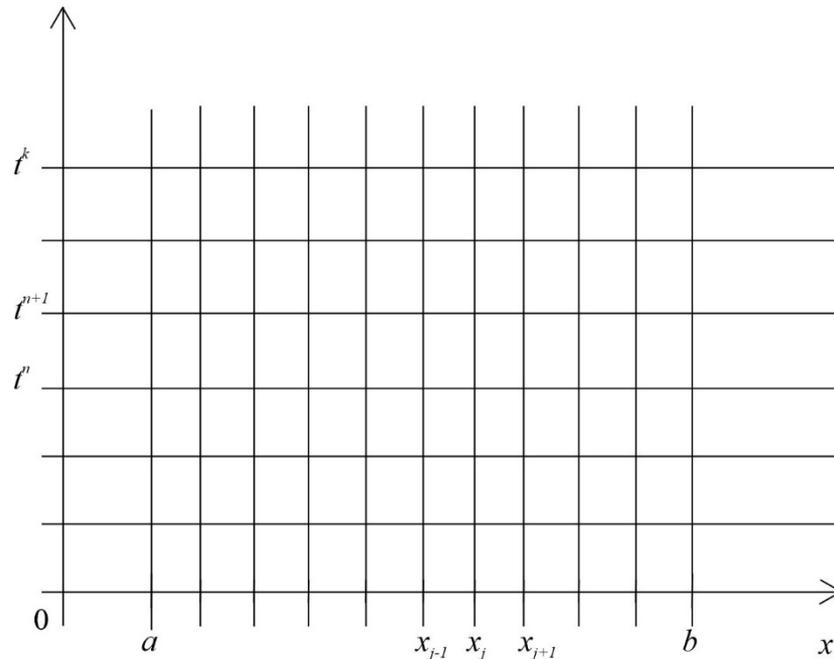


Рисунок 2.2 – Разностная сетка для функции двух независимых переменных

Введём следующие обозначения:

j – номер точек разбиения по оси x ;

n – номер точек разбиения по оси t ;

$x_{j+1} - x_j = \Delta x = h$ – длина интервала между точками разбиения по оси x ;

$t^{n+1} - t^n = \Delta t$ – длина интервала между точками разбиения по оси t ;

$u(t^n, x_j) = u_n^j$ – значение функции $u(t, x)$ в узле t^n, x^j .

Нумеровать узлы разностной сетки по каждой из осей будем следующим образом:

по оси $x - j = 1, 2, 3, \dots N$;

по оси $t - n = 0, 1, 2, \dots M$.

Заменим дифференциальные операторы исходного уравнения на конечно-разностные операторы. Первую производную по t будем аппроксимировать

правой конечной разностью, а вторую производную по x – соответствующей конечной разностью второго порядка. Подставив эти конечные разности в исходное дифференциальное уравнение, получим соотношение, аппроксимирующее это уравнение в точке (t^n, x_j) на разностной сетке. Это соотношение называется разностной схемой:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \sigma \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h} + f(u_j^n, t^n, x_j).$$

В полученной конечно-разностной схеме аппроксимация второй производной $u(t, x)$ по пространственной координате рассматривается на n -м шаге по времени, то есть в узле t^n , для которого рассматривается аппроксимация всего уравнения. В этом случае схему называют явной разностной схемой.

С другой стороны, аппроксимацию производной второго порядка функции $u(t, x)$ по пространственной координате можно брать и на $(n + 1)$ -м шаге по времени в узле t^{n+1} . В этом случае схему называют неявной разностной схемой:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \sigma \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h} + f(u_j^{n+1}, t^n, x_j).$$

Следует учитывать, что если в свободный член входит искомая функция $u(t, x)$, то её значение нужно брать на n -м шаге по времени при использовании явной конечно-разностной схемы и на $(n + 1)$ -м шаге при использовании неявной конечно-разностной схемы.

Представление узлов разностной сетки, связанных уравнением разностной схемы, в схематическом виде называют разностным шаблоном. Разностный шаблон является наглядным пособием при выборе метода решения конечно-разностной схемы и построении алгоритма ее решения. На рисунке 2.3 изображены разностные шаблоны для рассмотренных явной и неявной схем.

Явная конечно-разностная схема	Неявная конечно-разностная схема

Рисунок 2.3 – Разностные шаблоны для явной и неявной разностных схем

2.2.2 Конечно-разностные схемы расщепления для реализации двумерных моделей

В математических моделях реальных процессов, включая биологические, встречаются как одномерные, так и двумерные дифференциальные уравнения, для численного решения которых необходимы специальные подходы. В этом разделе мы приведем конечно-разностные схемы для решения двумерных по пространственной координате уравнений в частных производных параболического типа, не содержащих первых производных по пространственным координатам, которые особенно важны для целей настоящего исследования [4].

Пусть требуется решить дифференциальное уравнение:

$$\frac{\partial u}{\partial t} = \sigma \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f(x, y, t), \quad \sigma > 0.$$

Существуют различные конечно-разностные схемы, которые можно использовать для аппроксимации этого уравнения. Приведем их краткую характеристику.

Явная разностная схема [4]

$$\frac{u_{j,k}^{n+1} - u_{j,k}^n}{\Delta t} = \sigma \lambda_{xx} u_{j,k}^n + \sigma \lambda_{yy} u_{j,k}^n + f_{j,k}^n.$$

- Имеет порядок аппроксимации $O(\Delta t, h_x^2, h_y^2)$.
- Условно устойчива.
- Решается с помощью рекуррентного соотношения.

Схема дробных шагов Яненко [4, 13]

$$\frac{u_{j,k}^{n+1/2} - u_{j,k}^n}{\Delta t} = \sigma \lambda_{xx} u_{j,k}^{n+1/2} + f_{j,k}^n,$$

$$\frac{u_{j,k}^{n+1} - u_{j,k}^{n+1/2}}{\Delta t} = \sigma \lambda_{yy} u_{j,k}^{n+1}.$$

– Имеет порядок аппроксимации $O(\Delta t, h_x^2, h_y^2)$.

– Абсолютно устойчива.

– Каждая подсхема решается с помощью метода прогонки.

Метод переменных направлений Писмена–Рэчфорда [4, 10]

$$\frac{u_{j,k}^{n+1/2} - u_{j,k}^n}{\Delta t / 2} = \sigma \lambda_{xx} u_{j,k}^{n+1/2} + \sigma \lambda_{yy} u_{j,k}^n,$$

$$\frac{u_{j,k}^{n+1} - u_{j,k}^{n+1/2}}{\Delta t / 2} = \sigma \lambda_{xx} u_{j,k}^{n+1/2} + \sigma \lambda_{yy} u_{j,k}^{n+1} + f_{j,k}^{n+1/2}.$$

– Имеет порядок аппроксимации $O(\Delta t^2, h_x^2, h_y^2)$.

– Абсолютно устойчива (в двумерном случае).

– Каждая подсхема решается с помощью метода прогонки.

Существуют и другие схемы, такие как схема переменных направлений с экстраполяцией Формалева, схема со стабилизирующей поправкой, схема предиктор-корректор [9].

Рассмотрим применение схемы дробных шагов на примере решения дифференциального уравнения параболического типа с известным точным решением. Пусть необходимо решить дифференциальное уравнение:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x, y, t),$$

где $f(x, y, t) = 2 \cos(2t) \cdot x(x - L) \cdot y(y - L) - 2 \sin(2t) \cdot (x(x - L) + y(y - L))$, область изменения переменных:

$$x \in [0, L], \quad y \in [0, L], \quad t \in [0, T],$$

начальные и граничные условия:

$$u(x, y, 0) = 0,$$

$$u|_{\Gamma} = 0.$$

Для этого решения можно найти аналитическое решение, оно имеет вид:

$$u(x, y, t) = \sin(2t) \cdot x(x - L) \cdot y(y - L).$$

мы будем использовать это точное решение для оценки эффективности численного решения.

Введем следующую пространственно-временную сетку с шагами h_x , h_y , τ по переменным x , y , t :

$$\Omega_{h_x, h_y}^{\tau} = \left\{ x_i = ih_x, y_j = jh_y, t_k = j\tau, i = \overline{0, N}, j = \overline{0, M}, k = \overline{0, K} \right\},$$

на которой построим конечно-разностную схему решения дифференциального уравнения с расщеплением шага по времени методом дробных шагов. Член $f(x, y, t)$ будем учитывать как на первом, так и на втором полушаге по времени с коэффициентом $1/2$. Аппроксимация на первом полушаге по времени будет иметь вид:

$$\frac{u_{i,j}^{k+1/2} - u_{i,j}^k}{\tau} = \frac{u_{i+1,j}^{k+1/2} - 2u_{i,j}^{k+1/2} + u_{i-1,j}^{k+1/2}}{h_x^2} + \frac{1}{2} f_{i,j}^{k+1/2}.$$

Преобразуем уравнение к удобному для использования метода прогонки виду, а именно сгруппируем члены относительно неизвестных значений функции на шаге $k + 1/2$:

$$-\frac{\tau}{h_x^2} u_{i+1,j}^{k+1/2} + \left(1 + \frac{2\tau}{h_x^2}\right) u_{i,j}^{k+1/2} - \frac{\tau}{h_x^2} u_{i-1,j}^{k+1/2} = u_{i,j}^k + \frac{1}{2} \tau f_{i,j}^{k+1/2}.$$

Введем следующие обозначения:

$$a_i = -\frac{\tau}{h_x^2}, \quad b_i = 1 + \frac{2\tau}{h_x^2}, \quad c_i = -\frac{\tau}{h_x^2}, \quad \xi_i = u_{i,j}^k + \frac{1}{2} \tau f_{i,j}^{k+1/2}, \quad i = 1..N-1,$$

тогда прогоночные коэффициенты будут вычисляться по следующим формулам:

$$\alpha_i = \frac{-a_i}{b_i + c_i \alpha_{i-1}}, \quad \beta_i = \frac{\xi_i - c_i \beta_{i-1}}{b_i + c_i \alpha_{i-1}}, \quad i = 1..N-1.$$

из граничных условий получаем следующие значения начальных прогоночных коэффициентов и значение функции на правой границе:

$$\alpha_0 = 0, \quad \beta_0 = 0, \quad u_{N,j}^{k+1/2} = 0.$$

Обратная прогонка будет выполняться по следующей формуле:

$$u_{i,j}^{k+1/2} = \alpha_i u_{i+1,j}^{k+1/2} + \beta_i, \quad i = (N-1)..0,$$

при этом необходимо выполнять вычисления, перемещаясь по сетке в направлении оси y путем изменения значения j от 1 до $M-1$.

Перейдем к аппроксимации на втором полушаге по времени, схема будет иметь вид:

$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k+1/2}}{\tau} = \frac{u_{i,j+1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-1}^{k+1}}{h_y^2} + \frac{1}{2} f_{i,j}^{k+1}.$$

Аналогичным образом преобразуем уравнение для использования метода прогонки:

$$-\frac{\tau}{h_y^2} u_{i,j+1}^{k+1} + \left(1 + \frac{2\tau}{h_y^2}\right) u_{i,j}^{k+1} - \frac{\tau}{h_y^2} u_{i,j-1}^{k+1} = u_{i,j}^{k+1/2} + \frac{1}{2} \tau f_{i,j}^{k+1}.$$

Введем следующие обозначения:

$$\tilde{\alpha}_j = -\frac{\tau}{h_y^2}, \quad \tilde{b}_j = 1 + \frac{2\tau}{h_y^2}, \quad \tilde{c}_j = -\frac{\tau}{h_x^2}, \quad \tilde{\xi}_j = u_{i,j}^k + \frac{1}{2} \tau f_{i,j}^{k+1/2}, \quad j = 1..M-1,$$

тогда прогоночные коэффициенты будут вычисляться по следующим формулам:

$$\tilde{\alpha}_j = \frac{-\tilde{\alpha}_j}{\tilde{b}_j + \tilde{c}_j \tilde{\alpha}_{j-1}}, \quad \tilde{\beta}_j = \frac{\tilde{\xi}_j - \tilde{c}_j \tilde{\beta}_{j-1}}{\tilde{b}_j + \tilde{c}_j \tilde{\alpha}_{j-1}}, \quad j = 1..M-1.$$

из граничных условий получаем следующие значения начальных прогоночных коэффициентов и значение функции на правой границе:

$$\tilde{\alpha}_0 = 0, \quad \tilde{\beta}_0 = 0, \quad u_{i,M}^{k+1} = 0.$$

Обратная прогонка будет выполняться по следующей формуле:

$$u_{i,j}^{k+1} = \tilde{\alpha}_j u_{i,j+1}^{k+1} + \tilde{\beta}_j, \quad j = (M-1)..0,$$

при этом необходимо выполнять вычисления, перемещаясь по сетке в направлении оси x путем изменения значения i от 1 до $N-1$.

Реализация этой схемы была выполнена в среде Matlab на компьютере с четырёхъядерным процессором i7-3770K с тактовой частотой 3,50 ГГц. Значения параметров реализации: $T = 1$, $L = 1$, $N = M = 100$, $K = 5000$. Время выполнения оставило 394 секунды, максимальная погрешность на последнем шаге по времени – $4,2968 \cdot 10^{-5}$. Листинг программы приведен в приложении А. На ри-

сунке 2.4 представлен график численного решения на последнем шаге по времени.

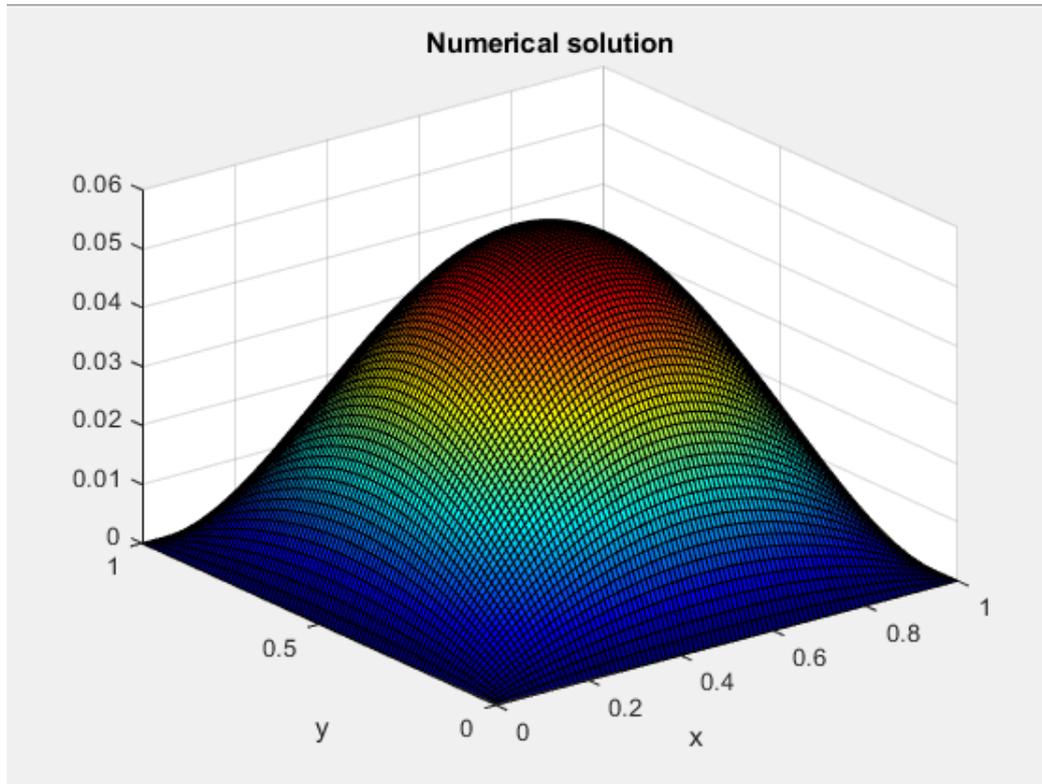


Рисунок 2.4 – График численного решения на последнем шаге по времени

Для иллюстрации сходимости численного решения приведем графики точного и приближенного решения при разном шаге разбиения. На рисунке 2.5 приведен график точного и приближенного решения на последнем шаге по времени на центральном срезе по $y = L/2$ при $M = N = K = 10$, в этом случае максимальная ошибка на последнем шаге по времени составила 0,0064. На рисунке 2.6 представлен тот же график при $M = N = 10$ и $K = 100$, в этом случае максимальная ошибка на последнем шаге по времени составила 0,0011. При $N = M = 100$, $K = 5000$ графики точного и приближенного решения сливаются в одну линию и неотличимы друг от друга. Видно, что приближенное решение приближается к точному при уменьшении шага разбиения.

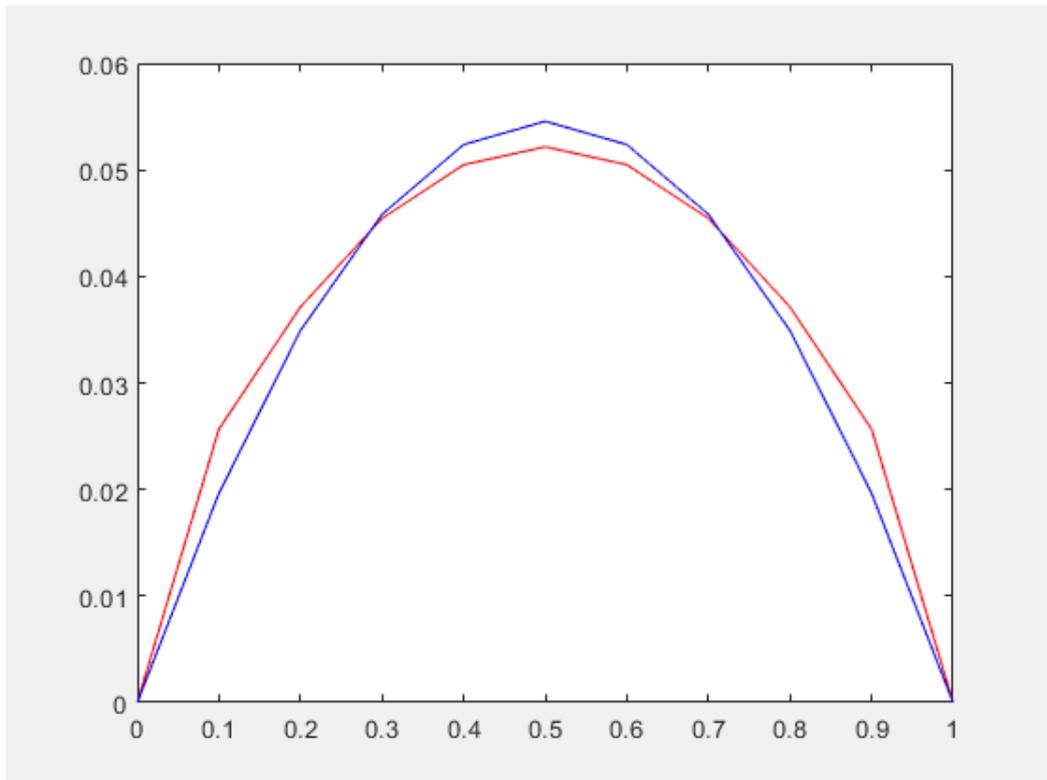


Рисунок 2.5 — Сравнение точного и приближенного решения на последнем шаге по времени на центральном срезе по $y = L/2$ при $M = N = K = 10$

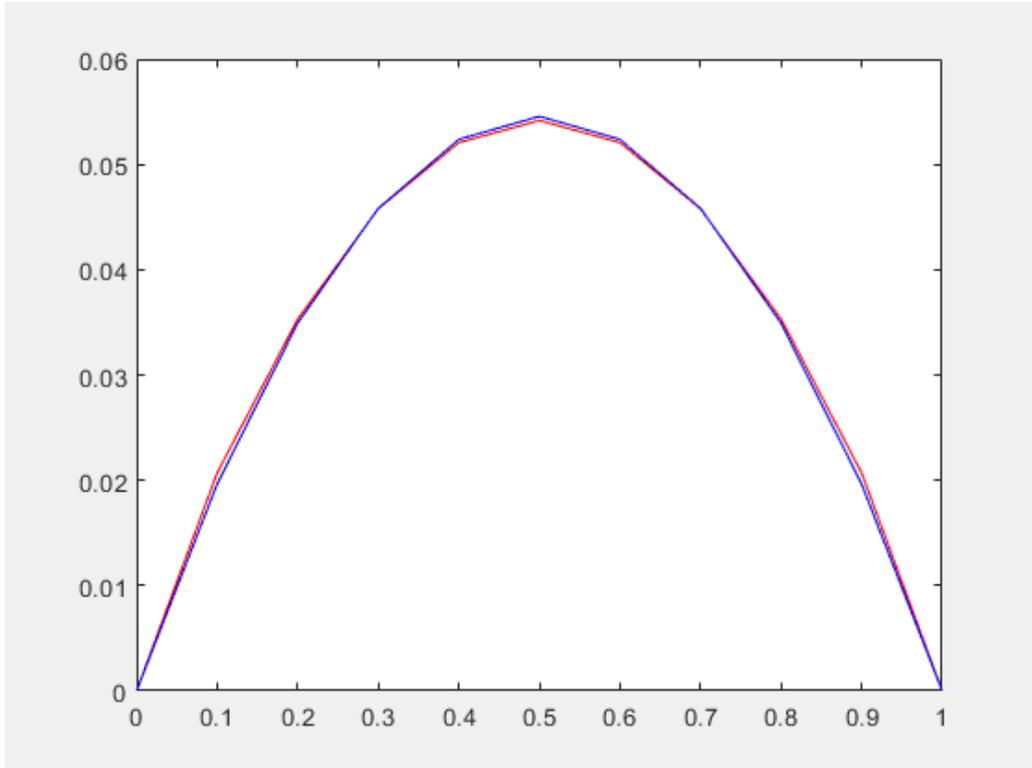


Рисунок 2.6 — Сравнение точного и приближенного решения на последнем шаге по времени на центральном срезе по $y = L/2$ при $M = N = 10, K = 100$

2.3 Программное обеспечение для решения задач математической физики

В большинстве случаев задачи математической физики, встречающиеся на практике, приводят к сложным, зачастую нелинейным, уравнениям в частных производных, которые невозможно или очень сложно решить аналитически. Кроме того, в силу многомерности и нелинейности этих задач, при их решении численными методами требуется большое количество вычислений, особенно если требуется достичь высокой точности. По этой причине появилось и активно используется специализированное программное обеспечение (ПО) для выполнения таких расчетов. Приведем описание некоторых наиболее популярных программных продуктов.

Пакет прикладных программ MATLAB (сокращение от "MATrix LABoratory") — это мощный многопарадигмальный язык программирования и среда для вычислений, разработанная компанией MathWorks. MATLAB предоставляет богатые возможности для работы с матрицами, построения графиков функций и данных, реализации вычислительных алгоритмов, создания пользовательских интерфейсов, а также для интеграции с программами, написанными на других языках программирования.

Несмотря на основное предназначение для числовых вычислений, MATLAB включает дополнительный набор инструментов, использующих символичный механизм MuPAD, который позволяет выполнять символьные вычисления. Кроме того, пакет расширения Simulink предоставляет возможности для графического комплексного моделирования и проектирования на основе моделей, что особенно полезно для работы с динамическими и встроенными системами.

В ППП MATLAB достаточно ограниченные возможности автоматизированного решения задач математической физики, однако его богатые возможности математического программирования значительно упрощают и ускоряют разработку собственных вычислительных схем. Кроме того, MATLAB обладает широкими возможностями по визуализации полученных решений.

COMSOL Multiphysics — это программный пакет конечно-элементного моделирования для различных физических и инженерных задач. Он обладает традиционным пользовательским интерфейсом, который упрощает решение физических задач и связанных систем уравнений в частных производных. COMSOL Multiphysics предоставляет интегрированную среду разработки и унифицированный рабочий процесс для электрических, механических, жидкостных, акустических и химических приложений.

Помимо классических задач, которые можно решить с помощью прикладных модулей, базовый пакет Multiphysics можно использовать для решения уравнений в слабой форме. API для Java и MATLAB можно использовать для внешнего управления программным обеспечением. Программа также служит сборщиком приложений для физических приложений. Для COMSOL доступно несколько модулей, классифицированных по областям применения: электрические, механические, гидравлические, акустические, химические, многоцелевые и интерфейсные.

Программное обеспечение ANSYS инженерного моделирования является одним из лидеров в этой области. Программный пакет ANSYS Mechanical для решения задач методом конечных элементов применяется для моделирования компьютерных моделей конструкций, электроники или машинных компонентов. Оно позволяет анализировать прочность, вязкость, эластичность, температурное поле, электромагнитные свойства, потоки жидких сред и другие характеристики. ANSYS помогает прогнозировать поведение продукта в различных условиях, исключая необходимость создания тестовых образцов или проведения прочностных испытаний. Например, можно смоделировать, как мост выдержит годы воздействия автомобильных пробок, как лучше всего перерабатывать лосося на консервном заводе, чтобы сократить количество отходов, или как спроектировать путепровод с меньшим использованием материалов без ущерба для безопасности.

Большинство симуляций ANSYS выполняется с использованием системы ANSYS Workbench. Обычно крупные части системы разделяются на более мел-

кие компоненты, каждый из которых моделируется и исследуется отдельно. Процесс начинается с определения размеров объекта моделирования, затем добавляются такие физические характеристики, как масса, давление, температура и другие параметры. Пакет ANSYS затем моделирует и исследует различные эффекты, такие как движение, усталость составляющих материалов, образование трещин, потоки жидких сред, температуру, электромагнитные характеристики и изменения других параметров во времени.

Перечисленные выше программные продукты являются коммерческими. Существуют и бесплатные, и даже с открытым исходным кодом, достаточно функциональные системы конечно-элементного моделирования, такие как FreeFEM и Elmer FEM.

Кроме того, для реализации вычислительных алгоритмов можно использовать языки программирования общего назначения, такие как C++, совместно с математическими библиотеками, такими как Kratos Multiphysics. Это, конечно, потребует больше усилий для написания программного кода но в то же время предоставляет разработчику полный контроль над всеми этапами вычислений и позволяет максимально гибко управлять своей программой.

В настоящей работе для реализации вычислительной схемы был выбран ППП MATLAB, для решения исследуемой задачи он предлагает оптимальный баланс между удобством разработки и гибкостью управления параметрами задачи.

2.4 Фрактальный анализ растровых изображений

Фракталы – это геометрические объекты, имеющие грубую или фрагментированную форму, которую можно разделить на части, каждая из которых является (по крайней мере приблизительно) уменьшенной копией целого. Это объекты извивающейся формы, которые не поддаются традиционным измерениям, таким как длина, и чаще всего характеризуются фрактальной размерностью. Эти математические объекты высокой степени геометрической сложности, которые позволяют моделировать многие природные явления. Почти во всех природных объектах наблюдается фрактальная структура (береговые ли-

нии, деревья, горы и облака, бактериальные колонии и т. д.). Фрактальная размерность строго превышает топологическую размерность [29].

Фрактальная размерность – это число, часто нецелое, часто единственная мера фракталов. Она измеряет степень фрагментации или неравномерности фрактальных границ в нескольких масштабах. Фрактальная размерность определяет, чем фрактал отличается от евклидовых объектов (точки, линии, плоскости, круга и т. д.).

Формальное определение фрактальной размерности можно задать в следующем виде [29]:

$$D = -\lim_{\varepsilon \rightarrow 0} \frac{\ln(N_\varepsilon)}{\ln(\varepsilon)},$$

где D – фрактальная размерность, N_ε – минимальное число n -мерных шаров радиуса ε , необходимых для покрытия множества. Согласно этому определению, фрактальная размерность евклидовых фигур (точки, линии, плоскости, круга, шара и т. д.) совпадает с их евклидовой размерностью.

На практике одним из самых простых и эффективных способов нахождения фрактальной размерности геометрических объектов является Метод покрытий (*box counting*) [30]. С помощью этого метода можно определить фрактальную размерность черно-белых оцифрованных изображений фракталов. Суть его заключается в покрытии фрактала (его изображения) прямоугольниками (квадратами), а затем оценке, сколько блоков необходимо, чтобы полностью покрыть фрактал. Повторение этого измерения с блоками разных размеров приведет к получению логарифмической функции размера блока (ось x) и количества блоков, необходимых для покрытия фрактала (ось y). Наклон этой функции называется размерностью покрытий (*box dimension*). Размерность покрытий принимается как подходящее приближение фрактальной размерности.

3 СИСТЕМА КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ПРОЦЕССОВ БАКТЕРИАЛЬНОГО РОСТА НА ПИТАТЕЛЬНЫХ СРЕДАХ

3.1 Концептуальная постановка задачи моделирования

В этой работе выполняется моделирование процессов, для которых известны экспериментальные микробиологические исследования, выполненные с использованием микроскопического наблюдения результатов. Краткое резюме этих экспериментов приведены в работе Кавасаки и др. [19], приведем их исходные данные.

В экспериментах исследовался бактериальный штамм сенной палочки (*Bacillus subtilis*), который имеет форму палочек со жгутиками (около 0,7 мкм в диаметре и 2 мкм в длину) и двигается в воде за счет вращения жгутиков. Выполнялся точечный посев бактерий в центр пластины с агаром, содержащей пептон в качестве питательного вещества, в пластиковой чашке Петри диаметром 88 мм.

В зависимости от концентрации питательного вещества C_n и плотности агаровой среды, выражаемой в виде $1/C_a$, где C_a – концентрация агара, развитие бактериальной колонии происходило с различным паттерном и с различной скоростью (см. рисунок 1).

В данной работе рассматривается случай, который на рисунке 3.1 выделен пунктирным прямоугольником., главным образом захватывающим область E . Область E характеризуется небольшим количеством питательных веществ и средней плотностью агара. В этих условиях колония росла радиально с плотной ветвящейся морфологией, и ее расширяющийся край был плавно закругленным. Типичный вид колонии такого типа показан на рисунке 3.2.

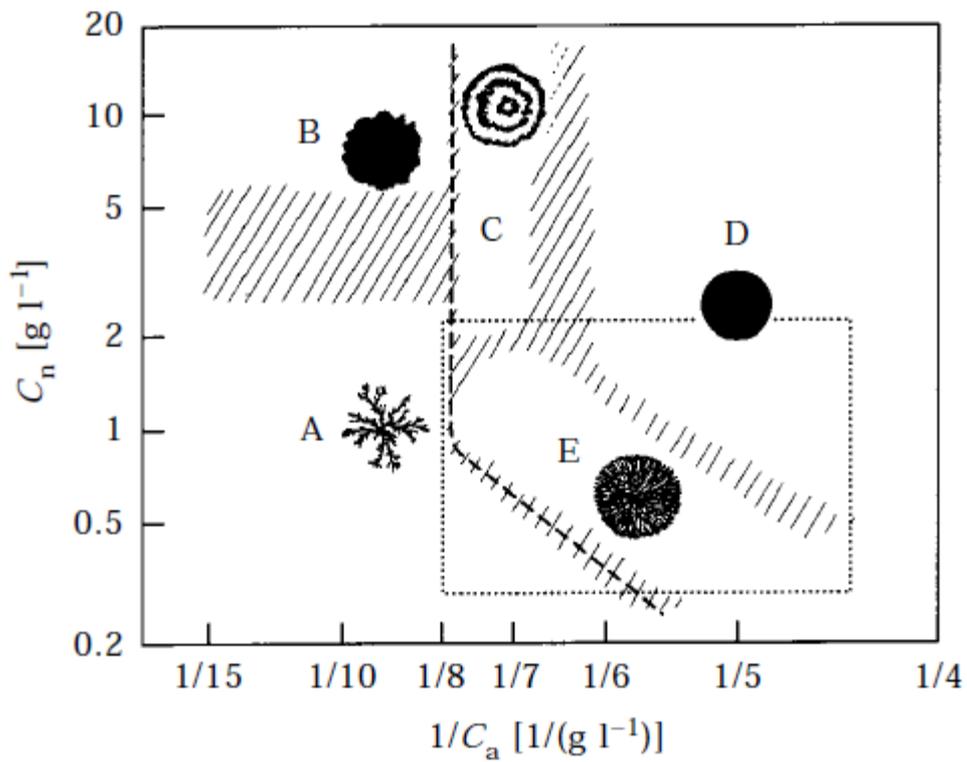


Рисунок 3.1 – Паттерны бактериальных колоний в зависимости от концентрации питательного вещества C_n и плотности агаровой среды $1/C_a$ [19]

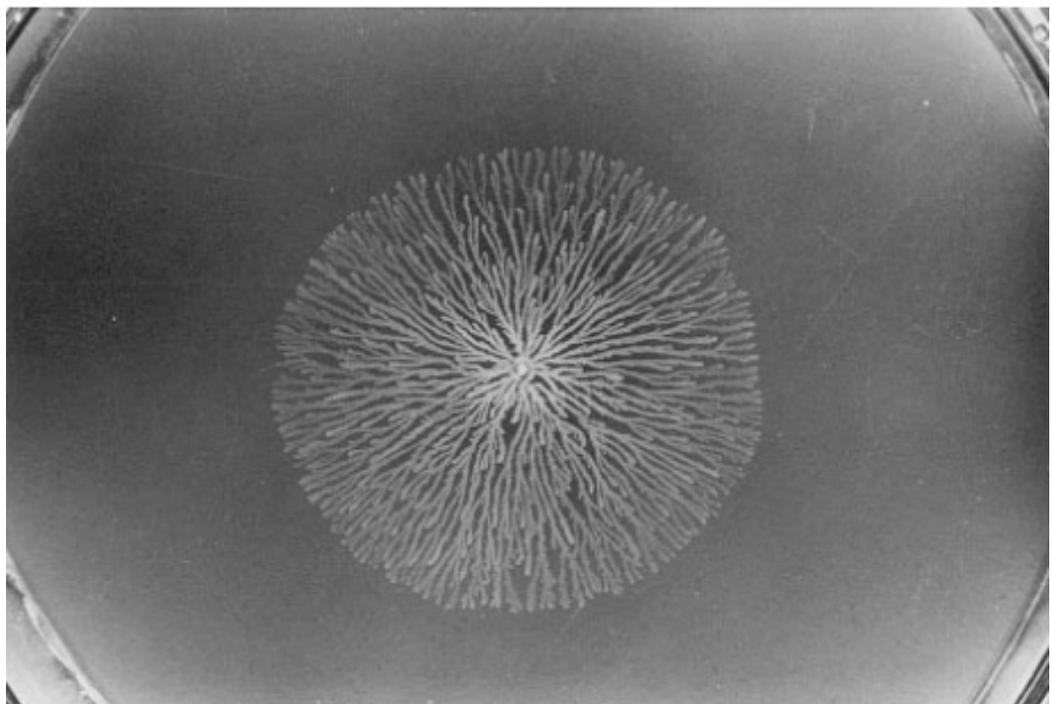


Рисунок 3.2 – Фотография типичной бактериальной колонии с плотной ветвящейся морфологией [19]

3.2 Математическая постановка задачи моделирования

Рассмотрим модель роста бактериальной колонии в условиях, указанных в предыдущем пункте, которая была предложена в работе Кавасаки и др. [19]. Эта модель описывает двумерную систему из бактериальной биомассы и питательного вещества (нутриента). Бактерии и питательное вещество подвергаются воздействию диффузии. В дополнение к этому бактериальные клетки размножаются, потребляя нутриент. Пусть $b(x, y, t)$ – это плотность популяции бактериальных клеток (в количестве клеток на единицу площади), которая зависит от координат x, y и от времени t , а $n(x, y, t)$ – это концентрация питательного вещества (в единицах массы на единицу площади). Тогда функции b и n будут связаны следующими соотношениями:

$$\begin{aligned}\frac{\partial n}{\partial t} &= D_n \nabla^2 n - f(n, b), \\ \frac{\partial b}{\partial t} &= \nabla (D_b \nabla b) + \theta f(n, b),\end{aligned}$$

где D_b и D_n – коэффициенты диффузии для бактериальной колонии и питательного вещества соответственно. При этом параметр D_n является постоянным, а параметр D_b одновременно зависит от плотности бактериальной массы и от концентрации питательного вещества. Конкретную форму D_b тяжело определить экспериментально, поэтому в рассматриваемой работе используется наиболее простой вид:

$$D_b = \sigma n b,$$

где $\sigma = \sigma_0(1 + \Delta)$, значение параметра σ_0 пропорционально концентрации питательного вещества, Δ задает случайные флуктуации, характерные для биологических систем. Член $f(n, b)$ задает скорость потребления нутриента бактериальной массой, а член $\theta f(n, b)$ – задает скорость роста клеток, где θ – скорость преобразования (урожай клеток) потребляемого питательного вещества в рост бак-

терий. В предложенной модели предполагается, что скорость потребления описывается уравнением Михаэлиса-Ментен:

$$f(n,b) = knb / (1 + \gamma n),$$

где k – собственная скорость потребления, а k/γ – максимальная скорость потребления одной клетки. Для рассматриваемого случая с относительно низкой концентрацией нутриента f можно упростить до следующего вида:

$$f(n,b) = knb.$$

Установим следующие начальные условия:

$$\begin{aligned} n(x, y, 0) &= n_0, \\ b(x, y, 0) &= b_0(x, y), \end{aligned}$$

где n_0 – начальная концентрация нутриента, равномерно распределенная по рассматриваемой области, а $b_0(x, y)$ – плотность посева, расположенного в центре области.

Для уменьшения количества параметров модели введем замену переменных:

$$\begin{aligned} n' &= \sqrt{\frac{\theta}{D_n}} n, & b' &= \sqrt{\frac{1}{\theta D_n}} b, \\ x' &= \sqrt[4]{\frac{\theta k^2}{D_n}} x, & y' &= \sqrt[4]{\frac{\theta k^2}{D_n}} y, & t' &= \sqrt{\theta d_n} t. \end{aligned}$$

Подставляя новые переменные и опуская штрих для удобства записи, получим:

$$\begin{aligned} \frac{\partial n}{\partial t} &= \nabla^2 n - bn, \\ \frac{\partial b}{\partial t} &= \nabla(\sigma n b \nabla b) + nb, \end{aligned} \tag{3.1}$$

с начальными условиями

$$n(x, y, 0) = \sqrt{\frac{\theta}{D_n}} n_0 \equiv v_0,$$

$$b(x, y, 0) = \sqrt{\frac{1}{\theta D_n}} b_0(x, y) \equiv \beta_0(x, y).$$

В итоге мы получили модель с тремя изменяемыми параметрами: σ , v_0 и β_0 , где v_0 и β_0 – относительные величины начальной концентрации питательного вещества и плотности бактерий соответственно. Для численного моделирования выберем в качестве β_0 нормальное распределение:

$$\beta_0(x, y) = \beta_M e^{-\frac{x^2+y^2}{6,25}},$$

где β_M – максимальная плотность в центре области.

3.3 Построение вычислительной схемы

Для численной реализации математической модели воспользуемся методом конечных разностей. Сконструируем неявную конечно-разностную схему, применяя метод дробных шагов Яненко, как мы это делали при решении тест-примера в п. 2.4. Сначала рассмотрим конечно-разностную аппроксимацию первого уравнения системы (3.1). Введем пространственно-временную сетку с шагами h_x, h_y, τ по переменным x, y, t соответственно:

$$\Omega_{h_x h_y}^\tau = \left\{ x_i = -L + ih_x, y_j = -L + jh_y, t_k = j\tau, i = \overline{0, N}, j = \overline{0, M}, k = \overline{0, K} \right\}$$

Аппроксимация распределения нутриента n на первом полушаге по времени выглядит следующим образом:

$$\frac{n_{i,j}^{k+1/2} - n_{i,j}^k}{\tau} = \frac{n_{i+1,j}^{k+1/2} - 2n_{i,j}^{k+1/2} + n_{i-1,j}^{k+1/2}}{h_x^2} - \frac{1}{2}b_{i,j}^k n_{i,j}^k \Rightarrow$$

$$-\frac{\tau}{h_x^2} n_{i+1,j}^{k+1/2} + \left(1 + 2\frac{\tau}{h_x^2}\right) n_{i,j}^{k+1/2} - \frac{\tau}{h_x^2} n_{i-1,j}^{k+1/2} = n_{i,j}^k - \frac{\tau}{2} b_{i,j}^k n_{i,j}^k.$$

Коэффициенты системы линейных алгебраических уравнений имеют вид:

$$a_i = c_i = -\frac{\tau}{h_x^2}, \quad b_i = 1 + 2\frac{\tau}{h_x^2}, \quad \xi_i = n_{i,j}^k - \frac{\tau}{2} b_{i,j}^k n_{i,j}^k.$$

Прогоночные коэффициенты запишем в форме:

$$\alpha_i = \frac{-a_i}{b_i + c_i \alpha_{i-1}}, \quad \beta_i = \frac{\xi_i - c_i \beta_{i-1}}{b_i + c_i \alpha_{i-1}}.$$

Граничные условия на левой границе:

$$\frac{n_{1,j}^{k+1/2} - n_{0,j}^{k+1/2}}{h_x^2} = 0 \Rightarrow n_{1,j}^{k+1/2} - n_{0,j}^{k+1/2} = 0 \Rightarrow n_{1,j}^{k+1/2} = n_{0,j}^{k+1/2} \Rightarrow \alpha_1 = 1, \beta_1 = 0.$$

На правой границе $\frac{\partial n}{\partial x} = \varphi(y, t) \equiv 0$, тогда значение аппроксимирующей

функции в узлах $n_{N,j}^{k+1/2}$ будет определяться соотношением

$$n_{N,j}^{k+1/2} = \frac{h_x \varphi_j^{k+1/2} + \beta_{N-1}}{1 - \alpha_{N-1}} = \frac{\beta_{N-1}}{1 - \alpha_{N-1}}.$$

Начальные условия: $n_{i,j}^0 = n_0$.

Аппроксимация n на втором полушаге по времени выглядит следующим образом:

$$\frac{n_{i,j}^{k+1} - n_{i,j}^{k+1/2}}{\tau} = \frac{n_{i,j+1}^{k+1} - 2n_{i,j}^{k+1} + n_{i,j-1}^{k+1}}{h_y^2} - \frac{1}{2}b_{i,j}^{k+1/2}n_{i,j}^{k+1/2} \Rightarrow$$

$$-\frac{\tau}{h_y^2}n_{i,j+1}^{k+1} + \left(1 + 2\frac{\tau}{h_y^2}\right)n_{i,j}^{k+1} - \frac{\tau}{h_y^2}n_{i,j-1}^{k+1} = n_{i,j}^{k+1/2} - \frac{\tau}{2}b_{i,j}^{k+1/2}n_{i,j}^{k+1/2}.$$

Коэффициенты системы линейных алгебраических уравнений имеют вид:

$$\tilde{\alpha}_j = \tilde{c}_j = -\frac{\tau}{h_y^2}, \quad \tilde{b}_j = 1 + 2\frac{\tau}{h_y^2}, \quad \tilde{\xi}_i = n_{i,j}^{k+1/2} - \frac{\tau}{2}b_{i,j}^{k+1/2}n_{i,j}^{k+1/2}.$$

Прогоночные коэффициенты:

$$\tilde{\alpha}_j = \frac{-\tilde{\alpha}_j}{\tilde{b}_j + \tilde{c}_j\tilde{\alpha}_{j-1}}, \quad \tilde{\beta}_i = \frac{\tilde{\xi}_j - \tilde{c}_j\tilde{\beta}_{j-1}}{\tilde{b}_j + \tilde{c}_j\tilde{\alpha}_{j-1}}.$$

Граничные условия на нижней границе:

$$\frac{n_{i,1}^{k+1} - n_{i,0}^{k+1}}{h_y^2} = 0 \Rightarrow n_{i,1}^{k+1} - n_{i,0}^{k+1} = 0 \Rightarrow n_{i,1}^{k+1} = n_{i,0}^{k+1} \Rightarrow \tilde{\alpha}_1 = 1, \quad \tilde{\beta}_1 = 0.$$

На верхней границе $\frac{\partial n}{\partial y} = \psi(x, t) \equiv 0$, тогда значение аппроксимирующей

функции в узлах $n_{i,M}^{k+1}$ будет определяться соотношением

$$n_{i,M}^{k+1} = \frac{h_y \psi_i^{k+1} + \tilde{\beta}_{M-1}}{1 - \tilde{\alpha}_{M-1}} = \frac{\tilde{\beta}_{M-1}}{1 - \tilde{\alpha}_{M-1}}.$$

Перейдем к построению аппроксимации для второго уравнения. Сложность его решения заключается в присутствии нелинейности коэффициента диффузии σnb , который зависит как от n , так и от b . Перед построением конечно-разностной схемы сначала раскроем оператор дивергенции, используя фор-

мулу производной произведения (параметр σ вынесем за знак дивергенции, так как он не зависит от n и b):

$$\operatorname{div}(n \cdot b \cdot \operatorname{grad}(b)) = \frac{\partial n}{\partial x} b \frac{\partial b}{\partial x} + n \frac{\partial b}{\partial x} \frac{\partial b}{\partial x} + nb \frac{\partial^2 b}{\partial x^2} + \frac{\partial n}{\partial y} b \frac{\partial b}{\partial y} + n \frac{\partial b}{\partial y} \frac{\partial b}{\partial y} + nb \frac{\partial^2 b}{\partial y^2}$$

Второе уравнение примет вид:

$$\frac{\partial b}{\partial t} = \sigma nb \left(\frac{\partial^2 b}{\partial x^2} + \frac{\partial^2 b}{\partial y^2} \right) + nb + \sigma \left(\frac{\partial n}{\partial x} b + n \frac{\partial b}{\partial x} \right) \frac{\partial b}{\partial x} + \sigma \left(\frac{\partial n}{\partial y} b + n \frac{\partial b}{\partial y} \right) \frac{\partial b}{\partial y}$$

При решении уравнений в частных производных, содержащих первую производную по координате, методом конечных разностей выбор правой или левой конечной разности для аппроксимации первых производных зависит от знака коэффициента при этой производной. Поскольку в нашем случае мы не можем заранее знать знак коэффициента, будем для первых производных использовать центральные конечные разности. Запишем конечно-разностную аппроксимацию полученного уравнения:

$$\begin{aligned} \frac{b_{i,j}^{k+1} - b_{i,j}^k}{\tau} &= \sigma n_{i,j}^k b_{i,j}^k \left(\frac{b_{i+1,j}^{k+1} - 2b_{i,j}^{k+1} + b_{i-1,j}^{k+1}}{h_x^2} + \frac{b_{i,j+1}^{k+1} - 2b_{i,j}^{k+1} + b_{i,j-1}^{k+1}}{h_y^2} \right) + \frac{1}{2} n_{i,j}^k b_{i,j}^k + \\ &+ \sigma \left(\frac{n_{i+1,j}^k - n_{i-1,j}^k}{2h_x} b_{i,j}^k + n_{i,j}^k \frac{b_{i+1,j}^k - b_{i-1,j}^k}{2h_x} \right) \frac{b_{i+1,j}^{k+1} - b_{i-1,j}^{k+1}}{2h_x} + \\ &+ \sigma \left(\frac{n_{i,j+1}^k - n_{i,j-1}^k}{2h_y} b_{i,j}^k + n_{i,j}^k \frac{b_{i,j+1}^k - b_{i,j-1}^k}{2h_y} \right) \frac{b_{i,j+1}^{k+1} - b_{i,j-1}^{k+1}}{2h_y}. \end{aligned}$$

Далее применим схему дробных шагов и запишем аппроксимацию на первом полушаге по времени:

$$\frac{b_{i,j}^{k+1/2} - b_{i,j}^k}{\tau} = \sigma n_{i,j}^k b_{i,j}^k \frac{b_{i+1,j}^{k+1/2} - 2b_{i,j}^{k+1/2} + b_{i-1,j}^{k+1/2}}{h_x^2} + \frac{1}{2} n_{i,j}^k b_{i,j}^k +$$

$$+ \sigma \left(\frac{n_{i+1,j}^k - n_{i-1,j}^k}{2h_x} b_{i,j}^k + n_{i,j}^k \frac{b_{i+1,j}^k - b_{i-1,j}^k}{2h_x} \right) \frac{b_{i+1,j}^{k+1/2} - b_{i-1,j}^{k+1/2}}{2h_x}.$$

Коэффициенты системы примут вид:

$$a_i = -\frac{\tau \sigma n_{i,j}^k b_{i,j}^k}{h_x^2} - \eta_{i,j}^k, \quad b_i = 1 + 2 \frac{\tau \sigma n_{i,j}^k b_{i,j}^k}{h_x^2}, \quad c_i = -\frac{\tau \sigma n_{i,j}^k b_{i,j}^k}{h_x^2} + \eta_{i,j}^k, \quad \xi_i = b_{i,j}^k + \frac{\tau}{2} n_{i,j}^k b_{i,j}^k,$$

$$\text{где } \eta_{i,j}^k = \frac{\sigma \tau}{4h_x^2} \left((n_{i+1,j}^k - n_{i-1,j}^k) b_{i,j}^k + n_{i,j}^k (b_{i+1,j}^k - b_{i-1,j}^k) \right).$$

$$\text{Прогоночные коэффициенты: } \alpha_i = \frac{-a_i}{b_i + c_i \alpha_{i-1}}, \quad \beta_i = \frac{\xi_i - c_i \beta_{i-1}}{b_i + c_i \alpha_{i-1}}.$$

Граничные условия на левой границе:

$$\frac{b_{1,j}^{k+1/2} - b_{0,j}^{k+1/2}}{h_x^2} = 0 \Rightarrow b_{1,j}^{k+1/2} - b_{0,j}^{k+1/2} = 0 \Rightarrow b_{1,j}^{k+1/2} = b_{0,j}^{k+1/2} \Rightarrow \alpha_1 = 1, \quad \beta_1 = 0.$$

На правой границе $\frac{\partial b}{\partial x} = \bar{\varphi}(y, t) \equiv 0$, тогда значение аппроксимирующей

функции в узлах $b_{N,j}^{k+1/2}$ будет определяться соотношением

$$b_{N,j}^{k+1/2} = \frac{h_x \bar{\varphi}_j^{k+1/2} + \beta_{N-1}}{1 - \alpha_{N-1}} = \frac{\beta_{N-1}}{1 - \alpha_{N-1}}. \quad \text{Начальные условия: } b_{i,j}^0 = \beta_m \exp\left(-\frac{x_i^2 + y_j^2}{C}\right).$$

Аппроксимация на втором полушаге по времени будет иметь вид:

$$\frac{b_{i,j}^{k+1} - b_{i,j}^{k+1/2}}{\tau} = \sigma n_{i,j}^k b_{i,j}^k \frac{b_{i,j+1}^{k+1} - 2b_{i,j}^{k+1} + b_{i,j-1}^{k+1}}{h_y^2} + \frac{1}{2} n_{i,j}^{k+1/2} b_{i,j}^{k+1/2} +$$

$$+ \sigma \left(\frac{n_{i,j+1}^{k+1/2} - n_{i,j-1}^{k+1/2}}{2h_y} b_{i,j}^{k+1/2} + n_{i,j}^{k+1/2} \frac{b_{i,j+1}^{k+1/2} - b_{i,j-1}^{k+1/2}}{2h_y} \right) \frac{b_{i,j+1}^{k+1} - b_{i,j-1}^{k+1}}{2h_y}$$

Коэффициенты системы:

$$\tilde{a}_i = -\frac{\tau\sigma n_{i,j}^{k+1/2} b_{i,j}^{k+1/2}}{h_y^2} - \tilde{\eta}_{i,j}^k, \quad \tilde{b}_i = 1 + 2\frac{\tau\sigma n_{i,j}^{k+1/2} b_{i,j}^{k+1/2}}{h_y^2},$$

$$\tilde{c}_i = -\frac{\tau\sigma n_{i,j}^{k+1/2} b_{i,j}^{k+1/2}}{h_y^2} + \tilde{\eta}_{i,j}^k, \quad \tilde{\xi}_i = b_{i,j}^{k+1/2} + \frac{\tau}{2} n_{i,j}^{k+1/2} b_{i,j}^{k+1/2},$$

где $\tilde{\eta}_{i,j}^k = \frac{\sigma\tau}{4h_y^2} \left((n_{i,j+1}^{k+1/2} - n_{i,j-1}^{k+1/2}) b_{i,j}^{k+1/2} + n_{i,j}^{k+1/2} (b_{i,j+1}^{k+1/2} - b_{i,j-1}^{k+1/2}) \right)$.

Прогоночные коэффициенты: $\tilde{\alpha}_j = \frac{-\tilde{a}_j}{\tilde{b}_j + \tilde{c}_j \tilde{\alpha}_{j-1}}, \quad \tilde{\beta}_j = \frac{\tilde{\xi}_j - \tilde{c}_j \tilde{\beta}_{j-1}}{\tilde{b}_j + \tilde{c}_j \tilde{\alpha}_{j-1}}$.

Граничные условия на нижней границе:

$$\frac{b_{i,1}^{k+1} - b_{i,0}^{k+1}}{h_y^2} = 0 \Rightarrow b_{i,1}^{k+1} - b_{i,0}^{k+1} = 0 \Rightarrow b_{i,1}^{k+1} = b_{i,0}^{k+1} \Rightarrow \tilde{\alpha}_1 = 1, \quad \tilde{\beta}_1 = 0.$$

На верхней границе $\frac{\partial b}{\partial y} = \psi(x, t) \equiv 0$, тогда значение аппроксимирующей

функции в узлах $b_{i,M}^{k+1}$ будет определяться соотношением

$$b_{i,M}^{k+1} = \frac{h_y \tilde{\psi}_i^{k+1} + \tilde{\beta}_{M-1}}{1 - \tilde{\alpha}_{M-1}} = \frac{\tilde{\beta}_{M-1}}{1 - \tilde{\alpha}_{M-1}}.$$

3.4 Вычислительные эксперименты и анализ результатов

Сконструированный вычислительный алгоритм был реализован программно с использованием среды ППП Matlab. В качестве параметра δ была взята случайная величина с треугольным распределением на отрезке $[-\rho, -\rho]$, значение δ генерировалось отдельно для каждого узла аппроксимации. Вычисления проводились при следующих нормированных значениях параметров модели и численных параметров сетки:

$$n_0 = 0.71, \beta_0 = 0.71,$$

$$\sigma_0 = \{0.1; 0.55; 1; 2\}, \rho = \{0; 0.5; 1\}, C^2 = 6.25, \beta_m = 0.71,$$

$$L = 200, T = 2000,$$

$$N = M = 1000, K = 2100.$$

Результаты вычислительных экспериментов по визуализации пространственной конфигурации биомассы в последние моменты времени наблюдения представлены на рисунке 3.3.

Моделирование проводилось при вариации двух управляющих параметров модели: σ_0 и ρ , отвечающих за изменение геометрической конфигурации бактериальных паттернов. Вариация параметра σ_0 (увеличение значения) приводит к нарушению гладкости границы бактериальной колонии и от круглой формы она трансформируется сначала к «цветочной форме» и, далее, к дендритоподобной разветвленной структуре. При этом параметр $\rho = 0$ соответствует полностью симметричным паттернам. Конфигурации более адекватные данным биологического эксперимента визуализируются при введении стохастической поправки и, соответственно, при $0 < \rho \leq 1$ фигуры имеют выраженную асимметрию.

В результате проведения серии вычислительных экспериментов выявлено, что фрактальная размерность увеличивается вместе с ростом параметров σ_0 и ρ . Максимальное значение фрактальной размерности границы $D_L = 1.47$ в этой серии экспериментов достигнуто при значениях параметров $\sigma_0 = 1.45$ и $\rho = 1$ (рисунок 3.4). Также отдельно проведено исследование зависимости фрактальной размерности границы от уровня начального питания n_0 (рисунок 3.5).

На рисунке 3.6 представлен результат численного моделирования геометрии бактериальных паттернов и пространственного распределения питания в финальный момент симуляции при изменении начального питания n_0 . Здесь наглядно виден процесс сначала усложнения ветвистой структуры, а затем «склеивания» ветвей и переход к гомогенной геометрии.

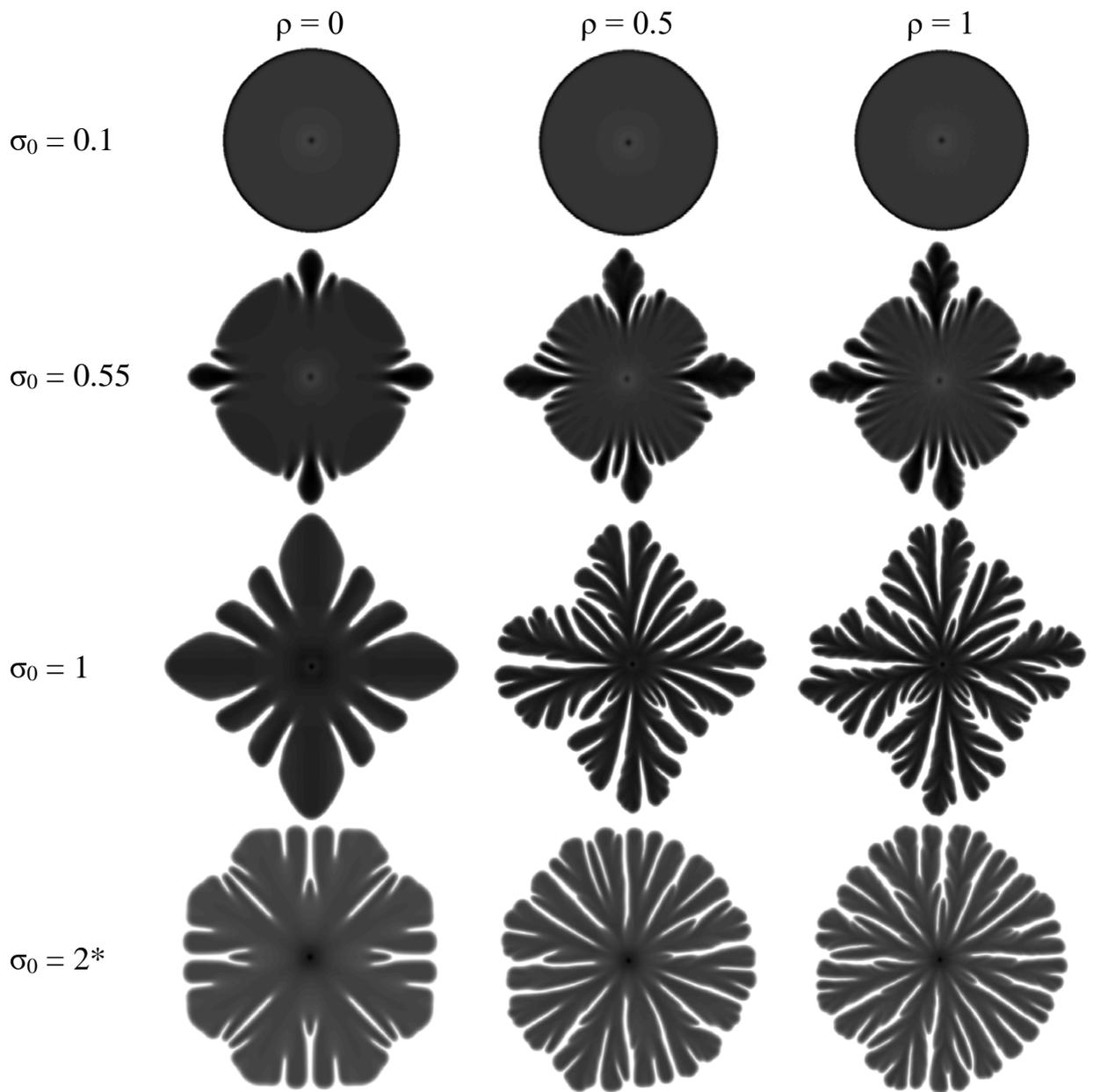


Рисунок 3.3 – Результаты моделирования при различных значениях параметров ρ и σ (* при $\sigma = 2$ рост колонии происходит слишком быстро, поэтому симуляция выполнялась при уменьшенном значении времени $T = 1000$).

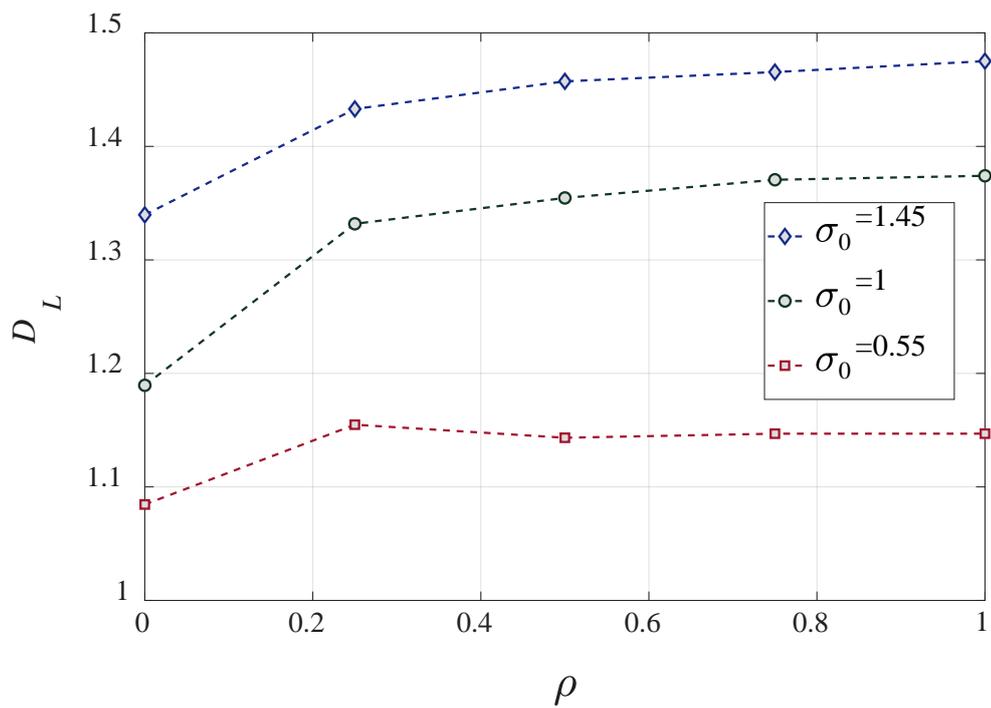


Рисунок 3.4 – Зависимость фрактальной размерности границы D_L от параметров σ_0 и ρ

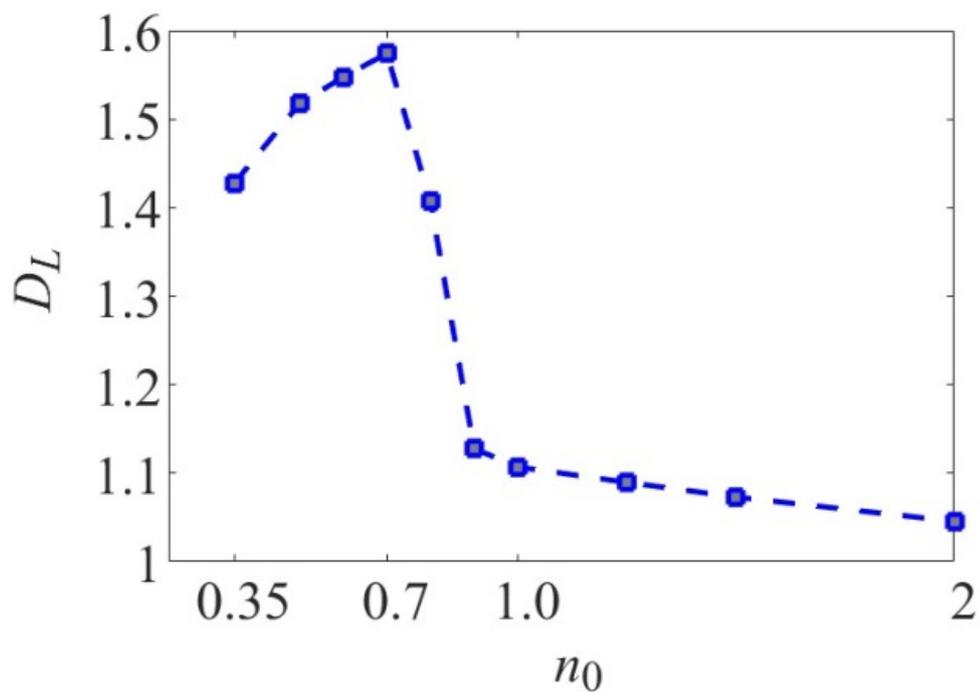


Рисунок 3.5 – Зависимость фрактальной размерности границы D_L от параметра n_0

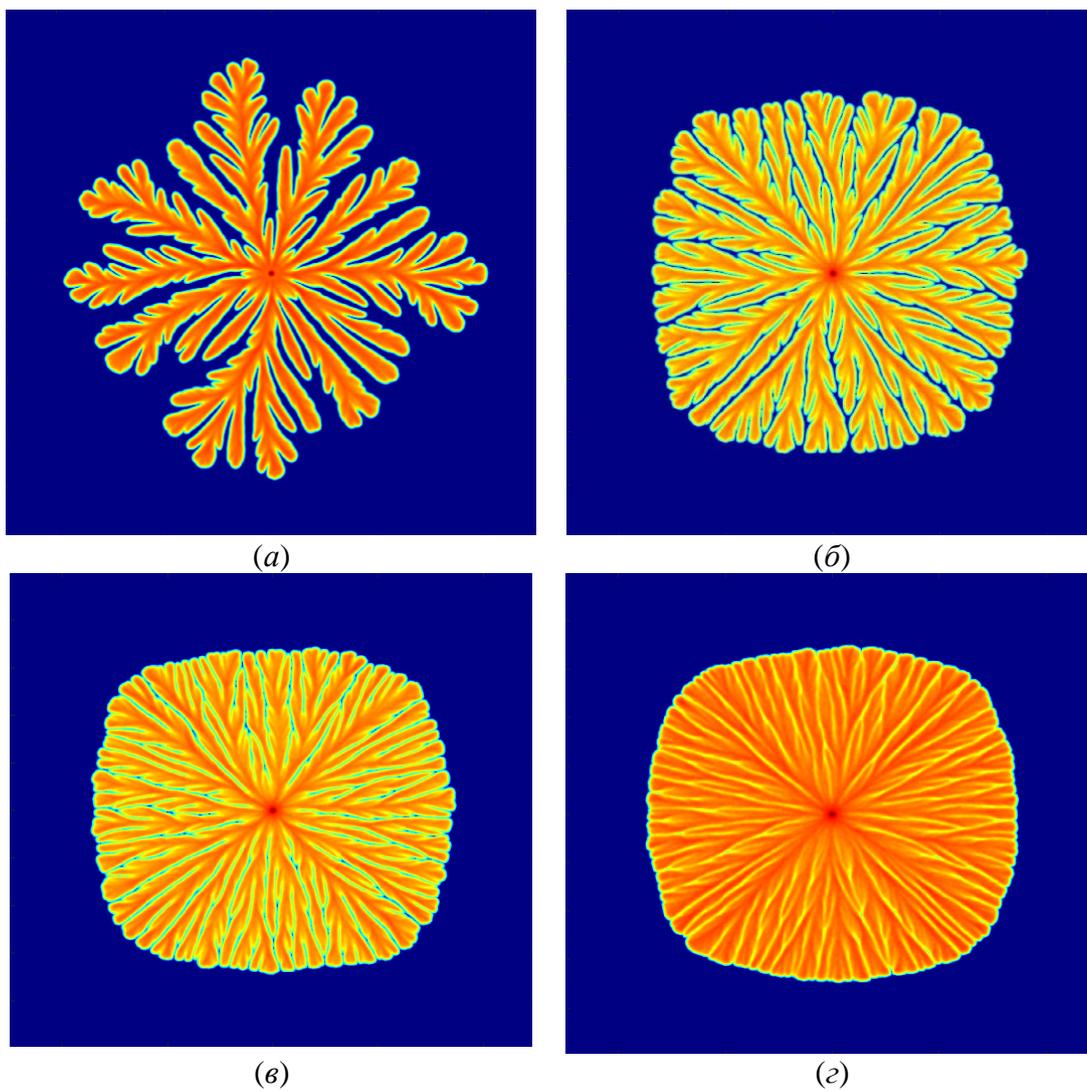


Рисунок 3.6 – Результаты численного моделирования пространственного распределения бактериальной биомассы в зависимости от начального питания:

$$n_0 = 0.4 \text{ (a)}, n_0 = 0.8 \text{ (б)}, n_0 = 1.0 \text{ (в)}, n_0 = 1.25 \text{ (г)}.$$

ЗАКЛЮЧЕНИЕ

Работа обобщает результат численной реализации дифференциальной модели формирования дендритоподобной геометрии бактериальных колоний, выращенных на питательных средах.

На основе подробного дифференциального представления была проведена модификация аналитической записи нелинейного диффузионного слагаемого уравнения системы, отвечающего за рост бактериальной массы. Применяя неявную конечно-разностную схему расщепления, сконструирован вычислительный алгоритм и проведена его программная реализация в пакете прикладных программ Matlab. Проведена серия вычислительных экспериментов, направленных на выявление управляющих параметров модели, которые определяют формирование дендритоподобной геометрии решения задачи. Установлено, что увеличение параметра, связанного с коэффициентом диффузии, приводит к усложнению структуры границы, а добавление стохастического механизма изменения этого параметра позволяет визуализировать структуры, соответствующие естественным бактериальным колониям, наблюдаемым в практике биологического эксперимента. Также установлено, что начальный уровень питания также напрямую влияет на геометрию решения, и при переходе от бедного питания к насыщенному сначала происходит усложнение ветвистой структуры, но на определенном уровне ветви «склеиваются» и происходит переход к однородной кругообразной форме.

В работе также был проведен количественный анализ указанных закономерностей с помощью вычисления фрактальной размерности границы получаемых бактериальных колоний методом покрытий (*box counting*), что позволяет формально измерить степень «ветвистости» геометрии, а также тонко настраивать параметры модели для наибольшего соответствия экспериментальным данным.

Результаты исследований, полученные в рамках этой работы, были апробированы на 7 научных конференциях регионального, всероссийского и меж-

дународного уровня и оформлены в 6 публикациях, в том числе в рецензируемых изданиях. Для реализации вычислительного алгоритма в пакете прикладных программ было получено свидетельство о государственной регистрации программы для ЭВМ.

Научно-исследовательская работа в 2023-2024 гг. была поддержана грантом АмГУ для выполнения научно-исследовательского проекта «Система моделирования бактериального роста на питательных средах: гибридный подход».

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации, проект № 122082400001-8.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Арзамасцев, А.А. Математические модели кинетики микробиологического синтеза: возможности использования и новые подходы к разработке / А. А. Арзамасцев, А. А. Андреев // Вестник ТГУ. – 2000. – Т.5 Вып.1. – С. 111–123.
- 2 Динамические модели в биологии, информационная система [Электронный ресурс] : МГУ им. Ломоносова. – М., 2022 – Режим доступа : <http://www.dmb.biophys.msu.ru>. – 17.12.2022.
- 3 Колмогоров, А.Н. Исследование уравнения диффузии, соединенной с возрастанием количества вещества и его применение к одной биологической проблеме / А.Н. Колмогоров. – М. : Онти. Глав. ред. техн.-теоретич. лит-ры, 1937. – 26 с.
- 4 Кольцова, Э.М. Численные методы решения уравнений математической физики и химии / Э. М. Кольцова, А. С. Скичко, А.В. Женса. – М. : Юрайт, 2022. – 220 с.
- 5 Масловская, А.Г. Численные методы: использование инструментальных средств и реализация алгоритмов на базе ППП MATLAB / А. Г. Масловская, А. В. Павельчук. – Благовещенск : Амурский гос. ун-т, 2020. – 214 с.
- 6 Медицинская микробиология, вирусология и иммунология / под ред. А. С. Быкова, В. В. Зверева. – М. : Медицинское информационное агентство, 2016. – 816 с.
- 7 Мороз, Л.И. Решение дифференциальных задач методом конечных элементов / Л. И. Мороз, А. Г. Масловская. – Благовещенск: Амурский гос. ун-т, 2019 - 71 с.
- 8 Ризниченко, Г. Ю. Лекции по математическим моделям в биологии / Г. Ю. Ризниченко. – М. : РХД, 2011. – 560 с.
- 9 Салмиянов, В. О. Адаптированные алгоритмы интеллектуальной спецификации скейлинговых характеристик морфологии культивируемых бакте-

рий / В. О. Салмиянов, И.А. Шевкун, А. Г. Масловская // Информатика и системы управления. – 2023. – С. 75-89.

10 Сегерлинд, Л. Применение метода конечных элементов / Л. Сегерлинд. – М. : Мир, 1976. – 393 с.

11 Формалев, В. Ф. Численные методы / В. Ф. Формалев, Д. Л. Ревизников. – М. : ФИЗМАТЛИТ, 2004. – 400 с.

12 Шевкун, И.А. Вариация диффузионных режимов при численном моделировании бактериального роста на питательных средах / И.А. Шевкун, А.Г. Масловская // Информационные технологии и высокопроизводительные вычисления: Материалы VII международной научно-практической конференции. – 2023. – С. 257-259.

13 Шевкун, И.А. Вариация моделей дендритоподобной морфологии бактериальных культур: численная реализация в рамках реакционно-диффузионного приближения / И.А. Шевкун, А.Г. Масловская // Вестник АмГУ. Серия: Естественные и экономические науки. – 2023. – № 101. – С. 30-39.

14 Шевкун, И.А. Диффузионные эффекты модели бактериального роста: детерминированный подход / И.А. Шевкун // День науки: Материалы XXXII научной конференции. – 2023. – С. 64-65.

15 Шевкун, И.А. Компьютерное моделирование роста бактериальных колоний на питательных средах: детерминированный подход/ И.А. Шевкун, А.П. Хмелев // Молодежь XXI века: шаг в будущее: материалы XXIV региональной научно-практической конференции. – 2023. – С. 200-201.

16 Шевкун, И.А. Характеристики геометрических паттернов для реакционно-диффузионной модели роста бактерий при культивировании на питательных средах / И.А. Шевкун, А.Г. Масловская // Вычислительные технологии и прикладная математика: Материалы II Международного семинара. –2023. – С. 232-234.

17 Шеннон, Р. Имитационное моделирование систем - искусство и наука / Р. Шеннон. – М. : Мир, 1978. – 421 с.

18 Яненко, Н. Н. Метод дробных шагов решения многомерных задач математической физики / Н. Н. Яненко. – Новосибирск : Наука, 1967. – 196 с.

19 Vazhlekov, I. Fractional derivative modeling of bioreaction-diffusion processes [Электронный ресурс] / I. Vazhlekov, E. Vazhlekova // AIP Publishing. – 08.03.2021. – Режим доступа : <https://aip.scitation.org/doi/10.1063/5.0041611>. – 20.12.2022.

20 Demirel, Y. Nonequilibrium Thermodynamics: Transport and Rate Processes in Physical, Chemical and Biological Systems, Fourth Edition / Y. Demirel, V. Gerbaud. – Elsevier. – 2019. – 854 с.

21 Eberl, H.J. A transient density-dependent diffusion-reaction model for the limitation of antibiotic penetration in biofilms / H. J. Eberl, M. A. Efendiev // Electronic Journal of Differential Equations. – 2003. – №10. – С. 123-142.

22 Eisenbach, M. Bacterial Chemotaxis / M. Eisenbach // Encyclopedia of Life Sciences. – 01.04.2001. – Режим доступа : https://www.researchgate.net/publication/228050570_Bacterial_Chemotaxis. – 18.12.2022.

23 El-Sayed, A.M.A. On the solutions of time-fractional bacterial chemotaxis in a diffusion gradient chamber / A.M.A. El-Sayed, S.Z. Rida, A.A.M. Arafa // International Journal of Nonlinear Science. – 2009. – №7. – С. 485-492.

24 Kawasaki, K. Modeling spatio-temporal patterns generated by *Bacillus subtilis* / K. Kawasaki, A. Mochizuki, M. Matsushita, T. Umeda, N. Shigesada // Journal of Theoretical Biology. – 1997. – №188. – С. 177-185.

25 Khassehkhan, H. A Nonlinear Master Equation for a Degenerate Diffusion Model of Biofilm Growth / H. Khassehkhan, T. Hillen, H.J. Eberl // Lecture Notes in Computer Science. – 2009. – №5. – С. 735-744.

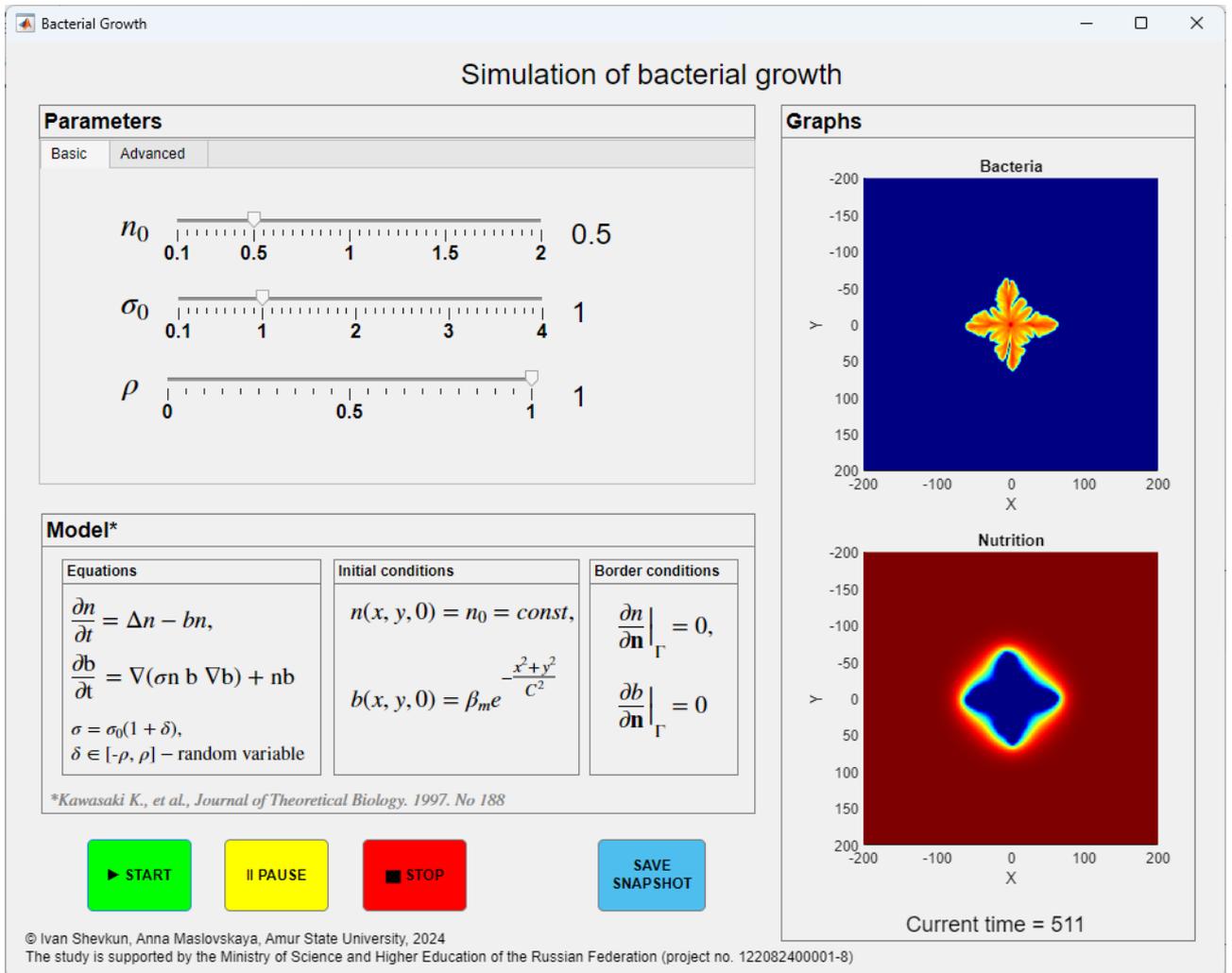
26 Lagziel, Sh. Mind your media / Sh. Lagziel, E. Gottlieb, T. Shlomi // Nature Metabolism. – 2020. – №2. – С. 1369-1372.

27 Madigan, J.M. Brock Biology Of Microorganisms 11th edition / J.M. Madigan, M.T. Martinko. – Pearson. – 2006. – 1056 с.

- 28 Magin R.L. Fractional calculus in bioengineering // Critical Reviews™ in Biomedical Engineering. - 2004. - №32(1). - С. 1-104.
- 29 Mathematical Modeling of Biological Systems, Volume II. / A. Deutsch и др. – Boston: Birkhäuser Boston, MA, 2007. – 386 с.
- 30 Mitra, K. Travelling waves in a PDE-ODE coupled system with nonlinear diffusion [Электронный ресурс] / K. Mitra, J. M. Hughes, S. Sonner, H. J. Eberl, J. D. Dockery // arXiv®. – 15.02.2022. – Режим доступа : <https://arxiv.org/abs/2202.07748>. – 20.12.2022.
- 31 Paintera, K. J. Mathematical models for chemotaxis and their applications in self-organisation phenomena / K. J. Paintera // Journal of Theoretical Biology. – 2019. – №481. – С. 162-182.
- 32 Parija, S.Ch. Microbiology and Immunology / S.Ch. Parija. – Elsevier. – 2012. – 682 с.
- 33 Rida, S. Z. Effect of Bacterial Memory Dependent Growth by Using Fractional Derivatives Reaction-Diffusion Chemotactic Model / S. Z. Rida, A. M. A. El-Sayed, A. A. M. Arafa // Journal of Statistical Physics. – 2010. – №140. – С. 797–811.
- 34 Shanmugavadivu, P. Fractal Dimension Based Texture Analysis of Digital Images / P. Shanmugavadivu, V. Sivakumar // Procedia Engineering. – 2012. – №38. – С. 2981-2986.
- 35 Zmeskal, O. Fractal Analysis of Image Structures / O. Zmeskal et. al // HarFA - Harmonic and Fractal Image Analysis. – 2001. – С. 3-5.

ПРИЛОЖЕНИЕ А

Интерфейс программы



Продолжение Приложения А

Bacterial Growth
— □ ×

Simulation of bacterial growth

Parameters

Basic
Advanced

L $-L < x, y < L$ β_m

T $0 < t \leq T$ C^2

Mesh

Rough (fast)
 Moderate
 Fine (slow)

Graphs

Bacteria

Nutr

Current time = 614

Model*

Equations	Initial conditions	Border conditions
$\frac{\partial n}{\partial t} = \Delta n - bn,$ $\frac{\partial b}{\partial t} = \nabla(\sigma n \mathbf{b} \nabla b) + nb$ <p>$\sigma = \sigma_0(1 + \delta),$ $\delta \in [-\rho, \rho]$ – random variable</p>	$n(x, y, 0) = n_0 = const,$ $b(x, y, 0) = \beta_m e^{-\frac{x^2+y^2}{C^2}}$	$\frac{\partial n}{\partial \mathbf{n}} \Big _{\Gamma} = 0,$ $\frac{\partial b}{\partial \mathbf{n}} \Big _{\Gamma} = 0$

*Kawasaki K., et al., Journal of Theoretical Biology. 1997. No 188

▶ START

⏸ PAUSE

■ STOP

SAVE SNAPSHOT

© Ivan Shevkun, Anna Maslovskaya, Amur State University, 2024
 The study is supported by the Ministry of Science and Higher Education of the Russian Federation (project no. 122082400001-8)

ПРИЛОЖЕНИЕ Б

Листинг программы

```
classdef BacterialGrowth < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        BacterialGrowthUIFigure matlab.ui.Figure
        ErrorsLabel              matlab.ui.control.Label
        STOPButton               matlab.ui.control.Button
        PAUSEButton              matlab.ui.control.Button
        SimulationofbacterialgrowthLabel matlab.ui.control.Label
        SAVESNAPSHOTButton      matlab.ui.control.Button
        ModelPanel               matlab.ui.container.Panel
        BorderconditionsPanel    matlab.ui.container.Panel
        fracpartialnpartialtDeltanbnLabel_5 matlab.ui.control.Label
        fracpartialnpartialtDeltanbnLabel_4 matlab.ui.control.Label
        InitialconditionsPanel  matlab.ui.container.Panel
        fracpartialnpartialtDeltanbnLabel_2 matlab.ui.control.Label
        fracpartialnpartialtDeltanbnLabel_3 matlab.ui.control.Label
        EquationsPanel           matlab.ui.container.Panel
        fracpartialnpartialtDeltanbnLabel_6 matlab.ui.control.Label
        fracpartialnpartialtDeltanbnLabel matlab.ui.control.Label
        textitKawasakiKetalJournalofTheoreticalBiology1997No188Label
        matlab.ui.control.Label
        IvanShevkunAnnaMaslovskaya2024Label matlab.ui.control.Label
        ParametersPanel          matlab.ui.container.Panel
        TabGroup                  matlab.ui.container.TabGroup
        BasicTab                  matlab.ui.container.Tab
        rho_Label                 matlab.ui.control.Label
        sigma0_Label             matlab.ui.control.Label
        n0_Label                  matlab.ui.control.Label
        rho_Slider                matlab.ui.control.Slider
        rhoLabel                  matlab.ui.control.Label
        sigma0_Slider             matlab.ui.control.Slider
        sigma_0Label              matlab.ui.control.Label
        n0_Slider                 matlab.ui.control.Slider
        n_0Label                  matlab.ui.control.Label
        AdvancedTab               matlab.ui.container.Tab
        tleqTLabel                matlab.ui.control.Label
        LxyLLabel                 matlab.ui.control.Label
        TEditField                matlab.ui.control.NumericEditField
        TLabel                    matlab.ui.control.Label
        LEditField                matlab.ui.control.NumericEditField
        LLabel                     matlab.ui.control.Label
        C2EditField               matlab.ui.control.NumericEditField
        C2Label                    matlab.ui.control.Label
        beta_mEditField           matlab.ui.control.NumericEditField
        beta_0Label               matlab.ui.control.Label
        MeshButtonGroup           matlab.ui.container.ButtonGroup
        FineButton                 matlab.ui.control.RadioButton
        ModerateButton            matlab.ui.control.RadioButton
        RoughButton               matlab.ui.control.RadioButton
        GraphsPanel               matlab.ui.container.Panel
        CurrenttimestepLabel      matlab.ui.control.Label
        UIAxes_n                  matlab.ui.control.UIAxes
        UIAxes_b                  matlab.ui.control.UIAxes
        STARTButton               matlab.ui.control.Button
    end

end
```

Продолжение Приложения Б

```

properties (Access = private)
    started = false;
    paused = false;
    tostop = false;
    T_cur = 0;
end

methods (Access = private)

function z=zerob(app, x,y,offset_x, offset_y, bm, delta)
    x = x - offset_x;
    y = y - offset_y;
    z = bm * exp(-(x*x + y*y)/delta);
end

function start_calc(app, n0, sigma0, rho, mesh, bm, C2, L, T)

    XY_coef = 1;
    T_coef = 1;

    % Parameter initialisation
    Dn = XY_coef/T_coef;
    vn = 1;
    vb = 1;

    % n0 = 0.5;
    % bm = 0.71;
    delta = C2*(XY_coef);
    sigma0 = sigma0*XY_coef/T_coef;
    % rho = 1; % The random variable varies in the range [-rho, rho]

    % Initializing a random number generator using
    % a triangular distribution to calculate sigma
    if rho > 0
        pd = makedist('Triangular','A',-rho,'B',0,'C',rho); rng('default');
    else
        pd = 0;
    end;

    % L = 100; % x, y from [-L, L]
    T = T*T_coef; % t from [0, T]

    Tmax = T; Xmax = L; Ymax = L; % Maximum Variable Values
    nt = T*mesh; nx = 2*L*mesh; ny = 2*L*mesh; % Number of intervals in the
mesh
    tau = Tmax/nt; hx = 2*Xmax/nx; hy = 2*Ymax/ny; % mesh step

    %
    t = 0:tau:Tmax;
    x = -Xmax:hx:Xmax;
    y = -Ymax:hy:Ymax;

    % n_aprx = zeros(nx+1,ny+1); % The value of n at the current integer time
step
    n_semisteps = zeros(nx+1,ny+1); % The value of n at the current time
semistep

    % b_aprx = zeros(nx+1,ny+1); % The value of b at the current integer time
step

```

Продолжение Приложения Б

```

semistep    b_semisteps = zeros(nx+1,ny+1); % The value of b at the current time

% Setting initial conditions
nt0 = zeros(nx+1,ny+1) + n0;
bt0 = zeros(nx+1,ny+1);
for i = 1:(nx+1)
    for j = 1:(ny+1)
        bt0(i, j) = zerob(app, x(i), y(j), 0, 0, bm, delta);
    end
end

n_aprx = nt0;
b_aprx = bt0;

% Calculating Linear equation coefficients for n
a_n_first = zeros(1, nx-1) - Dn*tau/(hx^2);
c_n_first = a_n_first;
b_n_first = zeros(1, nx-1) + 1 + Dn*2*tau/(hx^2);

a_n_second = zeros(1, ny-1) - Dn*tau/(hy^2);
c_n_second = a_n_second;
b_n_second = zeros(1, ny-1) + 1 + Dn*2*tau/(hy^2);

% Disable the visibility of graphs to save to file
f=figure('visible','off');

% Loop by t variable
for k = 1:nt

    if app.tostop
        app.started = false;
        return
    end

    while app.paused
        pause(1);
    end

    app.CurrenttimestepLabel.Text = strcat("Current time = ",
string(round(tau*k)));
    app.T_cur = round(tau*k);

    % First time semistep

    % Calculation of delta for each mesh node at the first time semistep
    sigma_random = 0;
    if rho > 0
        sigma_random = random(pd,nx+1,ny+1); % sigma_random its a random
parameter "delta"
    end;

    for j = 1:ny+1

        % Calculating n
        % Coefficients for Tridiagonal matrix algorithm
        clear beta_n alpha_n;

```

Продолжение Приложения Б

```

alpha_n = zeros(1, nx); alpha_n(1) = 1; % <-- Zero flow at the bor-
der
beta_n = zeros(1, nx); beta_n(1) = 0;
for i = 2:(nx)
    ksi_n = n_aprx(i,j) -
vn*(0.5/T_coef)*tau*b_aprx(i,j)*n_aprx(i,j);
    alpha_n(i) = -a_n_first(i-1)/(b_n_first(i-1) + c_n_first(i-
1)*alpha_n(i-1));
    beta_n(i) = (ksi_n - c_n_first(i-1)*beta_n(i-1))/(b_n_first(i-
1) + c_n_first(i-1)*alpha_n(i-1));
end
% Zero flow at the border
n_semisteps(nx+1,j) = beta_n(nx)/(1-alpha_n(nx));
% Reverse substitution
for i = nx:-1:1
    n_semisteps(i,j) = alpha_n(i)* n_semisteps(i+1,j) + beta_n(i);
end
% End of n calculation

% Calculating b

% Calculating of linear equations coefficients and
% Coefficients for Tridiagonal matrix algorithm
clear a_b b_b c_b;
a_b = zeros(1, nx-1); b_b = zeros(1, nx-1); c_b = zeros(1, nx-1);
clear beta_b alpha_b;
alpha_b = zeros(1, nx); alpha_b(1) = 1; % <-- Zero flow at the bor-
der
beta_b = zeros(1, nx); beta_b(1) = 0;

for i = 2:(nx)
    if rho > 0
        sigma = sigma0*(1 + sigma_random(i,j));
    else
        sigma = sigma0;
    end;

    eta = tau*sigma*(b_aprx(i,j)*(n_semisteps(i+1,j) -
n_semisteps(i-1,j)) + ...
        n_semisteps(i,j)*(b_aprx(i+1,j) - b_aprx(i-1,j)))/(2*hx^2);
    %eta = 0;
    temp = -tau*sigma*n_semisteps(i,j)*b_aprx(i,j)/(hx^2);
    a_b(i-1) = temp - eta;
    b_b(i-1) = 1 + (-2*temp);
    c_b(i-1) = temp + eta;
    ksi_b = b_aprx(i,j) +
vb*(0.5/T_coef)*tau*b_aprx(i,j)*n_semisteps(i,j);
    alpha_b(i) = -a_b(i-1)/(b_b(i-1) + c_b(i-1)*alpha_b(i-1));
    beta_b(i) = (ksi_b - c_b(i-1)*beta_b(i-1))/(b_b(i-1) + c_b(i-
1)*alpha_b(i-1));
end

% Zero flow at the border
b_semisteps(nx+1,j) = beta_b(nx)/(1-alpha_b(nx));
% Reverse substitution
for i = nx:-1:1
    b_semisteps(i,j) = alpha_b(i)* b_semisteps(i+1,j) + beta_b(i);
end
% End of b calculation

```

Продолжение Приложения Б

```

end

% Second time semistep

% Calculation of delta for each mesh node at the first time semistep
if rho > 0
    sigma_random = random(pd,nx+1,ny+1); % sigma_random its a random
parameter "delta"
end

for i = 1:nx+1

    % Calculating n
    % Coefficients for Tridiagonal matrix algorithm
    clear beta_n alpha_n;
    alpha_n = zeros(1, ny); alpha_n(1) = 1; % <-- Zero flow at the bor-
der

    beta_n = zeros(1, ny); beta_n(1) = 0;
    for j = 2:(ny)
        ksi_n = n_semisteps(i,j) -
vn*(0.5/T_coef)*tau*b_semisteps(i,j)*n_semisteps(i,j);
        alpha_n(j) = -a_n_second(j-1)/(b_n_second(j-1) + c_n_second(j-
1)*alpha_n(j-1));
        beta_n(j) = (ksi_n - c_n_second(j-1)*beta_n(j-
1))/(b_n_second(j-1) + c_n_second(j-1)*alpha_n(j-1));
    end
    % Zero flow at the border
    n_aprx(i,ny+1) = beta_n(ny)/(1-alpha_n(ny));
    % Reverse substitution
    for j = ny:-1:1
        n_aprx(i,j) = alpha_n(j)* n_aprx(i,j+1) + beta_n(j);
    end
    % End of n calculation

    % Calculating b

    % Calculating of linear equations coefficients and
    % Coefficients for Tridiagonal matrix algorithm
    clear a_b b_b c_b;
    a_b = zeros(1, ny-1); b_b = zeros(1, ny-1); c_b = zeros(1, ny-1);
    clear beta_b alpha_b;
    alpha_b = zeros(1, ny); alpha_b(1) = 1; % <-- Zero flow at the bor-
der

    beta_b = zeros(1, ny); beta_b(1) = 0;

    for j = 2:(ny)
        if rho > 0
            sigma = sigma0*(1 + sigma_random(i,j));
        else
            sigma = sigma0;
        end;

        eta = tau*sigma*(b_semisteps(i,j)*(n_aprx(i,j+1) - n_aprx(i,j-
1)) + ...
            n_aprx(i,j)*(b_semisteps(i,j+1) - b_semisteps(i,j-
1)))/(2*hy^2);

        %eta = 0;
        temp = -tau*sigma*n_aprx(i,j)*b_semisteps(i,j)/(hy^2);
        a_b(j-1) = temp - eta;
    end
end

```

Продолжение Приложения Б

```

        b_b(j-1) = 1 + (-2*temp);
        c_b(j-1) = temp + eta;
        ksi_b = b_semisteps(i,j) +
vb*(0.5/T_coef)*tau*b_semisteps(i,j)*n_aprx(i,j);
        alpha_b(j) = -a_b(j-1)/(b_b(j-1) + c_b(j-1)*alpha_b(j-1));
        beta_b(j) = (ksi_b - c_b(j-1)*beta_b(j-1))/(b_b(j-1) + c_b(j-
1)*alpha_b(j-1));
        end

        % Zero flow at the border
        b_aprx(i,ny+1) = beta_b(ny)/(1-alpha_b(ny));
        % Reverse substitution
        for j = ny:-1:1
            b_aprx(i,j) = alpha_b(j)* b_aprx(i,j+1) + beta_b(j);
        end
        % End of b calculation

    end

    imagesc(app.UIAxes_b, [-Xmax Xmax] , [-Ymax Ymax], b_aprx);
    imagesc(app.UIAxes_n, [-Xmax Xmax] , [-Ymax Ymax], n_aprx);
    drawnow;

end

app.started = false;
end
end

% Callbacks that handle component events
methods (Access = private)

% Button pushed function: STARTButton
function STARTButtonPushed(app, event)
    if app.started
        return;
    end
    app.tostop = false;
    app.started = true;
    mesh = 2;
    if app.RoughButton.Value
        mesh = 1;
    elseif app.ModerateButton.Value
        mesh = 2
    elseif app.FineButton.Value
        mesh = 4;
    end
    start_calc(app, app.n0_Slider.Value, app.sigma0_Slider.Value,
app.rho_Slider.Value, mesh, ...
        app.beta_mEditField.Value, app.C2EditField.Value, app.LEditField.Value,
app.TEditField.Value);
end

% Value changing function: n0_Slider
function n0_SliderValueChanging(app, event)
    value = event.Value;
    [~, minIdx] = min(abs(value - event.Source.MinorTicks(:)));
    event.Source.Value = event.Source.MinorTicks(minIdx);

```

Продолжение Приложения Б

```
    app.n0_Label.Text = string(event.Source.MinorTicks(minIdx));
end

% Value changing function: sigma0_Slider
function sigma0_SliderValueChanging(app, event)
    value = event.Value;
    [~, minIdx] = min(abs(value - event.Source.MinorTicks(:)));
    event.Source.Value = event.Source.MinorTicks(minIdx);
    app.sigma0_Label.Text = string(event.Source.MinorTicks(minIdx));
end

% Value changing function: rho_Slider
function rho_SliderValueChanging(app, event)
    value = event.Value;
    [~, minIdx] = min(abs(value - event.Source.MinorTicks(:)));
    event.Source.Value = event.Source.MinorTicks(minIdx);
    app.rho_Label.Text = string(event.Source.MinorTicks(minIdx));
end

% Value changed function: LEditField
function LEditFieldValueChanged(app, event)
    value = app.LEditField.Value;
    app.UIAxes_b.XLim = [-value value];
    app.UIAxes_b.YLim = [-value value];
    app.UIAxes_n.XLim = [-value value];
    app.UIAxes_n.YLim = [-value value];
end

% Button pushed function: SAVESNAPSHOTButton
function SAVESNAPSHOTButtonPushed(app, event)
    exportgraphics(app.UIAxes_b, strcat('b_T=',
num2str(app.T_cur, '%06.f'), '.jpg'), 'Resolution', 300);
    exportgraphics(app.UIAxes_n, strcat('n_T=',
num2str(app.T_cur, '%06.f'), '.jpg'), 'Resolution', 300);
end

% Button pushed function: STOPButton
function STOPButtonPushed(app, event)
    app.tostop = true;
    app.started = false;
    app.paused = false;
    app.PAUSEButton.Text = "|| PAUSE";
end

% Button pushed function: PAUSEButton
function PAUSEButtonPushed(app, event)
    if ~app.started
        return
    end

    if app.paused
        app.paused = false;
        app.PAUSEButton.Text = "|| PAUSE";
    else
        app.paused = true;
        app.PAUSEButton.Text = "|| RESUME";
    end
end
end
end
```

Продолжение Приложения Б

```
% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create BacterialGrowthUIFigure and hide until all components are created
    app.BacterialGrowthUIFigure = uifigure('Visible', 'off');
    colormap(app.BacterialGrowthUIFigure, 'jet');
    app.BacterialGrowthUIFigure.Position = [100 100 1005 779];
    app.BacterialGrowthUIFigure.Name = 'Bacterial Growth';

    % Create STARTButton
    app.STARTButton = uibutton(app.BacterialGrowthUIFigure, 'push');
    app.STARTButton.ButtonPushedFcn = createCallbackFcn(app,
@STARTButtonPushed, true);
    app.STARTButton.BusyAction = 'cancel';
    app.STARTButton.BackgroundColor = [0 1 0];
    app.STARTButton.FontWeight = 'bold';
    app.STARTButton.Position = [68 57 86 60];
    app.STARTButton.Text = '▶ START';

    % Create GraphsPanel
    app.GraphsPanel = uipanel(app.BacterialGrowthUIFigure);
    app.GraphsPanel.Title = 'Graphs';
    app.GraphsPanel.FontWeight = 'bold';
    app.GraphsPanel.FontSize = 18;
    app.GraphsPanel.Position = [640 32 342 696];

    % Create UIAxes_b
    app.UIAxes_b = uiaxes(app.GraphsPanel);
    title(app.UIAxes_b, 'Bacteria')
    xlabel(app.UIAxes_b, 'X')
    ylabel(app.UIAxes_b, 'Y')
    zlabel(app.UIAxes_b, 'Z')
    app.UIAxes_b.PlotBoxAspectRatio = [1.34562211981567 1.34562211981567 1];
    app.UIAxes_b.XLim = [-200 200];
    app.UIAxes_b.YLim = [-200 200];
    app.UIAxes_b.FontSize = 12;
    colormap(app.UIAxes_b, 'jet')
    app.UIAxes_b.Position = [21 355 300 300];

    % Create UIAxes_n
    app.UIAxes_n = uiaxes(app.GraphsPanel);
    title(app.UIAxes_n, 'Nutrition')
    xlabel(app.UIAxes_n, 'X')
    ylabel(app.UIAxes_n, 'Y')
    zlabel(app.UIAxes_n, 'Z')
    app.UIAxes_n.PlotBoxAspectRatio = [1 1 1];
    app.UIAxes_n.XLim = [-200 200];
    app.UIAxes_n.YLim = [-200 200];
    app.UIAxes_n.FontSize = 12;
    colormap(app.UIAxes_n, 'jet')
    app.UIAxes_n.Position = [21 44 300 300];

    % Create CurrenttimestepLabel
    app.CurrenttimestepLabel = uilabel(app.GraphsPanel);
    app.CurrenttimestepLabel.FontSize = 18;
    app.CurrenttimestepLabel.Position = [103 3 182 23];
```

Продолжение Приложения Б

```
app.CurrenttimestepLabel.Text = 'Current time = 0';

% Create ParametersPanel
app.ParametersPanel = uipanel(app.BacterialGrowthUIFigure);
app.ParametersPanel.Title = 'Parameters';
app.ParametersPanel.FontWeight = 'bold';
app.ParametersPanel.FontSize = 18;
app.ParametersPanel.Position = [28 412 591 316];

% Create TabGroup
app.TabGroup = uitabgroup(app.ParametersPanel);
app.TabGroup.Position = [0 0 591 287];

% Create BasicTab
app.BasicTab = uitab(app.TabGroup);
app.BasicTab.Title = 'Basic';

% Create n_0Label
app.n_0Label = uilabel(app.BasicTab);
app.n_0Label.Interpreter = 'latex';
app.n_0Label.HorizontalAlignment = 'right';
app.n_0Label.FontSize = 24;
app.n_0Label.FontWeight = 'bold';
app.n_0Label.Position = [60 195 33 36];
app.n_0Label.Text = '$n_0$';

% Create n0_Slider
app.n0_Slider = uislider(app.BasicTab);
app.n0_Slider.Limits = [0.1 2];
app.n0_Slider.MajorTicks = [0.1 0.5 1 1.5 2];
app.n0_Slider.ValueChangingFcn = createCallbackFcn(app,
@n0_SliderValueChanging, true);
app.n0_Slider.MinorTicks = [0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55
0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1 1.05 1.1 1.15 1.2 1.25 1.3 1.35 1.4 1.45 1.5 1.55
1.6 1.65 1.7 1.75 1.8 1.85 1.9 1.95 2];
app.n0_Slider.FontSize = 16;
app.n0_Slider.FontWeight = 'bold';
app.n0_Slider.Position = [114 218 300 3];
app.n0_Slider.Value = 0.5;

% Create sigma_0Label
app.sigma_0Label = uilabel(app.BasicTab);
app.sigma_0Label.Interpreter = 'latex';
app.sigma_0Label.HorizontalAlignment = 'right';
app.sigma_0Label.FontSize = 24;
app.sigma_0Label.FontWeight = 'bold';
app.sigma_0Label.Position = [60 130 33 36];
app.sigma_0Label.Text = '$\sigma_0$';

% Create sigma0_Slider
app.sigma0_Slider = uislider(app.BasicTab);
app.sigma0_Slider.Limits = [0.1 4];
app.sigma0_Slider.MajorTicks = [0.1 1 2 3 4];
app.sigma0_Slider.ValueChangingFcn = createCallbackFcn(app,
@sigma0_SliderValueChanging, true);
app.sigma0_Slider.MinorTicks = [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1
1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3 3.1 3.2 3.3 3.4
3.5 3.6 3.7 3.8 3.9 4];
app.sigma0_Slider.FontSize = 16;
```

Продолжение Приложения Б

```
app.sigma0_Slider.FontWeight = 'bold';
app.sigma0_Slider.Position = [115 153 300 3];
app.sigma0_Slider.Value = 1;

% Create rhoLabel
app.rhoLabel = uilabel(app.BasicTab);
app.rhoLabel.Interpreter = 'latex';
app.rhoLabel.HorizontalAlignment = 'right';
app.rhoLabel.FontSize = 24;
app.rhoLabel.FontWeight = 'bold';
app.rhoLabel.Position = [60 67 25 32];
app.rhoLabel.Text = '$\rho$';

% Create rho_Slider
app.rho_Slider = uislider(app.BasicTab);
app.rho_Slider.Limits = [0 1];
app.rho_Slider.MajorTicks = [0 0.5 1];
app.rho_Slider.ValueChangingFcn = createCallbackFcn(app,
@rho_SliderValueChanging, true);
app.rho_Slider.MinorTicks = [0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5
0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1];
app.rho_Slider.FontSize = 16;
app.rho_Slider.FontWeight = 'bold';
app.rho_Slider.Position = [106 86 300 3];
app.rho_Slider.Value = 1;

% Create n0_Label
app.n0_Label = uilabel(app.BasicTab);
app.n0_Label.FontSize = 24;
app.n0_Label.Position = [439 182 58 55];
app.n0_Label.Text = '0.5';

% Create sigma0_Label
app.sigma0_Label = uilabel(app.BasicTab);
app.sigma0_Label.FontSize = 24;
app.sigma0_Label.Position = [439 117 58 55];
app.sigma0_Label.Text = '1';

% Create rho_Label
app.rho_Label = uilabel(app.BasicTab);
app.rho_Label.FontSize = 24;
app.rho_Label.Position = [439 47 58 55];
app.rho_Label.Text = '1';

% Create AdvancedTab
app.AdvancedTab = uitab(app.TabGroup);
app.AdvancedTab.Title = 'Advanced';

% Create MeshButtonGroup
app.MeshButtonGroup = uibuttongroup(app.AdvancedTab);
app.MeshButtonGroup.Title = 'Mesh';
app.MeshButtonGroup.FontWeight = 'bold';
app.MeshButtonGroup.FontSize = 16;
app.MeshButtonGroup.Position = [93 35 387 78];

% Create RoughButton
app.RoughButton = uiradiobutton(app.MeshButtonGroup);
app.RoughButton.Text = 'Rough (fast)';
app.RoughButton.FontSize = 16;
```

Продолжение Приложения Б

```
app.RoughButton.Position = [29 18 110 22];

% Create ModerateButton
app.ModerateButton = uiradiobutton(app.MeshButtonGroup);
app.ModerateButton.Text = 'Moderate';
app.ModerateButton.FontSize = 16;
app.ModerateButton.Position = [155 18 90 22];
app.ModerateButton.Value = true;

% Create FineButton
app.FineButton = uiradiobutton(app.MeshButtonGroup);
app.FineButton.Text = 'Fine (slow)';
app.FineButton.FontSize = 16;
app.FineButton.Position = [260 18 100 22];

% Create beta_0Label
app.beta_0Label = uilabel(app.AdvancedTab);
app.beta_0Label.Interpreter = 'latex';
app.beta_0Label.HorizontalAlignment = 'right';
app.beta_0Label.FontSize = 18;
app.beta_0Label.Position = [369 190 29 26];
app.beta_0Label.Text = '$\beta_m$';

% Create beta_mEditField
app.beta_mEditField = uieditfield(app.AdvancedTab, 'numeric');
app.beta_mEditField.FontSize = 18;
app.beta_mEditField.Position = [413 192 79 24];
app.beta_mEditField.Value = 0.71;

% Create C2Label
app.C2Label = uilabel(app.AdvancedTab);
app.C2Label.Interpreter = 'latex';
app.C2Label.HorizontalAlignment = 'right';
app.C2Label.FontSize = 18;
app.C2Label.Position = [379 135 31 27];
app.C2Label.Text = '$C^2$';

% Create C2EditField
app.C2EditField = uieditfield(app.AdvancedTab, 'numeric');
app.C2EditField.FontSize = 18;
app.C2EditField.Position = [425 138 67 24];
app.C2EditField.Value = 6.25;

% Create LLabel
app.LLabel = uilabel(app.AdvancedTab);
app.LLabel.Interpreter = 'latex';
app.LLabel.HorizontalAlignment = 'right';
app.LLabel.FontSize = 18;
app.LLabel.Position = [58 191 25 24];
app.LLabel.Text = '$L$';

% Create LEditField
app.LEditField = uieditfield(app.AdvancedTab, 'numeric');
app.LEditField.ValueChangedFcn = createCallbackFcn(app,
@LEditFieldValueChanged, true);
app.LEditField.FontSize = 18;
app.LEditField.Position = [98 191 88 24];
app.LEditField.Value = 200;
```

Продолжение Приложения Б

```
% Create TLabel
app.TLabel = uilabel(app.AdvancedTab);
app.TLabel.Interpreter = 'latex';
app.TLabel.HorizontalAlignment = 'right';
app.TLabel.FontSize = 18;
app.TLabel.Position = [64 136 25 24];
app.TLabel.Text = '$T$';

% Create TEditField
app.TEditField = uieditfield(app.AdvancedTab, 'numeric');
app.TEditField.FontSize = 18;
app.TEditField.Position = [104 136 84 24];
app.TEditField.Value = 2000;

% Create LxyLLabel
app.LxyLLabel = uilabel(app.AdvancedTab);
app.LxyLLabel.Interpreter = 'latex';
app.LxyLLabel.FontSize = 14;
app.LxyLLabel.Position = [209 188 127 31];
app.LxyLLabel.Text = '$-L < x,y < L$';

% Create tleqTLabel
app.tleqTLabel = uilabel(app.AdvancedTab);
app.tleqTLabel.Interpreter = 'latex';
app.tleqTLabel.FontSize = 14;
app.tleqTLabel.Position = [212 131 127 31];
app.tleqTLabel.Text = '$0 < t \leq T$';

% Create IvanShevkunAnnaMaslovskaya2024Label
app.IvanShevkunAnnaMaslovskaya2024Label = uilabel(app.BacterialGrowthUIFigure);
app.IvanShevkunAnnaMaslovskaya2024Label.Position = [17 1 707 50];
app.IvanShevkunAnnaMaslovskaya2024Label.Text = {'@ Ivan Shevkun, Anna Maslovskaya, Amur State University, 2024'; 'The study is supported by the Ministry of Science and Higher Education of the Russian Federation (project no. 122082400001-8) '};

% Create ModelPanel
app.ModelPanel = uipanel(app.BacterialGrowthUIFigure);
app.ModelPanel.Title = 'Model*';
app.ModelPanel.FontWeight = 'bold';
app.ModelPanel.FontSize = 18;
app.ModelPanel.Position = [30 138 589 250];

% Create textitKawasakiKetalJournalofTheoreticalBiology1997No188Label
app.textitKawasakiKetalJournalofTheoreticalBiology1997No188Label = uilabel(app.ModelPanel);

app.textitKawasakiKetalJournalofTheoreticalBiology1997No188Label.Interpreter = 'latex';
app.textitKawasakiKetalJournalofTheoreticalBiology1997No188Label.FontColor = [0.502 0.502 0.502];
app.textitKawasakiKetalJournalofTheoreticalBiology1997No188Label.Position = [5 1 409 22];
app.textitKawasakiKetalJournalofTheoreticalBiology1997No188Label.Text = '\textit{*Kawasaki K., et al., Journal of Theoretical Biology. 1997. No 188}';

% Create EquationsPanel
app.EquationsPanel = uipanel(app.ModelPanel);
app.EquationsPanel.Title = 'Equations';
app.EquationsPanel.FontWeight = 'bold';
```

Продолжение Приложения Б

```

app.EquationsPanel.Position = [17 32 214 180];

% Create fracpartialnpartialtDeltanbnLabel
app.fracpartialnpartialtDeltanbnLabel = uilabel(app.EquationsPanel);
app.fracpartialnpartialtDeltanbnLabel.Interpreter = 'latex';
app.fracpartialnpartialtDeltanbnLabel.FontSize = 18;
app.fracpartialnpartialtDeltanbnLabel.Position = [6 58 194 94];
app.fracpartialnpartialtDeltanbnLabel.Text = {'$\frac{\partial n}{\partial t} = \Delta n - bn,'; '\frac{\partial b}{\partial t} = \nabla(\sigma n b \nabla b) + nb$'};

% Create fracpartialnpartialtDeltanbnLabel_6
app.fracpartialnpartialtDeltanbnLabel_6 = uilabel(app.EquationsPanel);
app.fracpartialnpartialtDeltanbnLabel_6.Interpreter = 'latex';
app.fracpartialnpartialtDeltanbnLabel_6.FontSize = 14;
app.fracpartialnpartialtDeltanbnLabel_6.Position = [6 -1 227 60];
app.fracpartialnpartialtDeltanbnLabel_6.Text = {'$\sigma = \sigma_0(1+\delta),'; '\delta \in [-\rho, \rho]$ - \text{random variable}'};

% Create InitialconditionsPanel
app.InitialconditionsPanel = uipanel(app.ModelPanel);
app.InitialconditionsPanel.Title = 'Initial conditions';
app.InitialconditionsPanel.FontWeight = 'bold';
app.InitialconditionsPanel.Position = [241 32 203 180];

% Create fracpartialnpartialtDeltanbnLabel_3
app.fracpartialnpartialtDeltanbnLabel_3 = uilabel(app.InitialconditionsPanel);
app.fracpartialnpartialtDeltanbnLabel_3.Interpreter = 'latex';
app.fracpartialnpartialtDeltanbnLabel_3.FontSize = 18;
app.fracpartialnpartialtDeltanbnLabel_3.Position = [12 49 187 53];
app.fracpartialnpartialtDeltanbnLabel_3.Text = '$b(x, y, 0) = \beta_{me}^{\frac{x^2+y^2}{C^2}}$';

% Create fracpartialnpartialtDeltanbnLabel_2
app.fracpartialnpartialtDeltanbnLabel_2 = uilabel(app.InitialconditionsPanel);
app.fracpartialnpartialtDeltanbnLabel_2.Interpreter = 'latex';
app.fracpartialnpartialtDeltanbnLabel_2.FontSize = 18;
app.fracpartialnpartialtDeltanbnLabel_2.Position = [12 122 194 26];
app.fracpartialnpartialtDeltanbnLabel_2.Text = '$n(x, y, 0) = n_{\theta} = const,$';

% Create BorderconditionsPanel
app.BorderconditionsPanel = uipanel(app.ModelPanel);
app.BorderconditionsPanel.Title = 'Border conditions';
app.BorderconditionsPanel.FontWeight = 'bold';
app.BorderconditionsPanel.Position = [452 32 123 180];

% Create fracpartialnpartialtDeltanbnLabel_4
app.fracpartialnpartialtDeltanbnLabel_4 = uilabel(app.BorderconditionsPanel);
app.fracpartialnpartialtDeltanbnLabel_4.Interpreter = 'latex';
app.fracpartialnpartialtDeltanbnLabel_4.FontSize = 18;
app.fracpartialnpartialtDeltanbnLabel_4.Position = [22 93 87 54];
app.fracpartialnpartialtDeltanbnLabel_4.Text = '$\left. \frac{\partial n}{\partial t} \right|_{\Gamma} = 0,$';

% Create fracpartialnpartialtDeltanbnLabel_5

```

Продолжение Приложения Б

```

    app.fracpartialnpartialtDeltanbnLabel_5 = ui-
label(app.BorderconditionsPanel);
    app.fracpartialnpartialtDeltanbnLabel_5.Interpreter = 'latex';
    app.fracpartialnpartialtDeltanbnLabel_5.FontSize = 18;
    app.fracpartialnpartialtDeltanbnLabel_5.Position = [22 27 82 54];
    app.fracpartialnpartialtDeltanbnLabel_5.Text = '$\left.\frac{\partial
b}{\partial \text{bf}{n}}\right|_{\Gamma} = 0$';

    % Create SAVESNAPSHOTButton
    app.SAVESNAPSHOTButton = uibutton(app.BacterialGrowthUIFigure, 'push');
    app.SAVESNAPSHOTButton.ButtonPushedFcn = createCallbackFcn(app,
@SAVESNAPSHOTButtonPushed, true);
    app.SAVESNAPSHOTButton.BackgroundColor = [0.302 0.7451 0.9333];
    app.SAVESNAPSHOTButton.FontWeight = 'bold';
    app.SAVESNAPSHOTButton.Position = [489 57 89 60];
    app.SAVESNAPSHOTButton.Text = {'SAVE'; 'SNAPSHOT'};

    % Create SimulationofbacterialgrowthLabel
    app.SimulationofbacterialgrowthLabel = ui-
label(app.BacterialGrowthUIFigure);
    app.SimulationofbacterialgrowthLabel.HorizontalAlignment = 'center';
    app.SimulationofbacterialgrowthLabel.VerticalAlignment = 'top';
    app.SimulationofbacterialgrowthLabel.FontSize = 24;
    app.SimulationofbacterialgrowthLabel.Position = [367 737 333 33];
    app.SimulationofbacterialgrowthLabel.Text = 'Simulation of bacterial
growth';

    % Create PAUSEButton
    app.PAUSEButton = uibutton(app.BacterialGrowthUIFigure, 'push');
    app.PAUSEButton.ButtonPushedFcn = createCallbackFcn(app,
@PAUSEButtonPushed, true);
    app.PAUSEButton.BackgroundColor = [1 1 0];
    app.PAUSEButton.FontWeight = 'bold';
    app.PAUSEButton.Position = [181 57 86 60];
    app.PAUSEButton.Text = '|| PAUSE';

    % Create STOPButton
    app.STOPButton = uibutton(app.BacterialGrowthUIFigure, 'push');
    app.STOPButton.ButtonPushedFcn = createCallbackFcn(app, @STOPButtonPushed,
true);
    app.STOPButton.BackgroundColor = [1 0 0];
    app.STOPButton.FontWeight = 'bold';
    app.STOPButton.Position = [295 57 86 60];
    app.STOPButton.Text = '■ STOP';

    % Create ErrorsLabel
    app.ErrorsLabel = uilabel(app.BacterialGrowthUIFigure);
    app.ErrorsLabel.VerticalAlignment = 'top';
    app.ErrorsLabel.Position = [20 736 337 34];
    app.ErrorsLabel.Text = '';

    % Show the figure after all components are created
    app.BacterialGrowthUIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

```

Продолжение Приложения Б

```
% Construct app
function app = BacterialGrowth

    % Create UIFigure and components
    createComponents(app)

    % Register the app with App Designer
    registerApp(app, app.BacterialGrowthUIFigure)

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.BacterialGrowthUIFigure)
end
end
end
```

ПРИЛОЖЕНИЕ В

Свидетельство о государственной регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024662124

Система моделирования и оценки геометрических характеристик дендритных структур при культивации бактерий

Правообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего образования "Амурский государственный университет" (RU)*

Авторы: *Шевкун Иван Александрович (RU), Масловская Анна Геннадьевна (RU)*



Заявка № 2024660621

Дата поступления 07 мая 2024 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 23 мая 2024 г.

*Руководитель Федеральной службы
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 429b6a0fe3853164ba196f83b73b4aa7
Владелец: *Зубов Юрий Сергеевич*
Действителен с 18.05.2023 по 02.08.2024

Ю.С. Зубов