

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Северо-Кавказский федеральный университет»**

На правах рукописи

Ярцева Елена Павловна

**РАЗРАБОТКА И ЧИСЛЕННЫЕ ИССЛЕДОВАНИЯ
РЕКУРСИВНО-ИТЕРАЦИОННЫХ МЕТОДОВ И АЛГОРИТМОВ В ЗАДАЧЕ
МОДЕЛИРОВАНИЯ ПЕРЕНОСА ПРИМЕСЕЙ В АТМОСФЕРЕ**

Специальность 05.13.18 – Математическое моделирование,
численные методы и комплексы программ

ДИССЕРТАЦИЯ

на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
доктор физико – математических
наук, доцент
Наац Виктория Игоревна

Ставрополь 2015

Содержание

Введение	5
Глава 1. Численные методы и математические модели в проблеме охраны окружающей среды в системе атмосфера - подстилающая поверхность	17
1.1 Общая характеристика проблемы охраны окружающей среды в системе атмосфера - подстилающая поверхность	17
1.2 Физические основы математической модели пограничного слоя атмосферы	20
1.3 Обзор гидродинамических моделей пограничного слоя атмосферы	25
1.4 Постановка задачи моделирования нестационарного переноса примесей в пограничном слое атмосферы и определение исходных данных на основе сопутствующих математических моделей	32
1.5 Численные методы и алгоритмы в моделях нестационарного переноса загрязняющих примесей в приземном слое атмосферы	45
Выводы по главе 1	58
Глава 2. Численная реализация итерационных алгоритмов и исследование моделей переноса загрязняющих веществ в атмосфере. Методы качественной интерпретации расчетных данных	59
2.1 Разработка методик исследования физических явлений на моделях. Выбор исходных данных	60
2.2 Анализ данных моделирования пространственно-временной изменчивости поля концентрации загрязнений на основе предложенных численных методов	65
2.3 Методика планирования вычислительных экспериментов и интерпретация их результатов. Оценка устойчивости алгоритмов и влияния ошибок вычислений	69
2.4 Качественные подходы к исследованию динамики полей концентрации аэрозольных примесей на основе фундаментальной системы решения уравнений параболического типа	75
2.5 Алгоритмизация, программирование и результаты вычислений для качественной модели оценки концентрации аэрозольных примесей на основе	

фундаментальной системы решения уравнений параболического типа	78
2.6 Разработка методики качественной оценки концентрации аэрозольных примесей в рамках итерационных алгоритмов	82
2.7 Анализ эффективности предлагаемых методик качественной оценки значений поля концентрации аэрозольных примесей на основе итерационных методов численного решения уравнения переноса.....	85
2.8 Вопросы сходимости рекурсивно-итерационных методов и алгоритмов в задаче моделирования переноса примесей в атмосфере	91
Выводы по главе 2	94
Глава 3. Разработка решающих алгоритмов для уравнений переноса в рамках итерационно-рекурсивного подхода. Вычислительные эксперименты и их анализ	96
3.1 Тестовые примеры для трехмерных задач теории переноса. Выбор и обоснование исходных данных	96
3.2 Построение рекурсивных алгоритмов для уравнения переноса в рамках конечно-разностного подхода	102
3.3 Структура решающего алгоритма для трехмерной задачи переноса в рамках рекурсивно-итерационного подхода	106
3.4 Результаты численных экспериментов и их анализ на основе созданного программного обеспечения для задач моделирования нестационарного переноса примесей в атмосфере	112
Выводы по главе 3	126
Глава 4. Разработка системы компьютерного моделирования пространственно-временной изменчивости поля концентрации загрязнений в атмосфере на основе предложенных численных методов и программного обеспечения	127
4.1 Разработка информационно-логической схемы системы компьютерного моделирования пространственно-временной изменчивости поля концентрации загрязнений в атмосфере	127
4.2 Разработка структуры системы компьютерного моделирования на основе модульного подхода	134
4.3 Программные модули алгоритмической системы компьютерного модели-	

рования	138
Выводы по главе 4	149
Заключение	150
Литература	151
Приложение 1. Свидетельства о регистрации программ для ЭВМ	161
Приложение 2. Таблицы значений метеорологических параметров, взятых из научных публикаций	163
Приложение 3. Листинг программ	166

Введение

Актуальность темы исследования и степень разработанности. Прогноз изменения экологических систем, подвергающихся воздействию антропогенных факторов, является важной и весьма актуальной задачей науки в настоящее время [16 – 18]. Свойства многомерности, неполной предсказуемости поведения, обусловленной стохастичностью происходящих в них процессов, существенно затрудняют проведение натуральных экспериментов с такими системами. Поэтому важную роль в их исследовании играют их математическое моделирование, проведение численных экспериментов и комплексный мониторинг – сложная система разномасштабных наблюдений различных характеристик среды с одновременным анализом полученных данных и прогнозом последующего изменения состояния среды. Накопленный в мире опыт решения научных и практических задач природоохранного направления [5, 9, 16 – 18, 32, 57] показывает, что математические модели и данные натуральных исследований являются равноправными и дополняющими друг друга при изучении природных процессов. При этом активно используются методы дистанционного зондирования в сочетании с различными методами контактных измерений [47 – 49]. В результате сбора данных к исследователям попадает разнородная информация, с разных сторон характеризующая наблюдаемые явления. В этом случае возникает задача совместного использования этой информации и математических моделей с целью восприятия исходных данных.

Для решения подобных задач необходимо разрабатывать математические модели исследуемых процессов, соответствующие эффективные алгоритмы на основе численных методов, позволяющие воспринимать данные мониторинга, измеренные в ходе натуральных экспериментов с погрешностями. Кроме того имеет место проблема, связанная с невозможностью непосредственных натуральных измерений некоторых характеристик природной среды, например атмосферной турбулентности. Также имеет место проблема оценки значений производных поля скорости ветра, поля турбулентности, концентрации загрязняющих примесей, начальных и краевых условий в уравнении массопереноса. Все это в целом составляет проблему неопределенности или недостаточности исходных данных, для решения которой требуется привлечение некоторой дополнительной информации или полуэмпирических качественных моделей. Актуальной задачей также является создание соответствующего информационного обеспечения на осно-

ве современных достижений в области вычислительной техники и информационных технологий, а именно автоматизированных систем моделирования соответствующих прикладных задач. При этом средства математического, вычислительного и программного обеспечения в совокупности будут представлять собой информационно-вычислительную технологию моделирования атмосферных процессов, в частности, процесса нестационарного диффузного переноса загрязняющих примесей в атмосфере.

Научные направления, связанные с математическим моделированием и разработкой соответствующего информационно-вычислительного обеспечения в задачах охраны окружающей среды, давно развиваются в нашей стране и за рубежом [24, 28 – 31, 53 – 56, 70, 88 – 92] и являются, несмотря на все выдающиеся достижения в этой области, актуальными и современными и в настоящее время. Основоположителем данного направления по праву считают академика Марчука Г.И.. Значительные достижения в этой области также принадлежат ученым НИИ Прикладной математики РАН, академику Дымникову В.П., профессорам Алояну А.Е., Пененко В.В. и др., которые были получены в рамках научных направлений «Вычислительная математика», «Модели и методы в задачах физики атмосферы и океана», «Проблемы охраны окружающей среды». В последнее время значительные научные результаты в области математического моделирования задач экологического мониторинга были получены учеными таких научных школ как, научная школа профессора Угольницкого Г.А. (Южный федеральный университет) [12, 68, 69], научная школа профессора Сухинова А.И. (Южный федеральный университет) [65 – 67], научная школа профессора Наац И.Э. (Северо-Кавказский федеральный университет) [48 – 51], научная школа профессора Семенчина Е.А. (Кубанский государственный университет) [62 - 64]. Тем не менее, несмотря на имеющиеся достижения, не все научные проблемы до конца еще решены.

Разработка вычислительных моделей для уравнения диффузного переноса загрязняющих примесей в турбулентной атмосфере является достаточно сложной задачей из-за его многомерности, нестационарности и неопределенности в исходных данных. Как уже отмечалось выше, существует много разработок в данной области. Однако следует отметить, что в основном предлагаемые вычислительные модели строятся на основе различного вида конечно-разностных методов. Подобные алгоритмы не эффективны, поскольку они не устойчивы к погрешностям в исходных данных, а также предполагают значительную размерность задачи. Требуется построение более экономичных и устой-

чивых вычислительных схем, способных воспринимать эмпирические данные. Кроме того, для решения проблемы неопределенности исходных данных, актуальной задачей является разработка различного вида качественных моделей и соответствующего программно-алгоритмического сопровождения.

Помимо проблемы математического моделирования, самостоятельной можно считать проблему информационного обеспечения задач экологического мониторинга на основе алгоритмического, программного обеспечения, базы данных и соответствующих технических средств. Такая система может стать неким ядром информационной системы экологического мониторинга и прогноза атмосферы в пределах ее пограничного слоя, работающей на станциях экологического контроля. Подобные информационные системы ранее разрабатывались как в России, так и за рубежом. Известными являются такие работы в данной области, как работа [53], в которой описана методика организации сценариев моделирования на основе данных Reanalysis NCEP/NCAR [118, 119], включающая в себя задачи, связанные с оценками экологической перспективы при различных вариантах антропогенных воздействий и масштабов взаимодействий в климатической системе типа источник-детектор, детектор-источник. Создаваемая для этих целей система моделирования [56] позволяет оперативно восстанавливать пространственно-временную структуру атмосферной циркуляции с заданным разрешением в режиме восприятия данных реанализа. Данная система моделирования представляет собой многофункциональный комплекс моделей и является открытой и развиваемой. Существуют способы повышения эффективности алгоритмов системы моделирования, одним из которых служит метод распараллеливания. Методика организации параллельных вычислений развивается в рамках нового фундаментального научного направления, связанного с совместным использованием численных методов и структур ЭВМ [27]. Оно получило название «Отображение проблем вычислительной математики на архитектуру вычислительных систем» и стало одним из ведущих направлений научных исследований в Отделе вычислительной математики АН СССР (в настоящее время Институт вычислительной математики РАН), созданном академиком Марчуком Г.И. в 1980 году. К настоящему времени известен цикл работ, содержащих последние достижения в этой области [10, 11, 27, 71, 93, 95]. В частности, в статье Воеводина В.В. [11] рассматриваются основные положения обозначенного выше фундаментального научного направления, анализируется связь этого направления с различными областями, так или иначе связанными

с вычислениями. Что касается вычислительных моделей, построенных на основе методов прямого и обратного моделирования [53], позволяющих использование в них измерительной информации, то их структура построена на принципах аддитивности. Выбранный в них способ дискретизации с помощью вариационного принципа и метода расщепления обеспечивает конструирование численных моделей для основных и сопряженных задач в виде схем расщепления, взаимно согласованных на всех этапах вычислений. Как следствие этого возможна многоуровневая схема распараллеливания алгоритмов. В итоге каждый этап технологии моделирования может реализовываться параллельно. При этом на нижнем системном уровне покоординатное расщепление по пространственным переменным также может быть выстроено в параллельную структуру. Обозначенные выше проблемы в той или иной мере решаются в данной диссертации, что обуславливает актуальность ее темы.

Объектом исследования является пространственно-временное распределение поля концентрации загрязняющих примесей, распространяющихся в пограничном слое атмосферы и соответствующее уравнение диффузного переноса, описывающее данный процесс.

Предметом исследования являются методы моделирования диффузного переноса примесей в пограничном слое атмосферы, соответствующие численные методы и модели, программно-алгоритмическое обеспечение.

Целью диссертационного исследования является разработка и исследование численных методов и алгоритмов в задаче моделирования диффузного переноса загрязняющих примесей в пограничном слое атмосферы, способных воспринимать приближенные данные экологического мониторинга, и создание системы компьютерного моделирования применительно к проблеме экологического прогноза состояния пограничного слоя атмосферы.

Методы исследования: теория и методы математического моделирования, численные методы, применяемые для решения линейных многомерных дифференциальных уравнений в частных производных и интегральных уравнений, методы теории аппроксимации, алгоритмов и программирования.

Задачи диссертационного исследования:
в области математического моделирования:

1. в рамках проблемы обеспечения математических моделей исходными данными для одномерного параметризованного уравнения переноса загрязняющих примесей в атмосфере предложить и построить первую качественную расчетно-аналитическую модель оценки концентрации примесей на основе фундаментальной системы решений однородного дифференциального уравнения второго порядка с постоянными коэффициентами; разработать вторую качественную модель расчета концентрации примесей на основе представления искомого решения интегральным уравнением Вольтерра второго рода и численного метода последовательных приближений;

в области численных методов и методов алгоритмизации:

2. для одномерного параметризованного уравнения переноса примесей построить итерационно-рекурсивные алгоритмы для первой и второй качественных моделей соответственно; на основе конечно-разностного метода и численных методов решения СЛАУ разработать алгоритмы тестовых примеров для одномерного и трехмерного уравнений переноса соответственно, а также алгоритмы численного исследования вычислительных схем на сходимость и устойчивость к погрешностям в исходных данных;

3. на основе численного метода покоординатного расщепления трехмерного параметризованного уравнения переноса и итерационно-рекурсивных методов и алгоритмов, построенных первоначально для одномерной задачи, выполнить построение вычислительных алгоритмов для решения пространственной задачи переноса;

в области разработки программного обеспечения, постановки и проведения вычислительного эксперимента:

4. на основе методики тестирования алгоритмов и программ выполнить постановку и провести вычислительный эксперимент по исследованию свойств рекурсивно-итерационных алгоритмов решения одномерной задачи переноса, таких как сходимость и устойчивость к погрешностям в исходных данных; по исследованию влияния различных значений скорости ветра, турбулентности и других характеристик состояния пограничного слоя атмосферы на пространственно-временное распределение поля концентрации загрязняющих примесей на примере решения пространственной задачи переноса;

5. предложить концептуальную схему информационно-вычислительного обеспечения задач переноса загрязняющих примесей в атмосфере и на ее основе разработать си-

стему компьютерного моделирования пространственно-временной изменчивости поля концентрации.

Научная новизна.

1. В рамках проблемы обеспечения математических моделей исходными данными для одномерного параметризованного уравнения переноса загрязняющих примесей в атмосфере предложены две качественные расчетно-аналитические модели, первая из которых строится на основе фундаментальной системы решений однородного дифференциального уравнения второго порядка с постоянными коэффициентами, а вторая – на основе представления искомого решения уравнения переноса интегральным уравнением Вольтерра второго рода и численного метода последовательных приближений, что позволяет выполнять качественную оценку значений параметров в задачах математического моделирования явления переноса примесей в системе атмосфера - подстилающая поверхность, решать проблему неопределенности исходных данных, получать простые аналитические решения и строить соответствующие алгоритмы исходной задачи, использовать полученные результаты моделирования в других более сложных вычислительных схемах, как некие предварительные исходные данные; разработаны соответствующие итерационные и рекурсивные вычислительные алгоритмы, устойчивые к погрешностям в исходных данных;
2. Для трехмерного параметризованного уравнения переноса примесей в атмосфере с учетом «уравнения неразрывности» построены вычислительные алгоритмы на основе численного метода покоординатного расщепления, в структуру которых включены итерационные и рекурсивные алгоритмы решения соответствующих одномерных задач;
3. На основе разработанного алгоритмического и программного обеспечения проведен вычислительный эксперимент, включающий в себя тестирование алгоритмов и программ с помощью специально созданных тестовых примеров, а также исследование влияния скорости ветра, турбулентности и других состояний пограничного слоя атмосферы на пространственно-временное распределение поля концентрации загрязняющих примесей, получены соответствующие результаты расчетов;
4. Предложена концепция информационно-вычислительного обеспечения задач экологического прогноза состояния воздушной среды, которая может быть включена, например, в состав системы «автоматизированное место специалиста-эколога» на станции экологического контроля, в соответствии с которой разработана система компьютерного

моделирования пространственно-временной изменчивости поля концентрации загрязняющих примесей.

Степень достоверности научных положений и выводов обусловлена корректной математической постановкой рассматриваемых задач, использованием известных теоретических положений и методов теории «Уравнения математической физики», «Численные методы» и др. Представленные в диссертации результаты имеют математическое обоснование, проведены численные исследования сходимости и устойчивости вычислительных алгоритмов, сопоставление приближенных решений с точными решениями, моделируемыми с помощью специально разработанных тестовых задач. Результаты численных экспериментов, полученные на основе разработанного комплекса программ, согласуются с известными по литературным источникам данными натурных экспериментов.

Научная и практическая значимость работы состоит в возможности дальнейшего использования созданного в ней математического, вычислительного, алгоритмического и программного обеспечения. Некоторые научные результаты диссертации использовались СРОО «Экологический Конгресс Ставрополя. Академия Экологической Безопасности» при разработке информационных систем мониторинга и прогноза экологического состояния атмосферы региона (Акт о внедрении от 26.09.2014 г.). Также результаты диссертационного исследования успешно апробированы в учебном процессе кафедры математического анализа Северо – Кавказского федерального университета (Акт о внедрении от 03.10.2014 г.). Программный комплекс зарегистрирован в Федеральной службе по интеллектуальной собственности и доступен другим пользователям [74, 75].

Положения, выносимые на защиту:

1. Расчетно-аналитическая качественная модель для параметризованного уравнения переноса загрязняющих примесей в пограничном слое атмосферы с приближенно заданными исходными данными, основанная на использовании фундаментальных решений однородного дифференциального уравнения второго порядка с постоянными коэффициентами, позволяющая выполнять качественную оценку значений параметров соответствующей математической модели, получать простые аналитические решения и строить соответствующие алгоритмы исходной задачи (п.1,2,3 паспорта специальности);

2. Расчетно-аналитическая качественная модель для параметризованного уравнения переноса примесей в атмосфере с приближенно заданными исходными данными, построенная на основе сведения систем дифференциальных уравнений к эквивалентным системам интегральных уравнений Вольтерра второго рода с последующим введением в структуру вычислительных моделей рекурсивно-итерационных алгоритмов, что повышает их устойчивость к погрешностям в исходных данных и эффективность (п.1,2,3 паспорта специальности);

3. Вычислительная модель для пространственной задачи переноса, основанная на численном методе покоординатного расщепления трехмерного параметризованного уравнения переноса с учетом в нем «уравнения неразрывности», в структурную схему которой включены итерационно-рекурсивные алгоритмы соответствующих расчетно-аналитических качественных моделей, построенных первоначально для одномерной задачи (п.1,2,3 паспорта специальности);

4. Методика, реализованная на основе разработанных тестовых примеров и программного обеспечения, и результаты вычислительного эксперимента по исследованию сходимости и устойчивости алгоритмов к погрешностям в исходных данных, а также исследованию влияния скорости ветра, турбулентности и других состояний атмосферы в пределах пограничного слоя на пространственно-временное распределение поля концентрации загрязняющих примесей (п.3,4,7 паспорта специальности);

5. Система компьютерного моделирования пространственно-временной изменчивости поля концентрации загрязнений в атмосфере (п.7 паспорта специальности).

Апробация работы и публикации. Результаты, полученные в рамках диссертационной работы, докладывались и обсуждались на конференциях и научных семинарах: Международной научно-практической конференции «Современные достижения в науке и образовании: математика и информатика» (г. Архангельск, 2010); XVII Международной конференции студентов, аспирантов и молодых ученых «Ломоносов - 2010» (г. Москва, 2010); VI Всероссийской открытой научно – практической конференции «Актуальные задачи математического моделирования и информационных технологий» (г. Сочи, 2010); XVIII Международной конференции студентов, аспирантов и молодых ученых «Ломоносов - 2011» (г. Москва, 2011); Международной заочной научно – практической конференции «Актуальные научные вопросы: реальность и перспективы» (г. Тамбов, 2012); Международной научной конференции, посвященной 35-летию математиче-

ского факультета и 25-летию факультета информатики и вычислительной техники Ярославского государственного университета им П.Г. Демидова «Моделирование и анализ информационных систем» (г. Ярославль, 2012); VII Международной научно-технической конференции «Аналитические и численные методы моделирования естественнонаучных и социальных проблем» (г. Пенза, 2012); Международной научно-практической конференции «Актуальные задачи математического моделирования и информационных технологий» (г. Сочи, 2013); VII Всероссийской научно-практической конференции молодых ученых «Наука и устойчивое развитие» (г. Нальчик, 2013); ежегодной научно – практической конференции Северо – Кавказского федерального университета «Университетская наука - региону»: «Естественные науки – основа настоящего и фундамент для будущего» (г. Ставрополь, 2013); Международной научно-практической конференция «Наука и образование XXI века» (г. Уфа, 2014).

Количество публикаций по теме диссертации составляет 21, из них 4 статьи в журналах, рекомендованных ВАК, 2 свидетельства о государственной регистрации программы для ЭВМ. К основным публикациям (статьи в рецензируемых изданиях, тезисы докладов на всероссийских и международных конференциях), в которых излагаются научные результаты диссертации, относятся 16 работ. Лично диссертантом опубликовано 11 научных работ. Общее число печатных листов публикаций – 6.55, из них на долю диссертанта приходится 4.47. Диссертант имеет индекс Хирша с учетом только статей в журналах, равный 1. Число цитирований статей диссертанта, опубликованных за последние 5 лет, составляет 3.

I. Работы, опубликованные в изданиях из перечня ВАК:

1. Наац, В.И. Численное исследование рекурсивных и итерационных алгоритмов в задаче моделирования переноса аэрозолей в атмосфере / В.И. Наац, **Е.П. Ярцева** // Вестник Ставропольского государственного университета. – 2011. – Выпуск 75 [4]. – С. 44 – 50.
2. Наац, В.И. Качественная модель оценки концентрации аэрозольных примесей в атмосфере, основанная на интегральном представлении решения уравнения турбулентной диффузии / В.И. Наац, **Е.П. Ярцева** // Известия высших учебных заведений. Северо – Кавказский регион. Естественные науки. – 2012. – № 1. – С. 38 – 43.
3. Наац, В.И. Разработка качественной модели и итерационного алгоритма для оценки концентрации загрязняющих примесей в атмосфере / В.И. Наац, **Е.П. Ярцева** //

Вестник Северо – Кавказского федерального университета. – 2013. – № 1 (34). – С. 15 – 21.

4. Наац, В.И. Построение рекурсивно-итерационного алгоритма для трехмерного нестационарного уравнения переноса примесей в атмосфере / В.И. Наац, **Е.П. Ярцева** // Вестник Северо – Кавказского федерального университета. – 2014. – № 2 (41). – С. 9 – 14.

II. Свидетельства о регистрации программ:

5. **Ярцева Е.П.** Качественные модели и вычислительные алгоритмы оценки концентрации загрязняющих примесей в турбулентной атмосфере. Свидетельство о государственной регистрации программы для ЭВМ № 2013614085. Зарегистрировано в Реестре программ для ЭВМ 23 апреля 2013.

6. **Ярцева Е.П.** Программа для реализации рекурсивных и итерационных вычислительных алгоритмов для пространственной задачи переноса загрязняющих примесей в турбулентной атмосфере. Свидетельство о государственной регистрации программы для ЭВМ № 2014661580. Зарегистрировано в Реестре программ для ЭВМ 31 октября 2014.

III. Публикации в других изданиях:

7. Наац, В.И. Построение и исследование вычислительного алгоритма для параметризованного нестационарного уравнения переноса примесей в атмосфере / В.И. Наац, **Е.П. Ярцева** // Современные достижения в науке и образовании: математика и информатика: материалы международной научно-практической конференции. – Архангельск. – 2010. – С. 561 – 562.

8. **Ярцева, Е.П.** Численные исследования алгоритма решения уравнения переноса субстанции в турбулентной атмосфере / Е.П. Ярцева // Ломоносов – 2010: материалы XVII Международной конференции студентов, аспирантов и молодых ученых. – М.: МГУ. – 2010. – 1 с.

9. **Ярцева, Е.П.,** Итерационный метод численного решения уравнения переноса примесей в атмосфере / Е.П. Ярцева, В.И. Наац // Актуальные задачи математического моделирования и информационных технологий: материалы VI Всероссийской открытой научно – практической конференции. – Сочи. – 2010. – С. 178 – 180.

10. **Ярцева, Е.П.** Численные исследования алгоритма решения нестационарного уравнения переноса загрязняющих веществ в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы 55 научно-методической конференции преподавателей

и студентов Ставропольского государственного университета. – Ставрополь. – 2010. – ч. 1. – С. 112 – 116.

11. **Ярцева, Е.П.** Рекурсивная вычислительная схема решения уравнения переноса примесей в турбулентной атмосфере / Е.П. Ярцева // Ломоносов – 2011: материалы XVIII Международной конференции студентов, аспирантов и молодых ученых. – М.: МГУ. – 2011. – 1 с.

12. **Ярцева, Е.П.** Метод качественной оценки значений параметров интегральных моделей в задачах моделирования переноса субстанции в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы 56 научно-методической конференции преподавателей и студентов Ставропольского государственного университета. – Ставрополь. – 2011. – ч. 2. – С. 144 – 147.

13. **Ярцева, Е.П.** Численные исследования вычислительного алгоритма в задаче переноса примесей в турбулентной атмосфере на основе нестационарного трехмерного уравнения / Е.П. Ярцева, В.И. Наац // Актуальные научные вопросы: реальность и перспективы: материалы Международной заочной научно – практической конференции. – Тамбов. – 2012. – С. 165 – 166.

14. Наац, В.И. Метод интегральных уравнений в схеме покоординатного расщепления уравнения переноса аэрозолей в атмосфере / В.И. Наац, **Е.П. Ярцева** // Актуальные задачи математического моделирования и информационных технологий: материалы международной научно – практической конференции. – Сочи. – 2012. – С. 80 – 81.

15. Наац, В.И. Вычислительный алгоритм в модели переноса примесей в турбулентной атмосфере на основе нестационарного трехмерного уравнения / В.И. Наац, **Е.П. Ярцева** // Моделирование и анализ информационных систем: труды международной научной конференции, посвященной 35-летию математического факультета и 25-летию факультета информатики и вычислительной техники Ярославского государственного университета им П.Г. Демидова. – Ярославль, ЯрГУ. – 2012. – С. 143 – 145.

16. **Ярцева, Е.П.** Оценка концентрации загрязняющих примесей в пограничном слое атмосферы на основе качественной модели / Е.П. Ярцева, В.И. Наац // Аналитические и численные методы моделирования естественнонаучных и социальных проблем: сборник статей VII Международной научно-технической конференции. – Пенза. – 2012. – С. 54 – 57.

17. **Ярцева, Е.П.** Построение и численные исследования трехмерной вычислительной модели переноса загрязняющих примесей в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы 57 научно-методической конференции преподавателей и студентов Ставропольского государственного университета. – Ставрополь. – 2012. – ч. 1. – С. 243 – 246.
18. **Yartseva, E.P.** Building a Computational Algorithm to Model Pollutant in the Turbulent Atmosphere on the Basis of Unsteady Three-Dimensional Equation / E.P. Yartseva // International Multidisciplinary Journal European Researcher. – 2013. – № 5. – P. 1215 – 1219.
19. **Ярцева, Е.П.** Качественная модель оценки концентрации аэрозольных примесей в атмосфере на основе итерационного метода / Е.П. Ярцева // Наука и устойчивое развитие: материалы VII Всероссийской научно-практической конференции молодых ученых. – Нальчик. – 2013. – С. 176 – 178.
20. **Ярцева, Е.П.** Построение итерационного вычислительного алгоритма качественной оценки концентрации загрязняющих примесей в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы ежегодной научно – практической конференции Северо – Кавказского федерального университета «Университетская наука - региону». – Ставрополь. – 2013. – С. 113 – 116.
21. **Ярцева, Е.П.** Построение модульной системы «Математические модели и вычислительные алгоритмы оценки концентрации загрязняющих примесей в турбулентной атмосфере» / Е.П. Ярцева // Наука и образование XXI века: сборник статей Международной научно – практической конференции. – Уфа. – 2014. – Часть 2. – С. 5 – 7.

Личный вклад автора в работах, опубликованных в соавторстве: (1 – 5) – построение моделей, алгоритмизация методов, проведение вычислительного эксперимента, разработка программного обеспечения, (7) – алгоритмизация метода, (9 – 20) – построение моделей, алгоритмизация методов, проведение вычислительного эксперимента, разработка программного обеспечения.

Структура и объем работы. Общий объем диссертации составляет 192 страницы, из них 160 страниц основного текста. Основная часть состоит из введения, четырех глав, заключения, списка литературы из 95 наименований. Диссертация содержит 33 рисунка, 12 таблиц и три приложения общим объемом 32 страницы.

Глава 1. Численные методы и математические модели в проблеме охраны окружающей среды в системе атмосфера - подстилающая поверхность

В главе выполняется обзор научной литературы по проблеме математического моделирования задач экологического мониторинга. Рассмотрены основные понятия и методы экологического мониторинга, физические основы явления переноса загрязняющих примесей в турбулентной атмосфере и соответствующие математические модели. Выполняется обзор численных методов, применяемых для построения вычислительных алгоритмов задачи переноса примесей. На основе данного обзора проводится анализ существующих проблем в данной научной области и обосновывается актуальность темы настоящего исследования.

1.1 Общая характеристика проблемы охраны окружающей среды в системе атмосфера - подстилающая поверхность

Понятие экологического мониторинга и проблема охраны окружающей среды. Экологический мониторинг - это система наблюдений, оценки и прогноза состояния природной среды, не включающая управление качеством окружающей среды, но дающая необходимую информацию для такого управления и выработки инженерных методов защиты окружающей среды [63]. Экологический мониторинг может быть реализован на основе системного подхода и проведения определенных организационно-технических мероприятий, цель которых – создание информационно-измерительной системы контроля и управления экологическим состоянием региона.

Проблема охраны окружающей среды является крайне наукоемкой и находится на стыке целого ряда областей физики, химии и математики. Она требует большого объема экспериментальных исследований, направленных на мониторинг состояния окружающей среды. Наряду с этим необходимы сложные теоретические исследования с целью построения адекватных математических моделей, включающих идентификацию источников загрязнения, оценку скорости выделения загрязнений, перемещение выбросов от источника к месту выпадения и знание физических и химических процессов преобразования выделенных веществ в течение этого перемещения [9, 15]. Как отмечалось выше, задачи мониторинга окружающей среды требуют привлечения современных вычислительных средств в связи с необходимостью обработки больших объемов информации, ее

сортировки, предварительной обработки для дальнейшего использования в сложных вычислительных алгоритмах. Создание моделей оценивания полей концентраций и параметров источников с использованием данных наблюдений и модельных представлений о процессах распространения примесей позволяет более надежно контролировать параметры техногенного загрязнения местности, что дает возможность определять информативность систем наблюдений и оптимизировать положение и количество точек отбора проб. Решение задачи идентификации источников позволяет установить вклад отдельных источников и учитывать это при создании новых – ввода новых промышленных предприятий и управление выбросами, чтобы их суммарное количество не превышало установленных норм. Определение возможных источников загрязнения использует методы, в основу которых заложено решение обратных задач переноса примеси, позволяющее по ограниченному числу точек наблюдений восстановить параметры источников и поля аэрозольного загрязнения местности. Здесь часто используют уравнения гидродинамики, требующие значительных усилий, как для аналитического исследования, так и для численного решения. Основоположником математических методов исследования по праву считают академика Марчука Г.И., значительные достижения в этой области принадлежат ученым НИИ Прикладной математики РАН (г. Москва), возглавляемого Марчуком Г.И. [23 – 31]. В работах этой научной школы создана методология математического моделирования, исследованы ее фундаментальные вопросы, разработаны оригинальные конструктивные подходы к изучению циркуляции атмосферы и океана, к решению задач прогноза погоды, теории климата и охраны окружающей среды с помощью математических моделей. Разработке и исследованию математических моделей транспортно – диффузионного распространения примесей в атмосфере посвящены многочисленные труды как российских, так и зарубежных ученых: Алояна А.Е. [1, 88, 89], Бабешко В.А. [3, 4], Бакланова А.А., Борисевича М.Д., Брандта Д., Головизнина В.М., Златева З., Нааца И.Э. [46 – 51], Пененко В.В. [53 – 56, 92], Печингера У., Семенчина Е.А. [62 – 64], Соренсона Д., Сухинова А.И. [65 – 67], Тишкина В.Ф. [71], Угольницкого Г.А. [12, 68, 69], Хирота М. и многих других. Также на сегодня разработано множество программ и программных комплексов на основе геоинформационных или экспертных систем, в структуре которых присутствуют моделирующие блоки. Расчеты проводятся с использованием моделей различной сложности, масштабности и оперативности. Например, экологический программный комплекс «Zone» [70] позволяет рассчи-

тывать рассеяние примеси в результате мгновенного или продолжительного выбросов и предназначен для оценки предельно допустимых концентраций. Разработанный группой сотрудников ИПМ РАН (г. Москва) интегрированный пакет «TIMES» [19] представляет собой визуально – прогностический пакет, позволяющий решать задачи о распространении загрязнений в воздушном бассейне с учетом ветрового поля и получать графические отображения процессов, как в условиях городской застройки, так и над местностью, имеющей сложный географический рельеф. Этот пакет является оригинальной программной системой как по применяемым адаптированным физико – математическим моделям, так и по методам решения систем уравнений, опирающихся на специально разработанные методики численного моделирования. Базовая модель этого комплекса оперативная конвективно – диффузионная или транспортная модель. На комплексе «TIMES» по заказу института гражданской авиации РФ были выполнены вычислительные эксперименты для задач безопасности окружающей среды при эксплуатации авиапромышленных комплексов. Каждая практическая задача требует разработки своей версии пакета, поэтому он создавался как модульная система.

Оценка и контроль загрязнения атмосферы, гидросферы и подстилающей поверхности в настоящее время основываются на результатах теоретического и экспериментального изучения распространения загрязняющих веществ от их источников. Основные работы по моделированию подобных задач обобщены в работе Кожевниковой М. Ф. [20]. По известным профилям ветра и различным полиномиальным формам распространения приземных концентраций на основе уравнения неразрывности массы построена простая модель загрязнения атмосферы города. Приблизительно связать максимальные концентрации примесей с вектором скорости ветра позволяют графические методы, когда на круговой диаграмме строят изоплеты повторяемости скорости ветра в полярных координатах [14]. Значительное развитие получила работа по прогнозу потенциала загрязнения воздуха Новикова Ю.В. [52]. В ее основу заложен учет статистической повторяемости условий, опасных с точки зрения формирования высоких уровней концентрации. В работах [58, 72, 87] на основе измерения концентраций загрязняющих ингредиентов строятся их эмпирические функции распределения, используемые для оценки вероятности превышения предельно допустимых концентраций. Широко известен подход к решению задачи о распространении атмосферных примесей от произвольной системы источников [2, 6]. Авторами работ [65 – 67] построены математические модели гидро-

физических процессов в мелководных водоемах при различных физических параметрах, а также процессов переноса загрязняющих веществ в многокомпонентной воздушной среде в прибрежной зоне.

1.2 Физические основы математической модели пограничного слоя атмосферы

Рассмотрим основные уравнения физической модели приземного слоя атмосферы. Состояние покоя атмосферы относительно земной (подстилающей) поверхности называется статическим состоянием, основное уравнение которого имеет вид [34]: $-\frac{\partial p}{\partial z} = g\rho$,

p - давление атмосферы на высоте z , g - ускорение свободного падения, ρ - плотность атмосферы. Первое начало термодинамики применительно к воздуху описывается уравнением [34]: $dq = c_v dT_i + p dv_i$, p - давление в окружающем воздухе, dq - количество тепла, сообщенное частице сухого воздуха единичной массы, dv_i - приращение объема частицы, dT_i - изменение внутренней энергии частицы, c_v - удельная теплоемкость при постоянном объеме. Термодинамический процесс, протекающий без теплообмена частицы с окружающей средой, называется адиабатическим. При адиабатическом процессе $dq = 0$. Отсюда следует, что при адиабатическом процессе расширение ($dv_i > 0$) или сжатие ($dv_i < 0$) частицы сухого воздуха единичной массы сопровождается соответственно уменьшением ($dT_i < 0$) или увеличением ($dT_i > 0$) ее внутренней энергии; при подъеме частицы объем ее увеличивается ($dv_i > 0$), а давление падает ($dp < 0$); кроме того, при адиабатическом подъеме температура частицы всегда уменьшается ($dT_i < 0$). Падение температуры сухого воздуха единичной массы при адиабатическом подъеме, отнесенное к единице высоты Z , называется сухоадиабатическим градиентом γ_α : $\gamma_\alpha = -\frac{dT_i}{dz}$. Легко

показать, что $\gamma_\alpha = \frac{g}{c_p} \frac{dT_i}{T_i}$, T_i - температура внешней среды, окружающей частицу, g -

ускорение свободного падения. В [63] показано, что γ_α мало изменяется с высотой: приблизительно на $0,0098 \text{ C}^\circ$ на каждый метр подъема (это означает, что через каждые 100 м подъема температура сухого воздуха единичной массы, адиабатически поднимающейся вверх, уменьшается примерно на $0,98 \text{ C}^\circ$). Выделим в атмосфере элементарный

объем влажного насыщенного воздуха единичной массы и обозначим через dq – количество сообщенного ей тепла. Оно будет затрачено на: изменение внутренней энергии (в количестве $c_v dT_i$), работу расширения (в количестве $p dv_i$), испарение содержащейся в частице влаги (в количестве $L dS_m$), где L - удельная теплота парообразования, dS_m - изменение доли пара путем испарения влаги. Уравнение первого начала термодинамики в этом случае имеет вид:

$$dq = c_v dT_i + p dv_i + L dS_m, \quad (1.1)$$

или

$$dq = c_p dT_i - RT_i \frac{dp}{p} + L dS_m, \quad (1.2)$$

R - удельная газовая постоянная влажного воздуха. При адиабатическом процессе, как и ранее, $dq=0$ в (1.1), (1.2). Величина $\gamma'_\alpha = -\frac{dT_i}{dz} = \frac{g}{c_p} \frac{T_i}{T_l} + \frac{L}{c_p} \frac{dS_m}{dz} = \gamma_\alpha + \frac{L}{c_p} \frac{dS_m}{dz}$ называется

влажноадиабатическим градиентом. Распределение по высоте температуры T_l и других метеорологических величин в окружающей элементарный объем воздуха внешней среде, называется стратификацией атмосферы. Распределение T_l в пограничном слое атмосферы характеризуется вертикальным градиентом температуры $\gamma = -\frac{\partial T_l}{\partial z}$. При адиабати-

ческом подъеме влажного насыщенного воздуха доля пара уменьшается, т.е. $\frac{dS_m}{dz} < 0$.

Кроме того, очевидно, что $\frac{L}{c_p} > 0$. Поэтому $\gamma'_\alpha < \gamma_\alpha$. В связи с этим возможны пять ви-

дов стратификации атмосферы: 1) сухо- и влажнонеустойчивая, или абсолютно неустойчивая, при $\gamma_\alpha > \gamma > \gamma'_\alpha$; 2) сухобезразличная и влажнонеустойчивая при $\gamma = \gamma_\alpha > \gamma'_\alpha$; 3) сухоустойчивая и влажнонеустойчивая, или условно устойчивая, при $\gamma_\alpha > \gamma > \gamma'_\alpha$; 4) сухоустойчивая и влажнобезразличная при $\gamma_\alpha > \gamma = \gamma'_\alpha$; 5) сухо- и влажноустойчивая, или абсолютно устойчивая при $\gamma_\alpha > \gamma'_\alpha > \gamma$.

Атмосферные течения могут находиться в двух режимах: ламинарном и турбулентном. Режим движения атмосферы, при котором малые объемы воздуха перемещаются по параллельным или слегка искривленным траекториям, называется ламинарным; режим, при котором малые объемы совершают неупорядоченные, хаотические движе-

ния, принято называть турбулентным. Характер режима (движения) газа определяется [63] двумя безразмерными параметрами:

1) числом Рейнольдса $R_l = \frac{ul}{\nu}$, u - скорость течения газа, l - характерный масштаб длин (средний размер тел, обтекаемых потоком, расстояние между твердыми стенками - границами и т.д.), ν - кинематический коэффициент молекулярной вязкости;

2) числом Ричардсона $R_i = \frac{g}{T} \frac{\gamma_\alpha - \gamma}{\beta^2}$, g - ускорение свободного падения, T - температура газа (воздуха) в К°, $\gamma = -\frac{\partial T}{\partial z}$ - вертикальный градиент температуры, γ_α - сухоадиабатический градиент, β - вертикальный градиент скорости ветра. При малом R_l движение газа носит ламинарный характер. Если же R_l превышает некоторое критическое значение $R_{lкр}$, то ламинарное течение переходит в турбулентное. Аналогично, если R_i меньше некоторого критического значения $R_{iкр}$, то интенсивность турбулентного движения (кинетическая энергия турбулентных частиц) увеличивается, если $R_i > R_{iкр}$ - ослабевает.

Обозначим через: S - удельное содержание примеси в атмосфере на высоте z , $S + \Delta S$ - удельное содержание примеси на высоте $z + \Delta z$, Q - массу примеси, которая переносится турбулентным потоком в единицу времени через единицу площади горизонтальной поверхности, расположенной между уровнями z и $z + \Delta z$ (Q называется турбулентным потоком). Предполагают, что

$$Q \approx -A \frac{\Delta S}{\Delta z}, \quad (1.3)$$

или в пределе

$$Q = -A \frac{\partial S}{\partial z}. \quad (1.4)$$

Коэффициент пропорциональности A в равенствах (1.3), (1.4) называется коэффициентом турбулентного обмена (перемешивания). Отношение $k = A/\rho$ называют коэффициентом турбулентности (вдоль оси Oz). Здесь введены коэффициенты турбулентного обмена A и турбулентности k вдоль оси Oz . Аналогично вводятся коэффициенты турбулентного обмена и турбулентности вдоль осей Ox , Oy . Вблизи земной поверхности молекулы воздуха, соударяясь с неровностями рельефа, теряют скорость поступательного движения. Сталкиваясь с другими молекулами, они также замедляют их движение. Воз-

никает сила трения между воздушным потоком и подстилающей поверхностью и наблюдается эффект прилипания потока к поверхности - скорость ветра в некотором слое (вязком подслое) у земной поверхности практически равна нулю. Толщину вязкого подслоя z_0 принято называть параметром шероховатости. Величина z_0 изменяется от сотых долей сантиметра (для пустыни $z_0 = 0,03$ см) до нескольких метров (для леса, города). Следует обратить внимание, что z_0 не совпадает с высотой неровностей рельефа подстилающей поверхности. Изменение температуры в любом объеме атмосферы определяется притоком тепла в этот объем: разностью входящих и уходящих потоков. Уравнение притоков тепла в атмосфере можно получить из первого начала термодинамики

(уравнения сохранения энергии) [63]: $\frac{dq}{dt} = c_p \frac{dT}{dt} - \frac{RT}{p} \frac{dp}{dt}$, T - температура единичной

массы воздуха, R - удельная газовая постоянная, p - давление, c_p - удельная теплоемкость, $\frac{dq}{dt}$ - приток тепла к единичной массе движущегося воздуха за единицу времени.

При этом $\frac{dq}{dt}$ представляет сумму удельных притоков тепла, обусловленных: турбулентным и молекулярным теплообменом ε_T , переносом лучистой энергии ε_l , фазовыми переходами воды в атмосфере ε_n , диссипацией кинетической энергии движения в тепловую в процессе молекулярного и турбулентного движений ε_k , т.е.

$\frac{dq}{dt} = \varepsilon_T + \varepsilon_l + \varepsilon_n + \varepsilon_k$. В общем виде уравнение притока тепла в турбулентной атмосфере

имеет вид [63]:

$$\frac{dT}{dt} = - \left(u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \right) + w(\gamma - \gamma_\alpha) + \frac{\partial}{\partial z} k \frac{\partial \theta}{\partial z} + \frac{\varepsilon_l + \varepsilon_n + \varepsilon_k}{c_p}, \quad (1.5)$$

$u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$, $w = \frac{dz}{dt}$ - проекции скорости единичной массы воздуха на оси координат

Ox , Oy , Oz ; k - коэффициент турбулентности вдоль оси Oz ; θ - потенциальная температура единичной массы (температура, которую примет единичная масса воздуха, если ее опустить или поднять сухоадиабатически с уровня ее положения до того уровня, где давление равно 1000 гПа). В приземном слое атмосферы уравнение (1.5) принимает вид [63]:

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial z} k \frac{\partial T}{\partial z} + \varepsilon', \quad (1.6)$$

$\varepsilon' = \varepsilon'(t, z)$ - приток тепла, обусловленный величинами ε_l , ε_n , ε_k (часто полагают, что $\varepsilon' = \varepsilon'(t, z)$ обусловлен только ε_l [63]). Пусть $T(t, z) = \bar{T}(z) + \theta(t, z)$, $\bar{T}(z)$ - среднее суточное значение температуры на высоте z (независящее от момента времени t), $\theta(t, z)$ - отклонение температуры от среднесуточной $\bar{T}(z)$ в момент времени t . Тогда (1.6) распадается на два уравнения:

$$\frac{\partial}{\partial z} k \frac{\partial \bar{T}}{\partial z} + \bar{\varepsilon}'(z) = 0, \quad (1.7)$$

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} k \frac{\partial \theta}{\partial z}, \quad (1.8)$$

$\bar{\varepsilon}'(z)$ - усредненный по времени приток тепла $\varepsilon'(t, z)$: $\varepsilon'(t, z) = \bar{\varepsilon}'(z) + \tilde{\varepsilon}'(t, z)$ флуктуации $\tilde{\varepsilon}'(t, z)$ притока $\varepsilon'(t, z)$ обычно не учитывают. Уравнение (1.7) при заданных k , $\bar{\varepsilon}'(z)$ и граничных условиях легко интегрируется, (1.8) имеет более сложный вид и при заданных k , начальном и граничных условиях далеко не всегда может быть проинтегрировано и решается численно. Оно позволяет описать суточный ход температуры приземного слоя воздуха. Граничные условия для $\theta(t, z)$ имеют вид: $\theta(t, \infty) = 0$, $\theta(t, 0) = \theta_m \cos(\omega t - \varphi)$, θ_m - амплитуда суточных колебаний на высоте $z=0$, $\omega = 2\pi/P$, P - период колебаний (сут), φ - начальная фаза. Распространение температуры в почве описывается [63] уравнением:

$\frac{\partial T}{\partial t} = \alpha^2 \frac{\partial^2 T}{\partial \zeta^2}$, α^2 - коэффициент температуропроводности почвы. Пусть

$T(t, \zeta) = \bar{T}(\zeta) + \tau(t, \zeta)$, $\bar{T}(\zeta)$ - среднее суточное значение температуры почвы на глубине ζ , $\tau(t, \zeta)$ - отклонение температуры почвы от среднесуточной $\bar{T}(\zeta)$ в момент времени t , тогда $\tau(t, \zeta)$ удовлетворяет уравнению:

$$\frac{\partial \tau}{\partial t} = \alpha^2 \frac{\partial^2 \tau}{\partial \zeta^2}. \quad (1.9)$$

Для двухслойной среды (почва - атмосфера) имеем систему уравнений (1.8), (1.9), граничные условия для которой имеют вид [63]:

$$\left. -k\rho c_p \frac{\partial \theta}{\partial z} - \alpha^2 \rho_1 c_1 \frac{\partial \tau}{\partial \zeta} = R_1 \cos \omega t \right|_{t=\zeta=0}, \quad \theta|_{z=0} = \tau|_{\zeta=0}, \quad \theta|_{z=\infty} = 0, \quad \tau|_{\zeta=\infty} = 0, \quad (1.10)$$

R_1 - суточная амплитуда колебаний радиационного баланса, ρc_p - объемная теплоемкость воздуха, $\rho_1 c_1$ - объемная теплоемкость почвы (воды). Граничные условия $\theta(t, \infty) = 0$, означают, что при неограниченном возрастании Z отклонения $\theta(t, z)$ и $\tau(t, \zeta)$ стремятся к нулю. Фактически условия $\theta(t, \infty) = 0$ начинают выполняться уже с высоты приземного слоя H , поэтому при численных расчетах оно заменяется условием: $\theta(t, H) = 0$. Аналогичное замечание справедливо для (1.10).

На прямоугольном бланке отложим по оси абсцисс температуру, а по оси ординат степенную функцию давления: $y \sim \left(P_0 \frac{\aleph - 1}{\aleph} - P \frac{\aleph - 1}{\aleph} \right)$, $\aleph = \frac{c_p}{c_v} = 1,40$. Прямые, параллельные оси абсцисс (проводятся через каждые 10 гПа для значений давлений от 1050 до 10 гПа) называются изобарами, а прямые, параллельные оси ординат (проводятся через 1 С° от -80 С° до 40 С°) называются изотермами [63]. Установившееся горизонтальное движение воздуха при прямолинейных равноотстоящих изобарах в отсутствии сил трения называется геострофическим ветром, а скорость такого движения воздуха - скоростью геострофического ветра G . В случае стационарных течений в пограничном слое атмосферы над плоской, однородной подстилающей поверхностью при прямолинейных и равномерно расположенных изобарах поле давлений $p(x, y, z)$ имеет вид [63]: $p(x, y, z) = p_1(z) + f\rho(V_g x - U_g y)$, V_g , U_g - компоненты геострофического ветра, определяемые равенствами $V_g = \frac{1}{f\rho} \frac{\partial p}{\partial x} = G \sin \alpha$, $U_g = -\frac{1}{f\rho} \frac{\partial p}{\partial y} = G \cos \alpha$, G - скорость геострофического ветра, α - угол между изобарой и осью абсцисс, направленной параллельно касательному напряжению трения вблизи подстилающей поверхности.

1.3 Обзор гидродинамических моделей пограничного слоя атмосферы

Как отмечалось во введении, в результате сбора данных к исследователям попадает разнородная информация, с разных сторон характеризующая наблюдаемые явления. При этом возникает задача совместного использования приближенных исходных данных математическими моделями, восстановления пространственно-временной структуры полей функций состояния, оценки параметров моделей и источников внешних воздействий, диагностики качества моделей и планирования наблюдений. Для определенности представляются задачи [34], связанные с оценками характеристик атмосферы с

использованием данных дистанционного зондирования и контактных наблюдений за компонентами функций состояния. В этом случае типичной является задача о нахождении распределения температуры и концентраций оптически активных субстанций в атмосфере. Основные элементы постановки задачи: модели процессов, модели измерений и функционалы для организации методов моделирования и усвоения данных наблюдений. Для описания процессов и их математических моделей вводятся три типа объектов: функции состояния $\vec{\varphi} = \{\varphi_i, i = \overline{1, n\varphi}\} \in Q(D_t, D_{t\nu})$, параметры моделей $\vec{Y} = \{Y_i, i = \overline{1, n\varphi}\} \in R(D_t, D_{t\nu})$, сопряженные функции $\vec{\varphi}^* = \{\varphi_i^*, i = \overline{1, n\varphi}\} \in Q^*(D_t, D_{t\nu})$. Функциональное пространство $Q^*(D_t, D_{t\nu})$ сопряжено по отношению к пространству функций состояния $Q(D_t, D_{t\nu})$, $R(D_t, D_{t\nu})$ - область допустимых значений параметров, $D_t, D_{t\nu}$ - области изменения пространственно-временных координат x, t , и частот ν . Структуры функций φ и φ^* идентичны, хотя информационное содержание их различается. Структура областей, выбор системы координат, основные обозначения и их смысл, постановки задач, краевых и начальных условий, описание функций состояния и параметров, структура функционалов, в зависимости от целей исследований, и способы построения интегрального тождества для вариационных формулировок моделей описаны в [34]. В качестве моделей процессов рассматриваются модели гидротермодинамики в климатической системе, модели переноса и трансформации влаги, химически и оптически активных загрязняющих примесей в газовом и аэрозольном состояниях. Функции источников в моделях параметрически учитывают действия естественных и антропогенных факторов. Для того, чтобы рассматривать совместно модели процессов и систему мониторинга с целью организации между ними взаимодействия в режиме прямых и обратных связей, предполагается, что все элементы комплекса (модели и наблюдения) могут содержать ошибки. В этом случае можно ставить вопрос о конструировании алгоритмов для реализации таких связей, исходя из условий минимизации ошибок. Общую структуру моделей для данного класса задач можно записать в следующей операторной форме:

$L_0(\vec{\varphi}) \equiv B \frac{\partial \vec{\varphi}}{\partial t} + G(\vec{\varphi}, \vec{Y}) - \vec{f} - \vec{\varepsilon} = 0$, B - диагональная матрица, $G(\vec{\varphi}, \vec{Y})$ - нелинейный матричный дифференциальный оператор, основной элемент в нем - адвективно-диффузионный оператор, действующий на различные компоненты функции состояния; \vec{f} - функции источников, $\vec{\varepsilon}$ - ошибки моделей. Операторы переноса в моделях процес-

сов симметризируются с помощью уравнения неразрывности [34]. Здесь рассмотрим только те модели из входящих в состав комплекса моделей климатической системы, которые непосредственно связаны с процессами распространения тепла, излучения и переноса оптически и химически активных субстанций в атмосфере.

1. Модель термодинамики атмосферы: $L_1(\vec{\varphi}) \equiv \frac{\partial \pi \theta}{\partial t} + \text{div}(\pi(\theta \vec{u} - \mu_\theta \text{grad} \theta)) - \pi Q_\theta - \varepsilon_\theta = 0$.

2. Модель переноса и трансформации влаги и оптически и химически активных примесей в газовом и аэрозольном состояниях:

$$L_2(\vec{\varphi}) \equiv \frac{\partial \pi \varphi_i}{\partial t} + \text{div}(\pi(\varphi_i \vec{u} - \mu_{\varphi_i} \text{grad} \varphi_i)) - \pi (R_i(\vec{\varphi}))_i - \pi Q_{\varphi_i} - \varepsilon_{\varphi_i} = 0, \quad i = \overline{1, na}.$$

Здесь θ - потенциальная температура, \vec{u} - вектор скорости, μ_θ , μ_{φ_i} - коэффициенты турбулентного обмена, π - функция, зависящая от давления, φ_i - отношения смеси для характеристик атмосферной влаги и концентрации примесей в атмосфере, na - общее число субстанций, Q_θ , Q_{φ_i} - источники тепла, влаги и примесей $(R_i(\vec{\varphi}))_i$ - операторы, описывающие процессы трансформации, ε_θ , ε_{φ_i} - функции, учитывающие ошибки и меру неопределенностей моделей процессов. Функции θ и φ_i , $i = \overline{1, na}$ включаются в число компонентов функции состояния $\vec{\varphi}$, а μ_θ , μ_{φ_i} , Q_θ , Q_{φ_i} и входные данные начальных и краевых условий - в компоненты вектора параметров \vec{Y} . Если учитывается аэрозолеобразование, то добавляется еще одна независимая переменная - размер частиц, и по этой переменной оператор трансформации становится интегро-дифференциальным. Содержательное и конструктивное описание операторов трансформации примесей осуществляется с помощью автоматизированной системы построения кинетических моделей атмосферной химии [34].

Значительные достижения в области математического моделирования атмосферных процессов содержатся в работах Алояна А.Е. и его учеников [1, 88, 89]. В частности, в статье [88] рассматривается математическая модель переноса многокомпонентной примеси с учетом фотохимической трансформации и образования аэрозолей в тропосфере северного полушария с учетом кинетических процессов нуклеации, конденсации и коагуляции. Модель рассматривается применительно к сферической Земле в системе координат (λ, ψ, z) , λ - долгота, ψ - дополнение до широты, z - высота от подстилающей по-

верхности. Основные уравнения для скорости изменения концентрации газовых примесей и аэрозолей записываются в следующем виде:

$$\begin{aligned} & \frac{\partial C_i}{\partial t} + \frac{u}{a \sin \psi} \frac{\partial C_i}{\partial \lambda} + \frac{v}{a} \frac{\partial C_i}{\partial \psi} + \omega \frac{\partial C_i}{\partial z} = \\ & = F_{gas} - P_{nucl} - P_{cond} + P_{phot} + \frac{\partial}{\partial z} v \frac{\partial C_i}{\partial z} + \frac{1}{a^2 \sin^2 \psi} \frac{\partial}{\partial \lambda} \mu \frac{\partial C_i}{\partial \lambda} + \frac{1}{a^2 \sin \psi} \frac{\partial}{\partial \psi} \mu \sin \psi \frac{\partial C_i}{\partial \psi}, \end{aligned} \quad (1.11)$$

$$\begin{aligned} & \frac{\partial \varphi_k}{\partial t} + \frac{u}{a \sin \psi} \frac{\partial \varphi_k}{\partial \lambda} + \frac{v}{a} \frac{\partial \varphi_k}{\partial \psi} + (\omega - \omega_g) \frac{\partial \varphi_k}{\partial z} = \\ & = F_{aer} - P_{cond} - P_{coag} + P_{nucl} + \frac{\partial}{\partial z} v \frac{\partial \varphi_k}{\partial z} + \frac{1}{a^2 \sin^2 \psi} \frac{\partial}{\partial \lambda} \mu \frac{\partial \varphi_k}{\partial \lambda} + \frac{1}{a^2 \sin \psi} \frac{\partial}{\partial \psi} \mu \sin \psi \frac{\partial \varphi_k}{\partial \psi}. \end{aligned} \quad (1.12)$$

Здесь C_i , ($i = \overline{1, N_g}$), φ_k , ($k = \overline{1, N_a}$) – концентрация газовых примесей и аэрозолей; N_g и N_a – число газовых компонент и аэрозольных фракций; (u, v, ω) – компоненты вектора скорости ветра в направлении (λ, ψ, z) ; ω_g – скорость гравитационного оседания; a – средний радиус земли; (μ, ν) – коэффициенты турбулентного обмена в горизонтальном и вертикальном направлениях; F_{gas} и F_{aer} – источники газовых примесей и аэрозолей; P_{nucl} , P_{cond} , P_{coag} , P_{phot} – операторы нуклеации, конденсации, коагуляции и фотохимической трансформации. Уравнения (1.11) и (1.12) рассматриваются в области $D_t = G \times [(0, T)]$, $G = S \times [0, H]$; $S = \{(\lambda, \psi): 0 \leq \lambda \leq 2\pi, 0 \leq \psi \leq \pi/2\}$; H – верхняя граница расчетной области:

$$\begin{aligned} \varphi(0, \psi, z, t) &= \varphi(2\pi, \psi, z, t), \quad \varphi(\lambda, -\psi, z, t) = \varphi(\lambda + \pi, \psi, z, t), \\ \varphi(\lambda, \pi + \psi, z, t) &= \varphi(\lambda + \pi, \pi - \psi, z, t), \quad \frac{\partial \varphi}{\partial \psi} = 0, \quad \psi = \pi/2. \end{aligned}$$

Для определения коэффициентов горизонтального турбулентного обмена используется метод, предложенный Смагоринским [24]. Метеорологические характеристики в пограничном слое рассчитываются с помощью параметризации планетарного пограничного слоя атмосферы. Краевые условия задачи (1.11)-(1.12) на первом расчетном уровне (на верхней границе приземного слоя атмосферы) $z = h$ и операторы P_{nucl} , P_{cond} , P_{coag} , P_{phot} рассматриваются в работе [29]. В качестве примера приведем вид оператора P_{coag} , соответствующего процессу коагуляции. Изменение концентрации и спектра размеров частиц описывается уравнением Смолуховского:

$$\frac{\partial \varphi_g}{\partial t} = \frac{1}{2} \int_0^g \tilde{K}(g, g_1) \varphi_{g-g_1} \varphi_{g_1} dg_1 - \varphi_g \int_0^\infty \tilde{K}(g, g_1) \varphi_{g_1} dg_1 + I(\varphi_g, t), \quad (1.13)$$

g_1 - текущая переменная (масса), по которой производится интегрирование, φ_g - концентрация частиц с массой, заключенной в интервале от g до $g + dg$; $\tilde{K}(x, y)$ - частота столкновения частиц с массами x и y ; $I(\varphi_g, t)$ - интенсивность образования новых частиц с массой g . Этот член добавляется в правую часть (1.13). Первый член в правой части (1.13) описывает приход частиц в размер g за счет коагуляции меньших частиц, второй – уход частиц из размера g за счет коагуляции со всеми остальными частицами. Решение (1.13) определяется на множестве неотрицательных решений. При этом предполагается, что сталкиваются частицы шарообразной формы, и длина свободного пробега в газе-носителе намного больше их размера. В этом случае $\tilde{K}(x, y)$ можно представить в следующем виде [29]: $\tilde{K}(x, y) = A(x^{1/3} + y^{1/3})(D_x + D_y)\beta_{xy}$, D_x - коэффициент диффузии частицы массы x : $D_x = \frac{k_B T}{6\pi\eta \cdot R_x} \left[1 + Kn_x (A + Q \cdot \exp\left(-\frac{b_3}{Kn_x}\right)) \right]$, k_B - постоянная Больцмана; b_3 , A , Q - эмпирические постоянные; $A = 1,25$, $Q = 0,4$, $b_3 = 1,1$. Фактор коррекции β_{xy} определяется из [38]:

$$\beta_{xy} = \left(\frac{R_x + R_y}{R_x + R_y + d_{xy}} + \frac{4(D_x + D_y)}{(V_x^2 + V_y^2)^{1/2}(R_x + R_y)} \right)^{-1}, \quad Kn_x = \frac{l_x}{R_x}; \quad d_{xy} = (d_x^2 + d_y^2)^{1/2};$$

$$d_x = \frac{1}{6R_x l_x} \left((2R_x + l_x)^3 - (4R_x^2 + l_x^2)^{3/2} \right) - 2R_x, \quad l_x = \frac{8D_x}{\pi V_x}, \quad V_x = \left(\frac{8kT}{\pi x} \right)^{1/2}, \quad x = \frac{4}{3} \pi R_x^3 \rho. \quad (1.14)$$

Здесь D_x - коэффициент диффузии частицы i с радиусом R_x ; T - температура; l_x - средняя длина пробега; V_x - средняя скорость частицы; ρ - плотность частицы; η - вязкость газа. Численные методы решения (1.13)-(1.14) приведены в [30].

Далее выполним обзор основных уравнений переноса и диффузии в атмосфере, используя источники [23, 24, 29 – 31]. Перенос загрязняющих субстанций в атмосфере осуществляется ветровыми потоками воздуха с учетом их мелкомасштабных флуктуаций. Осредненный поток субстанций имеет адвективную и конвективную составляющие, а осредненные флуктуационные их движения интерпретируют как диффузию на фоне основного осредненного движения, связанного с ним. Пусть $\varphi(x, y, z, t)$ - интенсивность аэрозольной субстанции, мигрирующей вместе с потоком воздуха в атмосфере.

Перенос субстанции вдоль траектории частиц воздуха с сохранением ее интенсивности с учетом уравнения неразрывности записывается следующим образом [24]:

$$\frac{\partial \varphi}{\partial t} + \operatorname{div} \bar{u} \varphi = 0. \quad (1.15)$$

К (1.15) присоединяются начальные данные и краевые условия. Точное решение (1.15) возможно тогда, когда известны значения функций u , v и w в пространстве и во все моменты времени. Если же информации о компонентах вектора скорости недостаточно, то пользуются различными приближениями. Процесс переноса при наличии источников загрязняющей субстанции φ , описываемых функцией $f(x, y, z, t)$, когда часть субстанции вступает в реакцию с внешней средой или распадается, описывается уравнением: $\frac{\partial \varphi}{\partial t} + \operatorname{div} \bar{u} \varphi + \alpha \varphi = f$. Для стационарного процесса коэффициенты u , v и w вместе с другими входными данными задачи f и φ не зависят от времени:

$$\operatorname{div} \bar{u} \varphi + \alpha \varphi = f, \quad \varphi = \varphi_S \quad \text{на } S \quad \text{при } u_n < 0. \quad (1.16)$$

Задача (1.16) описывает частный процесс переноса субстанций с неизменными во времени входными данными. Однако набор таких частных решений может использоваться и при описании более сложных физических ситуаций, реализуемых на практике. А именно, в различные периоды времени в атмосфере реализуются те или иные типы движений воздушных масс, которые за период характерного времени существования считают стационарными. После каждого такого периода происходит их перестройка и наступает новое стационарное состояние. Так как период перестройки циркуляции намного короче времени существования данного типа движений, то считается, что перемена типов движений происходит мгновенно. Пусть этих типов будет n , в результате приходят к системе независимых уравнений

$$\operatorname{div} \bar{u}_i \varphi_i + \alpha \varphi_i = f, \quad \varphi_i = \varphi_{iS} \quad \text{на } S \quad \text{при } u_{in} < 0, \quad i = \overline{1, n} \quad (1.17)$$

и решению задачи о распределении примеси в виде линейной комбинации

$$\tilde{\varphi} = \frac{1}{T} \sum_{i=1}^n \varphi_i \Delta t_i. \quad (1.18)$$

Задача (1.17) - (1.18) является статистической моделью. Решение стационарных задач имеет много общего с решением задачи о среднем за некоторый период времени T распределении субстанции на основе специальным образом поставленных нестационар-

ных задач: $\frac{\partial \varphi}{\partial t} + \operatorname{div} \bar{u} \varphi + \sigma \varphi = f$, $\varphi = \varphi_S$ на S при $u_n < 0$, здесь функции u и φ_S , как и ранее считаются независимыми от t . Процесс переноса и диффузии субстанции в исключительно диффузионной постановке имеет вид [24]: $\sigma \varphi = \mu \frac{d^2 \varphi}{dx^2} + Q \delta(x - x_0)$. Случай, когда скорость потока воздушных масс отлична от нуля [30], описывается уравнением:

$$u \frac{d\varphi}{dx} + \sigma \varphi = \mu \frac{d^2 \varphi}{dx^2} + Q \delta(x - x_0),$$

решение которого характеризует снос субстанции ветром с одновременной диффузией.

Особый интерес при изучении задач локального загрязнения окружающей среды представляют тяжелые аэрозоли. Распространяясь, они диффундируют и под действием силы тяжести опускаются на землю. Данную скорость находят из задачи Стокса, она является величиной постоянной, направленной вниз. Поэтому в уравнениях переноса появляется новое слагаемое $\bar{w}_g \partial \varphi / \partial z$, и задача принимает вид:

$$\frac{\partial \varphi}{\partial t} + \frac{\partial u \varphi}{\partial x} + \frac{\partial v \varphi}{\partial y} + \frac{\partial (w - \bar{w}_g) \varphi}{\partial z} + \sigma \varphi = \frac{\partial}{\partial z} v \frac{\partial \varphi}{\partial z} + \mu \Delta \varphi + f$$

$$\varphi = \varphi_0 \quad \text{при } t = 0, \quad \varphi = 0 \quad \text{на } \Sigma, \quad \frac{\partial \varphi}{\partial z} = \alpha \varphi \quad \text{на } \Sigma_0, \quad \varphi = 0 \quad \text{на } \Sigma_H.$$

Количество выпавшего аэрозоля за интервал времени находят из уравнения:

$$\frac{\partial \bar{\varphi}}{\partial t} + \frac{\partial u \bar{\varphi}}{\partial x} + \frac{\partial v \bar{\varphi}}{\partial y} + \sigma \bar{\varphi} = \mu \Delta \bar{\varphi} - (\bar{w}_g + v \alpha) \bar{\varphi}_g + F, \quad (1.19)$$

Из (1.19) следует, что в каждую единицу времени количество аэрозоля в атмосфере над точкой (x, y) уменьшается на величину $(\bar{w}_g + v \alpha) \bar{\varphi}_g$. При этом $\bar{w}_g \bar{\varphi}_g$ - часть аэрозолей, связанная с падением частиц под действием силы тяжести, а $v \alpha \bar{\varphi}_g$ - с турбулентным обменом в приграничной зоне около земной поверхности.

В работах [12, 68, 69] моделирование экологических объектов выполнено с помощью имитационных систем, позволяющих на базе средств современной вычислительной техники объединить концепции и фактический материал, накопленный естественными и общественными науками, математические методы и экспертные оценки, отражающие знания, опыт и интуицию специалистов – практиков. Авторами описана концептуальная модель оценки воздействия на экосистему региона, проведена ее математическая формализация, результатом которой явилась система конечно – разностных уравнений и некоторых вспомогательных соотношений.

Работы [65 – 67] основаны на модели движения многокомпонентной воздушной среды, которая базируется на:

- уравнении Навье – Стокса:
$$\frac{dv_j}{dt} = -\frac{1}{\rho} \frac{\partial p}{\partial x_j} + \operatorname{div}(\mu \operatorname{grad}(v_j)) - g_i,$$

- уравнении неразрывности:
$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{v}) = \operatorname{div}(\mu \operatorname{grad}(\rho)) + I_\rho,$$

- уравнении состояния:
$$P = \sum_i \frac{\rho_i}{M_i} RT,$$

- уравнении транспорта загрязняющих веществ:
$$\frac{d\varphi_i}{dt} = \operatorname{div}(\mu \operatorname{grad}(\varphi_i)) + I_\varphi,$$

- уравнении притока тепла:
$$\frac{dQ}{dt} = \operatorname{div}(\mu \operatorname{grad}(Q)) + \operatorname{div}(\lambda \operatorname{grad}(T)) + I_Q.$$

Здесь φ_i - объемные доли i -ой фазы ($i = 0$ - воздух, $i = 1$ - вода в газообразном состоянии, $i = 2$ - газ на источнике, $i = 3$ - вода в жидком состоянии, $i = 4$ - сажа), v_j - проекции компонент скорости на оси Ox_j , g - ускорение свободного падения, R - универсальная газовая постоянная, M - молярная масса, I - функция, описывающая распределение и мощность источников примесей, T - температура газовой фазы, Q - тепловая энергия, λ - коэффициент теплопроводности.

В работе [60] делается попытка обсуждения всего спектра проблем численного решения задач конвекции – диффузии. Рассматриваются вопросы построения дискретных аналогов, важнейшие вопросы сходимости приближенного решения к точному решению в различных нормах. Принципиальные вопросы устойчивости и корректности разностных схем для задач конвекции – диффузии рассмотрены с позиций общей теории устойчивости операторно – разностных схем, приведены оценки устойчивости по начальным данным и правой части.

1.4. Постановка задачи моделирования нестационарного переноса примесей в пограничном слое атмосферы и определение исходных данных на основе сопутствующих математических моделей

Вычисление концентрации загрязняющих примесей, переносимых в пределах пограничного слоя атмосферы на основе уравнения переноса. Математические модели, связанные с описанием явления диффузного переноса загрязнений в пределах по-

граничного слоя атмосферы и разработкой численных методов прогноза их распространения, основаны на нестационарных трехмерных уравнениях параболического типа. Трехмерное уравнение переноса субстанции в турбулентной среде можно записать в следующем виде [38]:

$$\frac{\partial q(P,t)}{\partial t} + \alpha(t) \cdot q(P,t) + \left(\frac{\partial}{\partial x} (V_x(P,t) \cdot q(P,t)) + \frac{\partial}{\partial y} (V_y(P,t) \cdot q(P,t)) + \frac{\partial}{\partial z} (V_z(P,t) \cdot q(P,t)) \right) - \left(\frac{\partial}{\partial x} \left(K_x(P,t) \cdot \frac{\partial q(P,t)}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y(P,t) \cdot \frac{\partial q(P,t)}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_z(P,t) \cdot \frac{\partial q(P,t)}{\partial z} \right) \right) = S(P,t), \quad (1.20)$$

в котором $P = P(x, y, z)$ - точка области пространства, $P \in \Omega \subset R_3$, $t \in [0, T]$,

$$\text{начальные условия} \quad q(P,t)|_{t=0} = q(P,0) = q_0(P) \quad \text{для } P \in \Omega, \quad (1.21)$$

$$\text{граничные условия} \quad q(P,t) = \bar{q}(P,t) \quad \text{для } P \in \bar{\Omega}, \quad (1.22)$$

$q(P,0) = q_0(P)$ и $\bar{q}(P,t)$ - заданные функции, а $\bar{\Omega}$ - граница области Ω . В уравнении (1.20) $q(P,t)$ - концентрация примесей, имеющих в точке пространства P в момент времени t , ($\text{кг}/\text{м}^3$); $\alpha(t)$ - коэффициент, характеризующий степень вывода или привнесения примесей в данный объем за счет химических или других процессов, протекающих в приземном слое атмосферы, ($1/\text{с}$); $V_x(P,t)$, $V_y(P,t)$, $V_z(t,P)$ - компоненты вектора скорости ветра, ($\text{м}/\text{с}$); $K_x(P,t)$, $K_y(P,t)$, $K_z(P,t)$ - турбулентность, характеризуемая коэффициентом турбулентной диффузии, ($\text{м}^2/\text{с}$), перенос осуществляется вдоль координатных осей \bar{Ox} , \bar{Oy} , \bar{Oz} [34, 35, 38 – 40]. В общем случае

$$K(P,t) = \begin{bmatrix} K_{xx}(P,t) & K_{xy}(P,t) & K_{xz}(P,t) \\ K_{yx}(P,t) & K_{yy}(P,t) & K_{yz}(P,t) \\ K_{zx}(P,t) & K_{zy}(P,t) & K_{zz}(P,t) \end{bmatrix}$$

- тензорная величина. Если положить, что

$$K(P,t) = \begin{bmatrix} K_{xx}(P,t) & 0 & 0 \\ 0 & K_{yy}(P,t) & 0 \\ 0 & 0 & K_{zz}(P,t) \end{bmatrix},$$

то можно записать $K_{xx}(P,t) = K_x(P,t)$, $K_{yy}(P,t) = K_y(P,t)$, $K_{zz}(P,t) = K_z(P,t)$. $S(P,t)$ - источник загрязняющих примесей, ($\text{кг}/(\text{м}^3 \cdot \text{с})$). Задача (1.20) – (1.22) решается относительно распределения $q(P,t)$, остальные функции являются исходными измерительными данными, значения которых считаются известными (например, это могут быть экспери-

ментальные измерения). Уравнение (1.20) представляет собой параболическое дифференциальное уравнение в частных производных, известное под названием уравнения теплопередачи (теплопроводности) или уравнения диффузии или уравнения массопереноса, связывающим распределения $q(P,t)$ и $S(P,t)$ при заданных распределениях $V_x(P,t)$, $V_y(P,t)$, $V_z(P,t)$ и $K_x(P,t)$, $K_y(P,t)$, $K_z(P,t)$. Ограничения, накладываемые на функции исходных данных, гарантирующие существование и единственность решения поставленной задачи хорошо известны и далее считаем, что они выполняются и решение (1.20) существует и единственно.

Относительно формы уравнения (1.20) можно далее заметить следующее. Не трудно видеть, что оно записано в скалярной форме. Если речь идет о векторном поле скорости ветра $\vec{V} = V_x \cdot \vec{i} + V_y \cdot \vec{j} + V_z \cdot \vec{k}$, то уравнение (1.20) может быть записано в векторном виде, а именно

$$\left. \begin{aligned} \dot{q} + \alpha q + \operatorname{div} \vec{J} &= S \\ \vec{J} &= \vec{V}q - \vec{K} \cdot \nabla q \end{aligned} \right\}, \quad (1.23)$$

\vec{J} - вектор потока переносимой субстанции, а \vec{K} - диада (тензор) с матрицей $\{K_{i,j}\}$, $i, j = 1, 2, 3$. Если турбулентным движением в среде можно пренебречь, то есть имеет место только адвективный перенос, то система (1.23) примет вид

$$\left. \begin{aligned} \dot{q} + \operatorname{div} \vec{J} &= S \\ \vec{J} &= \vec{V}q \end{aligned} \right\}.$$

Уравнение переноса субстанции в турбулентной среде, представленное в виде: $\dot{q} + Dq = S$, D - дифференциальный оператор второго порядка по пространственным переменным задачи, является моделью эволюционного типа, удобной при исследовании нестационарных задач [38] и далее рассматривается как самостоятельная типовая задача вычислительной математики:

$$\dot{\varphi} + A\varphi = f, \quad (1.24)$$

при $\varphi(0) = \varphi_0$, $\varphi \in \Phi$, Φ - множество решений (1.24), включающее в себя функции, непрерывные вместе со своими производными первого порядка. В качестве оператора A могут использоваться как дифференциальные операторы (параболическое уравнение), так и интегральные. В работе выполнено построение и обоснование решающих алгоритмов, которые носят рекурсивный характер и практически реализуются с помощью

операторов шага и источника: $T^j = \left(I + \frac{\tau}{2} \Lambda^j\right)^{-1} \left(I - \frac{\tau}{2} \Lambda^j\right)$, $S^j = \left(I + \frac{\tau}{2} \Lambda^j\right)$. Оператор T^j называется оператором шага, поскольку он определяет процесс пошагового определения φ^{j+1} по значениям φ^j , оператор S^j называют оператором источника, Λ^j - конечномерная аппроксимация оператора $A(t)$ на интервале $t_j \leq t \leq t_{j+1}$, $\tau = t_{j+1} - t_j$. Приведенные рассуждения позволяют записать конечномерное приближение для исходного эволюционного уравнения (1.24):

$$\varphi^{j+1} = T^j \varphi^j + \tau \cdot S^j f^j, \quad j = 0, 1, \dots \quad (1.25)$$

Выражение (1.25) определяет вычислительный процесс последовательного преобразования $\varphi^j \rightarrow \varphi^{j+1}$, где $j = 0, 1, \dots$, поэтому первым вопросом, связанным с исследованием этого процесса, является вопрос о его сходимости и соответствующих ограничениях на операторы T^j и S^j . Далее в работе [38] получено неравенство: $\|T^j\| < 1$, выступающее как ограничение на оператор шага. Для рекурсивной схемы (1.25) необходимы гарантии ограниченности члена $S^j f^j$ при любых значениях j . Ясно, что если норма $\|f\|_{\Phi}$ ограничена, допустим, константой M_f , то для ограниченности нормы $\|S^j f^j\|$ вполне достаточно ограниченности нормы оператора источника: $\|S^j f^j\| \leq \|S^j\| \cdot \|f^j\| \leq M_f$.

Вычисление компонент вектора скорости ветра на основе уравнения Навье-Стокса. Для определения, точнее вычисления, исходных данных в модели переноса (1.23), а именно компонент вектора скорости ветра и коэффициентов турбулентной диффузии возможно привлечение соответствующих математических моделей. В рамках данного параграфа делается попытка расширить представленную теорию переноса в пограничном слое атмосферы введением в нее векторного уравнения Навье-Стокса с обязательным учетом в них членов, определяемых турбулентным состоянием пограничного слоя атмосферы, с помощью которого возможна оценка поля скорости ветра в каждом конкретном случае [38]. При исследовании поля скорости ветра в приземном слое атмосферы естественно на первых порах обратиться к простейшей из возможных моделей, в основе которой лежит представление о движении «идеального» газа [34]. Соответствующая система уравнений, именуемая уравнениями Эйлера, имеет вид:

$$1. \text{ Уравнение Навье-Стокса: } \frac{\partial \vec{V}}{\partial t} + (\vec{V} \nabla) \vec{V} = -\frac{\nabla p}{\rho} + \vec{F}, \quad \vec{V} \cdot \nabla = V_x \frac{\partial}{\partial x} + V_y \frac{\partial}{\partial y} + V_z \frac{\partial}{\partial z}; \quad (1.26a)$$

$$2. \text{ Уравнение неразрывности: } \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{V}) = 0, \quad (1.26б)$$

$$3. \text{ Уравнение состояния (уравнение Клайперона): } p = f(\rho, T), \quad p = f(\rho, T) = R\rho T \quad (1.26в)$$

Все функции, входящие в систему уравнений (1.26) являются функциями координат (x, y, z) и времени t . Точка $M(x, y, z, t) \in D \otimes \Omega_t$, где $\Omega_t = [t_1, t_2]$. Система (1.26) при заданном температурном поле $T(x, y, z, t)$ в пределах указанной области определяет поле вектора $\vec{V}(V_x, V_y, V_z)$ и плотности газа $\rho(x, y, z, t)$ [кг/м³]. При рассмотрении локальных объемов в пределах пограничного слоя атмосферы в рамках данной модели движения воздушных масс источником этих движений следует считать поле давлений $p(x, y, z, t)$ [кг/м · с²], представленное в уравнениях градиентом ∇p (бариметрический градиент). На поле скорости ветра \vec{V} должны быть наложены ограничения $V_n = (\vec{V}, \vec{n}) = \varphi(M', t)$, $M' \in \partial D$, $\vec{V}(M, t = 0) = \vec{V}_0(M)$, $M \in D$, $n(M')$ - нормаль к границе области D . Вектор \vec{F} , представляет собой силовое поле, действующее в области $\bar{D} = D \cup \partial D$. Сила \vec{F} рассчитывается на единицу массы воздуха и потому $\vec{F}(F_x, F_y, F_z)$ имеет размерность [м/с²] (ускорения). В дальнейшем считаем, что $\vec{F}(0, 0, g)$. Кроме того, R - универсальная газовая постоянная. Следует помнить, что первое уравнение (1.26а) связано с интегральными равенствами

$$\iiint_V \rho \frac{d\vec{V}}{dt} dV = -\iiint_V \nabla p dV + \iint_{\partial V} \rho \vec{F} dS, \quad (1.27)$$

V - объем области D (ее мера). Интеграл слева есть сила, действующая на частицы среды в пределах области D в соответствии с законом Ньютона. Интеграл справа $\iint_{\partial V} \nabla p dV = \iint_{\partial V} p \vec{n} dS$, есть результирующая сил давления. И, наконец, последний интеграл справа является результирующим внешних сил. Таким образом (1.27) – уравнение движения субстанции в пределах исследуемого объема в интегральной форме. Второе уравнение системы (1.26) связано с интегральным равенством

$$\frac{d}{dt} \iiint_V \rho dV = \iint_{\partial V} \rho \vec{V} \vec{n} dS, \quad (1.28)$$

за которым стоит условие, что при отсутствии источников и стоков в пределах области V изменение количества вещества в единицу времени равно потоку этого вещества че-

рез границу S . Используем тождество $\iint_S \rho \vec{v} \vec{n} dS = \iiint_V \nabla(\rho \vec{v}) dV$, что позволяет равенство

(1.28) переписать в виде: $\iiint_V \left[\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{v}) \right] dV = 0$, откуда следует:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{v}) = 0. \quad (1.29)$$

В силу тождества $\nabla(\rho \vec{v}) = \vec{v} \cdot \nabla \rho + \rho \cdot \text{div} \vec{v}$, уравнение (1.29) запишем так:

$$\frac{\partial(\ln \rho)}{\partial t} + \vec{v} \cdot \nabla(\ln \rho) + \text{div} \vec{v} = 0. \quad (1.30)$$

Если принять, что $\text{div} \vec{v} = 0$ при переносе субстанции в пограничном слое атмосферы [24] (закон сохранения массы), то (1.30) примет наиболее простой вид, а именно:

$$\frac{\partial(\ln \rho)}{\partial t} = -\vec{v} \cdot \nabla(\ln \rho). \quad (1.31)$$

Соотношение (1.31) связывает временную и пространственную производные плотности $\rho(M, t)$ в пределах области D . При рассмотрении локальных объемов в рамках данной модели движения воздушных масс источником этих движений следует считать поле давлений, представленное в уравнениях градиентом ∇p . Ясно, что непосредственное определение поля ∇p (бариметрического градиента) вряд ли возможно. В связи с этим определяющим следует считать температурные градиенты, обусловленные температурным полем (тоже термодинамикой пограничного слоя атмосферы). Из уравнения состояния воздуха следует аналитическая связь градиентов ∇p и ∇T : $\frac{\nabla p}{\rho} = R \cdot T \cdot \frac{\nabla \rho}{\rho} + R \cdot \nabla T$,

или $\frac{\nabla p}{\rho} = R \cdot T \cdot \nabla(\ln \rho) + R \cdot \nabla T$. Приведенные рассуждения позволяют переписать систему

уравнений (1.26):

$$\begin{cases} \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} = -R \cdot T \cdot \nabla(\ln \rho) - R \cdot \nabla T + \vec{F} \\ \frac{\partial(\ln \rho)}{\partial t} + \vec{v} \cdot \nabla(\ln \rho) \end{cases}. \quad (1.32)$$

В дальнейшем считаем, что функция $T(x, y, z, t)$ определена и непрерывно дифференцируема всюду в $\bar{D} = D \cup \partial D$ и, следовательно, остается задать граничные условия на распределение $\rho(x, y, z, t)$. Конечно, выписать эти условия весьма трудно, да и следует учесть идеализированный характер рассматриваемой модели. Поэтому разумно, как это и делается в прикладных задачах, пространственно-временные вариации плотности воз-

духа в пределах пограничного слоя атмосферы считать малыми, что согласуется с принятым условием $div \vec{V} = 0$. Это условие заметно упрощает модель (1.32):

$$\frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla) \vec{V} = \vec{F} - R \cdot \nabla T. \quad (1.33)$$

Уравнение (1.33) позволяет методами математического моделирования в пределах соответствующего вычислительного эксперимента исследовать, каким образом температурный градиент ∇T в пределах области D определяет структуру поля скорости ветра \vec{V} . Учет вязкости воздуха в тех моделях, которые связаны с изучением явления перенос-диффузия в пределах пограничного слоя атмосферы, требует введения в уравнения дви-

жения дополнительного оператора $\tilde{\Delta} = \frac{\partial}{\partial x} \mu \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \mu \frac{\partial}{\partial y} + \frac{\partial}{\partial z} \nu \frac{\partial}{\partial z}$, μ и ν - горизонталь-

ные и вертикальные коэффициенты турбулентного обмена. Ниже будем использовать

запись $\tilde{\Delta} = \frac{\partial}{\partial x_1} K_1 \frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2} K_2 \frac{\partial}{\partial x_2} + \frac{\partial}{\partial x_3} K_3 \frac{\partial}{\partial x_3}$, или в более общей форме:

$\tilde{\Delta} = \sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial}{\partial x_i} K_{i,j} \frac{\partial}{\partial x_j}$. Последнее выражение учитывает возможность взаимного турбу-

лентного обмена между всеми координатными направлениями в трехмерном пространстве. С учетом всех замечаний исходную систему уравнений следует писать в виде

$$\begin{cases} \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla) \vec{V} = -\frac{\nabla p}{\rho} + \tilde{\Delta} \vec{V} + \vec{F} \\ \frac{\partial \rho}{\partial t} + div(\rho \cdot \vec{V}) = 0 \\ p = \tilde{f}(\rho, T) \end{cases}. \quad (1.34)$$

В случае учета вязкости в потоках воздуха предположение об «идеальном газе» естественно может быть принято более чем условно. В частности, последнее уравнение в (1.34) – уравнение состояния становится неопределенным. Если при этом используются некие полуэмпирические формулы, то возникают определенные трудности с установлением функциональных связей между градиентами $\nabla \rho$ и ∇T . Подобное обстоятельство требует применения соответствующих методик дифференцирования функций, заданных приближенно. Если считать, что выполнены те предположения, которые вели выше к уравнению (1.33), то соответствующим аналогом этого уравнения явится уравнение:

$$\frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla) \vec{V} - \tilde{\Delta} \vec{V} = \vec{F} - R \cdot \nabla T, \quad (1.35)$$

которое позволяет исследовать влияние градиента температуры ∇T на структуру векторного поля \vec{V} с учетом турбулентного характера движения воздуха. В случае двух пространственных переменных, скажем (x, z) уравнение (1.35) в координатной форме примет вид:

$$\begin{aligned} \frac{\partial V_1}{\partial t} + V_1 \frac{\partial V_1}{\partial x} + V_2 \frac{\partial V_1}{\partial z} - \frac{\partial}{\partial x} K_1 \frac{\partial V_1}{\partial x} - \frac{\partial}{\partial z} K_1 \frac{\partial V_1}{\partial z} &= R \frac{\partial T}{\partial x}, \\ \frac{\partial V_2}{\partial t} + V_1 \frac{\partial V_2}{\partial x} + V_2 \frac{\partial V_2}{\partial z} - \frac{\partial}{\partial x} K_2 \frac{\partial V_2}{\partial x} - \frac{\partial}{\partial z} K_2 \frac{\partial V_2}{\partial z} &= R \frac{\partial T}{\partial z} - g. \end{aligned}$$

В случае постановки краевой задачи для этой системы уравнений можно использовать условия:

$$\begin{aligned} V_1(P, t=0) &= V_{1,0}(P), \text{ при } P \in D, \quad V_2(P, t=0) = V_{2,0}(P), \text{ при } P \in D, \\ V_1(P, t=0) &= \bar{V}_1(P), \text{ при } P \in \partial D, \quad V_2(P, t=0) = \bar{V}_2(P), \text{ при } P \in \partial D. \end{aligned}$$

В работе [38] изложен еще один подход к построению вычислительной схемы для аэродинамической модели пограничного слоя атмосферы. Будем, как и раньше, исходить из уравнения Эйлера движения идеального газа (жидкости), которое записывается так: $\rho \cdot \frac{d\vec{V}}{dt} = \rho \cdot \vec{F} - \nabla p$. Это уравнение можно привести к виду:

$$\frac{\partial \vec{V}}{\partial t} + \frac{1}{2} \nabla \vec{V}^2 + [\text{rot} \vec{V}, \vec{V}] = \vec{F} - \frac{1}{\rho} \nabla p, \quad (1.36)$$

где $\text{rot} \vec{V} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ V_x & V_y & V_z \end{vmatrix} = \left(\frac{\partial V_z}{\partial y} - \frac{\partial V_y}{\partial z} \right) \cdot \vec{i} - \left(\frac{\partial V_z}{\partial x} - \frac{\partial V_x}{\partial z} \right) \cdot \vec{j} + \left(\frac{\partial V_y}{\partial x} - \frac{\partial V_x}{\partial y} \right) \cdot \vec{k}$. Уравнение (1.36)

дополняется членом, учитывающим вязкость воздуха, ведущую к явлению турбулентного обмена, а именно, $\frac{\mu}{\rho} \nabla^2 \vec{V}$, μ - динамический коэффициент вязкости [кг/м·с]. Помимо

μ может быть использована и величина $\nu = \frac{\mu}{\rho(P, t)}$ - кинематический коэффициент вязкости, [м²/с]. В результате аналитических преобразований [38], получим уравнение:

$$\frac{\partial \vec{V}}{\partial t} + A[\vec{V}] \vec{V} - \frac{\mu}{\rho} \nabla^2 \vec{V} = \vec{F} - \frac{1}{\rho} \nabla p - \frac{1}{2} \nabla \vec{V}^2 = \vec{F} - \frac{1}{\rho} \nabla \left(p + \frac{\rho}{2} \vec{V}^2 \right). \quad (1.37)$$

Заметим, что решением задачи (1.37) являются не только три функции V_x, V_y, V_z в точках $\{x_l, y_m, z_k, t_j\}$ заданной сетки, но и матрица $\left\{ \frac{\partial V_i}{\partial x}, \frac{\partial V_i}{\partial y}, \frac{\partial V_i}{\partial z} \right\}$, $i = 1, 2, 3$. В рассматриваемой здесь модели эта матрица имеет самостоятельный содержательный смысл, поскольку именно с ее помощью может осуществляться исследование влияния коэффициентов турбулентного обмена на характеристики поля скорости ветра. В работах [38, 59] приводятся выводы и обоснования рассмотренных выше аналитических выражений, составляющих структуру соответствующих моделей, а также численные методы и алгоритмы их программной реализации, результаты соответствующих вычислительных экспериментов.

В заключение приведем еще одну математическую модель, связанную с исследованием поля \vec{V} в пределах пограничного слоя атмосферы. Соответствующая система дифференциальных уравнений записывается в виде:

$$\begin{aligned} \frac{\partial V_1}{\partial t} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + l \cdot V_2 + \tilde{\Delta} V_1 + F_1, & \frac{\partial V_2}{\partial t} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} - l \cdot V_1 + \tilde{\Delta} V_2 + F_2, \\ \frac{\partial V_3}{\partial t} &= -\frac{1}{\rho} \frac{\partial p}{\partial z} + \tilde{\Delta} V_3 - g, & \operatorname{div} \vec{V} &= 0, \quad p = R \cdot \rho \cdot T, \end{aligned} \quad (1.38)$$

l - константа, $[1/c]$, порядка 10^{-4} [38], она предназначена для моделирования мезо- и крупномасштабных процессов типа перенос-диффузия в пограничном слое атмосферы. Математическая модель (1.38) по физическому содержанию существенно отлична от рассмотренных выше. Простейшим вариантом подобной системы является система:

$$\begin{cases} \frac{\partial V_1}{\partial t} = f \cdot V_2 + \frac{\partial}{\partial z} K \frac{\partial V_1}{\partial z} \\ \frac{\partial V_2}{\partial t} = -f \cdot (V_1 - G) + \frac{\partial}{\partial z} K \frac{\partial V_2}{\partial z} \end{cases},$$

G - скорость геофизического ветра вдоль оси \vec{Ox} , K - коэффициент турбулентного обмена и f - коэффициент Кориолиса, $[1/c]$. В этой модели результирующая структура поля определяется эффектом турбулентного «размывания» исходного потока воздуха, представленного внешним ветровым полем G . В этой модели источником движения воздуха является поле G , в отличие от предыдущих моделей. Учет термодинамических эффектов приводит к более сложной системе. Не трудно видеть, что система (1.38) является линейной и потому более простой как в анализе, так и в численной реализации. В

задачах моделирования поля скорости ветра \vec{V} могут использоваться практически все перечисленные выше модели. Действительно, если \vec{V} представить в виде суммы полей \vec{V}_{cp} и $\vec{V}_{лок}$, где первое поле соответствует крупномасштабным полям, а вторая компонента описывает вариации в пределах локальных объемов исследуемой среды, то для \vec{V}_{cp} потребуется модель (1.38), а для $\vec{V}_{лок}$ - первая из описанных моделей.

Оценка коэффициента турбулентной диффузии с учетом полуэмпирических моделей. Известно, что значение коэффициента турбулентной диффузии в пограничном слое атмосферы зависит помимо физических характеристик среды и от скорости движения воздушных масс [7, 8, 38]. Известна, например, полуэмпирическая формула:

$$K = cL^2 \left[\left(\frac{\partial V_x}{\partial x} \right)^2 + \left(\frac{\partial V_y}{\partial x} \right)^2 + \left(\frac{\partial V_z}{\partial x} \right)^2 + \left(\frac{\partial V_x}{\partial z} \right)^2 + \left(\frac{\partial V_y}{\partial z} \right)^2 + \left(\frac{\partial V_z}{\partial z} \right)^2 \right]^{1/2}, \quad (1.39)$$

$L=50-299$ м, $c=0.41$, она предполагает горизонтальную однородность пограничного слоя,

то есть $V_y \approx V_x$ и $\frac{\partial V_y}{\partial x} \approx \frac{\partial V_x}{\partial x}$, $\frac{\partial V_y}{\partial z} \approx \frac{\partial V_x}{\partial z}$. Обобщением (1.39) в чисто физическом плане

служит выражение $K_i = L^2 \left(\sum_{j=1}^3 \eta_{ij} \left(\frac{\partial V_i}{\partial x_j} \right)^2 \right)^{1/2}$, $i=1,2,3$, L - константа масштаба, а $\{\eta_{ij}\}_{3 \times 3}$ -

матрица весовых коэффициентов, характеризующая структуру турбулентных движений в пределах исследуемого объема. Выбор весовой матрицы $\{\eta_{ij}\}_{3 \times 3}$ может быть осуществлен по результатам предварительных численных экспериментов, согласованных с типами турбулентных состояний пограничного слоя атмосферы. В работах [34, 38] рассмотрены и другие модели оценки значений коэффициента турбулентной диффузии, в частности, обратная коэффициентная задача.

Определение начальных и краевых условий на основе полуэмпирических моделей. Во всех рассмотренных выше моделях требуется задание начальных значений всех распределений, а также значения соответствующих краевых условий. Для решения этой задачи можно использовать «Методику расчета концентраций в атмосферном воздухе вредных веществ, содержащихся в выбросах предприятий» [33]. В этой работе приведены основные расчетные полуэмпирические формулы. Также для этой цели можно использовать решение полуэмпирического уравнения турбулентной диффузии. Рассмотрим данную задачу. Пусть $q(P,t)$ - функция, значения которой в момент времени t

в точке $P(x, y, z)$ совпадают со значениями мгновенной концентрации примеси, переносимой в атмосфере потоками воздуха. Предполагается, что функция $q(P, t)$ - непрерывно дифференцируема по x, y, z, t . Полуэмпирическое уравнение турбулентной диффузии записывают в виде [34, 63]:

$$\frac{\partial q}{\partial t} + V_x \frac{\partial q}{\partial x} + V_y \frac{\partial q}{\partial y} + V_z \frac{\partial q}{\partial z} = \frac{\partial}{\partial x} K_x \frac{\partial q}{\partial x} + \frac{\partial}{\partial y} K_y \frac{\partial q}{\partial y} + \frac{\partial}{\partial z} K_z \frac{\partial q}{\partial z} + S, \quad (1.40)$$

S - источник примеси, находящийся в точке $P_0(x_0, y_0, z_0)$ и производящий мгновенный выброс загрязняющих примесей в момент времени t_0 в количестве Q_0 . Для расчета средних концентраций примеси в пограничном слое атмосферы от мгновенного точечного источника используется решение уравнения (1.40):

$$q(x, y, z, t) = \frac{Q(x_0, y_0, z_0, t_0)}{[4\pi(t-t_0)]^{3/2} [K_x K_y K_z]^{1/2}} \exp\left\{-\frac{[(x-x_0)-V_x(t-t_0)]^2}{4K_x(t-t_0)}\right\} \times \\ \exp\left\{-\frac{[(y-y_0)-V_y(t-t_0)]^2}{4K_y(t-t_0)}\right\} \times \exp\left\{-\frac{[(z-z_0)-V_z(t-t_0)]^2}{4K_z(t-t_0)}\right\}, \quad (1.41)$$

называемое гауссовой функцией распределения концентрации примеси. Решение (1.41) получено методом функции Грина [61].

Выбор исходных данных в математической модели переноса примесей в атмосфере на основе данных натуральных экспериментов. В завершение обсудим вопрос об оценке вероятных значений основных метеорологических параметров и их диапазонов изменения. В работе [34] выполнен обзор соответствующих литературных источников, из них осуществлен выбор таблиц с экспериментальными данными, их систематизация и анализ. Ниже приводятся кратко результаты этих исследований. При определении диапазона возможных значений указанных величин необходимо иметь в виду, что атмосфера по своим физическим свойствам (температура, давление, плотность и т.д.) неоднородна как по вертикали, так и по горизонтали. Наиболее сильно эти свойства изменяются по высоте. По признаку изменения температуры по высоте атмосферу делят на пять слоев, одним из которых является тропосфера (от 0 до 11 км). Внутри данного слоя выделяют пограничный слой атмосферы высотой до 1 – 1,5 км, а внутри пограничного – приземный слой высотой до 50 – 100 м. В пограничном слое на движение воздушных масс значительное влияние оказывает подстилающая поверхность и силы турбулентного трения. В пределах приземного слоя, и в целом пограничного слоя резко с высотой из-

меняются температура, скорость ветра и турбулентность. Кроме этого необходимо также учитывать тип стратификации пограничного слоя атмосферы, определяемого значением параметра стратификации. Выделяют условия безразличной или равновесной стратификации, когда вертикальный поток тепла равен нулю, а изменение температуры воздуха с высотой происходит по адиабатическому закону. Учитывая небольшую вертикальную протяженность приземного слоя, можно говорить о равновесной стратификации и в тех случаях, когда температура мало меняется с высотой, в частности при изотермии. Неравновесная стратификация характеризуется температурными градиентами, существенно отличными от нуля. Условия со сверхадиабатическими градиентами температуры относят к неустойчивому состоянию атмосферы. В таких случаях стратификация способствует развитию случайных возмущений в воздушном потоке и усилению турбулентного обмена. Инверсионное распределение, связанное с ростом температуры с высотой, определяется как устойчивая стратификация. При наличии инверсии температуры возмущения в потоке воздуха гасятся, и интенсивность турбулентности значительно ослабляется. В работе [34] приведены экспериментальные значения скорости ветра и коэффициента турбулентной диффузии на разных высотах и при различных типах стратификации. Так, на высоте 100 м скорость ветра $V^* \approx 10$ м/с, а коэффициент турбулентной диффузии $K^* \approx 17.5$ м²/с; соответствующие диапазоны изменения на высоте от 50 до 500 м.: для скорости ветра $[V_{\min}, V_{\max}] \approx [3.7; 16.4]$ м/с при нейтральной стратификации, $[V_{\min}, V_{\max}] \approx [4.4; 13.2]$ м/с при неустойчивой стратификации и $[V_{\min}, V_{\max}] \approx [1.2; 13.7]$ м/с при устойчивой стратификации; для коэффициента турбулентной диффузии $[K_{\min}, K_{\max}] \approx [8; 56]$ м²/с (Приложение 2, таблицы 1 и 2). Важное значение при выборе величины коэффициента турбулентной диффузии имеет тип облачной системы, наблюдаемой в верхних областях пограничного слоя от 1 до 1,5 км и выше. Так, например, для слоисто-дождевой облачности (Ns-As) рекомендуемые значения коэффициента турбулентной диффузии K^* меняются в пределах $[K_{\min}, K_{\max}] \approx [2; 25]$ м²/с, а для кучево-дождевой облачности (Cb) - $[K_{\min}, K_{\max}] \approx [50; 1500]$ м²/с. Вычислить значения V^* , K^* можно также с помощью полуэмпирических формул [34]. Здесь же также приведены возможные значения параметра α^* , определяющего скорость осаждения частиц на высоте 100 м. При этом учитываются размеры частиц и вид примеси: легкая, средняя и тяжелая. Так, например, для осаждающейся примеси со средним размером частиц реко-

мендуемое значение $\alpha^* \approx 0.3 \cdot 10^{-2}$ (1/с), соответствующий диапазон изменения $[\alpha_{\min}; \alpha_{\max}] \approx [0.13 \cdot 10^{-5}; 0.13 \cdot 10^{-1}]$ (1/с) (Приложение 2, таблица 3). Концентрация вредных веществ в воздухе, в частности ПДК, меняется в пределах $[q_{\min}; q_{\max}] \approx [0.5 \cdot 10^{-11}; 0.5 \cdot 10^{-8}]$ (кг/м³). Здесь же, в [34] приведены значения интенсивности источников. Например, для одного из источников интенсивность выброса в атмосферу вредных веществ составила $S^* \approx 0.04 \cdot 10^{-3}$ (кг/(с·м³)), соответствующий диапазон изменения составляет $[S_{\min}; S_{\max}] \approx [0.25 \cdot 10^{-11}; 0.4 \cdot 10^{-3}]$ (кг/(с·м³)) (Приложение 2, таблица 4).

Вычислительная схема решения поставленной задачи. В итоге можно построить следующую систему уравнений и соответствующих задач, а также алгоритм для решения поставленной задачи:

Задача I. Определение начальных и краевых условий на основе полуэмпирических моделей и данных натурных экспериментов. Исходные данные для моделей: значения констант R, g, μ, c, L^2 ; начальные условия $V_i(P, t=0), i=1,2,3$ и $q(P, t=0), \rho(P, t=0), p(P, t=0)$; краевые условия $V_i(P, t), i=1,2,3, q(P, t), \rho(P, t), p(P, t)$ для $P \in \bar{\Omega}$; распределения $T(P, t), S(P, t), F_i(P, t); P(x, y, z) \in \Omega, t \in [t_0, t_1]$.

Задача II. Расчет значений распределений $\rho(P, t)$ и $p(P, t)$, а также компонент вектора скорости ветра $V_i(P, t), i=1,2,3$ на основе векторного уравнения Навье-Стокса, уравнения неразрывности и уравнения Клайперона:

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{V}) = 0, \quad p = f(\rho, T) = R \rho T, \quad \frac{\partial \vec{V}}{\partial t} + (\vec{V} \nabla) \vec{V} = -\frac{\nabla p}{\rho} + \vec{F}, \quad \vec{V} \cdot \nabla = V_x \frac{\partial}{\partial x} + V_y \frac{\partial}{\partial y} + V_z \frac{\partial}{\partial z}.$$

Задача III. Оценка значений поля турбулентности, определяемого коэффициентом турбулентной диффузии на основе полуэмпирической формулы:

$$K = cL^2 \left[\left(\frac{\partial V_x}{\partial x} \right)^2 + \left(\frac{\partial V_x}{\partial z} \right)^2 + \left(\frac{\partial V_z}{\partial x} \right)^2 + \left(\frac{\partial V_z}{\partial z} \right)^2 \right]^{1/2} \quad \text{или} \quad K_i = L^2 \left(\sum_{j=1}^3 \eta_{ij} \left(\frac{\partial V_i}{\partial x_j} \right)^2 \right)^{1/2}, \quad i=1,2,3.$$

Задача IV. Повторный расчет значений распределений $\rho(P, t)$ и $p(P, t)$, а также компонент вектора скорости ветра $V_i(P, t), i=1,2,3$ на основе векторного уравнения Навье-Стокса, уравнения неразрывности и уравнения Клайперона с учетом в них членов, определяемых турбулентным состоянием пограничного слоя атмосферы:

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{V}) = 0, \quad p = f(\rho, T) = R\rho T, \quad \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla) \vec{V} = -\frac{\nabla p}{\rho} + \tilde{\Delta} \vec{V} + \vec{F}, \quad \text{где } \tilde{\Delta} = \sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial}{\partial x_i} K_{i,j} \frac{\partial}{\partial x_j}.$$

Задача V. Вычисление концентрации примесей $q(P, t)$, переносимых в пределах пограничного слоя атмосферы на основе уравнения переноса $\dot{q} + \alpha q + \operatorname{div} \vec{J} = S$, $\vec{J} = \vec{V}q - \vec{K} \cdot \nabla q$.

Замечания:

1. Данная система является открытой, последовательность решения задач может быть иной. Представленные модели не единственные, возможны и другие модификации с учетом различных предположений, допущений и т.п., рассмотренных выше;
2. Для оценки значений температурного поля $T(x, y, z, t)$ в пределах области Ω можно рассмотреть задачу, основанную на решении уравнения притока тепла (1.5);
3. Совокупность рассмотренных задач представляет собой достаточно сложную многомерную систему, и ее реализация в полном объеме представляется достаточно затруднительной в вычислительном отношении. Целесообразно для начала решать задачи по отдельности.

1.5. Численные методы и алгоритмы в моделях нестационарного переноса загрязняющих примесей в приземном слое атмосферы

Метод покоординатного расщепления и вычислительный алгоритм для трехмерного уравнения переноса субстанции в воздушной среде. Задачу для нестационарной модели (1.20) - (1.22) можно представить в виде трех последовательно решаемых подзадач, соответствующих переносу субстанции вдоль координатных осей \vec{O}_z , \vec{O}_x и \vec{O}_y в пределах элементарного временного интервала $t \in [t_j, t_{j+1}]$. Содержание данного утверждения лежит в основе метода, получившего название метода покоординатного расщепления [34]. Величина $\Delta t_j = t_{j+1} - t_j$ считается «достаточно малой». Понятие «малости» определяется исходными данными, а именно, $\alpha(t)$, $V_x(P, t)$, $V_y(P, t)$, $V_z(P, t)$ и $K_x(P, t)$, $K_y(P, t)$, $K_z(P, t)$, $S(P, t)$, характеризующими динамическое состояние среды. Оценка допустимой величины Δt в данной задаче осуществляется в вычислительном эксперименте. В рамках теории метода покоординатного расщепления сформулировано следующее утверждение: процесс решения задачи (1.20) - (1.22) может быть представлен в виде трех последовательных подзадач [38]:

$$t_j \leq t \leq t_{j+1}$$

$$\text{Задача I:} \quad \dot{q}_1 + \alpha \cdot q_1 + \frac{\partial}{\partial z}(V_z \cdot q_1) - \frac{\partial}{\partial z}\left(K_z \cdot \frac{\partial q_1}{\partial z}\right) = \omega_1 \cdot S(P, t), \quad (1.42a)$$

$$P_1 = P(z|x, y), P_4 = P(0|x, y), P_5 = P(Z|x, y), q_1(P_1, t=0) = q_0(P_1), \text{ если } t = 0,$$

$$q_1(P, t_j) = q_3(P, t_{j+1}), \text{ если } t > 0, q_1(P_4, t) = \bar{q}_4(x, y, t), q_1(P_5, t) = \bar{q}_5(x, y, t); \quad (1.42b)$$

$$\text{Задача II:} \quad \dot{q}_2 + \frac{\partial}{\partial x}(V_x \cdot q_2) - \frac{\partial}{\partial x}\left(K_x \cdot \frac{\partial q_2}{\partial x}\right) = \omega_2 \cdot S(P, t), \quad (1.43a)$$

$$P_2 = P(x|y, z), P_6 = P(0|y, z), P_7 = P(X|y, z)$$

$$q_2(P, t_j) = q_1(P, t_{j+1}), q_2(P_6, t) = \bar{q}_6(y, z, t), q_2(P_7, t) = \bar{q}_6(y, z, t); \quad (1.43b)$$

$$\text{Задача III:} \quad \dot{q}_3 + \frac{\partial}{\partial y}(V_y \cdot q_3) - \frac{\partial}{\partial y}\left(K_y \cdot \frac{\partial q_3}{\partial y}\right) = \omega_3 \cdot S(P, t), \quad (1.44a)$$

$$P_3 = P(y|x, z), P_8 = P(0|x, z), P_9 = P(Y|x, z), q_3(P, t_j) = q_2(P, t_{j+1}),$$

$$q_3(P_8, t) = \bar{q}_8(x, z, t), q_3(P_9, t) = \bar{q}_9(x, z, t); \omega_1 + \omega_2 + \omega_3 = 1. \quad (1.44b)$$

В качестве решения (1.20) принимается $q(P, t) = q_3(P, t)$ в пределах $t_j \leq t \leq t_{j+1}$. Введенные коэффициенты $\omega_1, \omega_2, \omega_3$ удовлетворяют условию нормировки $\omega_1 + \omega_2 + \omega_3 = 1$. Данный вариант схемы осуществляет последовательное покоординатное расщепление задачи, начиная с направления \overrightarrow{Oz} , далее \overrightarrow{Ox} и, наконец, \overrightarrow{Oy} в пределах каждого временного интервала $t_j \leq t \leq t_{j+1}$. Такая последовательность обусловлена наибольшей пространственно-временной изменчивостью характеристик пограничного слоя атмосферы в направлении вертикальной оси. Строгое обоснование вычислительного алгоритма (1.42) – (1.44) приведено в работе [38].

Методы параметризации вычислительных алгоритмов в задачах численного решения дифференциальных многомерных уравнений. Рассмотрим одномерный вариант задачи (1.20) – (1.22) [38]:

$$\frac{\partial q(x, t)}{\partial t} + \alpha(t) \cdot q(x, t) + \frac{\partial}{\partial x}(V(x, t) \cdot q(x, t)) - \frac{\partial}{\partial x}\left(K(x, t) \cdot \frac{\partial q(x, t)}{\partial x}\right) = S(x, t). \quad (1.45)$$

$$q(x, t=0) = q_0(x), q(x=0, t) = q_1(t), q(x=1, t) = q_2(t). \quad (1.46)$$

Если рассматривать уравнение (1.45) как математическую модель переноса дисперсных загрязнений в пограничном слое атмосферы, то следует особое внимание уделить вопросам обеспечения этого уравнения исходными данными и тем трудностям, ко-

торые при этом возникают. Первое состоит в том, что если поле скорости ветра $V(x, t)$ в принципе может быть определено, скажем в некоторой точке (x_k, t_j) , то поле коэффициента турбулентной диффузии $K(x, t)$ следует считать неизвестным. Иными словами, функцию $V(x, t)$ можно считать эмпирически вычисляемой, и исходить из того, что в вычислениях она может быть всегда представлена множеством значений $\{V(x_k, t_j)\}$, $k = \overline{0, K}$, $j = \overline{0, N}$, известных в пределах средней ошибки σ_V . Что касается соответствующих значений функции $K(x, t)$, то в лучшем случае они могут быть определены с использованием полуэмпирических формул, которые связывают в среднем наблюдаемые пространственно-временные вариации компонент скорости ветра со средними значениями $K(x, t)$ в пределах локальных объемов исследуемой среды. Таким образом, применение уравнения (1.45) к задачам переноса загрязнений в пограничном слое атмосферы сталкивается с определенными трудностями и, как следствие, недостаточной определенностью для того, чтобы гарантировать прогнозы пространственно-временных вариаций поля концентрации загрязнений. К этому следует добавить ряд других моментов, усугубляющих проблему обеспечения исходными данными расчеты с использованием решений уравнения переноса. Речь идет о том, что в (1.45) помимо $V(x, t)$ и $K(x, t)$ входят также и их частные производные $(V(x, t))'_x$ и $(K(x, t))'_x$. Что касается определения $(V(x, t))'_x$, то в принципе эта задача решается, если $V(x, t)$ представлена множеством приближенных данных $\{V(x_k, t_j)\}$, правда при использовании соответствующих расчетных методик. Оценка $(K(x, t))'_x$ заметно сложнее, поскольку операция дифференцирования полуэмпирических формул почти бессодержательна в соответствии с теорией аппроксимации функций. Указанные трудности в задачах моделирования нестационарного процесса диффузного переноса часто преодолеваются в рамках качественного подхода, основанного на введении турбулентных состояний пограничного слоя. Опуская физическое обоснование подобной классификации [22, 73], укажем лишь на то, что с формальной точки зрения речь идет об ограничениях на $V(x, t)$ и $K(x, t)$ типа:

$$V_1^{(l)} \leq V(x, t) \leq V_2^{(l)}, \quad K_1^{(l)} \leq K(x, t) \leq K_2^{(l)}, \quad l = 1, 2, \quad (1.47)$$

$V_1^{(l)}$, $V_2^{(l)}$, $K_1^{(l)}$, $K_2^{(l)}$ - некоторые числа, определяющие границы изменения полей $V(x, t)$, $K(x, t)$ и зависящие от типа состояния (индекса l). В пределах указанных границ можно

говорить о существовании определенной функциональной зависимости между $V(x,t)$ и $K(x,t)$, представленной той или иной полуэмпирической формулой в которую естественно входят и физические характеристики пограничного слоя атмосферы. Ясно, что говорить о неких средних значениях \bar{V} и \bar{K} можно в пределах определенного типа состояния пограничного слоя. Помимо рассмотренной проблемы, связанной с практическим применением уравнения переноса (1.45), не менее сложным является соответствующий выбор интервалов $[0, X]$ и $[0, T]$. Понятно, что эти интервалы связаны с заданием начальных и граничных условий (1.46). Ошибки задания функций $q_0(x)$, $\bar{q}_0(t)$ и $\bar{q}_X(t)$ естественно влекут ошибки в $q(x,t)$, получаемой при решении уравнения (1.45). В исследованиях чисто теоретического характера принято полагать в качестве указанных интервалов следующие $(-\infty, +\infty)$ и $(0, +\infty)$. В соответствии с этим указанные выше функции считаются тождественно равными нулю, за возможным исключением первой из них. Для стационарных задач подобное допущение возможно и приемлемо, однако для задач оперативного прогноза динамики поля загрязнений в атмосфере подобный подход необоснован. В связи с этим в пределах настоящего исследования выбор значений X и T будет считаться некой самостоятельной задачей теории моделирования явления переноса. В более широком смысле выбор масштабов изменения пространственной и временной переменных следует рассматривать как выделение некоторого локального «объема» исследуемой среды с учетом конкретных особенностей решаемой задачи. Подобная задача адекватно отвечает ситуации, связанной с аварийными выбросами загрязнений в атмосферу и оперативного прогноза его распространения в пределах пограничного слоя атмосферы. Одновременно следует иметь в виду, что задача выбора величин X и T тесно связана с оценкой двух важных характеристик качества получаемой информации о поле концентрации, это $(\Delta x)_{\min}$ и $(\Delta t)_{\min}$. Первая из них определяет «пространственное разрешение» при исследовании поля концентрации загрязнений, а вторая – «временное разрешение». Эти характеристики следует считать определяющими независимо от методов исследования пространственно-временной изменчивости физических полей, каким в частности является поле концентрации $q(x,t)$. Замечание касается не только эмпирических методов исследования, но и расчетно-аналитических, в последнем случае особенно при наличии неопределенности задания исходных данных. В любом случае необходима оценка погрешности получаемого результата.

Неопределенность исходной задачи требует построения адекватной параметризованной модели данного физического явления с последующей оптимизацией получаемого в рамках этой модели результата. В работе [38] этот подход условно назван «методом параметризованных моделей» и детально разрабатывается на примере задач теории переноса субстанции в турбулентных средах. Анализ исходных данных для уравнения (1.45) с начальными и граничными условиями (1.46) показывает, что все они имеют сильный разброс в масштабах единиц измерения. Для приведения исходных величин к единому масштабу применяется процедура перемасштабирования или построения параметризованной вычислительной модели. Пусть $V_{\min} \leq V(x,t) \leq V_{\max}$, разделим на V_{\max} :

$$\frac{V_{\min}}{V_{\max}} \leq \frac{V(x,t)}{V_{\max}} \leq \frac{V_{\max}}{V_{\max}}. \text{ Обозначим } V(x,t)/V_{\max} = \hat{V}(x,t), \text{ тогда } V(x,t) = \hat{V}(x,t) \cdot V_{\max}. \text{ Отсюда}$$

очевидно, что $0 \leq \hat{V}(x,t) \leq 1$. Аналогичным образом нормируем распределение коэффициента турбулентной диффузии: $0 \leq \hat{K}(x,t) \leq 1$. Нормируя распределение концентрации загрязняющих веществ, необходимо учесть, что $q(x,t)$ нам неизвестно, а заданы лишь начальные условия $q(x,t=0) = q_0(x)$ и граничные условия $q(x=0,t) = q_1(t)$, $q(x=X,t) = q_2(t)$. Поэтому, в качестве q_{\max} можно взять $q^* = \max_x q_0(x)$, например

$q_{\max} = q^*$. Далее воспользуемся аналогичным неравенством $q_{\min} \leq q(x,t) \leq q_{\max}$, в котором положим $q(x,t)/q_{\max} = \hat{q}(x,t)$, откуда следует $q(x,t) = \hat{q}(x,t) \cdot q_{\max}$. Таким образом, получаем $0 \leq \hat{q}(x,t) \leq 1$. Нормировать можно и переменные задачи: $\hat{x} = \frac{x-x_0}{X-x_0}$, $\hat{t} = \frac{t-t_0}{T-t_0}$.

Если здесь будем полагать, что $x_0 = 0$, $t_0 = 0$, то последние формулы упростятся: $\hat{x} = x/X$, $\hat{t} = t/T$. Описанная процедура позволяет получить все распределения исходного уравнения, изменяющимися в диапазоне $[0; 1]$. Подставим нормированные значения:

$$q(x,t) = q_{\max} \cdot \hat{q}(x,t), \quad V(x,t) = V_{\max} \cdot \hat{V}(x,t), \quad K(x,t) = K_{\max} \cdot \hat{K}(x,t), \quad x = X \cdot \hat{x}, \quad t = T \cdot \hat{t} \quad (1.48)$$

в (1.45), после несложных преобразований получим параметризованную модель уравнения переноса:

$$\frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{t}} + \alpha \cdot \hat{q}(\hat{x}, \hat{t}) + \beta \cdot \frac{\partial}{\partial \hat{x}} (\hat{V}(\hat{x}, \hat{t}) \cdot \hat{q}(\hat{x}, \hat{t})) - \theta \cdot \frac{\partial}{\partial \hat{x}} \left(\hat{K}(\hat{x}, \hat{t}) \cdot \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}} \right) = \xi \cdot \hat{S}(\hat{x}, \hat{t}) \quad (1.49)$$

где нормировочные коэффициенты $\beta = V_{\max} \cdot T / X$, $\theta = K_{\max} \cdot T / X^2$, $\xi = S_{\max} \cdot T / q_{\max}$ являются уже безразмерными параметрами. Убедимся в этом:

$$\beta = \frac{V_{\max} \cdot T}{X} = \frac{\frac{\text{м}}{\text{с}} \cdot \text{с}}{\text{м}} = 1, \quad \theta = \frac{K_{\max} \cdot T}{X^2} = \frac{\frac{\text{м}^2}{\text{с}} \cdot \text{с}}{\text{м}^2} = 1, \quad \xi = \frac{S_{\max} \cdot T}{q_{\max}} = \frac{\frac{\text{кг}}{\text{м}^3} \cdot \text{с}}{\frac{\text{кг}}{\text{м}^3}} = 1.$$

Обозначим $\alpha(t) = \alpha^* \cdot T \quad \forall t \in [0; T]$, где $\alpha^* - \text{const}$, например, максимальное значение, аналогично выясним размерность $\left[\frac{1}{\text{с}} \cdot \text{с} \right] = 1$. Таким образом, полученное уравнение является параметрическим, поскольку теперь зависит от параметров $\alpha(t)$, β , θ , ξ , которые связывают между собой исходные данные V_{\max} , K_{\max} , q_{\max} и S_{\max} . Так как значения величин V_{\max} , α^* , T , X , K_{\max} , q_{\max} , S_{\max} положительны по своему физическому смыслу, то и безразмерные параметры α , β , θ , ξ также принимают только положительные значения и должны выбираться так, чтобы левая и правая части уравнения (1.49) были положительны.

Вычислительный алгоритм решения параметризованного уравнения переноса на основе конечно-разностных методов. Выполним построение вычислительного алгоритма на основе конечно – разностного метода [38]. Далее будем полагать, что $x = \hat{x}$, $t = \hat{t}$, $q(x, t) = \hat{q}(\hat{x}, \hat{t})$, $V(x, t) = \hat{V}(\hat{x}, \hat{t})$, $K(x, t) = \hat{K}(\hat{x}, \hat{t})$, $S(x, t) = \hat{S}(\hat{x}, \hat{t})$, $\alpha(t) = \hat{\alpha}(\hat{t})$. Построение вычислительного алгоритма для исходной задачи осуществляется с введения сетки узлов $\{(x_i, t_j)\}$ на интервалах $x \in [0; 1]$, $t \in [0; 1]$, $i = \overline{0, m}$, $j = \overline{0, n}$ и задания шаблона, то есть множества узлов точек сетки, участвующих в аппроксимации дифференциального выражения. Шаг сетки при этом определяется так: $\{x_i\}$, $i = \overline{0, m}$, $x_i = a + \Delta x \cdot i$, $\Delta x = (b - a) / m$; $\{t_j\}$, $j = \overline{0, n}$, $t_j = c + \Delta t \cdot j$, $\Delta t = (d - c) / n$. Для аппроксимации первых производных полей исходных данных в точке (x_i, t_j) используем центральную разность:

$$\frac{\partial \varphi}{\partial x} \approx \frac{\varphi_{i+1, j} - \varphi_{i-1, j}}{2\Delta x}, \text{ порядок аппроксимации в этой схеме } O(\Delta x^2). \text{ Для аппроксимации}$$

второй производной используем шеститочечную схему [39]:

$$\frac{\partial^2 \varphi}{\partial x^2} \approx \frac{1}{2} \cdot \left(\frac{\varphi_{i+1, j+1} - 2\varphi_{i, j+1} + \varphi_{i-1, j+1}}{(\Delta x)^2} + \frac{\varphi_{i+1, j} - 2\varphi_{i, j} + \varphi_{i-1, j}}{(\Delta x)^2} \right), \text{ с погрешностью } O(\Delta x^2 + \Delta t^2).$$

Распределения $V(x, t)$, $K(x, t)$, $S(x, t)$ и $q(x, t)$ заменим приближенными значениями се-

точных функций $V(x_i, t_j) = V_{i,j}$, $K(x_i, t_j) = K_{i,j}$, $S(x_i, t_j) = S_{i,j}$ и $q(x_i, t_j) = q_{i,j}$, в итоге разностное уравнение для внутренних узлов сетки примет вид:

$$\begin{aligned} \frac{q_{i,j+1} - q_{i,j}}{\Delta t} + \alpha q_{i,j} + \beta \frac{V_{i+1,j} - V_{i-1,j}}{2\Delta x} \cdot q_{i,j} + \beta V_{i,j} \cdot \frac{q_{i+1,j} - q_{i-1,j}}{2\Delta x} - \theta \frac{K_{i+1,j} - K_{i-1,j}}{2\Delta x} \cdot \frac{q_{i+1,j} - q_{i-1,j}}{2\Delta x} - \\ - \frac{\theta K_{i,j}}{2} \left(\frac{q_{i+1,j+1} - 2q_{i,j+1} + q_{i-1,j+1}}{\Delta x^2} + \frac{q_{i+1,j} - 2q_{i,j} + q_{i-1,j}}{\Delta x^2} \right) = \xi S_{i,j}, \end{aligned} \quad (1.50)$$

$i = \overline{1, m-1}$, $j = \overline{0, n-1}$; начальные распределения $q_{i,0} = q(x_i) = q_i$, $q_{0,j} = q_1(t_j) = q_{1,j}$, $q_{m,j} = q_2(t_j) = q_{2,j}$ известны из условий (1.46). Преобразование (1.50) и с учетом обозначения $\lambda = \Delta t / \Delta x^2$ приводит к уравнению:

$$\begin{aligned} -\frac{\lambda \theta}{2} \cdot K_{i,j} \cdot q_{i-1,j+1} + (1 + \lambda \theta K_{i,j}) \cdot q_{i,j+1} - \frac{\lambda \theta}{2} \cdot K_{i,j} \cdot q_{i+1,j+1} = \\ = \xi \Delta t \cdot S_{i,j} + \left(\frac{\lambda \beta}{2} \cdot \Delta x \cdot V_{i,j} + \frac{\lambda \theta}{4} \cdot (K_{i-1,j} - K_{i+1,j}) + \frac{\lambda \theta}{2} \cdot K_{i,j} \right) \cdot q_{i-1,j} + \\ + \left(1 - \alpha \Delta t - \frac{\lambda \beta}{2} \Delta x (V_{i+1,j} - V_{i-1,j}) - \lambda \theta \cdot K_{i,j} \right) q_{i,j} + \left(\frac{\lambda \theta}{2} K_{i,j} + \frac{\lambda \theta}{4} (K_{i+1,j} - K_{i-1,j}) - \frac{\lambda \beta}{2} \Delta x V_{i,j} \right) q_{i+1,j}. \end{aligned} \quad (1.51)$$

Вычислительная схема (1.51) есть совокупность разностных уравнений, аппроксимирующих дифференциальное уравнение (1.49) во внутренних узлах сетки и представляет собой систему линейных алгебраических уравнений с трехдиагональной симметричной матрицей, для численного решения которой в работе применялся метод прогонки [38, 61]. С целью использования метода прогонки систему (1.51) перепишем так:

$$a_l y_{l-1} + b_l y_l + c_l y_{l+1} = f_l, \quad (1.52)$$

$l = \overline{1, m-1}$; $j = \overline{0, n-1}$ и коэффициенты системы получают представление:

$$\begin{aligned} a_l = -0.5\lambda\theta K_{i,j}, \quad b_l = 1 + \lambda\theta K_{i,j}, \quad c_l = -0.5\lambda\theta K_{i,j}, \\ f_l = \xi \Delta t \cdot S_{i,j} + \left(\frac{\lambda \beta}{2} \cdot \Delta x \cdot V_{i,j} + \frac{\lambda \theta}{4} \cdot (K_{i-1,j} - K_{i+1,j}) + \frac{\lambda \theta}{2} \cdot K_{i,j} \right) \cdot q_{i-1,j} + \\ + \left(1 - \alpha \Delta t - \frac{\lambda \beta}{2} \Delta x (V_{i+1,j} - V_{i-1,j}) - \lambda \theta K_{i,j} \right) q_{i,j} + \left(\frac{\lambda \theta}{2} K_{i,j} + \frac{\lambda \theta}{4} (K_{i+1,j} - K_{i-1,j}) - \frac{\lambda \beta}{2} \Delta x V_{i,j} \right) q_{i+1,j}, \\ k = m-1; \quad y_0 = q_{0,j}, \quad y_l = q_{i,j}; \quad l = \overline{1, k}, \quad y_{k+1} = y_m = q_{m,j}. \end{aligned} \quad (1.53)$$

Алгоритм метода прогонки включает в себя два этапа: 1) прямая прогонка - определение коэффициентов $\{A_l\}$ и $\{B_l\}$: $d_1 = f_1 - a_1 y_0$, $d_l = f_l$, $l = \overline{2, k-1}$, $d_k = f_k - c_k y_{k+1}$, $A_1 = -c_1/b_1$, $B_1 = d_1/b_1$, $A_l = -c_l/(a_l A_{l-1} + b_l)$, $B_l = (d_l - a_l B_{l-1})/(a_l A_{l-1} + b_l)$, $l = \overline{2, k}$; 2) обратная

прогонка - нахождение $\{y_l\}$: $y_k = A_k y_{k+1} + B_k$; $y_l = A_l y_{l+1} + B_l$, $l = \overline{k-1, 1}$. Для возможности применения метода прогонки необходимо и достаточно, чтобы коэффициенты системы удовлетворяли условиям:

$$a_l \neq 0 \text{ и } b_l \neq 0, \quad (1.54)$$

$$|b_l| \geq |a_l| + |c_l|, \quad l = \overline{1, m}. \quad (1.55)$$

Выполнение (1.54) и (1.55) гарантирует существование и единственность решения системы (1.52) и возможность его нахождения методом прогонки. Выполнение условий (1.54) определяется исходными данными, значения которых всегда положительны по их физическому смыслу. Эти условия накладывают дополнительные ограничения на выбор исходных данных, которые не могут задаваться произвольно, и более того, их значения взаимосвязаны. Поэтому, выбор значения λ и значений исходных данных должен быть осуществлен в вычислительном эксперименте. Условие (1.55) фактически означает следующее:

$$|1 + \lambda \theta \cdot K_{i,j}| \geq |-0.5 \cdot \lambda \theta \cdot K_{i,j}| + |-0.5 \cdot \lambda \theta \cdot K_{i,j}| \Rightarrow 2 \cdot \lambda \theta \cdot K_{i,j} \geq 1.$$

Данное условие обеспечено соответствующим выбором λ из интервала $0 < \lambda < 1$, так как значения $\theta = K_{\max} \cdot T / X^2$ и $K_{i,j}$ в последнем неравенстве также положительны по своему физическому содержанию.

Итерационные алгоритмы для решения нестационарных уравнений переноса субстанции в воздушной среде. Далее рассмотрим построение итерационного алгоритма, соответствующего интегральному представлению решения краевой задачи для уравнения диффузного переноса [38]. Вычислительный метод, лежащий в основе данных алгоритмов, условно назван «методом интегральных уравнений». Его применение осуществляется для параметризованных моделей уравнения переноса, построенных ранее. Вычислительные алгоритмы, основанные на конечно-разностных методах, позволяют учитывать влияние производных q'_t , q'_x , q''_{xx} , V'_x , K'_x , входящих в одномерное уравнение переноса. Может показаться, что учет указанных производных в конечно-разностных вычислительных схемах осуществляется относительно просто. В то же время удовлетворить известному из теории условию $\lambda = \Delta t / \Delta x^2 < 1$ при $\Delta t \rightarrow 0$ и $\Delta x \rightarrow 0$ на практике совсем не просто. Необходимость корректного вычисления производных V'_x и K'_x усугубляет трудности этого подхода. Становится ясно, что из вычислительных схем целе-

сообразно «исключить» q''_{xx} , а также по возможности V'_x и K'_x . При нахождении решения $q(x,t)$ в численном алгоритме желательно оставить лишь q'_t и q'_x . В этом случае требование к сходимости можно ослабить: $\Delta t/\Delta x \approx O(1)$ при $\Delta t \rightarrow 0$ и $\Delta x \rightarrow 0$.

Данный вычислительный метод, используемый в работе, основан на предварительном интегрировании дифференциального уравнения в частных производных по временной переменной и сведению решения задачи к решению интегро-дифференциальных уравнений, в которых определяющую роль играет интегральный оператор Вольтерра второго рода. Все это позволяет вводить в структуру решающих алгоритмов итерационные вычислительные схемы на основе метода последовательных приближений. При этом хорошо известно, что для операторов Вольтерра второго рода метод последовательных приближений сходится при определенных условиях. Преобразование исходного дифференциального уравнения в интегральное позволяет строить в теории переноса интегральные параметризованные модели. Опираясь на эти модели, в работе выполнено построение и исследование итерационных вычислительных схем и алгоритмов, проведено доказательство их сходимости, приведены результаты вычислительного эксперимента, проводимого на основе тестовой задачи. Развитие и изучение этого метода можно найти в работах [36, 39, 86]. Вернемся к рассмотрению уравнения переноса (1.49), и перепишем его следующим образом:

$$\frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{t}} + \hat{q}(\hat{x}, \hat{t}) \left\{ \alpha + \beta \frac{\partial \hat{V}(\hat{x}, \hat{t})}{\partial \hat{x}} \right\} + \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}} \left\{ \beta \hat{V}(\hat{x}, \hat{t}) - \theta \frac{\partial \hat{K}(\hat{x}, \hat{t})}{\partial \hat{x}} \right\} - \frac{\partial^2 \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}^2} \{ \theta \hat{K}(\hat{x}, \hat{t}) \} = \xi \hat{S}(\hat{x}, \hat{t}), \quad (1.56)$$

здесь как и ранее $x = \hat{x}$, $t = \hat{t}$, $q(x,t) = \hat{q}(\hat{x}, \hat{t})$, $V(x,t) = \hat{V}(\hat{x}, \hat{t})$, $K(x,t) = \hat{K}(\hat{x}, \hat{t})$, $S(x,t) = \hat{S}(\hat{x}, \hat{t})$, $\alpha(t) = \alpha(\hat{t})$ - нормированные величины, значения которых принимают значения из интервала $[0; 1]$, нормировочные безразмерные коэффициенты α , β , θ , ξ вычисляются согласно описанной выше методике. Уравнение переноса, как известно, при численных оценках поля $q(x,t)$ предполагает наличие начальных и граничных условий:

$$q(x, t = 0) = q_0(x), \quad q(x = 0, t) = q_1(t), \quad q(x = 1, t) = q_2(t), \quad (1.57)$$

Для уравнения (1.56) введем следующие обозначения:

$$p(\hat{x}, \hat{t}) = \alpha + \beta \frac{\partial \hat{V}(\hat{x}, \hat{t})}{\partial \hat{x}}, \quad Q(\hat{x}, \hat{t}) = \psi(\hat{x}, \hat{t}) - \xi \hat{S}(\hat{x}, \hat{t}),$$

$$\psi(\hat{x}, \hat{t}) = \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}} \left\{ \beta \hat{V}(\hat{x}, \hat{t}) - \theta \frac{\partial \hat{K}(\hat{x}, \hat{t})}{\partial \hat{x}} \right\} - \frac{\partial^2 \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}^2} \{ \theta \hat{K}(\hat{x}, \hat{t}) \}, \quad (1.58)$$

что дает возможность записать его в виде:

$$\dot{q}(\hat{x}, \hat{t}) + p(\hat{x}, \hat{t}) \cdot q(\hat{x}, \hat{t}) + Q(\hat{x}, \hat{t}) = 0. \quad (1.59)$$

Уравнение (1.59) рассматривается как дифференциальное уравнение первого порядка по переменной t при фиксированном значении переменной x . Известно, что для подобного вида уравнений существует аналитическое решение, которое запишем [13]:

$$q(t) = \exp \left\{ - \int_{t_0}^t p(u) du \right\} \left[q_0 - \int_{t_0}^t Q(v) \exp \left\{ \int_{t_0}^v p(u) du \right\} dv \right]. \quad (1.60)$$

Переходя к обозначениям (1.58) получим следующее интегральное представление:

$$q(x, t) = \exp \left\{ - \int_{t_0}^t \left(\alpha(t') + \beta \cdot \frac{\partial V(x, t')}{\partial x} \right) dt' \right\} \times \\ \times \left[q(x, t_0) - \int_{t_0}^t (\psi(x, t') - \xi \cdot S(x, t')) \cdot \exp \left\{ \int_{t_0}^{t'} \left(\alpha(t'') + \beta \cdot \frac{\partial V(x, t'')}{\partial x} \right) dt'' \right\} dt' \right]. \quad (1.61)$$

Выполняя преобразования (1.61) и вводя обозначения:

$$\tilde{K}(x, t, t') = \exp \left\{ - \int_{t'}^t \left(\alpha(t'') + \beta \cdot \frac{\partial V(x, t'')}{\partial x} \right) dt'' \right\}, \quad (1.62)$$

$$\varphi(x, t) = q(x, t_0) \cdot \exp \left\{ - \int_{t_0}^t \left(\alpha(t') + \beta \cdot \frac{\partial V(x, t')}{\partial x} \right) dt' \right\} + \int_{t_0}^t \xi \cdot S(x, t') \cdot \tilde{K}(x, t, t') dt'. \quad (1.63)$$

окончательно для функции $q(x, t)$ будем иметь:

$$q(x, t) = \varphi(x, t) - \int_{t_0}^t \tilde{K}(x, t, t') \psi(x, t') dt'. \quad (1.64)$$

Выражение (1.64) можно рассматривать как интегральное уравнение относительно функции $q(x, t)$. При построении метода численного решения (1.64) относительно искомой функции $q(x, t)$, удобно ввести функции $q(t|x)$, где $x \in [0; 1]$ и играет роль параметра. Тогда для каждого фиксированного x интегральное уравнение относительно $q(t|x)$ является интегральным уравнением Вольтерра второго рода, численное решение которого можно осуществить методом последовательных приближений. Данный метод реализуется в виде следующей итерационной схемы:

$$q^{(v)}(t|x) = \varphi(t|x) - \int_{t_0}^t \tilde{K}(t, t'|x) \psi(x, t', q^{(v-1)}(t'|x)) dt', \quad (1.65)$$

ν - номер итерации. Для вычисления значений выражения $\psi(x, t, q(t|x))$ на множестве функций $q(t|x)$ необходима аппроксимация производных $\frac{\partial q(x, t)}{\partial x}$, $\frac{\partial^2 q(x, t)}{\partial x^2}$. С этой целью введем равномерную сетку узлов $\{x_i\}$ на интервале $x \in [0, 1]$, $i = \overline{0, m}$. Шаг сетки при этом определяется так: $\Delta x = 1/m$. Внутренние узлы сетки - $\{x_i\}$, $i = \overline{1, m-1}$. Внешние узлы сетки - $\{x_0 = 0\}$, $\{x_m = 1\}$. Далее будем полагать, что на каждом шаге итерации ν вычисляется последовательность $q_k^{(\nu)}(t) = q^{(\nu)}(t|x_k)$, $k = \overline{0, m}$. Это позволяет для каждого ν воспользоваться следующими конечно-разностными аппроксимациями для производных $\frac{\partial q(t|x)}{\partial x}$ и

$$\frac{\partial^2 q(t|x)}{\partial x^2} : \left(\frac{\partial q(t|x)}{\partial x} \right)^{(\nu)} = \frac{q_k^{(\nu)}(t) - q_{k-1}^{(\nu)}(t)}{\Delta x}, \quad \left(\frac{\partial^2 q(t|x)}{\partial x^2} \right)^{(\nu)} = \frac{q_{k+1}^{(\nu)}(t) - 2q_k^{(\nu)}(t) + q_{k-1}^{(\nu)}(t)}{\Delta x^2}, \quad \text{где } k = \overline{1, m-1}.$$

При этом $q_0^{(\nu)}(t) = q(t|x_0 = 0)$ и $q_m^{(\nu)}(t) = q(t|x_m = 1)$ определены граничными условиями (1.57) для исходного уравнения (1.56). В итоге имеем формулу для расчета значений функции $\psi(x, t, q(x, t))$ в точке $x = x_k$:

$$\begin{aligned} \psi(x_k, t, q^{(\nu)}(t|x_k)) = & \left(\beta \cdot V(t|x_k) - \theta \cdot \frac{K(t|x_k) - K(t|x_{k-1})}{\Delta x} \right) \cdot \frac{q^{(\nu)}(t|x_k) - q^{(\nu)}(t|x_{k-1})}{\Delta x} - \\ & - \theta \cdot K(t|x_k) \cdot \frac{q^{(\nu)}(t|x_{k+1}) - 2q^{(\nu)}(t|x_k) + q^{(\nu)}(t|x_{k-1})}{\Delta x^2}. \end{aligned} \quad (1.66)$$

Таким образом, вычислительный процесс для итерационной схемы (1.65) полностью определен и его результатом является последовательность функций $\{q_k^{(\nu)}(t)\}$, $\nu = 1, 2, \dots$, $k = \overline{1, m-1}$. Отметим одно обстоятельство: интегральное уравнение (1.64) эквивалентно в математическом смысле исходному уравнению (1.56). Вместе с тем оно интересно в плане физических интерпретаций. Действительно, ядро $\tilde{K}(x, t, t')$ для двух моментов времени t и t' в каждой точке x выступает как интегральная характеристика временной изменчивости процесса переноса. Как показывает формула (1.62), по мере увеличения разности $(t - t')$, ядро $\tilde{K}(x, t, t')$ экспоненциально убывает. Обратим внимание на тот факт, что размерность величины $[\alpha(t) + V'_x(t|x)]$ есть $(1/c)$ и обратная к ней величина определяет время рассматриваемого физического процесса. Далее заменим функции $q^{(\nu)}(t|x)$, $\tilde{K}(t, t'|x)$, $\varphi(t|x)$ и $\psi(x_k, t, q^{(\nu)}(t|x_k))$ сеточными функциями $q^{(\nu)}(x_k, t_j) = q_{k,j}^{(\nu)}$,

$\tilde{K}(x_k, t_i, t_j) = \tilde{K}_{k,i,j}$, $\varphi(x_k, t_j) = \varphi_{k,j}$ и $\psi^{(\nu)}(x_k, t_j) = \psi_{k,j}^{(\nu)}$, где (x_k, t_j) - внутренние узлы равномерной сетки $k = \overline{1, m-1}$, $j = \overline{1, n}$, $\Delta x = 1/m$, $\Delta t = 1/n$, $\lambda = \Delta t / \Delta x^2$; $x_k = k \cdot \Delta x$, $k = \overline{0, m}$, $t_j = j \cdot \Delta t$, $j = \overline{0, n}$. Начальными условиями определяются значения $q^{(\nu)}(x_k, t_0 = 0)$, $k = \overline{0, m}$; граничными условиями - $q^{(\nu)}(x_0 = 0, t_j)$ и $q^{(\nu)}(x_m = 1, t_j)$, $j = \overline{0, n}$. Тогда (1.62), (1.63), (1.65) и (1.66) получают следующее представление:

$$\tilde{K}_{k,i,j} = \exp \left\{ -\lambda \sum_{l=i}^j \omega_l \Delta x (\alpha_l \Delta x + \beta (V_{k,l} - V_{k-1,l})) \right\} = \tilde{K}_{k,l,j}(\lambda), \quad (1.67)$$

$$\varphi_{k,j} = q_{k,0} \exp \left\{ -\lambda \sum_{i=0}^j \omega_i (\alpha_i \Delta x^2 + \beta \Delta x (V_{k,i} - V_{k-1,i})) \right\} + \xi \lambda \sum_{i=0}^j \omega_i S_{k,i} \tilde{K}_{k,i,j} \Delta x^2 = \varphi_{k,j}(\lambda), \quad (1.68)$$

$$\psi_{k,i}^{(\nu-1)} = \frac{1}{\Delta x^2} (\beta \Delta x V_{k,i} - \theta (K_{k,i} - K_{k-1,i})) \cdot (q_{k,i}^{(\nu-1)} - q_{k-1,i}^{(\nu-1)}) - \frac{\theta K_{k,i}}{\Delta x^2} (q_{k+1,i}^{(\nu-1)} - 2q_{k,i}^{(\nu-1)} + q_{k-1,i}^{(\nu-1)}) = \frac{\tilde{\psi}_{k,i}^{(\nu-1)}}{\Delta x^2}, \quad (1.69)$$

$$q_{k,j}^{(\nu)} = \varphi_{k,j}(\lambda) - \lambda \sum_{i=0}^j \omega_i \tilde{K}_{k,i,j}(\lambda) \tilde{\psi}_{k,i}^{(\nu-1)}, \quad (1.70)$$

$\nu = 0, 1, 2, \dots$, $q_{k,j}^{(0)} = \varphi_{k,j}$, $k = \overline{1, m-1}$, $j = \overline{1, n}$, интегралы в (1.62), (1.63), (1.65) и (1.66) приближенно заменены интегральными суммами с квадратурными коэффициентами $\{\omega_i\}$, $i = \overline{0, j}$, $j = \overline{1, n}$ и $\lambda = \Delta t / \Delta x^2$. Проверяется условие сходимости:

$$\rho < \varepsilon, \quad (1.71)$$

$$\rho = \frac{1}{n \cdot (m-1)} \sum_{k=1}^m \sum_{j=1}^n |q_{k,j}^{(\nu)} - q_{k,j}^{(\nu-1)}|, \quad (1.72)$$

если $\rho < \varepsilon$, то $\tilde{q}_{k,j} = q_{k,j}^{(\nu)}$, в противном случае $\nu = \nu + 1$. Сравнение полученного приближенного решения \tilde{q} с точным решением q_T выполняется по формуле:

$$\sigma = \frac{1}{n \cdot (m-1)} \sum_{k=1}^m \sum_{j=1}^n |(q_T)_{k,j} - \tilde{q}_{k,j}|. \quad (1.73)$$

В завершение обсудим вопрос сходимости итерационной схемы [38]. Уравнение переноса (1.45) представим в виде:

$$\dot{q} + \alpha q + \frac{\partial}{\partial x} J = S, \quad (1.74)$$

J - поток субстанции в направлении оси $O\vec{x}$, тогда для (1.74) $J = Vq - Kq'$. В трехмерном случае исходное уравнение переноса с привлечением потока \vec{J} следует писать в виде:

$$\left. \begin{aligned} \dot{q} + \alpha q + \operatorname{div} \vec{J} &= S, \\ \vec{J} &= \vec{V}q - \vec{K} \nabla q \end{aligned} \right\}, \quad (1.75)$$

\vec{K} - тензор, характеризующий турбулентные характеристики среды, $S(P, t)$ - функция источника субстанции ($P \in \Omega$, $t \in [t_0, T]$). В уравнении (1.75) функции $V(P, t)$, компоненты тензора $K_{ij}(P, t)$, $\alpha(t)$ считаются заданными. Требуется определить поток $\vec{J}(P, t)$ и скалярное поле $q(P, t)$. Уравнение (1.74) является дифференциальным уравнением первого порядка по переменной t и допускает интегральное представление для $q(P, t)$:

$$q(x, t) = \int_{t_0}^t F(x, t', q) \exp(-\alpha(t-t')) dt' + \varphi(x, t, S), \quad (1.76)$$

$$F(x, t, q) = \frac{\partial}{\partial x} \left(Vq - K \frac{\partial q}{\partial x} \right), \quad (1.77)$$

$$\varphi(x, t, S) = q(x, t_0) \exp(-\alpha(t-t_0)) - \int_{t_0}^t F(x, t', q) \exp(-\alpha(t-t')) dt'. \quad (1.78)$$

Выражение (1.77) можно представить как $(D_2 q)(x, t)$, D_2 - дифференциальный оператор второго порядка по переменной x , либо $(D_1 J)(x, t)$, D_1 дифференциальный оператор первого порядка. Система (1.76) – (1.78) справедлива $\forall x \in [x_0, X]$ и может рассматриваться как линейное интегральное уравнение Вольтерра второго рода относительно неизвестного распределения $q(t|x)$. Для его решения воспользуемся методом последовательных приближений, который приводит к следующей итерационной схеме:

$$\left\{ \begin{aligned} q^{(v)} &= K \psi^{(v-1)} + \varphi \\ \psi^{(v)} &= D_2 q^{(v)}, v = 1, \dots \end{aligned} \right., \quad (1.79)$$

K - интегральный оператор, соответствующий (1.76), а вспомогательная переменная связана с потоком J соотношением $\psi = D_1 J$. Вводя величины $\Delta_q^{(v, v-1)} = q^{(v)} - q^{(v-1)}$ и $\Delta_\psi^{(v, v-1)} = \psi^{(v)} - \psi^{(v-1)}$, систему (1.79) перепишем в виде:

$$\left\{ \begin{aligned} \Delta_q^{(v, v-1)} &= K \Delta_q^{(v-1, v-2)}, \\ \Delta_\psi^{(v)} &= D_2 \Delta_q^{(v, v-1)}, v = 1, \dots \end{aligned} \right. \quad (1.80)$$

тогда условие сходимости итерационной схемы (1.80) примет вид:

$$\|K\| \cdot \|D_2\| < 1. \quad (1.81)$$

Вводя сетку узлов $\{x_i, t_j\}_{n \times m}$ в области $[x_0, X] \times [t_0, T]$ можно линейные операторы K и D_2 редуцировать к соответствующим матрицам \hat{K} и \hat{D}_2 , и тогда условие (1.81) есть ограничение на исходные данные $\{V(x_i, t_j)\}_{n \times m}$ и $\{K(x_i, t_j)\}_{n \times m}$. Для задачи переноса это очень важно, поскольку матрица $\{K(x_i, t_j)\}_{n \times m}$ почти всегда вводится на основе соответствующих априорных моделей, и, следовательно, требуется соблюдать определенную осторожность.

Выводы по главе 1

1. На основе обзора и анализа научных публикаций по данной проблеме рассмотрена общая характеристика проблемы охраны окружающей среды в системе атмосфера – подстилающая поверхность, изложены основные понятия теории экологического мониторинга.
2. Изучены физические основы математической модели пограничного слоя атмосферы, приведен обзор известных гидродинамических моделей.
3. Выполнена постановка задачи моделирования нестационарного переноса примесей в пограничном слое атмосферы и определение исходных данных на основе сопутствующих математических моделей.
4. Изучены численные методы и алгоритмы в моделях нестационарного переноса загрязняющих примесей в приземном слое атмосферы, методы параметризации исходных математических моделей, организации вычислительного эксперимента.

Глава 2. Численная реализация итерационных алгоритмов и исследование моделей переноса загрязняющих веществ в атмосфере. Методы качественной интерпретации расчетных данных

В главе выполняется создание алгоритмического и программного обеспечения для одномерного параметризованного уравнения переноса загрязняющих примесей в атмосфере. Методика моделирования, а именно, конечно-разностные вычислительные схемы, так называемый «метод интегральных уравнений», метод построения параметризованной модели переноса на примере одномерного уравнения, выбор и обоснование исходных данных в модели рассмотрены в предыдущей главе, и служат основой для разработки в пределах настоящей главы алгоритма и программного обеспечения вычислительного эксперимента. На основе этого в главе далее выполняется вычислительный эксперимент, для которого специально разрабатывается тестовый пример, позволяющий провести численные исследования сходимости приближенных решений к точным, а также устойчивости приближенных решений к погрешностям в исходных данных. Получены результаты вычислительного эксперимента и на их основе выполнено сопоставление вычислительных алгоритмов и сформулированы необходимые рекомендации.

Далее развивается подход, согласно которому на основе итерационно-рекурсивного метода построены расчетно-аналитические качественные модели и соответствующие алгоритмы для оценки концентрации примесей. Основная цель при этом состоит в разработке методик качественной оценки значений параметров в задачах математического моделирования явления переноса в турбулентных средах. Это в свою очередь позволяет в какой-то мере решить проблему неопределенности некоторых исходных данных, кроме того позволяет получать простые аналитические решения исходной задачи, что упрощает задачу при построении алгоритмов. Полученные результаты моделирования затем могут быть использованы в других более сложных вычислительных схемах, как некие предварительные исходные данные. Также упрощается предварительный анализ получаемых результатов. Во второй части главы подробно излагаются этапы построения качественных моделей, выполняется их алгоритмизация и программирование, реализуется предварительно разработанная методика вычислительного эксперимента применительно к этим моделям, получены результаты расчетов и численных исследований, исследован вопрос сходимости вычислительных алгоритмов.

2.1 Разработка методик исследования физических явлений на моделях.

Выбор исходных данных

Как показано в п. 1.4, нестационарное параметризованное уравнение переноса примесей для одной пространственной переменной с заданными начальными и граничными условиями имеет вид:

$$\frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{t}} + \alpha \cdot \hat{q}(\hat{x}, \hat{t}) + \beta \cdot \frac{\partial}{\partial \hat{x}} (\hat{V}(\hat{x}, \hat{t}) \cdot \hat{q}(\hat{x}, \hat{t})) - \theta \cdot \frac{\partial}{\partial \hat{x}} \left(\hat{K}(\hat{x}, \hat{t}) \cdot \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}} \right) = \xi \cdot \hat{S}(\hat{x}, \hat{t}), \quad (2.1)$$

$$q(x, t=0) = q_0(x), \quad q(x=0, t) = q_1(t), \quad q(x=1, t) = q_2(t). \quad (2.2)$$

Основные вычислительные формулы конечно – разностного алгоритма решения задачи (2.1) – (2.2) в п. 1.5 получены в виде:

$$\begin{aligned} & -\frac{\lambda\theta}{2} \cdot K_{i,j} \cdot q_{i-1,j+1} + (1 + \lambda\theta K_{i,j}) \cdot q_{i,j+1} - \frac{\lambda\theta}{2} \cdot K_{i,j} \cdot q_{i+1,j+1} = \\ & = \xi\Delta t \cdot S_{i,j} + \left(\frac{\lambda\beta}{2} \cdot \Delta x \cdot V_{i,j} + \frac{\lambda\theta}{4} \cdot (K_{i-1,j} - K_{i+1,j}) + \frac{\lambda\theta}{2} \cdot K_{i,j} \right) \cdot q_{i-1,j} + \\ & + \left(1 - \alpha\Delta t - \frac{\lambda\beta}{2} \Delta x (V_{i+1,j} - V_{i-1,j}) - \lambda\theta \cdot K_{i,j} \right) q_{i,j} + \left(\frac{\lambda\theta}{2} K_{i,j} + \frac{\lambda\theta}{4} (K_{i+1,j} - K_{i-1,j}) - \frac{\lambda\beta}{2} \Delta x V_{i,j} \right) q_{i+1,j}. \end{aligned} \quad (2.3)$$

Основные вычислительные формулы итерационного алгоритма в п.1.5 записаны следующим образом:

$$\begin{aligned} \tilde{K}_{k,i,j} &= \exp \left\{ -\lambda \sum_{l=i}^j \omega_l \Delta x (\alpha_l \Delta x + \beta (V_{k,l} - V_{k-1,l})) \right\} = \tilde{K}_{k,l,j}(\lambda), \\ \varphi_{k,j} &= q_{k,0} \exp \left\{ -\lambda \sum_{i=0}^j \omega_i (\alpha_i \Delta x^2 + \beta \Delta x (V_{k,i} - V_{k-1,i})) \right\} + \xi \lambda \sum_{i=0}^j \omega_i S_{k,i} \tilde{K}_{k,i,j} \Delta x^2 = \varphi_{k,j}(\lambda), \\ \psi_{k,i}^{(v-1)} &= \frac{1}{\Delta x^2} (\beta \Delta x V_{k,i} - \theta (K_{k,i} - K_{k-1,i})) \cdot (q_{k,i}^{(v-1)} - q_{k-1,i}^{(v-1)}) - \frac{\theta K_{k,i}}{\Delta x^2} (q_{k+1,i}^{(v-1)} - 2q_{k,i}^{(v-1)} + q_{k-1,i}^{(v-1)}) = \frac{\tilde{\psi}_{k,i}^{(v-1)}}{\Delta x^2}, \\ q_{k,j}^{(v)} &= \varphi_{k,j}(\lambda) - \lambda \sum_{i=0}^j \omega_i \tilde{K}_{k,i,j}(\lambda) \tilde{\psi}_{k,i}^{(v-1)}, \end{aligned} \quad (2.4)$$

Условие выхода из итерационного вычислительного процесса формулируется так: согласно формуле:

$$\rho = \frac{1}{n \cdot (m-1)} \sum_{k=1}^m \sum_{j=1}^n |q_{k,j}^{(v)} - q_{k,j}^{(v-1)}| \quad (2.5)$$

вычисляется значение ρ и проверяется условие: если $\rho < \varepsilon$, то $\tilde{q}_{i,j} = q_{i,j}^{(\nu)}$, в противном случае $\nu = \nu + 1$.

Для реализации и исследования описанных выше вычислительных методов (п. 1.5) требуются исходные данные, генерируемые входящим в общую схему вычислительного алгоритма «Блоком исходных данных». Данный блок обеспечивает формирование и хранение исходных данных в соответствии с алгоритмом. Под исходными данными подразумеваются массивы значений всех переменных и распределений уравнения (2.1). Помимо формирования этих массивов, сюда также включаются процедуры нормировки исходных данных и вычисления значений их производных. Для проверки качества работы вычислительной схемы и программного комплекса далее описаны аналитические модели генерации значений полей исходных данных, которые используются в качестве тестовых примеров. Основная идея построения тестовых примеров состоит в следующем. Дискретные (измеренные экспериментально) значения исходных распределений уравнения (2.1) – концентрацию загрязняющих веществ, скорость ветра, коэффициент турбулентной диффузии аппроксимируем непрерывными функциями.

Моделирование распределений V_T , K_T , q_T и вычисление S_T . В настоящей работе для моделирования пространственно-временной распределенности исходных данных $V(x,t)$ - поля скорости ветра, $K(x,t)$ - поля турбулентной диффузии, $q(x,t)$ - поля концентрации примесей, переносимых в атмосфере, предлагается выбирать функции вида:

$$\begin{aligned}\hat{V}(\hat{x}, \hat{t}) &= V_0 \cdot (1 + av \cdot \sin(bv \cdot \hat{x} + cv)) \cdot (1 + dv \cdot \sin(rv \cdot \hat{t} + gv)), \\ \hat{K}(\hat{x}, \hat{t}) &= K_0 \cdot (1 + ak \cdot \sin(bk \cdot \hat{x} + ck)) \cdot (1 + dk \cdot \cos(rk \cdot \hat{t} + gk)), \\ \hat{q}(\hat{x}, \hat{t}) &= q_0 \cdot (1 + aq \cdot \cos(bq \cdot \hat{x} + cq)) \cdot (1 + dq \cdot \cos(rq \cdot \hat{t} + gq)),\end{aligned}\tag{2.6}$$

произвольные коэффициенты av , bv , cv , dv , rv , gv , ak , bk , ck , dk , rk , gk , aq , bq , cq , dq , rq , gq выбираются в вычислительном эксперименте таким образом, чтобы определяемая далее функция источника $S(x,t)$ загрязняющих веществ принимала только положительные значения, а константы V_0 , K_0 , q_0 считаются заданными и выбираются из таблиц (Приложение 2) в соответствии с теми или иными состояниями атмосферы, о чем подробно говорилось в главе 1, п. 1.1. Так как, $V_0 \in [V_{\min}; V_{\max}]$ (п. 1.4), то можно положить, например, что $V_0 = (V_{\min} + V_{\max})/2$. То же касается и K_0 , q_0 . В выражениях (2.6) для коэффициентов функции скорости ветра были выбраны следующие значения:

$av = 0.8$, $bv = 7$, $cv = 1.5$, $dv = 0.75$, $rv = 2$, $gv = 1.5$; для коэффициентов функции турбулентной диффузии: $ak = 0.75$, $bk = 7$, $ck = 3$, $dk = 0.75$, $rk = 7$, $gk = 3$; для коэффициентов функции источника загрязняющих примесей: $aq = 0.075$, $bq = -5$, $cq = 5$, $dq = -1$, $rq = 2.75$, $gq = 0.1$. Далее функциям (2.6) ставились в соответствие сеточные функции:

$$\hat{V}(\hat{x}_i, \hat{t}_j) = V_0 \cdot (1 + av \cdot \sin(bv \cdot \hat{x}_i + cv)) \cdot (1 + dv \cdot \sin(rv \cdot \hat{t}_j + gv)),$$

$$\hat{K}(\hat{x}_i, \hat{t}_j) = K_0 \cdot (1 + ak \cdot \sin(bk \cdot \hat{x}_i + ck)) \cdot (1 + dk \cdot \cos(rk \cdot \hat{t}_j + gk)),$$

$$\hat{q}(\hat{x}_i, \hat{t}_j) = q_0 \cdot (1 + aq \cdot \cos(bq \cdot \hat{x}_i + cq)) \cdot (1 + dq \cdot \cos(rq \cdot \hat{t}_j + gq)),$$

что позволило сформировать массивы дискретных значений, соответствующих распределениям исходных данных. На их основе построены профили соответствующих распределений, так на рисунке 2.1 приведено пространственно-временное распределение поля скорости ветра $\hat{V}(\hat{x}, \hat{t})$, на рисунке 2.2 – поля турбулентной диффузии $\hat{K}(\hat{x}, \hat{t})$, на рисунке 2.3 – поля концентрации загрязняющих веществ $\hat{q}(\hat{x}, \hat{t})$. Можно сделать вывод о том, что функции вида (2.6) позволяют варьировать значения констант V_0 , K_0 , q_0 и получать положительные значения для всех распределений.

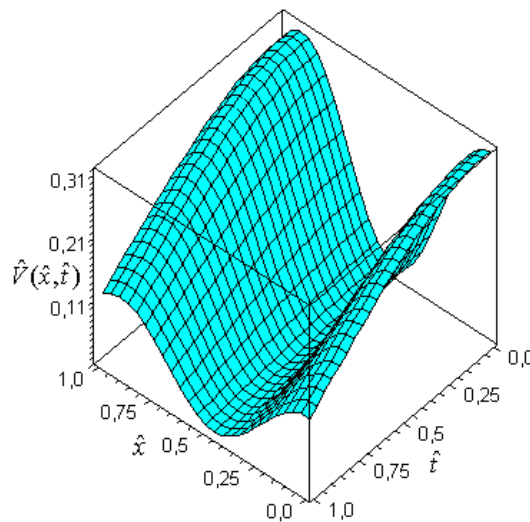


Рисунок 2.1 – Пространственно - временное распределение (профиль) поля скорости ветра $\hat{V}(\hat{x}, \hat{t})$ при $V_0 = 10$ м/с, значения $\hat{V}(\hat{x}, \hat{t})$, \hat{x} и \hat{t} - нормированные

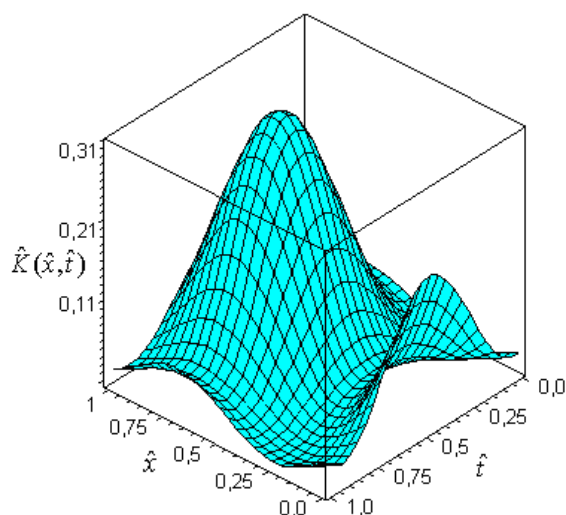


Рисунок 2.2 – Пространственно - временное распределение (профиль) поля турбулентной диффузии $\hat{K}(\hat{x}, \hat{t})$ при $K_0 = 100 \text{ м}^2/\text{с}$, значения \hat{x} и \hat{t} - нормированные

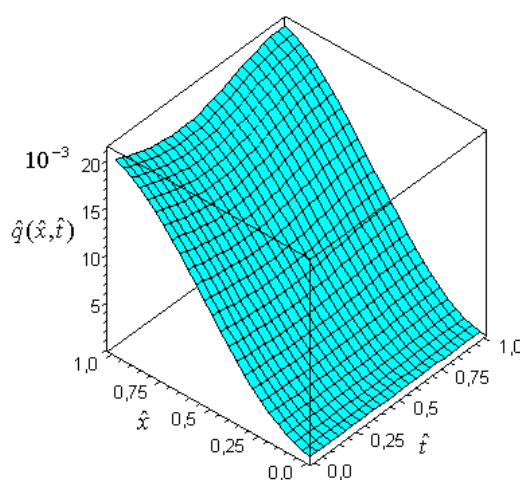


Рисунок 2.3 – Пространственно - временное распределение (профиль) поля концентрации загрязняющих веществ $\hat{q}(\hat{x}, \hat{t})$ при $q_0 = 0.001 \text{ кг/м}^3$, значения \hat{x} и \hat{t} - нормированные

Для определения массива значений функции источника $\hat{S}(\hat{x}_i, \hat{t}_j) = \hat{S}_{i,j}$, решаем уравнение (2.1) относительно $\hat{S}(\hat{x}, \hat{t})$, подставляя в него выражения (2.6). Кроме того, также вычисляем значения производных, входящих в уравнение переноса (2.1). Производные находим, непосредственно дифференцируя функции (2.6):

$$\frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{t}} = -(1 + aq \cdot \cos(bq \cdot \hat{x} + cq)) \cdot dq \cdot \sin(rq \cdot \hat{t} + gq) \cdot rq,$$

$$\frac{\partial}{\partial \hat{x}} (\hat{V}(\hat{x}, \hat{t}) \cdot \hat{q}(\hat{x}, \hat{t})) = av \cdot \cos(bv \cdot \hat{x} + cv) \cdot bv \cdot (1 + dv \cdot \sin(rv \cdot \hat{t} + gv)) \cdot$$

$$\cdot (1 + aq \cdot \cos(bq \cdot \hat{x} + cq)) \cdot (1 + dq \cdot \cos(rq \cdot \hat{t} + gq)) - (1 + av \cdot \sin(bv \cdot \hat{x} + cv)) \cdot$$

$$\cdot (1 + dv \cdot \sin(rv \cdot \hat{t} + gv)) \cdot aq \cdot \sin(bq \cdot \hat{x} + cq) \cdot bq \cdot (1 + dq \cdot \cos(rq \cdot \hat{t} + gq)),$$

$$\begin{aligned} \frac{\partial}{\partial \hat{x}} \left(\hat{K}(\hat{x}, \hat{t}) \cdot \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}} \right) = & -ak \cdot \cos(bk \cdot \hat{x} + ck) \cdot bk \cdot (1 + dk \cdot \cos(rk \cdot \hat{t} + gk)) \cdot \\ & aq \cdot \sin(bq \cdot \hat{x} + cq) \cdot bq \cdot (1 + dq \cdot \cos(rq \cdot \hat{t} + gq)) - (1 + ak \cdot \sin(bk \cdot \hat{x} + ck)) \cdot \\ & \cdot (1 + dk \cdot \cos(rk \cdot \hat{t} + gk)) \cdot aq \cdot \cos(bq \cdot \hat{x} + cq) \cdot bq^2 \cdot (1 + dq \cdot \cos(rq \cdot \hat{t} + gq)). \end{aligned}$$

Окончательно, значения функции источника определяем по формуле

$$\hat{S}(\hat{x}, \hat{t}) = \frac{\hat{Q}(\hat{x}, \hat{t}) \cdot q_{\max}}{S_{\max} \cdot T}, \quad (2.7)$$

$S_{\max} = q_{\max} / \xi \cdot T$, $\hat{Q}(\hat{x}, \hat{t})$ - левая часть (2.1):

$$\begin{aligned} \hat{Q}(\hat{x}, \hat{t}) = & 2.75(1 + 0.075 \cdot \cos(5\hat{x} - 5)) \cdot \sin(2.75 \cdot \hat{t} + 0.1) + 0.1 \cdot \\ & \cdot (1 + 0.075 \cos(5\hat{x} - 5)) \cdot (1 - \cos(2.75\hat{t} + 0.1)) + 0.56 \cos(7\hat{x} + 1.5) \cdot \\ & \cdot (1 + 0.75 \cdot \sin(2\hat{t} + 1.5)) \cdot (1 + 0.075 \cos(5\hat{x} - 5)) \cdot (1 - \cos(2.75\hat{t} + 0.1)) - \\ & - 0.0375 \cdot (1 + 0.8 \sin(7\hat{x} + 1.5)) \cdot (1 + 0.75 \sin(2\hat{t} + 1.5)) \cdot \sin(5\hat{x} - 5) \cdot \\ & \cdot (1 - \cos(2.75\hat{t} + 0.1)) + 0.00196875 \cos(7\hat{x} + 3) \cdot (1 + 0.75 \cos(7\hat{t} + 3)) \cdot \\ & \cdot \sin(5\hat{x} - 5) \cdot (1 - \cos(2.75\hat{t} + 0.1)) + 0.001875 \cdot (1 + 0.75 \sin(7\hat{x} + 3)) \cdot \\ & \cdot (1 + 0.75 \cos(7\hat{t} + 3)) \cdot \cos(5\hat{x} - 5) \cdot (1 - \cos(2.75\hat{t} + 0.1)). \end{aligned}$$

В результате получен массив значений функции $\hat{S}(\hat{x}_i, \hat{t}_j) = \hat{S}_{i,j}$, пространственно-временное распределение которого показано на рисунке 2.4. Таким образом, массивы значений всех исходных данных - $\{\hat{x}_i\}$, $\{\hat{t}_j\}$, $\{\hat{q}_{i,j}\}$, $\{\hat{v}_{i,j}\}$, $\{\hat{K}_{i,j}\}$, $\{\hat{S}_{i,j}\}$, $j = \overline{0, n}$, $i = \overline{0, m}$ получены и вычислены значения нормировочных констант q_{\max} , V_{\max} , K_{\max} и S_{\max} .

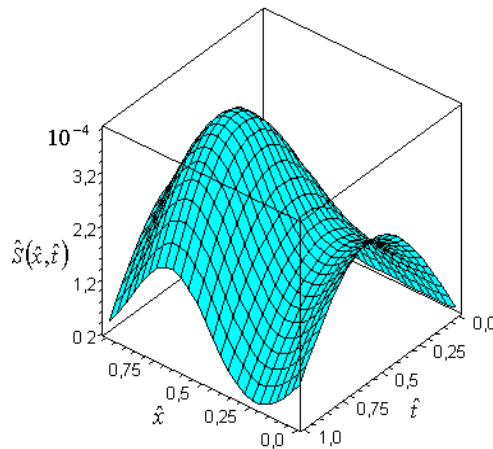


Рисунок 2.4 – Пространственно - временное распределение (профиль) поля источника загрязняющих веществ $\hat{S}(\hat{x}, \hat{t})$ при $S_0 = 0.0001 \text{ кг}/(\text{м}^3 \cdot \text{с})$, значения \hat{x} и \hat{t} - нормированные

Данную процедуру моделирования исходных данных можно представить следующим алгоритмом:

I. Блок исходных данных

1. Задаются значения $[a; b]$, $[c; d]$.
2. Задаются значения λ и m .
3. По заданным значениям в пунктах 1 и 2 считаются значения Δx , Δt , n .
4. Формируются массивы исходных данных $\{\hat{x}_i\}$, $\{\hat{t}_j\}$, $\{\hat{q}_{i,j}\} = \{\hat{q}_{i,j}\}_{toch}$, $\{\hat{V}_{i,j}\}$, $\{\hat{K}_{i,j}\}$, $\{\hat{S}_{i,j}\}$, $j = \overline{0, n}$, $i = \overline{0, m}$ и вычисляются значения нормировочных констант q_{\max} , V_{\max} , K_{\max} и S_{\max} .
5. Задаются начальные условия $\{\hat{q}_{i,0}\}$, $i = \overline{0, m}$.
6. Задаются краевые условия $\{\hat{q}_{0,j}\}$ и $\{\hat{q}_{m,j}\}$, $j = \overline{0, n}$.
7. Задаются значения α , β , θ , ξ , посчитанные ранее.
8. Вычисляются значения функции источника $\hat{S}(\hat{x}, \hat{t})$ по формуле (2.7).

2.2 Анализ данных моделирования пространственно-временной изменчивости поля концентрации загрязнений на основе предложенных численных методов

Далее приступим к реализации процедур «Блока моделирования» алгоритма вычислительной модели, а именно к нахождению неизвестных значений $\{\hat{q}_{i,j}\}$, $j = \overline{1, n}$, $i = \overline{1, m-1}$ при известных значениях $\{\hat{x}_i\}$, $\{\hat{t}_j\}$, $\{\hat{V}_{i,j}\}$, $\{\hat{K}_{i,j}\}$, $\{\hat{S}_{i,j}\}$, $j = \overline{0, n}$, $i = \overline{0, m}$; также известными считаются значения $\{\hat{q}_{i,0}\}$, $i = \overline{0, m}$ - начальные условия и граничные условия - $\{\hat{q}_{0,j}\}$ и $\{\hat{q}_{m,j}\}$, $j = \overline{0, n}$.

II. Блок моделирования

1. Рассчитываются значения $\tilde{q} = \{\tilde{q}_{i,j}\}$ для $i = \overline{1, m-1}$, $j = \overline{1, n}$ по формулам (2.3) с использованием метода прогонки (п. 1.5) для каждого временного слоя j (алгоритм на основе конечно – разностного метода) или по формулам (2.4), (2.5) (алгоритм на основе «метода интегральных уравнений»).
2. Выполняется сравнение полученного приближенного решения уравнения $\tilde{q} = \{\tilde{q}_{i,j}\}$ с точным решением по формуле

$$\sigma = \frac{1}{n} \sum_{j=1}^n \left(\frac{1}{m} \sum_{i=1}^{m-1} |\tilde{q}_{i,j} - (q_T)_{i,j}| \right) / \frac{1}{n} \sum_{j=1}^n \left(\frac{1}{m} \sum_{i=1}^{m-1} |(q_T)_{i,j}| \right), \quad (2.8)$$

итоговое значение погрешности определяется по формуле $\sigma = 1/n \sum_j \sigma_j$, расчет значений

$\tilde{q} = \{\tilde{q}_{i,j}\}$ осуществляется до того момента, пока $\sigma \leq \varepsilon$.

Программная реализация вычислительного алгоритма, соответствующего сеточным моделям (п. 1.5) проводилась при следующих значениях исходных данных: по заданным значениям $V_0 = 10$ м/с, $K_0 = 100$ м²/с, $q_0 = 0.001$ кг/м³, $S_0 = 0.0001$ кг/(м³·с), $\alpha = 0.1$, $\beta = 0.1$, $\theta = 0.001$, $\xi = 1$, $\lambda = 0.5$, вычисляются значения $\Delta x = 0.05$, $\Delta t = 0.00125$, $m = 20$, $n = 800$, $X = 972$ м, $T \approx 3.1$ с, $V_{\max} \approx 31.5$ м/с, $K_{\max} \approx 306$ м²/с, $q_{\max} \approx 0.002$ кг/м³, $S_{\max} \approx 0.000181$ кг/(м³·с). Значения $V_0 = 10$ м/с, $K_0 = 100$ м²/с, $q_0 = 0.001$ кг/м³, $S_0 = 0.0001$ кг/(м³·с) выбирались из таблиц 1 – 8 Приложения 2 и соответствуют следующим характеристикам состояния атмосферы: нейтральная стратификация, высота источника $Z = 500$ м, тяжелая примесь, слоисто-кучевая-дождевая облачность (Ns-As-Cb).

Далее приведены результаты расчетов по алгоритму, определяемому соотношениями (2.3), в виде массивов – массив $q_T = \{\hat{q}_T(i, j)\}$, $i = \overline{1, m, 2}$, $j = \overline{1, n, 80}$, который представляет собой точное решение уравнения переноса:

$$q_T = \begin{pmatrix} 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.006 \\ 0.071 & 0.068 & 0.066 & 0.065 & 0.064 & 0.065 & 0.067 & 0.069 & 0.072 & 0.074 & 0.075 \\ 0.208 & 0.200 & 0.194 & 0.189 & 0.189 & 0.192 & 0.198 & 0.205 & 0.212 & 0.217 & 0.219 \\ 0.407 & 0.392 & 0.379 & 0.370 & 0.369 & 0.374 & 0.386 & 0.400 & 0.414 & 0.424 & 0.428 \\ 0.651 & 0.628 & 0.606 & 0.593 & 0.590 & 0.599 & 0.618 & 0.641 & 0.664 & 0.679 & 0.686 \\ 0.924 & 0.890 & 0.860 & 0.841 & 0.837 & 0.850 & 0.876 & 0.909 & 0.941 & 0.964 & 0.972 \\ 1.203 & 1.159 & 1.121 & 1.096 & 1.091 & 1.108 & 1.142 & 1.185 & 1.226 & 1.256 & 1.267 \\ 1.469 & 1.416 & 1.368 & 1.338 & 1.332 & 1.352 & 1.394 & 1.446 & 1.497 & 1.533 & 1.548 \\ 1.702 & 1.639 & 1.585 & 1.549 & 1.543 & 1.566 & 1.614 & 1.675 & 1.734 & 1.776 & 1.791 \\ 1.883 & 1.815 & 1.753 & 1.714 & 1.707 & 1.733 & 1.786 & 1.854 & 1.918 & 1.965 & 1.982 \\ 1.999 & 1.927 & 1.862 & 1.820 & 1.812 & 1.840 & 1.897 & 1.968 & 2.037 & 2.087 & 2.105 \end{pmatrix}$$

и массив $\tilde{q}_1 = \{\tilde{q}_1(i, j)\}$, $i = \overline{1, m, 2}$, $j = \overline{1, n, 80}$, полученный в результате расчетов, проводимых по конечно-разностной шеститочечной схеме сеточной модели с применением метода прогонки:

$$\tilde{q}_1 = \begin{pmatrix} 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.006 \\ 0.071 & 0.068 & 0.066 & 0.064 & 0.064 & 0.065 & 0.067 & 0.069 & 0.072 & 0.074 & 0.075 \\ 0.208 & 0.199 & 0.193 & 0.189 & 0.188 & 0.191 & 0.197 & 0.204 & 0.212 & 0.217 & 0.219 \\ 0.407 & 0.389 & 0.377 & 0.369 & 0.367 & 0.373 & 0.385 & 0.399 & 0.413 & 0.424 & 0.428 \\ 0.651 & 0.620 & 0.605 & 0.596 & 0.589 & 0.598 & 0.616 & 0.639 & 0.662 & 0.681 & 0.686 \\ 0.924 & 0.874 & 0.859 & 0.840 & 0.836 & 0.849 & 0.874 & 0.907 & 0.939 & 0.968 & 0.972 \\ 1.203 & 1.129 & 1.119 & 1.096 & 1.091 & 1.106 & 1.139 & 1.181 & 1.223 & 1.264 & 1.267 \\ 1.469 & 1.368 & 1.366 & 1.339 & 1.333 & 1.351 & 1.391 & 1.443 & 1.494 & 1.548 & 1.548 \\ 1.708 & 1.575 & 1.579 & 1.552 & 1.544 & 1.565 & 1.612 & 1.672 & 1.731 & 1.798 & 1.791 \\ 1.883 & 1.735 & 1.744 & 1.716 & 1.708 & 1.733 & 1.785 & 1.852 & 1.918 & 1.995 & 1.982 \\ 1.999 & 1.837 & 1.848 & 1.822 & 1.814 & 1.841 & 1.897 & 1.968 & 2.039 & 2.123 & 2.105 \end{pmatrix}.$$

На рисунках 2.5 и 2.6 представлены графические изображения точного решения и решения, полученного по алгоритму конечно-разностной шеститочечной схемы. Точность расчетов при этом характеризуется величиной, вычисляемой по формуле (2.8) и определяющей отклонение в среднем полученных расчетных значений приближенного решения от точного решения, составила $\sigma_1(\hat{q}_T, \tilde{q}_1) = 0.0205$. На рисунке 2.7 показаны графики функций $\hat{q}_T(i, j)$ и $\tilde{q}_1(i, j)$ для фиксированного момента времени $\hat{t} = 0.5$.

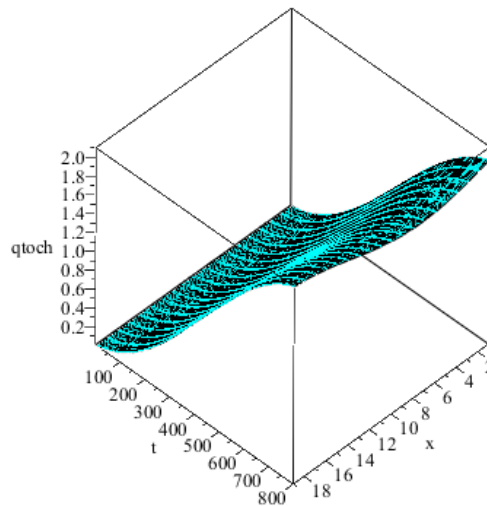


Рисунок 2.5 – Пространственно – временное распределение поля концентрации примесей, соответствующих точному решению $\hat{q}_T(i, j)$

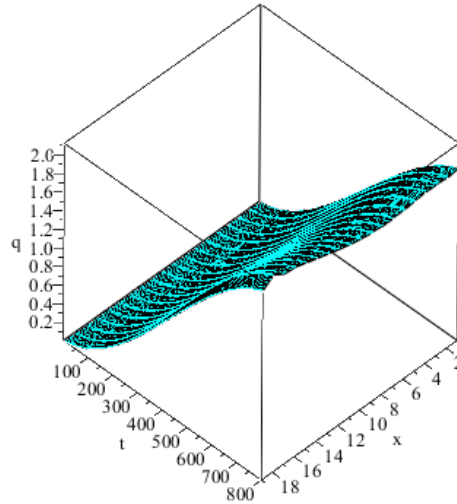


Рисунок 2.6 – Пространственно – временное распределение поля концентрации вредных примесей, соответствующих приближенному решению $\tilde{q}_1(i, j)$, полученному по алгоритму конечно-разностной шеститочечной схемы с погрешностью $\sigma_1 = 0.0205$

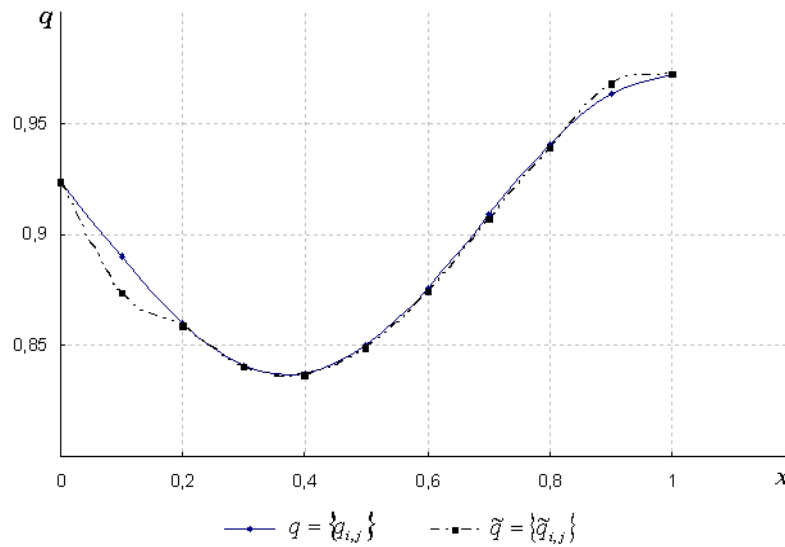


Рисунок 2.7 – Графики функций $\hat{q}_T(i, j)$ и $\tilde{q}_1(i, j)$ для фиксированного момента времени $\hat{t} = 0.5$ с с погрешностью $\sigma_1 = 0.0205$

Программная реализация вычислительного алгоритма, соответствующего итерационному методу (2.4), (2.5), проводилась при тех же исходных данных, описанных выше. Ниже представлены результаты численной реализации итерационного алгоритма в виде массива $\tilde{q}_2 = \{\tilde{q}_2(k, j)\}$, $k = \overline{1, m, 2}$, $j = \overline{1, n, 80}$,

$$\tilde{q}_2 = \begin{pmatrix} 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.006 \\ 0.071 & 0.069 & 0.066 & 0.065 & 0.064 & 0.065 & 0.068 & 0.069 & 0.072 & 0.075 & 0.075 \\ 0.208 & 0.202 & 0.194 & 0.189 & 0.189 & 0.192 & 0.198 & 0.205 & 0.213 & 0.218 & 0.219 \\ 0.406 & 0.392 & 0.378 & 0.371 & 0.368 & 0.375 & 0.385 & 0.401 & 0.414 & 0.424 & 0.428 \\ 0.651 & 0.627 & 0.606 & 0.593 & 0.591 & 0.599 & 0.617 & 0.641 & 0.664 & 0.679 & 0.685 \\ 0.923 & 0.890 & 0.860 & 0.841 & 0.838 & 0.850 & 0.877 & 0.909 & 0.940 & 0.964 & 0.972 \\ 1.203 & 1.159 & 1.121 & 1.096 & 1.091 & 1.107 & 1.142 & 1.185 & 1.225 & 1.256 & 1.267 \\ 1.469 & 1.416 & 1.369 & 1.337 & 1.332 & 1.352 & 1.393 & 1.446 & 1.497 & 1.533 & 1.546 \\ 1.702 & 1.639 & 1.586 & 1.549 & 1.543 & 1.567 & 1.615 & 1.675 & 1.733 & 1.776 & 1.792 \\ 1.883 & 1.814 & 1.753 & 1.715 & 1.707 & 1.733 & 1.786 & 1.854 & 1.918 & 1.965 & 1.982 \\ 1.999 & 1.927 & 1.862 & 1.822 & 1.813 & 1.842 & 1.897 & 1.969 & 2.037 & 2.085 & 2.105 \end{pmatrix}.$$

На рисунке 2.8 представлен «профиль» - пространственно – временное распределение поля концентрации загрязняющих веществ, соответствующее данному массиву. Значение $\sigma_2(q_T, \tilde{q}_2)$ - отклонение точного решения от расчетного, составило 0.0034.

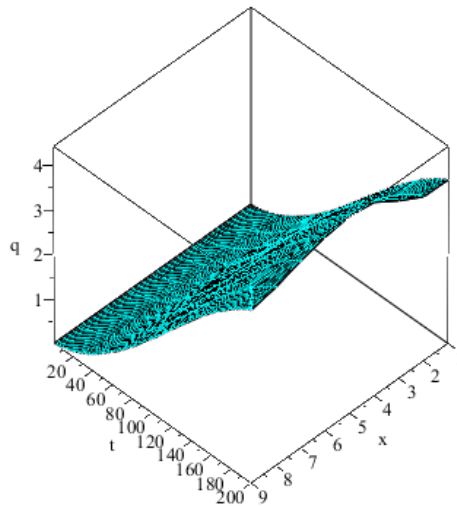


Рисунок 2.8 – Пространственно-временное распределение поля концентрации вредных примесей, соответствующих приближенному решению $\tilde{q}_2(k, j)$, полученному с использованием итерационного алгоритма с погрешностью $\sigma_2(\hat{q}_T, \tilde{q}_2) = 0.0034$

2.3 Методика планирования вычислительных экспериментов и интерпретация их результатов. Оценка устойчивости алгоритмов и влияния ошибок вычислений

Далее проводился вычислительный эксперимент по исследованию сходимости и устойчивости вычислительных конечно-разностных и итерационных алгоритмов, рассмотренных выше, к погрешностям в исходных данных.

В вычислительном эксперименте исследовалась сходимость приближенных решений к точному решению для каждого вычислительного алгоритма. В ходе численных

исследований было установлено, что важную роль в этом процессе играет соответствующий выбор значения параметра $\lambda = \Delta t / \Delta x^2$, связанного с размерностью задачи. Так, в таблице 2.1 приведены результаты исследования сходимости конечно – разностного алгоритма для различных значений параметра λ .

Таблица 2.1 – Влияние выбора параметра λ на сходимость конечно – разностного метода, характеризуемую величиной $\sigma_1(q_T, \tilde{q}_1)$

λ	m	n	$m \times n$	σ_1
1/2	10	200	2000	0.030
1/4	10	400	4000	0.029
1/6	10	600	6000	0.029
1/8	10	800	8000	0.028
1/2	20	800	16000	0.021
1/4	20	1600	32000	0.022
1/6	20	2400	48000	0.022
1/8	20	3200	64000	0.023

Примечание: $\lambda = \Delta t / \Delta x^2$ - параметр сетки, $m \times n$ - размерность задачи, σ - среднее отклонение приближенного решения \tilde{q}_1 от точного решения q_T .

Из таблицы 2.1 видно, каким должен быть параметр λ для достижения той или иной точности решения, наиболее приемлемым при этом значением является $\lambda = 0.5$, последующее его уменьшение приводит к увеличению размерности задачи и, в конечном итоге, к накоплению вычислительных ошибок, хотя и незначительному. Кроме того, применительно к решению конкретных прикладных задач, можно сказать, что обеспечить их большим объемом экспериментальных данных затруднительно. А с учетом того, что такие данные мониторинга могут нести в себе погрешности измерений, можно утверждать, что при большом объеме данных конечно – разностный алгоритм потеряет свойство устойчивости, поэтому требуются дополнительные численные исследования его устойчивости.

В таблице 2.2 показано, как влияет выбор параметра λ на сходимость итерационного алгоритма, характеризуемую величиной σ , определяемой по формуле (2.8). Из расчетных данных видно, что уменьшение параметра λ приводит к увеличению размерности сетки узлов и накоплению вычислительных ошибок, хотя и очень незначительному. Также видно, что с увеличением n значение параметра $\sigma_2(\hat{q}_T, \tilde{q}_2)$ практически не изменяется, то есть вычислительный процесс является стабильным, и поэтому приемле-

мыми значениями λ при конкретных значениях исходных данных можно считать значения $\lambda = \{0.5, 0.25\}$.

Таблица 2.2 – Влияние выбора параметра λ на сходимость итерационного метода, характеризуемую величиной $\sigma_2(\hat{q}_T, \tilde{q}_2)$

λ	m	n	$m \times n$	σ_2
1/2	10	200	2000	0.003
1/4	10	400	4000	0.003
1/6	10	600	6000	0.004
1/8	10	800	8000	0.005
1/2	20	800	16000	0.002
1/4	20	1600	32000	0.002
1/6	20	2400	48000	0.003
1/8	20	3200	64000	0.004

Сопоставим расчетные данные таблиц 2.1 и 2.2 с целью сравнения сходимости конечно – разностного и итерационного алгоритмов. Так, из рисунка 2.9 видно, что итерационный алгоритм обеспечивает точность решения на порядок выше, чем конечно - разностный при одинаковом количестве узлов расчетной сетки.

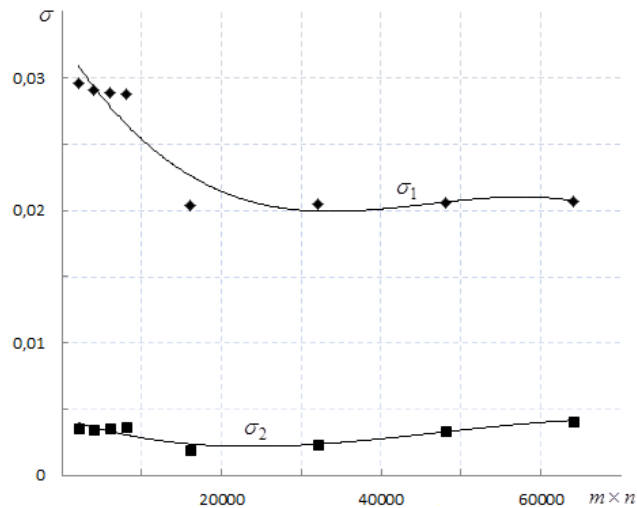


Рисунок 2.9 – Влияние размерности задачи на сходимость вычислительного процесса: σ_1 – отклонение приближенного решения от точного решения для конечно – разностного алгоритма, σ_2 – для итерационного алгоритма

Для итерационного метода в работе также выполнено исследование скорости сходимости, результаты представлены в таблице 2.3. Здесь величина ρ вычислялась по формуле (2.7), σ - по формуле (2.8). Величина $\delta\tilde{q}$ относительной погрешности метода определялась следующим образом:

$$\delta\tilde{q} = \frac{\sum_{k=1}^m \sum_{j=1}^n |(q_T)_{k,j} - \tilde{q}_{k,j}|}{\sum_{k=1}^m \sum_{j=1}^n |(q_T)_{k,j}|}. \quad (2.9)$$

Таблица 2.3 – Результаты исследования скорости сходимости итерационного метода, размерность задачи $m \times n = 2000$

ε	ν	ρ	σ	$\delta\tilde{q}$
0.1000	1	0.0211	0.0034	0.042
0.0100	2	0.0059	0.0033	0.041
0.0050	3	0.0024	0.0034	0.042
0.0010	4	0.0007	0.0034	0.042
0.0005	5	0.0002	0.0035	0.043
0.0001	6	0.0001	0.0036	0.044

Примечание: ε - задаваемая точность приближенного решения, ν - количество итераций, ρ - среднее отклонение приближенного решения от точного на текущей и предыдущей итерациях, σ - среднее отклонение приближенного решения от точного решения, $\delta\tilde{q}$ - относительная погрешность метода.

Анализ данных, приведенных в таблице, показывает, что представленный в работе итерационный метод при выбранных исходных данных обладает достаточно высокой скоростью сходимости.

Алгоритм исследования на устойчивость приближенных решений к возможным погрешностям в исходных данных, получаемых по конечно-разностному и итерационному алгоритмам соответственно, подробно описан в работах [38, 63] и в данной работе реализован программно для численных исследований. Согласно этому алгоритму предполагается, что вместо точного значения правой части $S^{(0)} = \{S_{k,j}^{(0)}\}$ исходного уравнения, известно его приближенное значение с погрешностью δ , то есть, имеем $S^{(\delta)} = \{S_{k,j}^{(\delta)}\}$. Здесь необходимо отметить, что вместо функции источника $S(x,t)$ в рассмотрение могут быть взяты и любые другие исходные данные, значения которых получены путем измерений при проведении соответствующих натуральных экспериментов. Величину погрешности δS и относительной погрешности $\overline{\delta S}$ функции источника $S(x,t)$ определим так:

$$\delta S = \|S^{(\delta)} - S^{(0)}\|, \quad (2.10)$$

$$\overline{\delta S} = \frac{\|S^{(\delta)} - S^{(0)}\|}{\|S^{(0)}\|}, \quad (2.11)$$

$$\|S^{(\delta)} - S^{(0)}\| = \sqrt{\frac{1}{n \cdot m} \sum_{j=1}^n \sum_{i=1}^m (S_{i,j}^{(\delta)} - S_{i,j}^{(0)})^2}, \quad \|S^{(0)}\| = \sqrt{\frac{1}{n \cdot m} \sum_{j=1}^n \sum_{i=1}^m (S_{i,j}^{(0)})^2}, \quad (2.12)$$

$$S^{(\delta)} = S^{(0)}(1 + \Theta \varepsilon), \quad (2.13)$$

Θ - случайные числа, равномерно распределенные на интервале $[-1; 1]$ (генерируются датчиком случайных чисел), $0 < \varepsilon < 1$. Аналогично определим величину погрешности δq и относительной погрешности $\overline{\delta q}$ функции распределения $q(x, t)$:

$$\delta q = \|q^{(\delta)} - q^{(0)}\|, \quad (2.14)$$

$$\overline{\delta q} = \frac{\|q^{(\delta)} - q^{(0)}\|}{\|q^{(0)}\|}, \quad (2.15)$$

$$\|q^{(\delta)} - q^{(0)}\| = \sqrt{\frac{1}{n \cdot m} \sum_{j=1}^n \sum_{i=1}^m (q_{i,j}^{(\delta)} - q_{i,j}^{(0)})^2}, \quad \|q^{(0)}\| = \sqrt{\frac{1}{n \cdot m} \sum_{j=1}^n \sum_{i=1}^m (q_{i,j}^{(0)})^2}. \quad (2.16)$$

Наряду с этими величинами введем в рассмотрение коэффициент усиления ошибки η , характеризующий степень отклонения получаемого решения $\tilde{q}^{(\delta)}$ от точного значения $q^{(0)}$ и коэффициент усиления относительной ошибки η_1 :

$$\eta = \delta S / \delta q, \quad (2.17)$$

$$\eta_1 = \overline{\delta S} / \overline{\delta q}. \quad (2.18)$$

Численные исследования устойчивости конечно – разностного и итерационного алгоритмов проведены на основе программной реализации алгоритма (2.10) – (2.18), результаты представлены в таблицах 2.4 и 2.5 соответственно.

Таблица 2.4 – Влияние величины погрешности в исходных данных на коэффициент усиления ошибки для конечно – разностного метода

ε	$\sigma_q^{(\delta)}$	$\overline{\delta S}$	$\overline{\delta q}$	η	η_1
0.010	0.020	0.001	0.0184	0.061	0.031
0.025	0.021	0.014	0.0183	1.531	0.789
0.050	0.021	0.029	0.0182	3.076	1.586
0.075	0.022	0.043	0.0182	4.633	2.389
0.100	0.023	0.058	0.0181	6.201	3.197
0.125	0.024	0.073	0.0188	7.776	4.009

Примечание: ε - погрешность, вносимая в источник $S(x,t)$, $\sigma_q^{(\delta)}$ - отклонение приближенного решения от точного, $\overline{\delta S}$ - величина относительной погрешности функции источника $S(x,t)$, $\overline{\delta q}$ - величина относительной погрешности функции распределения $q(x,t)$, η - коэффициент усиления ошибки, η_1 - коэффициент усиления относительной ошибки.

Расчетные данные таблицы 2.4 показывают, что погрешность в исходных данных свыше 5% не приемлема. Вычислительная схема (алгоритм) устойчива при величине погрешности в исходных данных (правая часть уравнения (2.1)) допустимой до 5%. В целом можно сделать вывод о неустойчивости конечно – разностного алгоритма к погрешностям в исходных данных. Результаты расчетов, представленные в таблице 2.5, свидетельствуют об устойчивости рассматриваемого в работе метода интегральных уравнений и соответствующей итерационной вычислительной схемы. При этом максимально возможной погрешностью в исходных данных (погрешностью, вносимой в правую часть уравнения переноса) можно считать погрешность порядка 7.5%.

Таблица 2.5 – Влияние величины погрешности в исходных данных на коэффициент усиления ошибки для итерационного метода

ε	$\sigma_q^{(\delta)}$	$\overline{\delta S}$	$\overline{\delta q}$	η	η_1
0.010	0.003	0.001	0.046	0.025	0.012
0.025	0.003	0.014	0.047	0.614	0.307
0.050	0.004	0.029	0.048	1.207	0.603
0.075	0.005	0.044	0.049	1.779	0.889
0.100	0.005	0.058	0.050	2.333	1.166
0.125	0.006	0.073	0.051	2.868	1.432

Сопоставление расчетных данных в таблицах 2.4 и 2.5 позволяет сделать вывод о большей устойчивости к погрешностям в исходных данных итерационного алгоритма, так как значение коэффициента усиления ошибки больше единицы можно считать не приемлемым.

Таким образом, в результате численных исследований выявлены особенности работы двух вычислительных алгоритмов, обосновано предпочтение алгоритма на основе метода интегральных уравнений по сравнению с алгоритмом на основе конечно – разностного метода. Эффективность итерационного алгоритма значительно выше, поскольку для его работы требуется значительно меньший объем вычислений и исходных дан-

ных, он обладает большей точностью получаемых решений и устойчивостью к погрешностям в исходных данных.

2.4 Качественные подходы к исследованию динамики полей концентрации аэрозольных примесей на основе фундаментальной системы решения уравнений параболического типа

Помимо исследуемой в работе известной итерационной схемы [38] решения уравнения переноса субстанции в турбулентной среде, могут быть построены и другие формы интегральных уравнений и соответствующие им преобразования интегральной модели. Предложенный ниже вариант будет носить сугубо качественный характер и его основное назначение состоит в разработке методик качественной оценки значений параметров в задачах математического моделирования явления переноса в турбулентных средах.

Уравнение переноса субстанции q в пограничном слое атмосферы основано на полуэмпирическом уравнении турбулентной диффузии (2.1), которое перепишем следующим образом:

$$\dot{q} + \alpha q + \beta V'q + \beta Vq' - \theta K'q' - \theta Kq'' = \xi \cdot S, \quad (2.19)$$

$$\dot{q} = \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{t}}, \quad q' = \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}}, \quad q'' = \frac{\partial^2 \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}^2}, \quad \hat{\alpha}(\hat{t}) = \alpha = const \quad \forall t, \quad V' = \frac{\partial \hat{V}(\hat{x}, \hat{t})}{\partial \hat{x}}, \quad K' = \frac{\partial \hat{K}(\hat{x}, \hat{t})}{\partial \hat{x}},$$

$V = \hat{V}(\hat{x}, \hat{t}), \quad K = \hat{K}(\hat{x}, \hat{t}), \quad S = \hat{S}(\hat{x}, \hat{t})$. При этом уравнение (2.19) дополняется условиями (ограничениями на коэффициенты) (п. 1.4):

$$\begin{cases} V_{\min} \leq V(x, t) \leq V_{\max}, \\ K_{\min} \leq K(x, t) \leq K_{\max}, \\ 0 \leq x \leq X, \quad 0 \leq t \leq T. \end{cases} \quad (2.20)$$

Следует также упомянуть о задании начальных и граничных условий на искомую функцию $q(x, t)$, а именно $q(x, t=0) = q_0(x)$, $q(x=0, t) = q_1(t)$ и $q(x=X, t) = q_2(t)$. Перейдем в (2.19) к нормированным функциям и независимым переменным:

$$\begin{cases} \hat{x} = \frac{x}{X}, \quad x = \hat{x} \cdot X, \quad 0 \leq \hat{x} \leq 1, \\ \hat{t} = \frac{t}{T}, \quad t = \hat{t} \cdot T, \quad 0 \leq \hat{t} \leq 1. \end{cases}$$

$$\hat{K}(\hat{x}, \hat{t}) = \frac{\hat{K}[\hat{x}, \hat{t}]}{K_{\max}}, \quad \hat{V}(\hat{x}, \hat{t}) = \frac{\hat{V}[\hat{x}, \hat{t}]}{V_{\max}}. \quad (2.21)$$

В ряде выражений ниже будут использоваться параметры

$$\omega_K = \frac{K_{\min}}{K_{\max}}, \quad \omega_V = \frac{V_{\min}}{V_{\max}}, \quad (2.22)$$

характеризующие размах функций $V(x, t)$ и $K(x, t)$. С учетом (2.21) и (2.22) можно писать ограничения на указанные функции в виде $\omega_K \leq \hat{K}(\hat{x}, \hat{t}) \leq 1$, $\omega_V \leq \hat{V}(\hat{x}, \hat{t}) \leq 1$. Если функция $V(x, t)$ помимо положительных принимает и отрицательные значения, первое неравенство в (2.14) следует заменить неравенством $V_{\min} \leq |V(x, t)| \leq V_{\max}$. В дальнейшем будем считать, что $\omega_K > 0$, $\omega_V > 0$. Выполним преобразования (2.13):

$$\begin{aligned} -\theta K q'' + q'(\beta V - \theta K') + q(\alpha + \beta V') &= \xi S - \dot{q}, \\ q'' - q' \cdot \frac{1}{\theta K}(\beta V - \theta K') - q \cdot \frac{1}{\theta K}(\alpha + \beta V') &= \frac{\xi S - \dot{q}}{-\theta K}. \end{aligned}$$

В итоге приходим к следующей параметризованной модели для исходного дифференциального уравнения (2.19):

$$q'' + \frac{\theta K' - \beta V}{\theta K} q' - \frac{\alpha + \beta V'}{\theta K} q = \frac{\dot{q} - \xi S}{\theta K}. \quad (2.23)$$

В выражении (2.23) коэффициенты α , β , θ , ξ как и прежде безразмерные величины. Для дальнейшего анализа удобно ввести следующие обозначения: $p_1(\hat{x}, \hat{t}) = (\theta K' - \beta V)/\theta K$, $p_2(\hat{x}, \hat{t}) = -(\alpha + \beta V')/\theta K$, $f(\hat{x}, \hat{t}) = (\dot{q} - \xi S)/\theta K$, в соответствии с которыми уравнение (2.23) примет вид

$$\hat{q}'' + p_1 \hat{q}' + p_2 \hat{q} = f(\hat{x}|\hat{t}), \quad (2.24)$$

и является, как известно, обыкновенным дифференциальным неоднородным уравнением второго порядка при фиксированном значении \hat{t} . Решение (2.24) есть сумма фундаментального решения однородного дифференциального уравнения, соответствующего (2.24) и частного решения неоднородного уравнения. Выполним построения, которые в соответствии с анализом начинаются с поиска фундаментальных решений однородного уравнения:

$$\hat{q}'' + p_1 \hat{q}' + p_2 \hat{q} = 0. \quad (2.25)$$

Здесь положим, что $p_1(\hat{x}|\hat{t})$ и $p_2(\hat{x}|\hat{t})$ есть постоянные, для этого допустим, что $\hat{V}(\hat{x}|\hat{t}) = 1$, $\hat{K}(\hat{x}|\hat{t}) = 1 \quad \forall \hat{x} \in [0; 1]$ и $\forall \hat{t} \in [0; 1]$, тогда получим $p_1 = -\beta/\theta$, $p_2 = -\alpha/\theta$. Если фундамен-

тальные решения обозначить через $u_1(\hat{x}) = e^{r_1 \hat{x}}$ и $u_2(\hat{x}) = e^{r_2 \hat{x}}$, r_1 и r_2 - корни характеристического уравнения $r^2 + p_1 r + p_2 = 0$: $r_1 = \frac{\beta}{2\theta} + \sqrt{\frac{\beta^2}{4\theta^2} + \frac{\alpha}{\theta}}$, $r_2 = \frac{\beta}{2\theta} - \sqrt{\frac{\beta^2}{4\theta^2} + \frac{\alpha}{\theta}}$, то для неоднородного уравнения (2.24), используя метод вариации произвольных постоянных, получим:

$$\begin{cases} C_1'(\hat{x})e^{r_1 \hat{x}} + C_2'(\hat{x})e^{r_2 \hat{x}} = 0, \\ r_1 C_1'(\hat{x})e^{r_1 \hat{x}} + r_2 C_2'(\hat{x})e^{r_2 \hat{x}} = f(\hat{x}). \end{cases} \quad (2.26)$$

Из первого уравнения системы (2.26) выразим $C_1'(\hat{x})$:

$$C_1'(\hat{x}) = -\frac{C_2'(\hat{x})e^{r_2 \hat{x}}}{e^{r_1 \hat{x}}} = -C_2'(\hat{x})e^{(r_2 - r_1)\hat{x}}, \quad (2.27)$$

и подставим во второе уравнение системы (2.26), в результате будем иметь: $C_2'(\hat{x})e^{r_2 \hat{x}}(r_2 - r_1) = f(\hat{x})$. Окончательно, запишем для $C_2'(\hat{x})$ следующее выражение:

$$C_2'(\hat{x}) = \frac{f(\hat{x})}{r_2 - r_1} e^{-r_2 \hat{x}}, \text{ подставляя которое в (2.27), получим для функции } C_1'(\hat{x}):$$

$$C_1'(\hat{x}) = -\frac{f(\hat{x})}{r_2 - r_1} e^{-r_1 \hat{x}}. \text{ Интегрируя эти выражения, находим функции } C_1(\hat{x}) \text{ и } C_2(\hat{x}) \text{ в виде}$$

следующих интегралов:

$$C_1(\hat{x}) = -\frac{1}{r_2 - r_1} \int_0^{\hat{x}} e^{-r_1 \hat{x}'} \cdot f(\hat{x}') d\hat{x}' + C_1(0), \quad C_2(\hat{x}) = \frac{1}{r_2 - r_1} \int_0^{\hat{x}} e^{-r_2 \hat{x}'} \cdot f(\hat{x}') d\hat{x}' + C_2(0).$$

Таким образом, частное решение исходного уравнения (2.24) принимает вид:

$$\begin{aligned} \tilde{q}(\hat{x}) &= C_1(\hat{x})u_1(\hat{x}) + C_2(\hat{x})u_2(\hat{x}) = \\ &= -\frac{1}{r_2 - r_1} \int_0^{\hat{x}} e^{-r_1 \hat{x}'} \cdot f(\hat{x}') d\hat{x}' \cdot e^{r_1 \hat{x}} + \frac{1}{r_2 - r_1} \int_0^{\hat{x}} e^{-r_2 \hat{x}'} \cdot f(\hat{x}') d\hat{x}' \cdot e^{r_2 \hat{x}} + C(0) = \\ &= \frac{1}{r_2 - r_1} \int_0^{\hat{x}} f(\hat{x}') [e^{r_2(\hat{x} - \hat{x}')} - e^{r_1(\hat{x} - \hat{x}')}] d\hat{x}' + \tilde{q}(0) = \frac{1}{r_2 - r_1} \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') f(\hat{x}') d\hat{x}' + \tilde{q}(0). \end{aligned} \quad (2.28)$$

В интеграл (2.28) остается ввести функцию $f(\hat{x}|\hat{t}) = \frac{\hat{q}(\hat{x}, \hat{t}) - \xi \cdot \hat{S}(\hat{x}|\hat{t})}{\theta \cdot \hat{K}}$, которая при

$\hat{K} = 1$ записывается следующим образом:

$$f(\hat{x}|\hat{t}) = \frac{1}{\theta} \cdot (\hat{q}(\hat{x}, \hat{t}) - \xi \cdot \hat{S}(\hat{x}|\hat{t})), \quad (2.29)$$

и тогда получаем, вновь введя новую переменную \hat{t} и подставляя (2.29) в (2.28):

$$\begin{aligned}\tilde{q}(\hat{x}|\hat{t}) &= \frac{1}{r_2 - r_1} \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') \cdot \left(\frac{1}{\theta} \cdot (\dot{q}(\hat{x}, \hat{t}) - \xi \cdot \hat{S}(\hat{x}|\hat{t})) \right) d\hat{x}' + \tilde{q}(0) = \\ &= \frac{1}{r_2 - r_1} \cdot \frac{1}{\theta} \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') \cdot \dot{q}(\hat{x}'|\hat{t}) d\hat{x}' - \frac{1}{r_2 - r_1} \cdot \frac{1}{\theta} \cdot \xi \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') \cdot \hat{S}(\hat{x}'|\hat{t}) d\hat{x}' + \tilde{q}(0|\hat{t}).\end{aligned}\quad (2.30)$$

Для выражения (2.30) введем обозначения:

$$\lambda = \frac{1}{\theta} \cdot \frac{1}{r_2 - r_1}, \quad \varphi(\hat{x}|\hat{t}) = \tilde{q}(0|\hat{t}) - \lambda \xi \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') \cdot \hat{S}(\hat{x}'|\hat{t}) d\hat{x}',$$

что позволяет записать (2.28) в виде:

$$\tilde{q}(\hat{x}|\hat{t}) = \lambda \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') \cdot \dot{q}(\hat{x}'|\hat{t}) d\hat{x}' + \varphi(\hat{x}|\hat{t}),\quad (2.31)$$

то есть свести его к интегро – дифференциальному уравнению.

2.5 Алгоритмизация, программирование и результаты вычислений для качественной модели оценки концентрации аэрозольных примесей на основе фундаментальной системы решения уравнений параболического типа

Таким образом, основные расчетные формулы метода (п. 2.4) можно представить в виде следующей расчетно-аналитической модели:

$$\begin{aligned}\tilde{q}(\hat{x}|\hat{t}) &= \lambda \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') \cdot \dot{q}(\hat{x}'|\hat{t}) d\hat{x}' + \varphi(\hat{x}|\hat{t}), \\ \varphi(\hat{x}|\hat{t}) &= \tilde{q}(0|\hat{t}) - \lambda \cdot \xi \int_0^{\hat{x}} \tilde{K}(\hat{x}, \hat{x}') \cdot \hat{S}(\hat{x}'|\hat{t}) d\hat{x}', \quad \tilde{K}(\hat{x}, \hat{x}') = e^{r_2(\hat{x}-\hat{x}')} - e^{r_1(\hat{x}-\hat{x}')}, \\ r_1 &= \frac{\beta}{2\theta} + \sqrt{\frac{\beta^2}{4\theta^2} + \frac{\alpha}{\theta}}, \quad r_2 = \frac{\beta}{2\theta} - \sqrt{\frac{\beta^2}{4\theta^2} + \frac{\alpha}{\theta}}, \quad \lambda = \frac{1}{\theta} \cdot \frac{1}{r_2 - r_1},\end{aligned}$$

которую далее для краткости будем называть первой качественной моделью.

Обратимся к уравнению (2.31) и выполним для него дискретизацию на сетке узлов $\{(\hat{x}_i, \hat{t}_j)\}$ в области $\Omega = [0; 1] \times [0; 1]$, полагая, что для оценки значений $\dot{q}(\hat{x}_i, \hat{t}_j)$ используется конечно – разностный подход. Имеем

$$\hat{q}(\hat{x}_i, \hat{t}_j) = \lambda \sum_{k=0}^i \omega_k \tilde{K}(\hat{x}_i, \hat{x}_k) \cdot \frac{\hat{q}(\hat{x}_k, \hat{t}_j) - \hat{q}(\hat{x}_k, \hat{t}_{j-1})}{\Delta \hat{t}} \cdot \Delta \hat{x} + \varphi(\hat{x}_i, \hat{t}_j) =$$

$$= \lambda \frac{\Delta \hat{x}}{\Delta \hat{t}} \sum_{k=0}^i \left[\omega_k \tilde{K}(\hat{x}_i, \hat{x}_k) (\hat{q}(\hat{x}_k, \hat{t}_j) - \hat{q}(\hat{x}_k, \hat{t}_{j-1})) \right] + \varphi(\hat{x}_i, \hat{t}_j). \quad (2.32)$$

Введем следующие обозначения применительно к (2.32) $\mu = \lambda \Delta \hat{x} / \Delta \hat{t}$, $\eta = \lambda \cdot \xi \cdot \Delta \hat{x}$, $\tilde{K}_{ik} = \tilde{K}(\hat{x}_i, \hat{x}_k)$, $\tilde{q}(\hat{x}_i, \hat{t}_j) = \tilde{q}_{ij}$, $\varphi(\hat{x}_i, \hat{t}_j) = \varphi_{ij}$, $k \leq i$, $k = \overline{0, m-1}$, $i = \overline{1, m}$, $j = \overline{1, n}$ и получим простую рекурсивную схему:

$$\tilde{q}_{i,j} = \mu \sum_{k=0}^i \omega_k \tilde{K}_{i,k} \cdot (\tilde{q}_{k,j} - \tilde{q}_{k,j-1}) + \varphi_{i,j}, \quad (2.33)$$

$$\varphi_{i,j} = q_{0,j} - \eta \sum_{k=0}^i \omega_k \tilde{K}_{i,k} S_{k,j}, \quad (2.34)$$

$$\tilde{K}_{i,k} = e^{r_2(x_i - x_k)} - e^{r_1(x_i - x_k)}, \text{ здесь } x_k \leq x_i, \quad (2.35)$$

$$\lambda = 1/\theta(r_2 - r_1), \quad \mu = \lambda \Delta \hat{x} / \Delta \hat{t}, \quad \eta = \lambda \cdot \xi \cdot \Delta \hat{x}, \quad k \leq i, \quad k = \overline{0, m-1}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Необходимо отметить, что данный метод (схема) работает в предположении о том, что $\hat{K} = 1$ и $\hat{V} = 1$. Тогда исходная модель должна быть такой:

$$\frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{t}} + \hat{\alpha}(\hat{t}) \cdot \hat{q}(\hat{x}, \hat{t}) + \beta \cdot \frac{\partial \hat{q}(\hat{x}, \hat{t})}{\partial \hat{x}} - \theta \cdot \frac{\partial \hat{q}^2(\hat{x}, \hat{t})}{\partial \hat{x}^2} = \xi \cdot \hat{S}(\hat{x}, \hat{t}).$$

Исходя из этого уравнения, был построен новый тестовый пример и получены новые распределения $(q_T)_{i,j}$ и $S_{i,j}$. Программная реализация данного алгоритма выполнялась при следующих значениях исходных данных: $V_0 = 10$ м/с, $K_0 = 100$ м²/с, $q_0 = 0.001$ кг/м³, $S_0 = 0.0001$ кг/(м³ · с). Значение V_0 определяется как среднее значение на интервале $[V_{\min}, V_{\max}]$, аналогично задаются значения K_0 , q_0 и S_0 на соответствующих интервалах. На основе этих констант с помощью алгоритма, подробно описанного выше в предыдущем параграфе, генерируются массивы дискретных данных $\{q_{ij}\}_{m \times n}$, $\{V_{ij}\}_{m \times n}$, $\{K_{ij}\}_{m \times n}$ и $\{S_{ij}\}_{m \times n}$. Далее выполняется нормирование заданных распределений в соответствии с процедурой, описанной выше, в результате чего получаем массивы $\{\hat{q}_{ij}\}_{m \times n}$, $\{\hat{V}_{ij}\}_{m \times n}$, $\{\hat{K}_{ij}\}_{m \times n}$ и $\{\hat{S}_{ij}\}_{m \times n}$, элементы которых принимают значения из интервала $[0; 1]$, а также значения V^* , q^* , K^* и S^* . Затем задаются значения коэффициентов $\alpha = 0.1$, $\beta = 1$, $\theta = 1$, $m = 10$, $n = 100$, вычисляются значения $\Delta x = 0.067$, $\Delta t = 0.05$, $\xi = 1.931$, $\lambda = 0.845$, $\mu = 1.127$, $\eta = 0.109$, $r_1 = -0.092$, $r_2 = 1.092$ и определяются значения $X = 10$ м и $T = 5$ с. Ниже приведены результаты расчетов для рекурсивной вычислительной схемы (2.33) –

(2.35) в виде массивов - массив $\hat{q}_T = \{\hat{q}_{i,j}\}$, $i = \overline{1,m}$, $j = \overline{1,n,10}$ с соответствующим шагом, который представляет собой точное решение уравнения переноса (2.1) в нормированном виде:

$$q_T = \begin{pmatrix} 0.534 & 0.575 & 0.615 & 0.654 & 0.691 & 0.726 & 0.759 & 0.789 & 0.817 & 0.842 & 0.864 \\ 0.559 & 0.602 & 0.643 & 0.684 & 0.723 & 0.759 & 0.794 & 0.826 & 0.855 & 0.881 & 0.904 \\ 0.581 & 0.625 & 0.669 & 0.711 & 0.751 & 0.789 & 0.825 & 0.858 & 0.888 & 0.915 & 0.939 \\ 0.598 & 0.644 & 0.689 & 0.732 & 0.774 & 0.813 & 0.850 & 0.885 & 0.916 & 0.944 & 0.968 \\ 0.611 & 0.658 & 0.704 & 0.748 & 0.790 & 0.831 & 0.868 & 0.903 & 0.935 & 0.964 & 0.988 \\ 0.618 & 0.665 & 0.711 & 0.756 & 0.799 & 0.839 & 0.878 & 0.913 & 0.945 & 0.974 & 0.999 \\ 0.618 & 0.666 & 0.712 & 0.757 & 0.799 & 0.840 & 0.879 & 0.914 & 0.946 & 0.975 & 1.000 \\ 0.612 & 0.659 & 0.705 & 0.749 & 0.792 & 0.832 & 0.870 & 0.905 & 0.937 & 0.966 & 0.990 \\ 0.600 & 0.646 & 0.691 & 0.735 & 0.777 & 0.816 & 0.853 & 0.887 & 0.919 & 0.947 & 0.971 \\ 0.583 & 0.628 & 0.672 & 0.714 & 0.754 & 0.793 & 0.829 & 0.862 & 0.892 & 0.919 & 0.943 \\ 0.562 & 0.605 & 0.647 & 0.688 & 0.727 & 0.764 & 0.798 & 0.830 & 0.859 & 0.886 & 0.909 \end{pmatrix}$$

и массив $\tilde{q}_3 = \{\tilde{q}_{i,j}\}$, $i = \overline{1,m}$, $j = \overline{1,n,10}$, полученный в результате расчетов, проводимых по рекурсивному алгоритму (2.33) - (2.35):

$$\tilde{q}_3 = \begin{pmatrix} 0.510 & 0.549 & 0.587 & 0.624 & 0.659 & 0.692 & 0.724 & 0.753 & 0.779 & 0.803 & 0.823 \\ 0.513 & 0.554 & 0.591 & 0.628 & 0.663 & 0.696 & 0.727 & 0.756 & 0.782 & 0.805 & 0.826 \\ 0.516 & 0.562 & 0.599 & 0.636 & 0.670 & 0.704 & 0.734 & 0.762 & 0.787 & 0.810 & 0.829 \\ 0.517 & 0.574 & 0.611 & 0.647 & 0.681 & 0.713 & 0.743 & 0.770 & 0.795 & 0.817 & 0.835 \\ 0.516 & 0.592 & 0.626 & 0.661 & 0.695 & 0.726 & 0.755 & 0.781 & 0.805 & 0.826 & 0.843 \\ 0.511 & 0.617 & 0.645 & 0.679 & 0.712 & 0.743 & 0.770 & 0.795 & 0.818 & 0.837 & 0.852 \\ 0.499 & 0.653 & 0.667 & 0.702 & 0.734 & 0.763 & 0.789 & 0.813 & 0.833 & 0.849 & 0.863 \\ 0.482 & 0.707 & 0.693 & 0.729 & 0.759 & 0.787 & 0.811 & 0.833 & 0.851 & 0.866 & 0.877 \\ 0.456 & 0.777 & 0.719 & 0.762 & 0.789 & 0.815 & 0.837 & 0.856 & 0.872 & 0.884 & 0.892 \\ 0.420 & 0.882 & 0.744 & 0.801 & 0.825 & 0.848 & 0.868 & 0.883 & 0.895 & 0.904 & 0.909 \\ 0.372 & 1.000 & 0.759 & 0.848 & 0.865 & 0.885 & 0.902 & 0.914 & 0.923 & 0.926 & 0.927 \end{pmatrix}$$

Графическое представление данных распределений показано на рисунках 2.10 и 2.11 соответственно. Отклонение приближенного решения от точного определялось по формуле (2.8) и составило 0.082. Полученный результат можно считать вполне приемлемым для данного качественного метода и соответствующего рекурсивного алгоритма. Приближенное качественное решение \tilde{q}_3 можно далее использовать как исходные данные для других более точных алгоритмов, описанных в [38]. Ниже приводятся результаты численного исследования алгоритма (2.33) - (2.35).

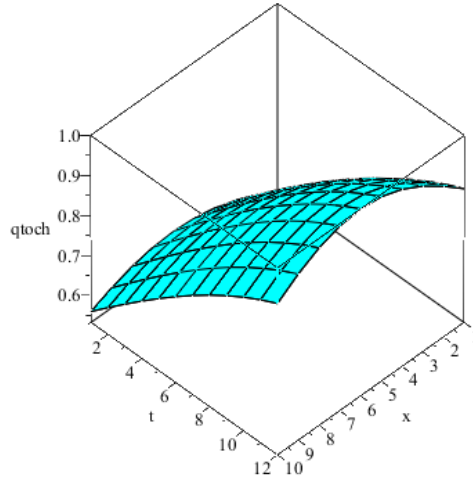


Рисунок 2.10 – Пространственно-временное распределение поля концентрации вредных примесей, соответствующих точному решению $\hat{q}_T(i, j)$

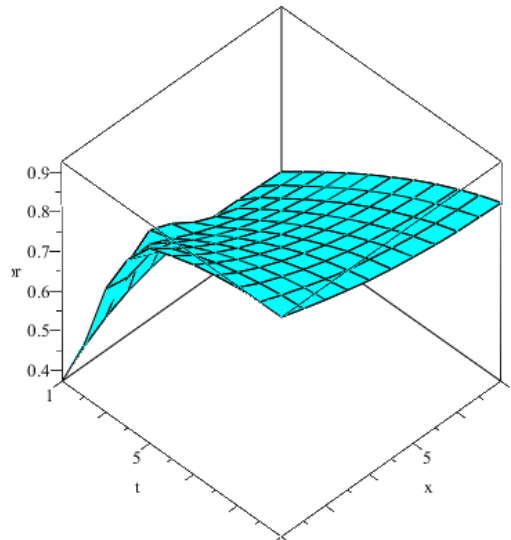


Рисунок 2.11 – Пространственно-временное распределение поля концентрации вредных примесей, соответствующих приближенному решению $\tilde{q}_3(i, j)$, полученному с использованием рекурсивного алгоритма первой качественной модели одномерного уравнения переноса с погрешностью $\sigma_3(\hat{q}_T, \tilde{q}_3) = 0.082$

Таблица 2.6 – Влияние размерности задачи на сходимость вычислительного алгоритма первой качественной модели

m	n	$m \times n$	μ	η	σ_3
5	100	500	1.690	0.325	0.059
10	100	1000	0.845	0.163	0.082
10	200	2000	1.690	0.161	0.095
15	200	3000	1.127	0.109	0.124

Примечание: $m \times n$ - размерность задачи, $\mu = \lambda \Delta \hat{x} / \Delta \hat{t}$, $\eta = \lambda \cdot \xi \cdot \Delta \hat{x}$, σ_3 - отклонение приближенного решения \tilde{q}_3 от точного решения q_T .

Анализ данных таблицы 2.6 показывает, что увеличение размерности задачи приводит к увеличению объема вычислений и, как следствие, накоплению вычислительной погрешности. Приемлемой размерностью можно считать размерность 10×100 . Кроме сходимости, также исследовалась устойчивость вычислительного алгоритма к возможным погрешностям в исходных данных уравнения (2.1), а именно исследовалась устойчивость по правой части уравнения переноса. Алгоритм исследования на устойчивость подробно изложен в предыдущем параграфе. Результаты вычислительного эксперимента представлены в таблице 2.7.

Таблица 2.7 – Влияние величины погрешности в исходных данных на коэффициент усиления ошибки для первой качественной модели

ε	$\sigma_q^{(\delta)}$	$\overline{\delta S}$	$\overline{\delta q}$	η	η_1
0.010	0.014	0.006	0.0002	0.077	0.106
0.025	0.015	0.014	0.0004	0.193	0.264
0.050	0.015	0.028	0.0008	0.386	0.527
0.075	0.016	0.044	0.0012	0.578	0.789
0.100	0.018	0.057	0.0016	0.769	1.052

Примечание: ε - погрешность, вносимая в источник $S(x, t)$, $\sigma_q^{(\delta)}$ - отклонение приближенного решения от точного, $\overline{\delta S}$ - величина относительной погрешности функции источника $S(x, t)$, $\overline{\delta q}$ - величина относительной погрешности функции распределения $q(x, t)$, η - коэффициент усиления ошибки, η_1 - коэффициент усиления относительной ошибки.

Анализируя данные таблицы 2.7, можно сделать вывод об устойчивости исследуемого алгоритма к возможным погрешностям в исходных данных. При этом максимально возможной погрешностью, вносимой в правую часть (2.1), можно считать погрешность порядка 7.5 % и не выше.

2.6 Разработка методики качественной оценки концентрации аэрозольных примесей в рамках итерационных алгоритмов

Изложенный выше в п. 1.5 подход к построению вычислительных алгоритмов в теории переноса, основанный на предварительном преобразовании дифференциального уравнения в частных производных в интегро – дифференциальное может быть одновременно использован для построения простых расчетно – аналитических моделей. Эта возможность связана с тем, что интегральные уравнения Вольтерра второго рода эффек-

тивно решаются методом последовательных приближений, то есть алгоритмически сводятся к итерационным вычислительным схемам. Известно, что если интегральное уравнение Вольтерра второго рода построено достаточно качественно (его ядро подобрано надлежащим образом), то уже первое приближение в итерационном процессе дает неплохое представление об аналитических свойствах искомого решения. Важно также заметить, что во многих случаях в качестве нулевого приближения в этих задачах достаточно использовать свободный член интегрального уравнения. Остается лишь в рассматриваемую задачу ввести те или иные частные допущения, чтобы интеграл, связанный с вычислением первого приближения, брался аналитическими методами. То, что получается в результате подобного подхода ниже будет называться приближенной расчетно – аналитической моделью в теории переноса субстанции в пограничном слое атмосферы. В качестве примера рассмотрим итерационный процесс, подробно изложенный в п. 1.5:

$$\begin{cases} q^{(v)}(t|x) = \varphi(t|x) - \int_0^t \tilde{K}(t, t', x) \cdot \psi^{(v-1)}(t'|x) dt' \\ \psi^{(v)}(t|x) = (V(x, t) - K'(x, t)) \cdot (q'(x, t))^{(v)} - K(x, t) \cdot (q''(x, t))^{(v)}, \end{cases} \quad (2.36)$$

для уравнения турбулентной диффузии, записанного в форме:

$$\dot{q} + (\alpha + V')q = S(x, t) - [(V - K')q' - Kq''].$$

Первое приближение для этого процесса дано интегралом:

$$q^{(1)}(t|x) = \int_0^t \tilde{K}(t, t', x) \cdot \psi[t', \varphi(t', x), x] dt' + \varphi(t', x), \quad 0 \leq x \leq X, \quad (2.37)$$

который и постараемся свести ниже к достаточно простому аналитическому выражению, позволяющему последовательно исследовать влияние тех или иных параметров задачи на искомое решение. Ясно, что упрощение аналитических выражений, включающих в себя операцию интегрирования или дифференцирования функции, требует ввести допущение о характере поведения этих функций, которые далее бы исключили указанные операции в этих выражениях. Если речь идет о первой итерационной схеме (2.37), то упростить ядро

$$\tilde{K}(x, t, t') = \exp \left\{ - \int_{t'}^t \left(\alpha(t'') + \beta \cdot \frac{\partial V(x, t'')}{\partial x} \right) dt'' \right\} \quad (2.38)$$

можно, если положить $V(x,t) = const$, $V'_x = 0$, а также считать $\alpha(t)$ константой, равной α .

Обратимся к полученному ранее в п. 1.5 выражению (1.64) для функции $q(x,t)$:

$$q(x,t) = \varphi(x,t) - \int_{t_0}^t \tilde{K}(x,t,t') \cdot \psi(x,t') dt', \quad (2.39)$$

$$\varphi(x,t) = q(x,t_0) \cdot \exp \left\{ - \int_{t_0}^t \left(\alpha(t') + \beta \cdot \frac{\partial V(x,t')}{\partial x} \right) dt' \right\} + \int_{t_0}^t \xi \cdot S(x,t') \cdot \tilde{K}(x,t,t') dt'. \quad (2.40)$$

Итак, приведенные условия позволяют записать (2.38) следующим образом:

$$\tilde{K}(t,t') = \exp \left\{ - \int_{t'}^t \alpha(t'') dt'' \right\} = \exp \left\{ - \alpha \cdot t'' \Big|_{t'}^t \right\} = e^{-\alpha(t-t')}, \quad 0 \leq t' \leq t. \quad (2.49)$$

Здесь ядро интегрального уравнения (2.39) перестает зависеть от переменной x . Подобное допущение применим и к функционалу $\psi = q'_x(\beta V - \theta K'_x) - q''_{xx}(\theta K)$, то есть, полагаем $K(x,t) = const$, $K'_x = 0$, в результате чего придем к выражению $\psi(x,t,q',q'') = \beta V q'_x - \theta K q''_{xx}$. Свободный член (2.40) в уравнении (2.39) при условиях $V' = 0$, $K' = 0$ уже исследованных ранее, принимает вид

$$\varphi(x,t) = q(x,0) \cdot \tilde{K}(t,0) + \int_0^t \xi \cdot S(x,t') \cdot \tilde{K}(t,t') dt' = q(x,0) \cdot e^{-\alpha t} + \int_0^t \xi \cdot e^{-\alpha(t-t')} \cdot S(x,t') dt'. \quad (2.41)$$

Таким образом, основные расчетные формулы метода приняли следующий вид:

$$q(x,t) = \varphi(x,t) - \int_0^t \tilde{K}(t,t') \psi(x,t') dt', \quad (2.42)$$

$$\varphi(x,t) = q(x,0) \cdot e^{-\alpha t} + \int_0^t \xi \cdot e^{-\alpha(t-t')} \cdot S(x,t') dt', \quad (2.43)$$

$$\tilde{K}(t,t') = e^{-\alpha(t-t')}, \quad (2.44)$$

$$\psi(x,t,q',q'') = \beta V q'_x(x,t) - \theta K q''_{xx}(x,t). \quad (2.45)$$

Данный метод реализуется в виде следующей итерационной схемы:

$$q^{(\nu)}(t|x) = \varphi(t|x) - \int_0^t \tilde{K}(t,t') \cdot \psi(x,t',q^{(\nu-1)}(t'|x)) dt'. \quad (2.46)$$

Здесь x фиксировано и играет роль параметра, ν - номер итерации.

2.7 Анализ эффективности предлагаемых методик качественной оценки значений поля концентрации аэрозольных примесей на основе итерационных методов численного решения уравнения переноса

Для вычисления значений выражения $\psi(x, t, q(t|x))$ в (2.46) на множестве функций $q(t|x)$ необходима аппроксимация производных $q'_x(x, t)$ и $q''_{xx}(x, t)$. С этой целью введем равномерную сетку узлов $\{x_i\}$ на интервале $x \in [0; 1]$, $i = \overline{0, m}$. Шаг сетки при этом определяется так: $\Delta x = 1/m$. Внутренние узлы сетки - $\{x_i\}$, $i = \overline{1, m-1}$. Внешние узлы сетки - $\{x_0 = 0\}$, $\{x_m = 1\}$. Далее будем полагать, что на каждом шаге итерации ν вычисляется последовательность $q_k^{(\nu)}(t) = q^{(\nu)}(t|x_k)$, $k = \overline{0, m}$. Это позволяет для каждого ν воспользоваться следующими конечно-разностными аппроксимациями для производных $q'_x(x, t)$ и

$$q''_{xx}(x, t): \left(\frac{\partial q(t|x)}{\partial x} \right)^{(\nu)} = \frac{q_k^{(\nu)}(t) - q_{k-1}^{(\nu)}(t)}{\Delta x}, \quad \left(\frac{\partial^2 q(t|x)}{\partial x^2} \right)^{(\nu)} = \frac{q_{k+1}^{(\nu)}(t) - 2q_k^{(\nu)}(t) + q_{k-1}^{(\nu)}(t)}{\Delta x^2}, \quad k = \overline{1, m-1}.$$

При этом $q_0^{(\nu)}(t) = q(t|x_0 = 0)$ и $q_m^{(\nu)}(t) = q(t|x_m = 1)$ определены граничными условиями. В итоге имеем формулу для расчета значений функции $\psi(x, t, q(x, t))$ в точке $x = x_k$:

$$\psi(x_k, t, q^{(\nu)}(t|x_k)) = \beta V \cdot \frac{q^{(\nu)}(t|x_k) - q^{(\nu)}(t|x_{k-1})}{\Delta x} - \theta K \cdot \frac{q^{(\nu)}(t|x_{k+1}) - 2q^{(\nu)}(t|x_k) + q^{(\nu)}(t|x_{k-1})}{\Delta x^2}. \quad (2.47)$$

Таким образом, вычислительный процесс для итерационной схемы (2.46) полностью определен и его результатом является последовательность функций $\{q_k^{(\nu)}(t)\}$, $\nu = 1, 2, \dots$, $k = \overline{1, m-1}$. Заменяем в выражениях (2.43) - (2.47) функции $q^{(\nu)}(t|x)$, $\tilde{K}(t, t')$, $\varphi(t|x)$ и $\psi(x_k, t, q^{(\nu)}(t|x_k))$ сеточными функциями $q^{(\nu)}(x_k, t_j) = q_{k,j}^{(\nu)}$, $\tilde{K}(t_i, t_j) = \tilde{K}_{i,j}$, $\varphi(x_k, t_j) = \varphi_{k,j}$ и $\psi^{(\nu)}(x_k, t_j) = \psi_{k,j}^{(\nu)}$, (x_k, t_j) - внутренние узлы равномерной сетки $k = \overline{1, m-1}$, $j = \overline{1, n}$, $\Delta x = 1/m$, $\Delta t = 1/n$, $\lambda = \Delta t / \Delta x^2$; $x_k = k \cdot \Delta x$, $k = \overline{0, m}$, $t_j = j \cdot \Delta t$, $j = \overline{0, n}$. Начальными условиями определяются значения $q^{(\nu)}(x_k, t_0 = 0)$, $k = \overline{0, m}$; граничными условиями задаются значения $q^{(\nu)}(x_0 = 0, t_j)$ и $q^{(\nu)}(x_m = 1, t_j)$, $j = \overline{0, n}$. Тогда выражения (2.43) - (2.47) получают следующее представление:

$$\tilde{K}(t_i, t_j) = e^{-\alpha(t_j - t_i)} = \tilde{K}_{i,j}, \quad (2.48)$$

$$\varphi(x_k, t_j) = q(x_k, t_0) e^{(-\alpha_j)} + \xi \sum_{i=0}^j \omega_i S(x_k, t_i) \tilde{K}(t_i, t_j) \Delta t = q_{k,0} e^{(-\alpha_j)} + \xi \sum_{i=0}^j \omega_i S_{k,i} e^{(-\alpha(t_j - t_i))} \Delta t = \varphi_{k,j}, \quad (2.49)$$

$$\begin{aligned} \psi^{(v-1)}(x_k, t_i) &= \beta V \cdot \frac{q^{(v-1)}(x_k, t_i) - q^{(v-1)}(x_{k-1}, t_i)}{\Delta x} - \\ &- \theta K \cdot \frac{q^{(v-1)}(x_{k+1}, t_i) - 2q^{(v-1)}(x_k, t_i) + q^{(v-1)}(x_{k-1}, t_i)}{\Delta x^2} = \psi_{k,i}^{(v-1)}. \end{aligned} \quad (2.50)$$

$$q^{(v)}(x_k, t_j) = \varphi(x_k, t_j) - \sum_{i=0}^j \omega_i \tilde{K}(t_i, t_j) \psi^{(v-1)}(x_k, t_i) \Delta t = \varphi_{k,j} - \sum_{i=0}^j \omega_i e^{(-\alpha(t_j - t_i))} \psi_{k,i}^{(v-1)} \Delta t = q_{k,j}^{(v)}, \quad (2.51)$$

$\nu = 0, 1, 2, \dots$, $q_{k,j}^{(0)} = \varphi_{k,j}$, $k = \overline{1, m-1}$, $j = \overline{1, n}$, интегралы, входящие в выражения (2.43) – (2.47) приближенно заменены интегральными суммами с квадратурными коэффициентами $\{\omega_i\}$, $i = \overline{0, j}$, $j = \overline{1, n}$.

Преобразуем выражения (2.48) – (2.51) следующим образом:

$$\tilde{K}_{i,j} = \exp\{-\alpha(t_j - t_i)\}, \quad (2.52)$$

$$\varphi_{k,j} = q_{k,0} \exp\{-\alpha t_j\} + \xi \lambda \sum_{i=0}^j \omega_i S_{k,i} \exp\{-\alpha(t_j - t_i)\} \Delta x^2 = \varphi(\lambda), \quad (2.53)$$

$$\psi_{k,i}^{(v-1)} = \frac{1}{\Delta x^2} \beta V \Delta x (q_{k,i}^{(v-1)} - q_{k-1,i}^{(v-1)}) - \frac{1}{\Delta x^2} \theta K (q_{k+1,i}^{(v-1)} - 2q_{k,i}^{(v-1)} + q_{k-1,i}^{(v-1)}) = \frac{1}{\Delta x^2} \cdot \tilde{\psi}_{k,i}^{(v-1)}, \quad (2.54)$$

$$q_{k,j}^{(v)} = \varphi_{k,j} - \lambda \sum_{i=0}^j \omega_i \exp\{-\alpha(t_j - t_i)\} \tilde{\psi}_{k,i}^{(v-1)}. \quad (2.55)$$

Проверяется условие сходимости:

$$\rho < \varepsilon, \quad (2.56)$$

$$\rho = \frac{1}{n \cdot (m-1)} \sum_{k=1}^m \sum_{j=1}^n |q_{k,j}^{(v)} - q_{k,j}^{(v-1)}|, \quad (2.57)$$

если $\rho < \varepsilon$, то $\tilde{q}_{k,j} = q_{k,j}^{(v)}$, в противном случае $\nu = \nu + 1$. Сравнение полученного приближенного решения \tilde{q} с точным решением q_T (точное решение получается в «Блоке исходных данных» и считается в данном тестовом примере известным) выполняется по следующей формуле:

$$\sigma = \frac{1}{n \cdot (m-1)} \sum_{k=1}^m \sum_{j=1}^n |(q_T)_{k,j} - \tilde{q}_{k,j}|. \quad (2.58)$$

Основное назначение данной модели – это возможность расчетно – аналитическим путем изучить влияние перечисленных выше параметров модели на пространственно –

временную изменчивость поля концентрации загрязняющих веществ в условиях турбулентной диффузии. Важным здесь является то, что за каждым параметром модели стоит вполне определенный физический фактор, влияющий на формирование поля $q(x, t)$ в области Ω (в пределах локального объема среды). Конечно, модель носит сугубо приближенный характер, и в случае необходимости уточнить тот или иной результат следует обратиться к итерационной схеме (2.36) и найти второе приближение $q^{(2)}(x, t)$ включая при этом уже процедуры численного дифференцирования и интегрирования. Для краткости далее будем называть модель (2.52) – (2.55) второй качественной моделью.

Программная реализация вычислительного алгоритма, соответствующего описанному выше итерационному методу, проводилась при следующих значениях исходных данных: по заданным значениям $V_0 = 10$ м/с, $K_0 = 70$ м²/с, $q_0 = 0.001$ кг/м³, $S_0 = 0.0001$ кг/(м³·с), $\alpha = 0.1$, $\beta = 0.1$, $\theta = 0.001$, $\lambda = 0.5$, определяются значения $\Delta x = 0.04$, $\Delta t = 0.0008$, $m = 10$, $n = 200$, $X = 700$ м, $T = 7$ с, $q_{\max} \approx 0.0001$ кг/м³, $S_{\max} \approx 0.000009$ кг/(м³·с). Значения V_0 , K_0 , q_0 , S_0 выбирались из таблиц 1 – 8 Приложения 2 и соответствуют следующим характеристикам состояния атмосферы: нейтральная стратификация, высота источника $Z = 100$ м, тяжелая примесь, мощно-кучевая облачность (CU-Cong).

Ниже приводятся результаты расчетов в виде следующих массивов – массив $q_T = \{q_{k,j}\}$, $k = \overline{1, m}$, $j = \overline{1, n, 20}$, который представляет собой точное решение уравнения переноса:

$$q_T = \begin{pmatrix} 0.124 & 0.144 & 0.176 & 0.189 & 0.215 & 0.242 & 0.270 & 0.299 & 0.331 & 0.364 & 0.399 \\ 0.124 & 0.145 & 0.167 & 0.191 & 0.216 & 0.243 & 0.272 & 0.302 & 0.334 & 0.367 & 0.402 \\ 0.125 & 0.146 & 0.168 & 0.192 & 0.218 & 0.245 & 0.274 & 0.304 & 0.336 & 0.369 & 0.404 \\ 0.126 & 0.147 & 0.169 & 0.194 & 0.219 & 0.247 & 0.275 & 0.306 & 0.338 & 0.372 & 0.407 \\ 0.127 & 0.148 & 0.171 & 0.195 & 0.221 & 0.248 & 0.277 & 0.308 & 0.340 & 0.374 & 0.409 \\ 0.128 & 0.149 & 0.171 & 0.196 & 0.222 & 0.250 & 0.279 & 0.309 & 0.342 & 0.376 & 0.412 \\ 0.128 & 0.150 & 0.173 & 0.197 & 0.223 & 0.251 & 0.281 & 0.312 & 0.344 & 0.379 & 0.415 \\ 0.129 & 0.151 & 0.174 & 0.198 & 0.225 & 0.253 & 0.282 & 0.314 & 0.347 & 0.381 & 0.417 \\ 0.130 & 0.152 & 0.175 & 0.199 & 0.226 & 0.254 & 0.284 & 0.316 & 0.349 & 0.393 & 0.419 \\ 0.131 & 0.153 & 0.176 & 0.201 & 0.228 & 0.256 & 0.286 & 0.317 & 0.351 & 0.386 & 0.422 \\ 0.132 & 0.153 & 0.177 & 0.202 & 0.229 & 0.257 & 0.288 & 0.319 & 0.353 & 0.388 & 0.425 \end{pmatrix}$$

и массив $\tilde{q}_4 = \{\tilde{q}_{k,j}\}$, $k = \overline{1,m}$, $j = \overline{1,n,20}$, полученный в результате расчетов, проводимых по итерационному алгоритму (2.52) - (2.55):

$$\tilde{q}_4 = \begin{pmatrix} 0.135 & 0.158 & 0.182 & 0.208 & 0.235 & 0.264 & 0.295 & 0.327 & 0.362 & 0.398 & 0.417 \\ 0.137 & 0.160 & 0.185 & 0.212 & 0.240 & 0.269 & 0.301 & 0.334 & 0.368 & 0.404 & 0.422 \\ 0.138 & 0.162 & 0.187 & 0.214 & 0.243 & 0.274 & 0.306 & 0.340 & 0.376 & 0.414 & 0.433 \\ 0.139 & 0.163 & 0.188 & 0.216 & 0.245 & 0.276 & 0.308 & 0.342 & 0.378 & 0.416 & 0.436 \\ 0.140 & 0.163 & 0.189 & 0.217 & 0.246 & 0.277 & 0.310 & 0.345 & 0.381 & 0.419 & 0.439 \\ 0.141 & 0.165 & 0.191 & 0.218 & 0.248 & 0.279 & 0.312 & 0.347 & 0.383 & 0.421 & 0.441 \\ 0.142 & 0.166 & 0.192 & 0.220 & 0.249 & 0.281 & 0.314 & 0.349 & 0.386 & 0.424 & 0.444 \\ 0.142 & 0.167 & 0.193 & 0.221 & 0.251 & 0.283 & 0.316 & 0.351 & 0.388 & 0.427 & 0.447 \\ 0.143 & 0.168 & 0.194 & 0.222 & 0.252 & 0.284 & 0.318 & 0.353 & 0.390 & 0.429 & 0.449 \\ 0.145 & 0.169 & 0.195 & 0.223 & 0.253 & 0.285 & 0.319 & 0.353 & 0.391 & 0.429 & 0.450 \\ 0.145 & 0.170 & 0.195 & 0.224 & 0.253 & 0.286 & 0.320 & 0.353 & 0.391 & 0.430 & 0.451 \end{pmatrix}.$$

Графическое представление данных распределений представлено на рисунках 2.12 и 2.13 соответственно.

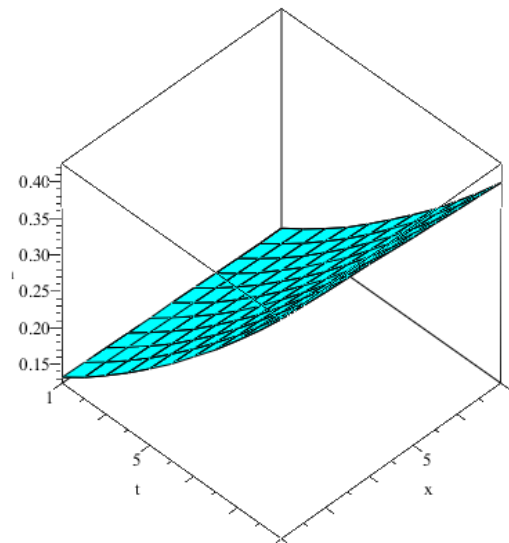


Рисунок 2.12 – Пространственно-временное распределение поля концентрации вредных примесей, соответствующих точному решению $\hat{q}_T(k, j)$

Результаты численного исследования скорости сходимости данного качественного метода приведены в таблице 2.8.

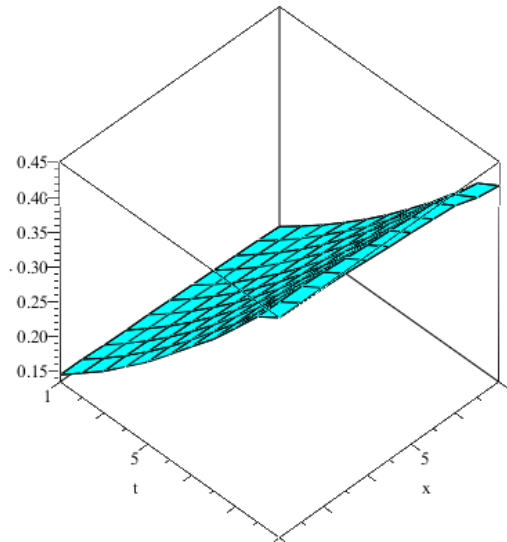


Рисунок 2.13 – Пространственно-временное распределение поля концентрации вредных примесей, соответствующих приближенному решению $\tilde{q}_4(k, j)$, полученному с использованием итерационного алгоритма второй качественной модели одномерного уравнения переноса с погрешностью

$$\sigma_4(\hat{q}_T, \tilde{q}_4) = 0.021$$

Таблица 2.8 – Результаты исследования скорости сходимости итерационного метода, размерность задачи $m \times n = 2000$

ε	ν	ρ	σ	$\delta\tilde{q}$
0.1000	1	0.0011	0.0208	0.101
0.0100	18	0.0009	0.0211	0.104
0.0050	20	0.0004	0.0211	0.105
0.0010	21	0.0002	0.0211	0.106
0.0005	23	0.0001	0.0211	0.106

Примечание: ε - задаваемая точность приближенного решения, ν - количество итераций, ρ - среднее отклонение приближенного решения от точного на текущей и предыдущей итерациях, σ - среднее отклонение приближенного решения от точного решения, $\delta\tilde{q}$ - относительная погрешность метода.

Анализ данных, приведенных в таблице, показывает, что представленный в работе итерационный метод при выбранных исходных данных обладает не очень высокой скоростью сходимости. Здесь величина ρ вычислялась по формуле (2.57), σ - по формуле (2.58). Величина $\delta\tilde{q}$ относительной погрешности метода определялась по формуле

$$\delta\tilde{q} = \frac{\sum_{k=1}^m \sum_{j=1}^n |(q_T)_{k,j} - \tilde{q}_{k,j}|}{\sum_{k=1}^m \sum_{j=1}^n |(q_T)_{k,j}|}$$

и составила 0.11, что для данного качественного метода и соответствующего итерационного алгоритма можно считать вполне приемлемым. Однако, приближенное качественное решение \tilde{q} , полученное в результате прове-

денных расчетов, может быть использовано в виде исходных данных в работе других более точных алгоритмов.

В таблице 2.9 показано, как влияет выбор параметра λ на точность вычислительного процесса, характеризуемую величиной σ , определяемой по формуле (2.58). Расчетные данные таблицы показывают, что уменьшение параметра λ приводит к увеличению размерности сетки узлов и накоплению вычислительных ошибок, хотя и очень незначительному. Также видно, что с увеличением n значение параметра $\sigma_4(q_T, \tilde{q}_4)$ практически не изменяется, то есть вычислительный процесс является стабильным, и поэтому приемлемыми значениями λ при конкретных значениях исходных данных можно считать значения $\lambda = \{0.5, 0.25\}$. Таким образом, из таблицы видно, каким должен быть параметр λ для достижения той или иной точности решения.

Таблица 2.9 – Влияние выбора параметра λ на сходимость итерационного метода, характеризуемую величиной $\sigma_4(q_T, \tilde{q}_4)$

λ	m	n	$m \times n$	σ_4
1/2	10	200	2000	0.021
1/4	10	400	4000	0.043
1/2	15	450	6750	0.062
1/4	15	900	13500	0.091

Примечание: $\lambda = \Delta t / \Delta x^2$ - параметр сетки, $m \times n$ - размерность задачи, σ_4 - среднее отклонение приближенного решения \tilde{q}_4 от точного решения q_T .

Проводимое в работе исследование устойчивости данного итерационного метода проводилось согласно методике, описанной в п. 2.3. Расчетные данные величин, определяемых по формулам (2.11), (2.15), (2.17) и (2.18) и представленные в таблице 2.10, свидетельствуют об устойчивости рассматриваемого в работе метода интегральных уравнений и соответствующей итерационной вычислительной схемы. Из таблицы 2.10 видно, как влияет величина погрешности в исходных данных на устойчивость вычислительной схемы (алгоритма) (2.52) – (2.55) при той или иной размерности задачи. Очевидно, что наиболее приемлемой в этом смысле является размерность сетки, равная $m \times n = 2000$ узлам, при этом максимально допустимая погрешность в исходных данных составляет до 10 % и не выше. Таким образом, вычислительный алгоритм качественной модели оценки концентрации аэрозольных примесей на основе итерационного метода обладает несколько большей устойчивостью к погрешностям в натурных экспериментах по срав-

нению с вычислительной схемой качественной модели на основе фундаментальной системы решений однородного дифференциального уравнения второго порядка с постоянными коэффициентами.

Таблица 2.10 – Влияние величины погрешности в исходных данных на коэффициент усиления ошибки для второй качественной модели

ε	$\sigma_q^{(\delta)}$	$\overline{\delta S}$	$\overline{\delta q}$	η	η_1
$m \times n = 1000$					
0.010	0.237	0.001	0.217	0.003	0.073
0.025	0.236	0.014	0.218	0.067	1.826
0.050	0.235	0.029	0.219	0.133	3.634
0.075	0.235	0.044	0.219	0.199	5.429
0.100	0.236	0.058	0.221	0.264	7.206
$m \times n = 2000$					
0.010	0.796	0.001	0.755	0.001	0.013
0.025	0.798	0.014	0.757	0.018	0.332
0.050	0.801	0.028	0.759	0.037	0.662
0.075	0.802	0.042	0.761	0.055	0.990
0.100	0.805	0.056	0.763	0.073	1.317
$m \times n = 4000$					
0.010	0.305	0.001	0.318	0.002	0.045
0.025	0.306	0.014	0.319	0.045	1.115
0.050	0.306	0.029	0.319	0.091	2.229
0.075	0.307	0.043	0.320	0.136	3.342
0.100	0.308	0.058	0.322	0.181	4.454

Примечание: ε - погрешность, вносимая в источник $S(x, t)$, $\sigma_q^{(\delta)}$ - отклонение приближенного решения от точного, $\overline{\delta S}$ - величина относительной погрешности функции источника $S(x, t)$, $\overline{\delta q}$ - величина относительной погрешности функции распределения $q(x, t)$, η - коэффициент усиления ошибки, η_1 - коэффициент усиления относительной ошибки.

2.8 Вопросы сходимости рекурсивно-итерационных методов и алгоритмов в задаче моделирования переноса примесей в атмосфере

В п. 1.4 приведены известные оценки для оператора шага и источника, обеспечивающие сходимость вычислительного процесса. Для исследования сходимости рекурсивно-итерационных методов и алгоритмов, построенных в диссертации применительно

к задаче моделирования переноса примесей в атмосфере, обратимся далее к уравнению (2.32) и с учетом введенного обозначения $\hat{q}(\hat{x}_k, \hat{t}_j) = \hat{q}_k^{(j)}$ перепишем его в виде:

$$q_k^{(j)} = \mu \sum_{i=1}^k K_{ki} q_i^{(j)} - \mu \sum_{i=1}^k K_{ki} q_i^{(j-1)} + \varphi_k^{(j)}. \quad (2.59)$$

Это выражение по индексу j определяет рекурсивную вычислительную схему. Действительно, при $j=1$ имеем $q_k^{(1)} = \mu \sum_{i=1}^k K_{ki} q_i^{(1)} - \mu \sum_{i=1}^k K_{ki} q_i^{(0)} + \varphi_k^{(1)}$. Значения $\hat{q}_i^{(0)} = \hat{q}(\hat{x}_i, \hat{t} = 0)$ заданы начальными условиями на исходное распределение $q(x, t)$, и потому вектор $\vec{q}^{(1)} = (q_1^{(1)}, q_2^{(1)}, \dots, q_n^{(1)})$ определяется из решения системы линейных уравнений с треугольной матрицей. При любом значении индекса j имеем:

$$\vec{q}^{(j+1)} = \mu K \vec{q}^{(j+1)} - \mu K \vec{q}^{(j)} + \vec{\varphi}^{(j)}, \text{ или } \vec{q}^{(j+1)} = \mu (I - \mu K)^{-1} K \vec{q}^{(j)} + (I - \mu K)^{-1} \vec{\varphi}^{(j)}, \quad j = 0, 1, \dots$$

Построенные рекурсивные схемы указывают на возможность построения операторов шага. Действительно, при дискретизации уравнения (2.31) вместо $\hat{q}(\hat{x}_k, \hat{t}_j)$ в левой его части запишем $\frac{1}{2}(\hat{q}(\hat{x}_k, \hat{t}_{j+1}) + \hat{q}(\hat{x}_k, \hat{t}_j))$, оставляя все остальное без изменения, как и ранее при выводе (2.32), что приводит к следующему векторному выражению

$$\frac{1}{2} \cdot (\vec{q}^{(j+1)} + \vec{q}^{(j)}) = \mu K \vec{q}^{(j+1)} - \mu K \vec{q}^{(j)} + \vec{\varphi}^{(j)},$$

которое можно записать в виде

$$\vec{q}^{(j+1)} = (I - 2\mu K)^{-1} (I + 2\mu K) \vec{q}^{(j)} + 2(I - 2\mu K)^{-1} \vec{\varphi}^{(j)}, \text{ или } \vec{q}^{(j+1)} = T \vec{q}^{(j)} + 2S \vec{\varphi}^{(j)},$$

где обозначено

$$T = (I - 2\mu K)^{-1} (I + 2\mu K), \quad (2.60)$$

$$S = (I - 2\mu K)^{-1}. \quad (2.61)$$

Выражение (2.60) определяет оператор шага в нестационарных задачах (задачах эволюционного типа). Уместно упомянуть, что $\|T\| < 1$. Это вытекает из леммы Келлога [25], в соответствии с которой $\|(I - \sigma A)^{-1} (I - \sigma A)\| < 1$ при $\forall \sigma \geq 0$ и $A \geq 0$. Последнее условие положительной определенности оператора (матрицы) A далеко не тривиально.

Поскольку на каждом временном слое рекурсивных вычислительных схем реализуется итерационный процесс, то выполним исследование его сходимости. Если ввести линейный функционал $\psi(\hat{x}, \hat{t}, \hat{q}) = \hat{q}'_i(\hat{x}, \hat{t})$, как это делалось выше в пределах первой гла-

вы, то рассмотренные ранее итерационные схемы численного решения интегрального уравнения применены в полной мере и к (2.31). Если считать, что модуль $|\hat{q}'_t(\hat{x}, \hat{t})|$ ограничен во всех точках области $\Omega = [0;1] \times [0;1]$, то характер сходимости соответствующего итерационного процесса в полной мере согласуется со сходимостью, присущей Вольтерровским интегральным уравнениям второго рода при их решении методами последовательных приближений. Напомним, что при $\nu \rightarrow \infty$ указанная сходимость определяется величиной $\frac{\lambda^\nu}{\sqrt{\nu!}}$ в метрике L_2 ($\lambda < 1$, $\nu > 0$). Если $\lambda < 1$, то независимо от величины $\frac{1}{\sqrt{\nu!}}$ сходимость гарантирована и носит характер сходимости геометрической прогрессии со знаменателем λ . Характерной особенностью интегральной модели (2.31) является зависимость λ от параметров. Поскольку $\lambda = 1/(\theta(r_2 - r)) = -1/\sqrt{\beta^2 + 4\alpha\theta}$, то условие $|\lambda| < 1$ ведет к неравенству $\sqrt{\beta^2 + 4\alpha\theta} > 1$, которое связывает параметры интегральной модели (2.31) α , β , θ и может служить условием их выбора в тех или иных вычислительных экспериментах. В этом случае критерием выбора параметров модели является скорость сходимости соответствующих итерационных схем численного решения интегрального уравнения (2.31). Здесь также необходимо учитывать и зависимость ядра уравнения $K(\hat{x}, \hat{x}')$ от названных выше параметров, поэтому требуемый выбор параметров модели несколько сложнее и требует проведения соответствующих вычислительных экспериментов.

Кроме того, интегральная модель (2.31) изучена в предположении, что коэффициенты $p_1(\hat{x}, \hat{t})$ и $p_2(\hat{x}, \hat{t})$ являются константами по крайней мере по переменной \hat{x} . Вместе с тем, подобное предположение не является строго обязательным. Определить фундаментальную систему решений $u_1(\hat{x})$ и $u_2(\hat{x})$ однородного уравнения (2.25) и построить затем интеграл $\hat{q}(\hat{x}, C_1, C_2)$ (2.28) зачастую можно и в тех случаях, когда $p_1(\hat{x}, \hat{t})$ и $p_2(\hat{x}, \hat{t})$ не являются константами. В частности эти коэффициенты могут быть полиномами, либо другими элементарными функциями. Важно при этом отметить, что если функции $V(x, t)$ и $K(x, t)$ определены табличными значениями, скажем в узлах (\hat{x}_k, \hat{t}_j) , то, используя методы численного дифференцирования, нетрудно рассчитать значения выражений $(\theta K' - \beta V)/\theta K$ и $-(\alpha + \beta V')/\theta K$ в указанных узлах, а затем для них построить надлежащие аппроксимационные многочлены. Подобное направление расчетно – аналитических ис-

следований следует считать качественным подходом в теории турбулентного переноса субстанции применительно к задачам экологии.

Выводы по главе 2

1. Выполнена разработка тестового примера, осуществлен выбор и обоснование исходных данных, выполнена программная реализация конечно – разностного и итерационного алгоритмов для одномерной параметризованной модели переноса примесей в атмосфере. На этой основе далее выполнено программирование указанных алгоритмов, составляющих методику вычислительного эксперимента. Методика проведения численных исследований включает в себя: выбор значений q_0 , V_0 , K_0 , α_0 , X и T в соответствии со спецификой конкретной прикладной задачи, масштабами и соответствующими диапазонами изменения этих констант, установленными на основе анализа табличных данных, опубликованных в научной печати; моделирование полей исходных данных с пространственно-временной распределенностью по алгоритмам тестовых задач, разработанных в диссертации; нормирование всех переменных и полей, входящих в уравнения математических моделей, построение соответствующих им параметризованных моделей; получение приближенных решений по алгоритмам вычислительных моделей; численные исследования сходимости приближенного решения к точному решению; численное исследование устойчивости вычислительных методов к погрешностям в исходных данных; моделирование решений при различных исходных данных. С учетом полученных результатов вычислительного эксперимента проведено сопоставление вычислительных алгоритмов и сформулированы необходимые рекомендации.

2. На основе фундаментальной системы решений однородного дифференциального уравнения второго порядка с постоянными коэффициентами разработана первая качественная модель оценки концентрации аэрозольных примесей. Идея построения модели состоит в следующем – на основе анализа возможных значений состояния атмосферы, проведенного в первой главе, в параметризованную модель переноса вводятся условия, представляющие собой некие ограничения на значения функции скорости ветра и атмосферной турбулентности. Подобные ограничения позволили свести исходное уравнение переноса к обыкновенному дифференциальному неоднородному уравнению второго порядка при фиксированном значении \hat{t} для которого существует аналитиче-

ское решение. Предложенная модель носит сугубо качественный характер и ее основное назначение состоит в разработке методик качественной оценки значений параметров в задачах математического моделирования явления переноса в турбулентных средах. Построен вычислительный алгоритм модели с последующей программной реализацией, проведен вычислительный эксперимент, выполнена визуализация результатов моделирования.

3. На основе представления искомого решения интегральным уравнением Вольтерра второго рода и метода последовательных приближений разработана вторая качественная модель оценки концентрации аэрозольных примесей. Идея построения модели состоит в том, что первое приближение итерационного процесса, используемого в работе ранее, сведено к достаточно простому аналитическому выражению через упрощение его ядра при условии $V(x,t) = const$, $K(x,t) = const$ и $V'_x = 0$, $K'_x = 0$, а также $\alpha(t) = \alpha = const$, что позволяет последовательно исследовать влияние тех или иных параметров задачи на искомое решение. Основное назначение второй качественной модели заключается в возможности расчетно – аналитическим путем изучить влияние параметров модели на пространственно – временную изменчивость поля концентрации загрязняющих веществ в условиях турбулентной диффузии. Выполнены алгоритмизация модели и программирование, а также визуализация результатов моделирования, реализована предварительно разработанная методика вычислительного эксперимента, получены результаты расчетов и численных исследований. Рассмотрены вопросы сходимости рекурсивно – итерационных вычислительных схем, получены оценки параметров предложенных моделей.

Результаты, полученные в данной главе, опубликованы в работах диссертанта [37, 41, 43, 45, 74, 76, 77, 78, 80, 82, 84, 86, 94].

Глава 3. Разработка решающих алгоритмов для уравнений переноса в рамках итерационно-рекурсивного подхода. Вычислительные эксперименты и их анализ

Глава посвящена разработке вычислительных схем для пространственной задачи переноса и реализации на их основе вычислительного эксперимента по исследованию влияния различных значений скорости ветра, турбулентности и других характеристик состояния пограничного слоя атмосферы на пространственно-временное распределение поля концентрации загрязняющих примесей в атмосфере. Первоначальным этапом является построение тестового примера для параметризованной модели пространственной задачи переноса, на основе которого в заключительной части главы выполняется тестирование алгоритмического и программного обеспечения. Далее в главе выполняется построение вычислительных алгоритмов на основе конечно-разностного и рекурсивно-итерационного методов, ранее разработанных и исследованных для одномерного параметризованного уравнения переноса в предыдущей главе. Отличительной особенностью при этом является то, что трехмерное уравнение переноса рассматривается совместно с уравнением неразрывности, что позволяет построить упрощенную модель по сравнению с той, которая рассмотрена в первой главе. Соответствующие одномерной задаче переноса вычислительные алгоритмы встраиваются в схему по координатному расщеплению исходного уравнения, подробно изложенную в первой главе. В главе излагаются этапы построения вычислительных алгоритмов, для которых затем создается программное обеспечение. На основе разработанного программно-алгоритмического обеспечения проводится вычислительный эксперимент, результаты которого представлены в виде графиков и таблиц, проведен анализ результатов, сделаны соответствующие выводы.

3.1 Тестовые примеры для трехмерных задач теории переноса. Выбор и обоснование исходных данных

Как отмечалось в п.1.4 трехмерное уравнение переноса субстанции в турбулентной среде можно записать в следующем виде [38]:

$$\frac{\partial q(P, t)}{\partial t} + \alpha(t) \cdot q(P, t) + \left(\frac{\partial}{\partial x} (V_x(P, t) \cdot q(P, t)) + \frac{\partial}{\partial y} (V_y(P, t) \cdot q(P, t)) + \frac{\partial}{\partial z} (V_z(P, t) \cdot q(P, t)) \right) - \left(\frac{\partial}{\partial x} \left(K_x(P, t) \cdot \frac{\partial q(P, t)}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y(P, t) \cdot \frac{\partial q(P, t)}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_z(P, t) \cdot \frac{\partial q(P, t)}{\partial z} \right) \right) = S(P, t). \quad (3.1)$$

Как отмечалось ранее в первой главе, для нижней части атмосферы с хорошей точностью выполняется закон сохранения массы, выраженный уравнением неразрывности $\text{div}\vec{V} = 0$. В этой ситуации выражение

$$\frac{\partial}{\partial x}(V_x(P,t) \cdot q(P,t)) + \frac{\partial}{\partial y}(V_y(P,t) \cdot q(P,t)) + \frac{\partial}{\partial z}(V_z(P,t) \cdot q(P,t)),$$

входящее в (3.1), может быть заменено на выражение

$$V_x(P,t) \cdot \frac{\partial q(P,t)}{\partial x} + V_y(P,t) \cdot \frac{\partial q(P,t)}{\partial y} + V_z(t,P) \cdot \frac{\partial q(P,t)}{\partial z}.$$

Последнее дает возможность исходное уравнение (3.1) привести к следующему виду:

$$\begin{aligned} & \frac{\partial q(P,t)}{\partial t} + \alpha(t) \cdot q(P,t) + \left(V_x(P,t) \cdot \frac{\partial q(P,t)}{\partial x} + V_y(P,t) \cdot \frac{\partial q(P,t)}{\partial y} + V_z(P,t) \cdot \frac{\partial q(P,t)}{\partial z} \right) - \\ & - \left(\frac{\partial}{\partial x} \left(K_x(P,t) \cdot \frac{\partial q(P,t)}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_y(P,t) \cdot \frac{\partial q(P,t)}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_z(P,t) \cdot \frac{\partial q(P,t)}{\partial z} \right) \right) = S(P,t), \end{aligned} \quad (3.2)$$

$P = P(x, y, z)$ - точка области пространства, $P \in \Omega \subset R_3$, $t \in [0, T]$, начальные условия:

$$q(P,t)|_{t=0} = q(P,0) = q_0(P) \quad \text{для } P \in \Omega, \quad (3.3)$$

граничные условия:

$$q(P,t) = \bar{q}(P,t) \quad \text{для } P \in \bar{\Omega}, \quad (3.4)$$

$q(P,0) = q_0(P)$ и $\bar{q}(P,t)$ - заданные функции, а $\bar{\Omega}$ - граница области Ω . В уравнении (3.2) $q(P,t)$ - концентрация примесей в точке пространства P в момент времени t (п. 1.4).

Кроме того, в данной постановке задачи полагается, что поле турбулентности, определяемое коэффициентом турбулентной диффузии, задается следующим образом

$$K(P,t) = \begin{bmatrix} K_{xx}(P,t) & 0 & 0 \\ 0 & K_{yy}(P,t) & 0 \\ 0 & 0 & K_{zz}(P,t) \end{bmatrix},$$

где обозначено $K_{xx}(P,t) = K_x(P,t)$, $K_{yy}(P,t) = K_y(P,t)$, $K_{zz}(P,t) = K_z(P,t)$.

Согласно теории метода покоординатного расщепления (п. 1.4) сформулировано следующее утверждение: процесс решения задачи (3.2) - (3.4) может быть представлен в виде трех последовательных подзадач:

$$t_j \leq t \leq t_{j+1}$$

$$\text{Задача I:} \quad \dot{q}_1 + \alpha \cdot q_1 + \frac{\partial}{\partial z}(V_z \cdot q_1) - \frac{\partial}{\partial z} \left(K_z \cdot \frac{\partial q_1}{\partial z} \right) = \omega_1 \cdot S(P,t), \quad (3.5a)$$

$$P_1 = P(z|x, y), \quad P_4 = P(0|x, y), \quad P_5 = P(Z|x, y), \quad q_1(P_1, t=0) = q_0(P_1), \quad \text{если } t=0,$$

$$q_1(P, t_j) = q_3(P, t_{j+1}), \text{ если } t > 0, \quad q_1(P_4, t) = \bar{q}_4(x, y, t), \quad q_1(P_5, t) = \bar{q}_5(x, y, t); \quad (3.5b)$$

$$\text{Задача II:} \quad \dot{q}_2 + \frac{\partial}{\partial x}(V_x \cdot q_2) - \frac{\partial}{\partial x}\left(K_x \frac{\partial q_2}{\partial x}\right) = \omega_2 \cdot S(P, t), \quad (3.6a)$$

$$P_2 = P(x|y, z), \quad P_6 = P(0|y, z), \quad P_7 = P(X|y, z),$$

$$q_2(P, t_j) = q_1(P, t_{j+1}), \quad q_2(P_6, t) = \bar{q}_6(y, z, t), \quad q_2(P_7, t) = \bar{q}_6(y, z, t); \quad (3.6b)$$

$$\text{Задача III:} \quad \dot{q}_3 + \frac{\partial}{\partial y}(V_y \cdot q_3) - \frac{\partial}{\partial y}\left(K_y \cdot \frac{\partial q_3}{\partial y}\right) = \omega_3 \cdot S(P, t), \quad (3.7a)$$

$$P_3 = P(y|x, z), \quad P_8 = P(0|x, z), \quad P_9 = P(Y|x, z), \quad q_3(P, t_j) = q_2(P, t_{j+1}),$$

$$q_3(P_8, t) = \bar{q}_8(x, z, t), \quad q_3(P_9, t) = \bar{q}_9(x, z, t); \quad \omega_1 + \omega_2 + \omega_3 = 1. \quad (3.7b)$$

В качестве решения исходного уравнения (3.2) принимается $q(P, t) = q_3(P, t)$.

Для уравнения (3.2) построение параметризованной модели выполняется согласно алгоритму, рассмотренному в первой главе (п. 1.4) для одномерного уравнения переноса, и включает в себя следующие этапы:

1) нормирование переменных

$$\hat{x} = x/X, \quad \hat{y} = y/Y, \quad \hat{z} = z/Z, \quad \hat{t} = t/T, \quad \hat{x} \in [0,1], \quad \hat{y} \in [0,1], \quad \hat{z} \in [0,1], \quad \hat{t} \in [0,1];$$

2) нормирование функций исходных данных

$$V_i^{(0)}(t) = \max_P V_i(P, t), \quad K_i^{(0)}(t) = \max_P K_i(P, t), \quad S_i^{(0)}(t) = \max_P S_i(P, t),$$

$$\hat{V}_i = V_i/V_i^{(0)}, \quad \hat{K}_i = K_i/K_i^{(0)}, \quad \hat{S}_i = S_i/S_i^{(0)},$$

$$\hat{V}_i(\hat{P}, \hat{t}) \in [0,1], \quad \hat{K}_i(\hat{P}, \hat{t}) \in [0,1], \quad \hat{S}_i(\hat{P}, \hat{t}) \in [0,1], \quad \hat{P} = P(\hat{x}, \hat{y}, \hat{z}),$$

$$\hat{P}_1 = P(\hat{z}|\hat{x}, \hat{y}), \quad \hat{P}_2 = P(\hat{x}|\hat{y}, \hat{z}), \quad \hat{P}_3 = P(\hat{y}|\hat{x}, \hat{z}), \quad i = \{1, 2, 3\},$$

здесь подразумевается соответствие V_1 компоненте поля скорости ветра V_z в направлении оси \vec{Oz} , соответствие V_2 компоненте поля скорости ветра V_x в направлении оси \vec{Ox} , соответствие V_3 компоненте поля скорости ветра V_y в направлении оси \vec{Oy} . То же касается коэффициентов турбулентности, а именно K_1 соответствует K_z , K_2 соответствует K_x и K_3 соответствует K_y ;

3) нормирование начальных и граничных условий

$$q(P, t=0) = q_0(P), \quad q^{(0)} = \max_P q_0(P), \quad \hat{q} = q/q^{(0)}, \quad \hat{q}_k(\hat{P}_k, \hat{t}) = \bar{q}_k(P_k, t)/q^{(0)}, \quad k = \overline{4,9},$$

$$\hat{q}_0(\hat{P}) \in [0,1], \quad \hat{q}(\hat{P}, \hat{t}) \in [0,1], \quad \hat{q}_k(\hat{P}_k, \hat{t}) \in [0,1], \quad \hat{P}_4 = P(0|\hat{x}, \hat{y}), \quad \hat{P}_5 = P(1|\hat{x}, \hat{y}),$$

$$\hat{P}_6 = P(0|\hat{y}, \hat{z}), \hat{P}_7 = P(1|\hat{y}, \hat{z}), \hat{P}_8 = P(0|\hat{x}, \hat{z}), \hat{P}_9 = P(1|\hat{x}, \hat{z});$$

4) вычисление нормировочных коэффициентов – параметров в параметризованной модели

$$\begin{aligned} \hat{\alpha}(\hat{t}) &= \alpha(\hat{t}) \cdot T, \beta_z(\hat{t}) = V_z^{(0)}(t) \cdot T/Z, \theta_z(\hat{t}) = K_z^{(0)}(t) \cdot T/Z^2, \xi_1(\hat{t}) = T \cdot S^{(0)}(t)/q^{(0)}, \\ \beta_x(\hat{t}) &= V_x^{(0)}(t) \cdot T/X, \theta_x(\hat{t}) = K_x^{(0)}(t) \cdot T/X^2, \xi_2(\hat{t}) = \xi_1(\hat{t}), \beta_y(\hat{t}) = V_y^{(0)}(t) \cdot T/Y, \\ \theta_y(\hat{t}) &= K_y^{(0)}(t) \cdot T/Y^2, \xi_3(\hat{t}) = \xi_2(\hat{t}), S_i(P, t) = \omega_i \cdot S(P, t), i = \overline{1,3}. \end{aligned}$$

С учетом нормирования переменных и функций, уравнения (3.5) - (3.7) получают представление:

$$\hat{t}_j \leq \hat{t} \leq \hat{t}_{j+1}$$

Задача I:
$$\frac{\partial \hat{q}_1}{\partial \hat{t}} + \hat{\alpha} \hat{q}_1 + \beta_z \cdot \hat{V}_z \cdot \frac{\partial \hat{q}_1}{\partial \hat{z}} - \theta_z \cdot \frac{\partial}{\partial \hat{z}} \left(\hat{K}_z \cdot \frac{\partial \hat{q}_1}{\partial \hat{z}} \right) = \xi_1 \cdot \hat{S}_1(\hat{P}_1, \hat{t})$$

$$\hat{q}_1(\hat{P}_1, 0) = \hat{q}_0(\hat{P}_1), \text{ если } \hat{t} = 0, \hat{q}_1(\hat{P}, \hat{t}_j) = \hat{q}_3(\hat{P}, \hat{t}_{j+1}),$$

если $\hat{t} > 0, \hat{q}_1(\hat{P}_4, \hat{t}) = \hat{q}_4(\hat{P}_4, \hat{t}), \hat{q}_1(\hat{P}_5, \hat{t}) = \hat{q}_5(\hat{P}_5, \hat{t});$ (3.8)

Задача II:
$$\frac{\partial \hat{q}_2}{\partial \hat{t}} + \beta_x \cdot \hat{V}_x \cdot \frac{\partial \hat{q}_2}{\partial \hat{x}} - \theta_x \cdot \frac{\partial}{\partial \hat{x}} \left(\hat{K}_x \cdot \frac{\partial \hat{q}_2}{\partial \hat{x}} \right) = \xi_2 \cdot \hat{S}_2(\hat{P}_2, \hat{t}),$$

$$\hat{q}_2(\hat{P}, \hat{t}_j) = \hat{q}_1(\hat{P}, \hat{t}_{j+1}), \hat{q}_2(\hat{P}_6, \hat{t}) = \hat{q}_6(\hat{P}_6, \hat{t}), \hat{q}_2(\hat{P}_7, \hat{t}) = \hat{q}_7(\hat{P}_7, \hat{t});$$
 (3.9)

Задача III:
$$\frac{\partial \hat{q}_3}{\partial \hat{t}} + \beta_y \cdot \hat{V}_y \cdot \frac{\partial \hat{q}_3}{\partial \hat{y}} - \theta_y \cdot \frac{\partial}{\partial \hat{y}} \left(\hat{K}_y \cdot \frac{\partial \hat{q}_3}{\partial \hat{y}} \right) = \xi_3 \cdot \hat{S}_3(\hat{P}_3, \hat{t}),$$

$$\hat{q}_3(\hat{P}, \hat{t}_j) = \hat{q}_2(\hat{P}, \hat{t}_{j+1}), \hat{q}_3(\hat{P}_8, \hat{t}) = \hat{q}_8(\hat{P}_8, \hat{t}), \hat{q}_3(\hat{P}_9, \hat{t}) = \hat{q}_9(\hat{P}_9, \hat{t}).$$
 (3.10)

Каждое из уравнений (3.8) - (3.10) является фактически одномерным уравнением, численное решение которого может быть выполнено методами, изложенными [38]. Особенным моментом здесь является переход от решения одного уравнения к другому в общей схеме расщепления с учетом специфики каждого вычислительного алгоритма.

Далее разрабатывается тестовый пример, позволяющий моделировать пространственно-временные распределения исходных данных, начальных и граничных условий, входящих в вычислительные параметризованные модели, разработанные выше. Моделирование распределений предваряется выбором начальных значений для соответствующих полей из таблиц экспериментальных данных, взятых из литературных источников (Приложение 2). На основе разработанного тестового примера далее в главе осуществляется постановка и реализация вычислительного эксперимента. Как отмечалось ранее в

п. 2.1, основная идея построения тестовых примеров состоит в том, что дискретные (измеренные экспериментально) значения исходных распределений, входящих в уравнения переноса (3.8) – (3.10) – концентрацию загрязняющих веществ, скорость ветра, коэффициент турбулентной диффузии задаем непрерывными функциями, аналогично тому, как это делалось в предыдущей главе. Для задания полей скорости ветра, турбулентной диффузии, концентрации примесей были выбраны функции вида:

$$\hat{V}(\hat{x}, \hat{y}, \hat{z}, \hat{t}) = V_0(1 + 0.25 \sin(av \cdot \hat{x})) \cdot (1 + 0.25 \sin(bv \cdot \hat{y})) \cdot (1 + 0.25 \cos(cv \cdot \hat{z})) \cdot (1 + 0.25 \cos(dv \cdot \hat{t})), \quad (3.11)$$

$$\hat{K}(\hat{x}, \hat{y}, \hat{z}, \hat{t}) = K_0(1 + 0.25 \sin(ak \cdot \hat{x})) \cdot (1 + 0.25 \sin(bk \cdot \hat{y})) \cdot (1 + 0.25 \sin(ck \cdot \hat{z})) \cdot (1 + 0.25 \cos(dk \cdot \hat{t})), \quad (3.12)$$

$$\hat{q}(\hat{x}, \hat{y}, \hat{z}, \hat{t}) = q_0(1 + \sin \hat{x}) \cdot (1 + 2 \cos \hat{y}) \cdot (1 + \sin \hat{z}) \cdot (1 + 1.5 \cos \hat{t}); \quad (3.13)$$

где произвольные коэффициенты av , bv , cv , dv , ak , bk , ck , dk подбирались в вычислительном эксперименте таким образом, чтобы определяемая функция источника $\hat{S}(\hat{x}, \hat{y}, \hat{z}, \hat{t})$ принимала только положительные значения. Распределение $\hat{S}(\hat{x}, \hat{y}, \hat{z}, \hat{t})$ на данном этапе считается неизвестным, его вычисление предполагается с помощью соотношений (3.8) – (3.10) – уравнений параметризованной модели.

Здесь необходимо сделать следующие пояснения. Соотношения (3.11) - (3.13) позволяют моделировать пространственно-временные распределения поля скорости ветра в точке $P = P(x, y, z)$, $P \in \Omega \subset R_3$, $t \in [0, T]$, а также для других распределений. При этом реальные величины значений распределений и их размерности определяются константами V_0 , K_0 , q_0 . При построении параметризованной модели (п. 2.1) отмечалось, что существуют определенные границы изменения значений указанных полей для каждого определенного типа состояния атмосферы, например для того или иного типа стратификации атмосферы или типа облачности. В приложении 2 приведены таблицы, взятые из литературных источников, из которых можно выбрать те или иные значения указанных констант, представляющих собой, по сути, некие усредненные значения соответствующих физических характеристик пограничного слоя атмосферы. В результате, с помощью выражений (3.11) – (3.13) удастся фактически осуществить варьирование соответствующих реальных констант в определенных пределах, например, для поля скорости ветра $V_0 \in [V_{\min}, V_{\max}]$, можно рассчитать V_0 следующим образом $V_0 = (V_{\min} + V_{\max})/2$. То же справедливо и для остальных распределений.

Для решения первого уравнения в общей схеме расщепления были выбраны следующие коэффициенты: $av_1 = 2$, $bv_1 = 5$, $cv_1 = 1.5$, $dv_1 = 3$, $ak_1 = 5$, $bk_1 = 1$, $ck_1 = 3$, $dk_1 = 2$;

для второго уравнения соответствующие коэффициенты имели вид: $av_2 = -2$, $bv_2 = 5$, $cv_2 = -1.5$, $dv_2 = 3$, $ak_2 = 5$, $bk_2 = -1$, $ck_2 = 3$, $dk_2 = -2$; и для третьего уравнения $av_3 = -1$, $bv_3 = 0.5$, $cv_3 = -2$, $dv_3 = 1.5$, $ak_3 = 2.5$, $bk_3 = -0.1$, $ck_3 = 2$, $dk_3 = -2$. Для выполнения требуемых вычислений и получения значений искомым распределений по формулам (3.11) – (3.13) введем, как и раньше, равномерную сетку узлов $\{(x_i, y_k, z_l, t_j)\}$ на интервалах $x \in [0; 1]$, $y \in [0; 1]$, $z \in [0; 1]$, $t \in [0; 1]$, $i = \overline{0, m}$, $k = \overline{0, m}$, $l = \overline{0, m}$, $j = \overline{0, n}$ и зададим шаблон, то есть множество узлов точек сетки, участвующих далее в аппроксимации дифференциального выражения. Шаг сетки при этом определяется так: $\{x_i\}$, $i = \overline{0, m}$, $x_i = a + \Delta x \cdot i$, $\Delta x = (b - a)/m$; $\{y_k\}$, $k = \overline{0, m}$, $y_k = a + \Delta y \cdot k$, $\Delta y = (b - a)/m$, $\{z_l\}$, $l = \overline{0, m}$, $z_l = a + \Delta z \cdot l$, $\Delta z = (b - a)/m$, $\{t_j\}$, $j = \overline{0, n}$, $t_j = c + \Delta t \cdot j$, $\Delta t = (d - c)/n$. Выполняя вычисления по формулам (3.11) – (3.13) для каждой такой точки, получим массивы $\hat{V}(\hat{x}_i, \hat{y}_k, \hat{z}_l, \hat{t}_j)$, $\hat{K}(\hat{x}_i, \hat{y}_k, \hat{z}_l, \hat{t}_j)$, $\hat{q}(\hat{x}_i, \hat{y}_k, \hat{z}_l, \hat{t}_j)$ для каждого момента времени t_j , которые обозначим соответственно $\{\hat{x}_i\}$, $\{\hat{y}_k\}$, $\{\hat{z}_l\}$, $\{\hat{t}_j\}$, $\{\hat{q}_{i,k,l,j}\}$, $\{\hat{V}_{i,k,l,j}\}$, $\{\hat{K}_{i,k,l,j}\}$, $i = \overline{0, m}$, $k = \overline{0, m}$, $l = \overline{0, m}$, $j = \overline{0, n}$. Для каждого массива определим начальные и граничные условия, так, как это описано в пункте 2.1. Далее, определив максимальный элемент в каждом массиве, выполним нормирование исходных распределений и вычислим нормировочные коэффициенты параметризованной модели (3.8) – (3.10) (4 этап процесса нормирования уравнений (3.5) - (3.7)).

Для вычисления значений распределения источника загрязняющих веществ обратимся к параметризованной модели (3.8) – (3.10) и подставим в ее уравнения вычисленные значения нормированных коэффициентов и нормированные распределения полей исходных данных. Предварительно введем обозначения для уравнения (3.8):

$$\frac{\partial \hat{q}_1(\hat{P}_1, \hat{t})}{\partial \hat{t}} = u_t(\hat{P}_1, \hat{t}), \quad \frac{\partial \hat{q}_1(\hat{P}_1, \hat{t})}{\partial \hat{z}} = u_z(\hat{P}_1, \hat{t}), \quad \frac{\partial \hat{K}_1(\hat{P}_1, \hat{t})}{\partial \hat{z}} = v_z(\hat{P}_1, \hat{t}), \quad \frac{\partial^2 \hat{q}_1(\hat{P}_1, \hat{t})}{\partial \hat{z}^2} = u_{zz}(\hat{P}_1, \hat{t});$$

для уравнения (3.9):

$$\frac{\partial \hat{q}_2(\hat{P}_2, \hat{t})}{\partial \hat{t}} = u_t(\hat{P}_2, \hat{t}), \quad \frac{\partial \hat{q}_2(\hat{P}_2, \hat{t})}{\partial \hat{x}} = u_x(\hat{P}_2, \hat{t}), \quad \frac{\partial \hat{K}_2(\hat{P}_2, \hat{t})}{\partial \hat{x}} = v_x(\hat{P}_2, \hat{t}), \quad \frac{\partial^2 \hat{q}_2(\hat{P}_2, \hat{t})}{\partial \hat{x}^2} = u_{xx}(\hat{P}_2, \hat{t});$$

для уравнения (3.10):

$$\frac{\partial \hat{q}_3(\hat{P}_3, \hat{t})}{\partial \hat{t}} = u_t(\hat{P}_3, \hat{t}), \quad \frac{\partial \hat{q}_3(\hat{P}_3, \hat{t})}{\partial \hat{y}} = u_y(\hat{P}_3, \hat{t}), \quad \frac{\partial \hat{K}_3(\hat{P}_3, \hat{t})}{\partial \hat{y}} = v_y(\hat{P}_3, \hat{t}), \quad \frac{\partial^2 \hat{q}_3(\hat{P}_3, \hat{t})}{\partial \hat{y}^2} = u_{yy}(\hat{P}_3, \hat{t}).$$

С их учетом параметризованная модель (3.8) – (3.10) получит представление:

$$u_t(\hat{P}_1, \hat{t}) + \alpha \hat{q}_1(\hat{P}_1, \hat{t}) + (\beta_z \cdot \hat{V}_z(\hat{P}_1, \hat{t}) - \theta_z \cdot v_z(\hat{P}_1, \hat{t})) \cdot u_x(\hat{P}_1, \hat{t}) - \theta_z \cdot \hat{K}_z(\hat{P}_1, \hat{t}) \cdot u_{zz}(\hat{P}_1, \hat{t}) = \xi_1 \cdot \hat{S}_1(\hat{P}_1, \hat{t}), \quad (3.14)$$

$$u_t(\hat{P}_2, \hat{t}) + (\beta_x \cdot \hat{V}_x(\hat{P}_2, \hat{t}) - \theta_x \cdot v_x(\hat{P}_2, \hat{t})) \cdot u_x(\hat{P}_2, \hat{t}) - \theta_x \cdot \hat{K}_x(\hat{P}_2, \hat{t}) \cdot u_{xx}(\hat{P}_2, \hat{t}) = \xi_2 \cdot \hat{S}_2(\hat{P}_2, \hat{t}), \quad (3.15)$$

$$u_t(\hat{P}_3, \hat{t}) + (\beta_y \cdot \hat{V}_y(\hat{P}_3, \hat{t}) - \theta_y \cdot v_y(\hat{P}_3, \hat{t})) \cdot u_y(\hat{P}_3, \hat{t}) - \theta_y \cdot \hat{K}_y(\hat{P}_3, \hat{t}) \cdot u_{yy}(\hat{P}_3, \hat{t}) = \xi_3 \cdot \hat{S}_3(\hat{P}_3, \hat{t}). \quad (3.16)$$

Выполняя вычисления согласно (3.14) – (3.16), определим значения правой части каждого из уравнений, то есть значения функции источника $\hat{S}_1(\hat{P}_1, \hat{t})$, $\hat{S}_2(\hat{P}_2, \hat{t})$, $\hat{S}_3(\hat{P}_3, \hat{t})$ и $\hat{S}(\hat{P}, \hat{t})$. Здесь необходимо отметить, что значения поля $\hat{S}(\hat{P}, \hat{t})$ определяются выражением $\hat{S}(\hat{P}, \hat{t}) = \omega_1 \hat{S}_1(\hat{P}, \hat{t}) + \omega_2 \hat{S}_2(\hat{P}, \hat{t}) + \omega_3 \hat{S}_3(\hat{P}, \hat{t})$ в котором значения параметров ω_1 , ω_2 , ω_3 должны определяться в вычислительном эксперименте с учетом значений параметров ξ_1 , ξ_2 и ξ_3 . В результате выполненных расчетов все распределения, начальные и граничные условия, входящие в исходную вычислительную параметризованную модель (3.8) – (3.10) становятся известны. Распределение $\hat{q}(\hat{P}, \hat{t})$ считаем точным решением и обозначим его $\hat{q}_T(\hat{P}, \hat{t})$. Дальнейшая реализация вычислительного алгоритма проводится в предположении о том, что распределение $\hat{q}(\hat{P}, \hat{t})$ не известно и должно определяться в ходе вычислений при известных исходных данных, которое обозначим $\hat{q}_M(\hat{P}, \hat{t})$.

3.2 Построение рекурсивных алгоритмов для уравнения переноса в рамках конечно-разностного подхода

На основе тестового примера, условно называемом в данной работе блоком исходных данных и подробно рассмотренном в предыдущем параграфе, сформированы массивы значений исходных данных, а именно, начальные и граничные условия $\hat{q}(\hat{P}, \hat{t})$, а также значения распределений $\hat{V}_x(\hat{P}, \hat{t})$, $\hat{V}_y(\hat{P}, \hat{t})$, $\hat{V}_z(\hat{P}, \hat{t})$, $\hat{K}_x(\hat{P}, \hat{t})$, $\hat{K}_y(\hat{P}, \hat{t})$, $\hat{K}_z(\hat{P}, \hat{t})$, $\hat{S}_x(\hat{P}, \hat{t})$, $\hat{S}_y(\hat{P}, \hat{t})$, $\hat{S}_z(\hat{P}, \hat{t})$ в узлах сетки $\{\hat{x}_i\}$, $\{\hat{y}_k\}$, $\{\hat{z}_l\}$, $\{\hat{t}_j\}$, $\{\hat{q}_{i,k,l,j}\}$, $\{\hat{V}_{i,k,l,j}\}$, $\{\hat{K}_{i,k,l,j}\}$, $\{\hat{S}_{i,k,l,j}\}$, $i = \overline{0, m}$, $k = \overline{0, m}$, $l = \overline{0, m}$, $j = \overline{0, n}$ и вычислены значения нормировочных констант q_{\max} , V_{\max} , K_{\max} и S_{\max} и параметров модели.

Далее приступим к реализации процедур «Блока моделирования» алгоритма вычислительной модели – нахождению неизвестных значений приближенного решения $\{\hat{q}_{i,k,l,j}\}$ во внутренних узлах сетки, $i = \overline{1, m-1}$, $k = \overline{1, m-1}$, $l = \overline{1, m-1}$, $j = \overline{1, n}$ при извест-

ных значениях исходных данных $\{\hat{x}_i\}$, $\{\hat{y}_k\}$, $\{\hat{z}_l\}$, $\{\hat{t}_j\}$, $\{\hat{q}_{i,k,l,j}\}$, $\{\hat{V}_{i,k,l,j}\}$, $\{\hat{K}_{i,k,l,j}\}$, $\{\hat{S}_{i,k,l,j}\}$, $i = \overline{0,m}$, $k = \overline{0,m}$, $l = \overline{0,m}$, $j = \overline{0,n}$; также известными считаются значения $\{\hat{q}_{i,k,l,0}\}$, $i = \overline{0,m}$, $k = \overline{0,m}$, $l = \overline{0,m}$ - начальные условия и граничные условия $\{\hat{q}_{0,k,l,j+1}\}$, $\{\hat{q}_{m,k,l,j+1}\}$ при $k = \overline{1,m-1}$, $l = \overline{1,m-1}$, $j = \overline{0,n-1}$; $\{\hat{q}_{i,0,l,j+1}\}$, $\{\hat{q}_{i,m,l,j+1}\}$ при $i = \overline{1,m-1}$, $l = \overline{1,m-1}$, $j = \overline{0,n-1}$; $\{\hat{q}_{i,k,0,j+1}\}$, $\{\hat{q}_{i,k,m,j+1}\}$ при $i = \overline{1,m-1}$, $k = \overline{1,m-1}$, $j = \overline{0,n-1}$.

В работе [38] в обобщенном виде разработан подобный алгоритм на основе неявной разностной схемы. В данной работе аналогичный алгоритм строится на основе центральной и шеститочечной разностных схем, имеющих, как известно, более высокую степень точности, а именно, центральная разность вида $\frac{\partial \varphi}{\partial x}(x_i, t_j) \approx \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x}$, имеет порядок аппроксимации $O(\Delta x^2)$; для аппроксимации второй производной используем шеститочечную схему вида $\frac{\partial^2 \varphi}{\partial x^2}(x_i, t_j) \approx \frac{1}{2} \left(\frac{\varphi_{i+1,j+1} - 2\varphi_{i,j+1} + \varphi_{i-1,j+1}}{(\Delta x)^2} + \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{(\Delta x)^2} \right)$, имеющую порядок аппроксимации $O(\Delta x^2 + \Delta t^2)$ (п. 1.5). Кроме того, для упрощения вычислительно алгоритма, положим, что $\alpha(t) = \alpha = const$, $\beta_z(t) = \beta_z = const$, $\beta_x(t) = \beta_x = const$, $\beta_y(t) = \beta_y = const$, $\theta_z(t) = \theta_z = const$, $\theta_x(t) = \theta_x = const$, $\theta_y(t) = \theta_y = const$, $\xi_1(t) = \xi_1 = const$, $\xi_2(t) = \xi_2 = const$, $\xi_3(t) = \xi_3 = const$. С учетом вышесказанного, выражения для аппроксимации производных полей исходных данных в точке (x_i, y_k, z_l, t_j) , входящих в (3.8) – (3.10), получают следующее представление:

$$\frac{\partial q}{\partial t} \approx \frac{q_{i,k,l,j+1} - q_{i,k,l,j}}{\Delta t},$$

$$\frac{\partial q}{\partial x} \approx \frac{q_{i+1,k,l,j+1} - q_{i-1,k,l,j+1}}{2\Delta x}, \quad \frac{\partial q}{\partial y} \approx \frac{q_{i,k+1,l,j+1} - q_{i,k-1,l,j+1}}{2\Delta y}, \quad \frac{\partial q}{\partial z} \approx \frac{q_{i,k,l+1,j+1} - q_{i,k,l-1,j+1}}{2\Delta z},$$

$$\frac{\partial^2 q}{\partial x^2} \approx \frac{1}{2} \left(\frac{q_{i+1,k,l,j+1} - 2q_{i,k,l,j+1} + q_{i-1,k,l,j+1}}{(\Delta x)^2} + \frac{q_{i+1,k,l,j} - 2q_{i,k,l,j} + q_{i-1,k,l,j}}{(\Delta x)^2} \right),$$

$$\frac{\partial^2 q}{\partial y^2} \approx \frac{1}{2} \left(\frac{q_{i,k+1,l,j+1} - 2q_{i,k,l,j+1} + q_{i,k-1,l,j+1}}{(\Delta y)^2} + \frac{q_{i,k+1,l,j} - 2q_{i,k,l,j} + q_{i,k-1,l,j}}{(\Delta y)^2} \right),$$

$$\frac{\partial^2 q}{\partial z^2} \approx \frac{1}{2} \left(\frac{q_{i,k,l+1,j+1} - 2q_{i,k,l,j+1} + q_{i,k,l-1,j+1}}{(\Delta z)^2} + \frac{q_{i,k,l+1,j} - 2q_{i,k,l,j} + q_{i,k,l-1,j}}{(\Delta z)^2} \right).$$

В результате конечно-разностное уравнение для внутренних узлов сетки, соответствующее (3.8), примет вид:

$$\begin{aligned} & \frac{q_1(x_i, y_k, z_l, t_{j+1}) - q_1(x_i, y_k, z_l, t_j)}{\Delta t} + a_z(x_i, y_k, z_l, t_{j+1}) \cdot q_1(x_i, y_k, z_l, t_{j+1}) + \\ & + b_z(x_i, y_k, z_l, t_{j+1}) \cdot \frac{q_1(x_i, y_k, z_{l+1}, t_{j+1}) - q_1(x_i, y_k, z_{l-1}, t_{j+1})}{2\Delta z} - \\ & - c_z(x_i, y_k, z_l, t_{j+1}) \cdot \frac{1}{2} \left(\frac{q_1(x_i, y_k, z_{l+1}, t_{j+1}) - 2q_1(x_i, y_k, z_l, t_{j+1}) + q_1(x_i, y_k, z_{l-1}, t_{j+1})}{\Delta z^2} + \right. \\ & \left. + \frac{q_1(x_i, y_k, z_{l+1}, t_j) - 2q_1(x_i, y_k, z_l, t_j) + q_1(x_i, y_k, z_{l-1}, t_j)}{\Delta z^2} \right) = \xi_1 \cdot S_1(x_i, y_k, z_l, t_{j+1}). \end{aligned} \quad (3.17)$$

В выражении (3.17) предполагается, что все переменные и функции нормированные. Точке пространства $P(x, y, z)$ ставится в соответствие узловая точка $P(x_i, y_k, z_l)$, $i = \overline{0, M}$, $k = \overline{0, K}$, $l = \overline{0, L}$. Коэффициенты в выражении (3.17) имеют представление

$$\begin{aligned} a_z(x_i, y_k, z_l, t_{j+1}) &= \alpha \cdot T, \quad c_z(x_i, y_k, z_l, t_{j+1}) = \theta_z \cdot K_z(x_i, y_k, z_l, t_{j+1}), \\ b_z(x_i, y_k, z_l, t_{j+1}) &= \beta_z \cdot V_z(x_i, y_k, z_l, t_{j+1}) - \theta_z \cdot \frac{K_z(x_i, y_k, z_{l+1}, t_{j+1}) - K_z(x_i, y_k, z_{l-1}, t_{j+1})}{2\Delta z}. \end{aligned}$$

Выражение (3.17) преобразуется к виду

$$\begin{aligned} & \lambda_1 A_1(x_i, y_k, z_l, t_{j+1}) \cdot q_1(x_i, y_k, z_{l-1}, t_{j+1}) - (1 + \lambda_1 B_1(x_i, y_k, z_l, t_{j+1})) \cdot q_1(x_i, y_k, z_l, t_{j+1}) + \\ & + \lambda_1 C_1(x_i, y_k, z_l, t_{j+1}) \cdot q_1(x_i, y_k, z_{l+1}, t_{j+1}) = -(D_1(x_i, y_k, z_l, t_{j+1}) + q_1(x_i, y_k, z_l, t_j) - \\ & - \frac{1}{2} \lambda_1 c_z(x_i, y_k, z_l, t_{j+1}) \cdot (q_1(x_i, y_k, z_{l+1}, t_j) - 2q_1(x_i, y_k, z_l, t_j) + q_1(x_i, y_k, z_{l-1}, t_j))) \end{aligned} \quad (3.18)$$

$$A_1(x_i, y_k, z_l, t_{j+1}) = \frac{1}{2} (b_z(x_i, y_k, z_l, t_{j+1}) \cdot \Delta z + c_z(x_i, y_k, z_l, t_{j+1})),$$

$$B_1(x_i, y_k, z_l, t_{j+1}) = a_z(x_i, y_k, z_l, t_{j+1}) \cdot \Delta z^2 + 2c_z(x_i, y_k, z_l, t_{j+1}),$$

$$C_1(x_i, y_k, z_l, t_{j+1}) = -b_z(x_i, y_k, z_l, t_{j+1}) \cdot \Delta z + c_z(x_i, y_k, z_l, t_{j+1}),$$

$$D_1(x_i, y_k, z_l, t_{j+1}) = \xi_1 S_1(x_i, y_k, z_l, t_{j+1}) \Delta t, \quad \lambda_1 = \Delta t / \Delta z^2, \quad i = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{1, L-1}, \quad j = \overline{0, N-1},$$

$$q_1(x_i, y_k, z_l, 0) = q_0(x_i, y_k, z_l), \quad \text{если } j = 0, \quad i = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{0, L},$$

$$q_1(x_i, y_k, z_l, t_j) = q_3(x_i, y_k, z_l, t_j), \quad \text{если } j > 0, \quad i = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{0, L},$$

$$q_1(x_i, y_k, z_0, t_{j+1}) = \bar{q}_4(x_i, y_k, 0, t_{j+1}), \quad q_1(x_i, y_k, z_L, t_{j+1}) = \bar{q}_5(x_i, y_k, 1, t_{j+1}), \quad i = \overline{0, M}, \quad k = \overline{0, K}.$$

Аналогичным образом строятся конечно-разностные схемы для уравнений (3.9) и (3.10). Соответственно получаем для второго уравнения (3.9):

$$\begin{aligned} & \frac{q_2(x_i, y_k, z_l, t_{j+1}) - q_2(x_i, y_k, z_l, t_j)}{\Delta t} + b_x(x_i, y_k, z_l, t_{j+1}) \cdot \frac{q_2(x_{i+1}, y_k, z_l, t_{j+1}) - q_2(x_{i-1}, y_k, z_l, t_{j+1})}{2\Delta x} \\ & - c_x(x_i, y_k, z_l, t_{j+1}) \cdot \frac{1}{2} \left(\frac{q_2(x_{i+1}, y_k, z_l, t_{j+1}) - 2q_2(x_i, y_k, z_l, t_{j+1}) + q_2(x_{i-1}, y_k, z_l, t_{j+1})}{\Delta x^2} + \right. \\ & \left. + \frac{q_2(x_{i+1}, y_k, z_l, t_j) - 2q_2(x_i, y_k, z_l, t_j) + q_2(x_{i-1}, y_k, z_l, t_j)}{\Delta x^2} \right) = \xi_2 \cdot S_2(x_i, y_k, z_l, t_{j+1}), \end{aligned} \quad (3.19)$$

$$\begin{aligned} b_x(x_i, y_k, z_l, t_{j+1}) &= \beta_x \cdot V_x(x_i, y_k, z_l, t_{j+1}) - \theta_x \cdot \frac{K_x(x_{i+1}, y_k, z_l, t_{j+1}) - K_x(x_{i-1}, y_k, z_l, t_{j+1})}{2\Delta x}, \\ c_x(x_i, y_k, z_l, t_{j+1}) &= \theta_x \cdot K_x(x_i, y_k, z_l, t_{j+1}). \end{aligned}$$

Система линейных алгебраических уравнений, соответствующих (3.19), имеет вид:

$$\begin{aligned} & \lambda_2 A_2(x_i, y_k, z_l, t_{j+1}) \cdot q_2(x_{i-1}, y_k, z_l, t_{j+1}) - (1 + \lambda_2 B_2(x_i, y_k, z_l, t_{j+1})) \cdot q_2(x_i, y_k, z_l, t_{j+1}) + \\ & + \lambda_2 C_2(x_i, y_k, z_l, t_{j+1}) \cdot q_2(x_{i+1}, y_k, z_l, t_{j+1}) = -(D_2(x_i, y_k, z_l, t_{j+1}) + q_2(x_i, y_k, z_l, t_j) - \\ & - \frac{1}{2} \lambda_2 c_x(x_i, y_k, z_l, t_{j+1}) \cdot (q_2(x_{i+1}, y_k, z_l, t_j) - 2q_2(x_i, y_k, z_l, t_j) + q_2(x_{i-1}, y_k, z_l, t_j))) \end{aligned} \quad (3.20)$$

$$A_2(x_i, y_k, z_l, t_{j+1}) = \frac{1}{2} (b_x(x_i, y_k, z_l, t_{j+1}) \cdot \Delta x + c_x(x_i, y_k, z_l, t_{j+1})), \quad B_2(x_i, y_k, z_l, t_{j+1}) = 2c_x(x_i, y_k, z_l, t_{j+1}),$$

$$C_2(x_i, y_k, z_l, t_{j+1}) = -b_x(x_i, y_k, z_l, t_{j+1}) \cdot \Delta x + c_x(x_i, y_k, z_l, t_{j+1}),$$

$$D_2(x_i, y_k, z_l, t_{j+1}) = \xi_2 S_2(x_i, y_k, z_l, t_{j+1}) \Delta t, \quad \lambda_2 = \Delta t / \Delta x^2, \quad i = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{1, L-1}, \quad j = \overline{0, N-1},$$

$$q_2(x_i, y_0, z_l, t_{j+1}) = \bar{q}_6(x_i, 0, z_l, t_{j+1}), \quad q_2(x_i, y_K, z_l, t_{j+1}) = \bar{q}_7(x_i, 1, z_l, t_{j+1}), \quad i = \overline{0, M}, \quad k = \overline{0, K}.$$

Для уравнения (3.10) получим следующую вычислительную схему:

$$\begin{aligned} & \frac{q_3(x_i, y_k, z_l, t_{j+1}) - q_3(x_i, y_k, z_l, t_j)}{\Delta t} + b_y(x_i, y_k, z_l, t_{j+1}) \cdot \frac{q_3(x_i, y_{k+1}, z_l, t_{j+1}) - q_3(x_i, y_{k-1}, z_l, t_{j+1})}{2\Delta y} \\ & - c_y(x_i, y_k, z_l, t_{j+1}) \cdot \frac{1}{2} \left(\frac{q_3(x_i, y_{k+1}, z_l, t_{j+1}) - 2q_3(x_i, y_k, z_l, t_{j+1}) + q_3(x_i, y_{k-1}, z_l, t_{j+1})}{\Delta y^2} + \right. \\ & \left. + \frac{q_3(x_i, y_{k+1}, z_l, t_j) - 2q_3(x_i, y_k, z_l, t_j) + q_3(x_i, y_{k-1}, z_l, t_j)}{\Delta y^2} \right) = \xi_3 \cdot S_3(x_i, y_k, z_l, t_{j+1}). \end{aligned} \quad (3.21)$$

В уравнении (3.21) коэффициенты имеют представление:

$$b_y(x_i, y_k, z_l, t_{j+1}) = \beta_y \cdot V_y(x_i, y_k, z_l, t_{j+1}) - \theta_y \cdot \frac{K_y(x_i, y_{k+1}, z_l, t_{j+1}) - K_y(x_i, y_{k-1}, z_l, t_{j+1})}{2\Delta y},$$

$$c_y(x_i, y_k, z_l, t_{j+1}) = \theta_y \cdot K_y(x_i, y_k, z_l, t_{j+1}).$$

Соответствующая (3.21) система линейных алгебраических уравнений записывается следующим образом:

$$\begin{aligned} & \lambda_3 A_3(x_i, y_k, z_l, t_{j+1}) \cdot q_3(x_{i-1}, y_k, z_l, t_{j+1}) - (1 + \lambda_3 B_3(x_i, y_k, z_l, t_{j+1})) \cdot q_3(x_i, y_k, z_l, t_{j+1}) + \\ & + \lambda_3 C_3(x_i, y_k, z_l, t_{j+1}) \cdot q_3(x_i, y_{k+1}, z_l, t_{j+1}) = -(D_3(x_i, y_k, z_l, t_{j+1}) + q_3(x_i, y_k, z_l, t_j) - \\ & - \frac{1}{2} \lambda_3 c_y(x_i, y_k, z_l, t_{j+1}) \cdot (q_3(x_i, y_{k+1}, z_l, t_j) - 2q_3(x_i, y_k, z_l, t_j) + q_3(x_i, y_{k-1}, z_l, t_j))) \end{aligned} \quad (3.22)$$

В выражении (3.22) коэффициенты определяются соотношениями вида:

$$A_3(x_i, y_k, z_l, t_{j+1}) = \frac{1}{2} (b_y(x_i, y_k, z_l, t_{j+1}) \cdot \Delta y + c_y(x_i, y_k, z_l, t_{j+1})),$$

$$B_3(x_i, y_k, z_l, t_{j+1}) = 2c_y(x_i, y_k, z_l, t_{j+1}), \quad C_3(x_i, y_k, z_l, t_{j+1}) = -b_y(x_i, y_k, z_l, t_{j+1}) \cdot \Delta y + c_y(x_i, y_k, z_l, t_{j+1}),$$

$$D_3(x_i, y_k, z_l, t_{j+1}) = \xi_3 S_3(x_i, y_k, z_l, t_{j+1}) \Delta t, \quad \lambda_3 = \Delta t / \Delta y^2, \quad i = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{1, L-1}, \quad j = \overline{0, N-1},$$

$$q_3(x_0, y_k, z_l, t_{j+1}) = \bar{q}_8(0, y_k, z_l, t_{j+1}), \quad q_3(x_M, y_k, z_l, t_{j+1}) = \bar{q}_9(1, y_k, z_l, t_{j+1}), \quad i = \overline{0, M}, \quad k = \overline{0, K}.$$

Системы линейных алгебраических уравнений (3.18), (3.20) и (3.22) с трехдиагональной симметричной матрицей далее решаются методом прогонки, рассмотренным в п. 1.5 и используемым при решении одномерного нестационарного уравнения переноса в п. 2.2.

3.3 Структура решающего алгоритма для трехмерной задачи переноса в рамках рекурсивно-итерационного подхода

Выполним построение вычислительной схемы решения трехмерного параметризованного уравнения переноса (3.2), в которой итерационные алгоритмы «встраиваются» в схему его покоординатного расщепления. Как отмечалось ранее, каждое уравнение в схеме (3.8) - (3.10) является одномерным с соответствующими начальными и граничными условиями, и к нему можно применить итерационную вычислительную процедуру, подробно изложенную в предыдущей главе. В работе [38] описан подобный обобщенный алгоритм. В настоящей работе принимаются допущения, такие же как и в предыдущем параграфе, а именно $\alpha(t) = \alpha = const$, $\beta_z(t) = \beta_z = const$, $\beta_x(t) = \beta_x = const$, $\beta_y(t) = \beta_y = const$, $\theta_z(t) = \theta_z = const$, $\theta_x(t) = \theta_x = const$, $\theta_y(t) = \theta_y = const$, $\xi_1(t) = \xi_1 = const$, $\xi_2(t) = \xi_2 = const$, $\xi_3(t) = \xi_3 = const$. Это позволяет упростить вычислительный алгоритм.

Кроме того, поскольку в рамках метода расщепления трехмерного уравнения переноса задача решается на каждом интервале $[t_j, t_{j+1}]$, то вычисление интегралов, входящих в основные расчетные формулы итерационно-рекурсивного вычислительного метода, требует значений подынтегральных функций, соответствующих исходным распределениям. Это в свою очередь обуславливает включение в общую схему алгоритмов процедур аппроксимации на заданном интервале исходных функций. В результате в обобщенном виде алгоритм получает представление:

для первого уравнения (3.8):

$$t_j \leq t' \leq t \leq t_{j+1}, \quad q_1(P_1, t) = \varphi_1(P_1, t) - \int_{t_j}^t \tilde{K}(t, t') \psi_1(P_1, t') dt', \quad (3.23)$$

$$\tilde{K}(t, t') = \exp\{-\alpha(t-t')\}, \quad \alpha = \alpha \cdot T,$$

$$\psi_1(P_1, t') = \left(\beta_z V_z(P_1, t') - \theta_z \frac{\partial K_z(P_1, t')}{\partial z} \right) \cdot \frac{\partial q_1(P_1, t')}{\partial z} - \theta_z K_z(P_1, t') \frac{\partial^2 q_1(P_1, t')}{\partial z^2},$$

$$\varphi_1(P_1, t) = q_1(P_1, t_j) \cdot \exp\{-\alpha(t_j - t)\} + \xi_1 \int_{t_j}^t S_1(P_1, t') \tilde{K}(t, t') dt',$$

$$q_1(P_1, t_j) = q_0(P_1), \quad \text{если } j=0, \quad q_1(P_1, t_j) = q_3(P_1, t_{j+1}), \quad \text{если } j>0,$$

$$P = P(x, y, z), \quad P_1 = P(z|x, y), \quad q_1(z_0|x, y, t) = \bar{q}_4(x, y, t), \quad q_1(z_L|x, y, t) = \bar{q}_5(x, y, t).$$

Соответствующий итерационный алгоритм состоит в следующем:

$$q_1^{(v)}(P_1, t) = \varphi_1(P_1, t) - \int_{t_j}^t \tilde{K}(t, t') \psi_1^{(v-1)}(P_1, t', q_1^{(v-1)}(P_1, t')) dt', \quad (3.24)$$

$$\psi_1^{(v-1)}(P_1, t') = \left(\beta_z V_z(P_1, t') - \theta_z \frac{\partial K_z(P_1, t')}{\partial z} \right) \cdot \frac{\partial q_1^{(v-1)}(P_1, t')}{\partial z} - \theta_z K_z(P_1, t') \frac{\partial^2 q_1^{(v-1)}(P_1, t')}{\partial z^2},$$

$$q_1^{(0)} = \varphi_1(P_1, t), \quad v=0,1,2,\dots; \quad \text{если } |q_1^{(v)}(P_1, t) - q_1^{(v-1)}(P_1, t)| \leq \varepsilon, \quad \text{то } q_1(P_1, t) = q_1^{(v)}(P_1, t).$$

Для второго уравнения (3.9) аналогичные построения приводят нас к следующему вычислительному алгоритму:

$$t_j \leq t' \leq t \leq t_{j+1}, \quad q_2(P_2, t) = \varphi_2(P_2, t) - \int_{t_j}^t \psi_2(P_2, t') dt', \quad (3.25)$$

$$\psi_2(P_2, t') = \left(\beta_x V_x(P_2, t') - \theta_x \frac{\partial K_x(P_2, t')}{\partial x} \right) \cdot \frac{\partial q_2(P_2, t')}{\partial x} - \theta_x K_x(P_2, t') \frac{\partial^2 q_2(P_2, t')}{\partial x^2},$$

$$\varphi_2(P_2, t) = q_2(P_2, t_j) + \xi_2 \int_{t_j}^t S_2(P_2, t') dt', \quad q_2(P, t_j) = q_1(P, t_{j+1}),$$

$$P = P(x, y, z), \quad P_2 = P(x|y, z), \quad q_2(x_0|y, z, t) = \bar{q}_6(y, z, t), \quad q_2(x_M|y, z, t) = \bar{q}_7(y, z, t).$$

Итерационная схема имеет вид:

$$q_2^{(\nu)}(P_2, t) = \varphi_2(P_2, t) - \int_{t_j}^t \psi_2^{(\nu-1)}(P_2, t', q_2^{(\nu-1)}(P_2, t')) dt', \quad (3.26)$$

$$\psi_2^{(\nu-1)}(P_2, t') = \left(\beta_x V_x(P_2, t') - \theta_x \frac{\partial K_x(P_2, t')}{\partial x} \right) \cdot \frac{\partial q_2^{(\nu-1)}(P_2, t')}{\partial x} - \theta_x K_x(P_2, t') \frac{\partial^2 q_2^{(\nu-1)}(P_2, t')}{\partial x^2},$$

$$q_2^{(0)} = \varphi_2(P_2, t), \quad \nu = 0, 1, 2, \dots; \quad \text{если } \left| q_2^{(\nu)}(P_2, t) - q_2^{(\nu-1)}(P_2, t) \right| \leq \varepsilon, \text{ то } q_2(P_2, t) = q_2^{(\nu)}(P_2, t).$$

Для третьего уравнения (3.10) имеем:

$$t_j \leq t' \leq t \leq t_{j+1}, \quad q_3(P_3, t) = \varphi_3(P_3, t) - \int_{t_j}^t \psi_3(P_3, t') dt', \quad (3.27)$$

$$\psi_3(P_3, t') = \left(\beta_y V_y(P_3, t') - \theta_y \frac{\partial K_y(P_3, t')}{\partial y} \right) \cdot \frac{\partial q_3(P_3, t')}{\partial y} - \theta_y K_y(P_3, t') \frac{\partial^2 q_3(P_3, t')}{\partial y^2},$$

$$\varphi_3(P_3, t) = q_3(P_3, t_j) + \xi_3 \int_{t_j}^t S_3(P_3, t') dt', \quad q_3(P, t_j) = q_2(P, t_{j+1}),$$

$$P = P(x, y, z), \quad P_3 = P(y|x, z), \quad q_3(y_0|x, z, t) = \bar{q}_8(x, z, t), \quad q_3(y_K|x, z, t) = \bar{q}_9(x, z, t).$$

Итерационная схема имеет представление:

$$q_3^{(\nu)}(P_3, t) = \varphi_3(P_3, t) - \int_{t_j}^t \psi_3^{(\nu-1)}(P_3, t', q_3^{(\nu-1)}(P_3, t')) dt', \quad (3.28)$$

$$\psi_3^{(\nu-1)}(P_3, t') = \left(\beta_y V_y(P_3, t') - \theta_y \frac{\partial K_y(P_3, t')}{\partial y} \right) \cdot \frac{\partial q_3^{(\nu-1)}(P_3, t')}{\partial y} - \theta_y K_y(P_3, t') \frac{\partial^2 q_3^{(\nu-1)}(P_3, t')}{\partial y^2}$$

$$q_3^{(0)} = \varphi_3(P_3, t), \quad \nu = 0, 1, 2, \dots; \quad \text{если } \left| q_3^{(\nu)}(P_3, t) - q_3^{(\nu-1)}(P_3, t) \right| \leq \varepsilon, \text{ то } q_3(P_3, t) = q_3^{(\nu)}(P_3, t).$$

Далее в работе выполняется детальная алгоритмизация (3.23) - (3.28). Заменяя в вычислительной схеме (3.24), (3.26) и (3.28) все функции их сеточными аналогами и ставя в соответствие точке пространства $P = P(x, y, z)$ точку $P(x_m, y_k, z_l)$ на сетке $\{x_m, y_k, z_l\}$, $m = \overline{0, M}$, $k = \overline{0, K}$, $l = \overline{0, L}$ для каждого фиксированного момента времени t_j , $j = \overline{0, N}$, а также, заменяя интегралы суммами, т.е. применяя процедуры численного

интегрирования с соответствующими квадратурными коэффициентами, и включая процедуры аппроксимации исходных данных на временном интервале, окончательно приходим к следующему вычислительному алгоритму решения нестационарного уравнения переноса для случая трех пространственных переменных:

Запуск алгоритма по моментам времени

$$1) j = \overline{0, N-1}; \tilde{N} = N.$$

$$2) h = \frac{\Delta t}{\tilde{N}}.$$

3) Формирование массива квадратурных коэффициентов, соответствующих методу трапеций $\omega_0 = \omega_{\tilde{N}} = 0.5$, $\omega_i = 1$, $i = \overline{1, \tilde{N}-1}$.

4) Первая подзадача

Формирование начальных и граничных условий

$$q_1(x_m, y_k, z_l, t_j) = q_0(x_m, y_k, z_l), \text{ если } j = 0, m = \overline{0, M}, k = \overline{0, K}, l = \overline{0, L};$$

$$q_1(x_m, y_k, z_l, t_j) = q_3(x_m, y_k, z_l, t_{j+1}), \text{ если } j > 0, m = \overline{0, M}, k = \overline{0, K}, l = \overline{0, L};$$

$$q_1(x_m, y_k, z_0, t_i) = \bar{q}_4(x_m, y_k, t_i), \quad q_1(x_m, y_k, z_L, t_i) = \bar{q}_5(x_m, y_k, t_i), \quad m = \overline{0, M}, k = \overline{0, K}.$$

Формирование ядра интегрального уравнения

$$5) \tilde{K}(t_i, t_r) = \exp\{-\alpha \cdot h \cdot (i-r)\}, \quad i = \overline{0, \tilde{N}}.$$

$$6) i = \overline{0, \tilde{N}}.$$

7) Формирование функции $q_1^{(0)}$

$$r = \overline{0, i}, m = \overline{0, M}, k = \overline{0, K}, l = \overline{1, L-1},$$

$$S_1(x_m, y_k, z_l, t_r) = S_1(x_m, y_k, z_l, t_j) + (S_1(x_m, y_k, z_l, t_{j+1}) - S_1(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$\varphi_1(x_m, y_k, z_l, t_i) = q_1(x_m, y_k, z_l, t_j) \cdot \exp\{-\alpha \cdot h \cdot i\} + \xi_1 \cdot \sum_{r=0}^i \omega_r \cdot S_1(x_m, y_k, z_l, t_r) \cdot \tilde{K}(t_i, t_r) \cdot h,$$

$$q_1^{(0)}(x_m, y_k, z_l, t_i) = \varphi_1(x_m, y_k, z_l, t_i).$$

8) Итерационный алгоритм

$$v = v + 1, \quad v = 0, 1, 2, \dots$$

$$V_z(x_m, y_k, z_l, t_r) = V_z(x_m, y_k, z_l, t_j) + (V_z(x_m, y_k, z_l, t_{j+1}) - V_z(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$K_z(x_m, y_k, z_l, t_r) = K_z(x_m, y_k, z_l, t_j) + (K_z(x_m, y_k, z_l, t_{j+1}) - K_z(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$\begin{aligned} \psi_1^{(v-1)}(x_m, y_k, z_l, t_r) = & \left[\beta_z V_z(x_m, y_k, z_l, t_r) - \theta_z \cdot \frac{K_z(x_m, y_k, z_l, t_r) - K_z(x_m, y_k, z_{l-1}, t_r)}{\Delta z} \right] \times \\ & \times \left(\frac{q_1^{(v-1)}(x_m, y_k, z_l, t_r) - q_1^{(v-1)}(x_m, y_k, z_{l-1}, t_r)}{\Delta z} \right) - \theta_z K_z(x_m, y_k, z_l, t_r) \cdot \\ & \cdot \left(\frac{q_1^{(v-1)}(x_m, y_k, z_{l+1}, t_r) - 2q_1^{(v-1)}(x_m, y_k, z_l, t_r) + q_1^{(v-1)}(x_m, y_k, z_{l-1}, t_r)}{(\Delta z)^2} \right), \end{aligned}$$

$$q_1^{(v)}(x_m, y_k, z_l, t_i) = \varphi_1(x_m, y_k, z_l, t_i) - \sum_{r=0}^i \omega_r \cdot \tilde{K}(t_i, t_r) \cdot \psi_1^{(v-1)}(x_m, y_k, z_l, t_r) \cdot h,$$

если $\left| q_1^{(v)}(x_m, y_k, z_l, t_i) - q_1^{(v-1)}(x_m, y_k, z_l, t_i) \right| \leq \varepsilon$, то $q_1(x_m, y_k, z_l, t_i) = q_1^{(v)}(x_m, y_k, z_l, t_i)$.

9) Условия перехода между первой и второй подзадачей

$$q_2(x_m, y_k, z_l, t_j) = q_1(x_m, y_k, z_l, t_{j+1}), \quad m = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{0, L}.$$

10) Вторая подзадача

$$i = \overline{0, \tilde{N}}.$$

Формирование начальных и граничных условий

$$q_2(x_0, y_k, z_l, t_i) = \bar{q}_6(y_k, z_l, t_i), \quad q_2(x_M, y_k, z_l, t_i) = \bar{q}_7(y_k, z_l, t_i), \quad l = \overline{0, L}, \quad k = \overline{0, K}.$$

11) Формирование функции $q_2^{(0)}$

$$l = \overline{0, L}, \quad k = \overline{0, K}, \quad m = \overline{1, M-1},$$

$$S_2(x_m, y_k, z_l, t_r) = S_2(x_m, y_k, z_l, t_j) + (S_2(x_m, y_k, z_l, t_{j+1}) - S_2(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$\varphi_2(x_m, y_k, z_l, t_i) = q_2(x_m, y_k, z_l, t_j) + \xi_2 \sum_{r=0}^i \omega_r \cdot S_2(x_m, y_k, z_l, t_r) \cdot h,$$

$$q_2^{(0)}(x_m, y_k, z_l, t_i) = \varphi_2(x_m, y_k, z_l, t_i), \quad \nu = 0, 1, 2, \dots$$

12) Итерационный алгоритм

$$\nu = \nu + 1, \quad \nu = 0, 1, 2, \dots$$

$$r = \overline{0, i}, \quad k = \overline{0, K}, \quad l = \overline{0, L}, \quad m = \overline{1, M-1},$$

$$V_x(x_m, y_k, z_l, t_r) = V_x(x_m, y_k, z_l, t_j) + (V_x(x_m, y_k, z_l, t_{j+1}) - V_x(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$K_x(x_m, y_k, z_l, t_r) = K_x(x_m, y_k, z_l, t_j) + (K_x(x_m, y_k, z_l, t_{j+1}) - K_x(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$\psi_2^{(v-1)}(x_m, y_k, z_l, t_r) = \left[\beta_x V_x(x_m, y_k, z_l, t_r) - \theta_x \cdot \frac{K_x(x_m, y_k, z_l, t_r) - K_x(x_{m-1}, y_k, z_l, t_r)}{\Delta x} \right] \times$$

$$\times \left(\frac{q_2^{(v-1)}(x_m, y_k, z_l, t_r) - q_2^{(v-1)}(x_{m-1}, y_k, z_l, t_r)}{\Delta x} \right) - \theta_x K_x(x_m, y_k, z_l, t_r) \cdot$$

$$\cdot \left(\frac{q_2^{(v-1)}(x_{m+1}, y_k, z_l, t_r) - 2q_2^{(v-1)}(x_m, y_k, z_l, t_r) + q_2^{(v-1)}(x_{m-1}, y_k, z_l, t_r)}{(\Delta x)^2} \right),$$

$$q_2^{(v)}(x_m, y_k, z_l, t_i) = \varphi_2(x_m, y_k, z_l, t_i) - \sum_{r=0}^i \omega_r \psi_2^{(v-1)}(x_m, y_k, z_l, t_r) \cdot h,$$

если $\left| q_2^{(v)}(x_m, y_k, z_l, t_i) - q_2^{(v-1)}(x_m, y_k, z_l, t_i) \right| \leq \varepsilon$, то $q_2(x_m, y_k, z_l, t_i) = q_2^{(v)}(x_m, y_k, z_l, t_i)$.

13) Условия перехода между второй и третьей подзадачей

$$q_2(x_m, y_k, z_l, t_j) = q_2(x_m, y_k, z_l, t_{j+1}), \quad m = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{0, L}.$$

14) Третья подзадача

$$i = \overline{0, \tilde{N}}.$$

Формирование начальных и граничных условий

$$q_3(x_m, y_0, z_l, t_i) = \bar{q}_8(x_m, z_l, t_i), \quad q_3(x_m, y_K, z_l, t_i) = \bar{q}_9(x_m, z_l, t_i), \quad l = \overline{0, L}, \quad m = \overline{0, M}.$$

15) Формирование функции $q_3^{(0)}$

$$l = \overline{0, L}, \quad m = \overline{0, M}, \quad k = \overline{1, K-1},$$

$$S_3(x_m, y_k, z_l, t_r) = S_3(x_m, y_k, z_l, t_j) + (S_3(x_m, y_k, z_l, t_{j+1}) - S_3(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$\varphi_3(x_m, y_k, z_l, t_i) = q_3(x_m, y_k, z_l, t_j) + \xi_3 \sum_{r=0}^i \omega_r \cdot S_3(x_m, y_k, z_l, t_r) \cdot h,$$

$$q_3^{(0)}(x_m, y_k, z_l, t_i) = \varphi_3(x_m, y_k, z_l, t_i).$$

16) Итерационный алгоритм

$$\nu = \nu + 1, \quad \nu = 0, 1, 2, \dots$$

$$r = \overline{0, i}, \quad m = \overline{0, M}, \quad l = \overline{0, L}, \quad k = \overline{1, K-1},$$

$$V_y(x_m, y_k, z_l, t_r) = V_y(x_m, y_k, z_l, t_j) + (V_y(x_m, y_k, z_l, t_{j+1}) - V_y(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$K_y(x_m, y_k, z_l, t_r) = K_y(x_m, y_k, z_l, t_j) + (K_y(x_m, y_k, z_l, t_{j+1}) - K_y(x_m, y_k, z_l, t_j)) \cdot h \cdot r,$$

$$\psi_3^{(v-1)}(x_m, y_k, z_l, t_r) = \left[\beta_y V_y(x_m, y_k, z_l, t_r) - \theta_y \cdot \frac{K_y(x_m, y_k, z_l, t_r) - K_y(x_m, y_{k-1}, z_l, t_r)}{\Delta y} \right] \times$$

$$\times \left(\frac{q_3^{(v-1)}(x_m, y_k, z_l, t_r) - q_3^{(v-1)}(x_m, y_{k-1}, z_l, t_r)}{\Delta y} \right) - \theta_y K_y(x_m, y_k, z_l, t_r).$$

$$\left(\frac{q_3^{(v-1)}(x_m, y_{k+1}, z_l, t_r) - 2q_3^{(v-1)}(x_m, y_k, z_l, t_r) + q_3^{(v-1)}(x_m, y_{k-1}, z_l, t_r)}{(\Delta y)^2} \right),$$

$$q_3^{(v)}(x_m, y_k, z_l, t_i) = \varphi_3(x_m, y_k, z_l, t_i) - \sum_{r=0}^i \omega_r \psi_3^{(v-1)}(x_m, y_k, z_l, t_r) \cdot h,$$

если $\left| q_3^{(v)}(x_m, y_k, z_l, t_i) - q_3^{(v-1)}(x_m, y_k, z_l, t_i) \right| \leq \varepsilon$, то $q_3(x_m, y_k, z_l, t_i) = q_3^{(v)}(x_m, y_k, z_l, t_i)$.

17) Условия перехода между третьей и первой подзадачей

$$q_1(x_m, y_k, z_l, t_j) = q_3(x_m, y_k, z_l, t_{j+1}), \quad m = \overline{0, M}, \quad k = \overline{0, K}, \quad l = \overline{0, L}.$$

18) Если $j = N$, то $q(x_m, y_k, z_l, t_j) = q_3(x_m, y_k, z_l, t_j)$, $m = \overline{0, M}$, $k = \overline{0, K}$, $l = \overline{0, L}$.

19) Конец алгоритма.

3.4 Результаты численных экспериментов и их анализ на основе созданного программного обеспечения для задач моделирования нестационарного переноса примесей в атмосфере

Программная реализация разработанных в п. 3.2 и п. 3.3 рекурсивного алгоритма на основе конечно – разностного метода и рекурсивно – итерационной схемы, в которой итерационные алгоритмы «встраиваются» в схему покоординатного расщепления (3.8) – (3.10), проводилась при следующих значениях исходных данных: по заданным значениям $q_0 = 0.01 \text{ кг/м}^3$, $\alpha = 0.02 \text{ 1/с}$, $\beta_x = 1.75$, $\beta_y = 1.75$, $\beta_z = 1.75$, $\theta_x = 0.03$, $\theta_y = 0.03$, $\theta_z = 0.03$, $\lambda_1 = 0.5$, $\lambda_2 = 0.5$, $\lambda_3 = 0.5$, $M = 10$, $K = 10$, $L = 10$, $N = 10$, $V_{x,0} = 6 \text{ м/с}$, $V_{y,0} = 7 \text{ м/с}$, $V_{z,0} = 5 \text{ м/с}$, $K_{x,0} = 35 \text{ м}^2/\text{с}$, $K_{y,0} = 45 \text{ м}^2/\text{с}$, $K_{z,0} = 40 \text{ м}^2/\text{с}$ вычислялись значения величин для первого уравнения $X_1 = 411 \text{ м}$, $T_1 = 66 \text{ с}$, $V_{\min 1} = 4.6 \text{ м/с}$, $V_{\max 1} = 11.9 \text{ м/с}$, $K_{\min 1} = 37.3 \text{ м}^2/\text{с}$, $K_{\max 1} = 92.4 \text{ м}^2/\text{с}$, $S_{\min 1} = 0.00015 \text{ кг/}(\text{м}^3 \cdot \text{с})$, $S_{\max 1} = 0.0008 \text{ кг/}(\text{м}^3 \cdot \text{с})$, для второго уравнения - $X_2 = 395 \text{ м}$, $T_2 = 52 \text{ с}$, $V_{\min 2} = 4.2 \text{ м/с}$, $V_{\max 2} = 11.5 \text{ м/с}$, $K_{\min 2} = 25.8 \text{ м}^2/\text{с}$, $K_{\max 2} = 66.8 \text{ м}^2/\text{с}$, $S_{\min 2} = 0.00014 \text{ кг/}(\text{м}^3 \cdot \text{с})$, $S_{\max 2} = 0.0006 \text{ кг/}(\text{м}^3 \cdot \text{с})$, и для третьего уравнения - $X_3 = 360 \text{ м}$, $T_3 = 63 \text{ с}$, $V_{\min 3} = 5.2 \text{ м/с}$, $V_{\max 3} = 10.5 \text{ м/с}$, $K_{\min 3} = 41.8 \text{ м}^2/\text{с}$, $K_{\max 3} = 68.3 \text{ м}^2/\text{с}$, $S_{\min 3} = 0.00012 \text{ кг/}(\text{м}^3 \cdot \text{с})$, $S_{\max 3} = 0.0008 \text{ кг/}(\text{м}^3 \cdot \text{с})$; значения для концентрации загрязняющих примесей $q_{\min} = 0.001 \text{ кг/м}^3$,

$q_{\max} = 0.007 \text{ кг/м}^3$. Выбранные исходные данные (значения метеорологических параметров) соответствуют следующему состоянию атмосферы в пределах пограничного слоя: нейтральная стратификация, высота источника $Z = 500 \text{ м}$, тяжелая примесь, мощно-кучевая облачность (CU-Cong) (Приложение 2).

Точность расчетов характеризуется величиной отклонения приближенного решения $\hat{q}_M(i, k, l, j+1)$ от точного $\hat{q}_T(i, k, l, j+1)$, генерируемого в «Блоке исходных данных», по формуле

$$(\sigma_{\text{отн}})_{j+1} = \sqrt{\frac{\sum_i \sum_k \sum_l ((\hat{q}_T)_{i,k,l,j+1} - (\hat{q}_M)_{i,k,l,j+1})^2}{\sum_i \sum_k \sum_l ((\hat{q}_T)_{i,k,l,j+1})^2}}. \quad (3.29)$$

Величина отклонения приближенного решения $\hat{q}_M(i, k)$ от точного $\hat{q}_T(i, k)$ для момента времени $\hat{t} = 1.0$ составила $\sigma(j=10) = 0.18$, итоговое значение погрешности (усредненное по времени) определялось по формуле

$$\sigma_{\text{отн}} = \frac{1}{n} \sum_{j=1}^n (\sigma_{\text{отн}})_j \quad (3.30)$$

и составило $\sigma = 0.12$.

Ниже приведены результаты расчетов по рекурсивно-итерационному алгоритму в виде массивов – массив $\hat{q}_T(i, k)$, $i = \overline{0, M}$, $k = \overline{0, K}$ который представляет собой точное решение уравнения переноса (массивы представлены при фиксированных значениях $l = 5$ (данное значение соответствует высоте 250 м.) и $\hat{t}(j=10) = 1.0$):

$$\hat{q}_T = \begin{pmatrix} 0.472 & 0.468 & 0.463 & 0.455 & 0.447 & 0.436 & 0.425 & 0.412 & 0.398 & 0.382 & 0.366 \\ 0.501 & 0.496 & 0.491 & 0.483 & 0.474 & 0.463 & 0.451 & 0.437 & 0.423 & 0.406 & 0.389 \\ 0.529 & 0.524 & 0.519 & 0.510 & 0.501 & 0.489 & 0.477 & 0.462 & 0.447 & 0.429 & 0.411 \\ 0.557 & 0.552 & 0.546 & 0.537 & 0.528 & 0.515 & 0.502 & 0.487 & 0.471 & 0.452 & 0.432 \\ 0.585 & 0.579 & 0.573 & 0.564 & 0.554 & 0.541 & 0.527 & 0.511 & 0.493 & 0.474 & 0.453 \\ 0.611 & 0.605 & 0.599 & 0.589 & 0.579 & 0.565 & 0.550 & 0.534 & 0.515 & 0.495 & 0.474 \\ 0.637 & 0.631 & 0.624 & 0.614 & 0.603 & 0.588 & 0.573 & 0.556 & 0.537 & 0.516 & 0.494 \\ 0.666 & 0.657 & 0.647 & 0.636 & 0.625 & 0.610 & 0.595 & 0.576 & 0.558 & 0.535 & 0.512 \\ 0.694 & 0.683 & 0.670 & 0.658 & 0.647 & 0.632 & 0.616 & 0.596 & 0.577 & 0.554 & 0.530 \\ 0.710 & 0.701 & 0.691 & 0.679 & 0.666 & 0.651 & 0.635 & 0.615 & 0.595 & 0.571 & 0.547 \\ 0.726 & 0.719 & 0.711 & 0.699 & 0.687 & 0.670 & 0.653 & 0.633 & 0.612 & 0.588 & 0.563 \end{pmatrix}$$

и массив $\hat{q}_M(i, k)$, $i = \overline{0, M}$, $k = \overline{0, K}$ - приближенное решение

$$\hat{q}_M = \begin{pmatrix} 0.567 & 0.582 & 0.597 & 0.596 & 0.594 & 0.584 & 0.573 & 0.556 & 0.542 & 0.520 & 0.498 \\ 0.619 & 0.639 & 0.661 & 0.660 & 0.659 & 0.648 & 0.637 & 0.618 & 0.599 & 0.574 & 0.548 \\ 0.670 & 0.697 & 0.724 & 0.725 & 0.726 & 0.712 & 0.700 & 0.679 & 0.657 & 0.628 & 0.598 \\ 0.712 & 0.742 & 0.772 & 0.774 & 0.775 & 0.760 & 0.746 & 0.722 & 0.698 & 0.666 & 0.634 \\ 0.754 & 0.787 & 0.820 & 0.822 & 0.824 & 0.808 & 0.792 & 0.765 & 0.738 & 0.704 & 0.669 \\ 0.791 & 0.826 & 0.861 & 0.863 & 0.865 & 0.847 & 0.829 & 0.801 & 0.772 & 0.736 & 0.698 \\ 0.827 & 0.864 & 0.901 & 0.903 & 0.905 & 0.886 & 0.867 & 0.837 & 0.806 & 0.767 & 0.727 \\ 0.861 & 0.899 & 0.937 & 0.938 & 0.941 & 0.921 & 0.899 & 0.868 & 0.836 & 0.795 & 0.753 \\ 0.894 & 0.934 & 0.973 & 0.974 & 0.975 & 0.954 & 0.932 & 0.899 & 0.865 & 0.822 & 0.778 \\ 0.921 & 0.961 & 1.002 & 1.001 & 1.001 & 0.979 & 0.957 & 0.922 & 0.886 & 0.842 & 0.797 \\ 0.948 & 0.988 & 1.026 & 1.027 & 1.028 & 1.005 & 0.981 & 0.945 & 0.908 & 0.862 & 0.816 \end{pmatrix}.$$

На рисунках 3.1 и 3.2 представлены соответствующие этим массивам графические изображения точного решения и приближенного решения соответственно.

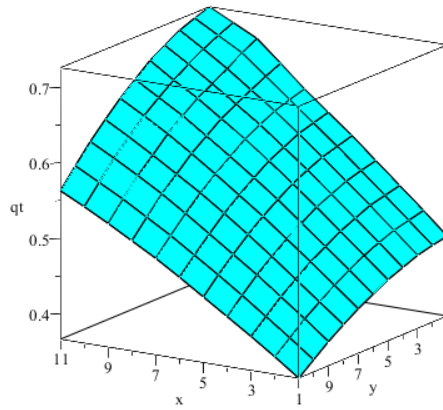


Рисунок 3.1 – Пространственно – временное распределение поля концентрации вредных примесей, соответствующих точному решению $\hat{q}_T(i, k)$ (значения нормированные) при $l = 5$ и $j = 10$

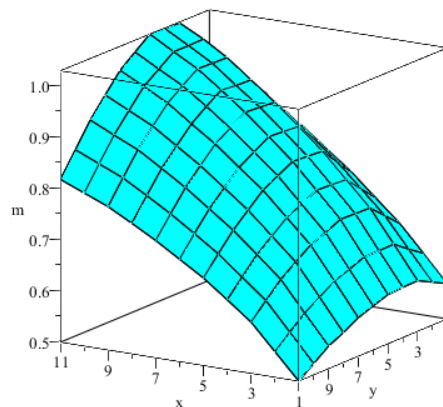


Рисунок 3.2 – Пространственно – временное распределение поля концентрации вредных примесей, соответствующих приближенному решению $\hat{q}_M(i, k)$ (значения нормированные) при фиксированных $l = 5$ и $j = 10$, с погрешностью 0.12

Также приведем результаты расчетов в виде массивов – массива $\hat{q}_T(i, k)$, $i = \overline{0, M}$, $k = \overline{0, K}$ - точного решения уравнения переноса (массив получен при фиксированных значениях $l = 10$ (изменение высоты, что соответствует высоте 500 м.) и $\hat{t}(j = 10) = 1.0$):

$$\hat{q}_T = \begin{pmatrix} 0.586 & 0.580 & 0.574 & 0.564 & 0.554 & 0.541 & 0.527 & 0.511 & 0.494 & 0.474 & 0.454 \\ 0.619 & 0.614 & 0.609 & 0.598 & 0.588 & 0.574 & 0.559 & 0.542 & 0.524 & 0.503 & 0.482 \\ 0.652 & 0.648 & 0.644 & 0.633 & 0.622 & 0.607 & 0.592 & 0.573 & 0.554 & 0.532 & 0.509 \\ 0.689 & 0.684 & 0.678 & 0.666 & 0.655 & 0.639 & 0.623 & 0.603 & 0.583 & 0.560 & 0.536 \\ 0.726 & 0.719 & 0.711 & 0.699 & 0.687 & 0.670 & 0.653 & 0.633 & 0.612 & 0.588 & 0.563 \\ 0.758 & 0.751 & 0.743 & 0.730 & 0.717 & 0.699 & 0.682 & 0.661 & 0.639 & 0.614 & 0.588 \\ 0.790 & 0.782 & 0.774 & 0.761 & 0.747 & 0.729 & 0.711 & 0.689 & 0.666 & 0.639 & 0.612 \\ 0.819 & 0.811 & 0.803 & 0.789 & 0.775 & 0.757 & 0.738 & 0.714 & 0.691 & 0.663 & 0.635 \\ 0.849 & 0.840 & 0.831 & 0.817 & 0.803 & 0.784 & 0.764 & 0.739 & 0.715 & 0.687 & 0.658 \\ 0.875 & 0.866 & 0.857 & 0.842 & 0.828 & 0.808 & 0.788 & 0.762 & 0.737 & 0.708 & 0.678 \\ 0.900 & 0.891 & 0.882 & 0.867 & 0.852 & 0.832 & 0.811 & 0.785 & 0.759 & 0.729 & 0.698 \end{pmatrix}$$

и массива $\hat{q}_M(i, k)$, $i = \overline{0, M}$, $k = \overline{0, K}$ - приближенного решения уравнения переноса

$$\hat{q}_M = \begin{pmatrix} 0.649 & 0.656 & 0.662 & 0.657 & 0.651 & 0.639 & 0.626 & 0.608 & 0.590 & 0.567 & 0.544 \\ 0.701 & 0.712 & 0.722 & 0.718 & 0.712 & 0.699 & 0.684 & 0.659 & 0.644 & 0.618 & 0.591 \\ 0.753 & 0.768 & 0.782 & 0.778 & 0.773 & 0.758 & 0.742 & 0.719 & 0.697 & 0.668 & 0.638 \\ 0.798 & 0.815 & 0.832 & 0.833 & 0.823 & 0.806 & 0.789 & 0.764 & 0.739 & 0.707 & 0.675 \\ 0.843 & 0.862 & 0.881 & 0.877 & 0.872 & 0.854 & 0.835 & 0.808 & 0.781 & 0.746 & 0.711 \\ 0.884 & 0.904 & 0.924 & 0.919 & 0.914 & 0.895 & 0.874 & 0.845 & 0.816 & 0.779 & 0.741 \\ 0.924 & 0.945 & 0.966 & 0.961 & 0.956 & 0.935 & 0.913 & 0.882 & 0.851 & 0.812 & 0.773 \\ 0.961 & 0.982 & 1.004 & 0.998 & 0.992 & 0.971 & 0.947 & 0.915 & 0.882 & 0.841 & 0.812 \\ 0.996 & 1.018 & 1.041 & 1.035 & 1.029 & 1.005 & 0.981 & 0.947 & 0.912 & 0.869 & 0.827 \\ 1.026 & 1.047 & 1.073 & 1.065 & 1.057 & 1.032 & 1.007 & 0.972 & 0.936 & 0.892 & 0.848 \\ 1.055 & 1.078 & 1.101 & 1.094 & 1.086 & 1.059 & 1.033 & 0.996 & 0.959 & 0.914 & 0.869 \end{pmatrix}.$$

Соответствующие этим массивам графические изображения точного и приближенного решений представлены на рисунках 3.3 и 3.4.

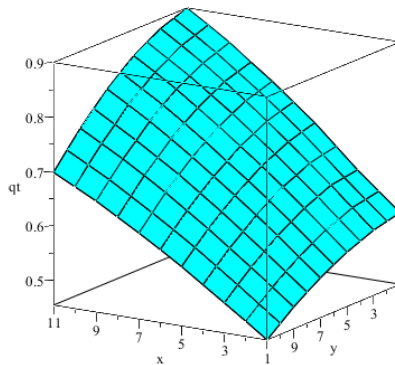


Рисунок 3.3 – Пространственно – временное распределение поля концентрации вредных примесей, соответствующих точному решению $\hat{q}_T(i, k)$ (значения нормированные) при $l = 10$ и $j = 10$

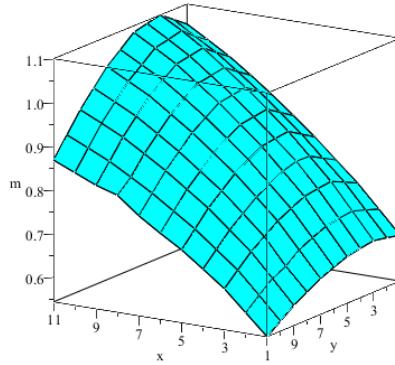


Рисунок 3.4 – Пространственно – временное распределение поля концентрации вредных примесей, соответствующих приближенному решению $\hat{q}_M(i,k)$ (значения нормированные) при фиксированных $l = 10$ и $j = 10$, с погрешностью 0.12

В таблице 3.1 приведены результаты расчетов при различных значениях размерности задачи (на примере конечно-разностного алгоритма).

Таблица 3.1 – Влияние размерности задачи на точность решения трехмерного уравнения переноса с использованием рекурсивного алгоритма на основе конечно – разностного метода

j - номер временного слоя	$(\sigma_{\text{отн}})_j$ - относительная погрешность (отклонение приближенного решения \hat{q}_M от точного \hat{q}_T)
$M = K = L = 5, N = 10$	
0	0.039
5	0.162
10	0.207
$M = K = L = 7, N = 10$	
0	0.022
5	0.111
10	0.161
$M = K = L = 5, N = 15$	
0	0.036
5	0.148
10	0.196
15	0.213
$M = K = L = 10, N = 10$	
0	0.033
5	0.135
10	0.178

Анализ расчетных значений погрешности вычислений показывает не высокую точность, хотя и приемлемую, с учетом многомерности и нестационарности исходной задачи. Очевидно, что в данном случае необходимо было бы реализовать технологию распараллеливания вычислительного процесса, что должно явиться в свою очередь следующим шагом в решении указанных проблем.

Разработанное программное обеспечение позволяет моделировать процесс нестационарного переноса примесей в атмосфере с учетом различных состояний атмосферы в пограничном слое. На основе данных мониторинга, приведенных в приложении 2, осуществлялся выбор значений соответствующих параметров. Так, при следующих значениях исходных данных $X = 648$ м, $T = 60$ с, $V_{x,0} = 10.8$ м/с, $V_{y,0} = 10.8$ м/с, $V_{z,0} = 10.8$ м/с, $K_{x,0} = 40$ м²/с, $K_{y,0} = 40$ м²/с, $K_{z,0} = 4$ м²/с, $\alpha = 0.013$ с⁻¹, $q_0 = 0.005$ кг/м³, что соответствует условиям состояния атмосферы: нейтральная стратификация, высота источника $Z = 100$ м, тяжелая примесь, слоисто-дождевая облачность среднего яруса (Ns-As-Ac) (Приложение 2) получено распределение поля концентрации примесей $\tilde{q}(P,t)$, представленное на рисунке 3.5 и соответствующие ему линии уровней – рисунок 3.6.

При значениях исходных данных: $X = 810$ м, $T = 60$ с, $V_{x,0} = 13.5$ м/с, $V_{y,0} = 13.5$ м/с, $V_{z,0} = 13.5$ м/с, $K_{x,0} = 25$ м²/с, $K_{y,0} = 25$ м²/с, $K_{z,0} = 5$ м²/с, $\alpha = 0.013$ с⁻¹, $q_0 = 0.005$ кг/м³, что соответствует условиям состояния атмосферы: нейтральная стратификация, высота источника $Z = 200$ м, тяжелая примесь, слоисто-дождевая облачность (Ns-As) (Приложение 2) с использованием разработанного программного комплекса получено распределение поля концентрации примесей $\tilde{q}(P,t)$, показанное на рисунке 3.7, соответствующие ему линии уровней – на рисунке 3.8.

При значениях исходных данных: $X = 642$ м, $T = 60$ с, $V_{x,0} = 10.7$ м/с, $V_{y,0} = 10.7$ м/с, $V_{z,0} = 10.7$ м/с, $K_{x,0} = 150$ м²/с, $K_{y,0} = 150$ м²/с, $K_{z,0} = 15$ м²/с, $\alpha = 0.013$ с⁻¹, $q_0 = 0.005$ кг/м³, что соответствует условиям состояния атмосферы: устойчивая стратификация, высота источника $Z = 200$ м, тяжелая примесь, мощно-кучевая облачность (CU-Cong) (Приложение 2) с использованием разработанного программного комплекса рассчитано распределение поля концентрации примесей $\tilde{q}(P,t)$ – рисунок 3.9, соответствующие ему линии уровней – рисунок 3.10.

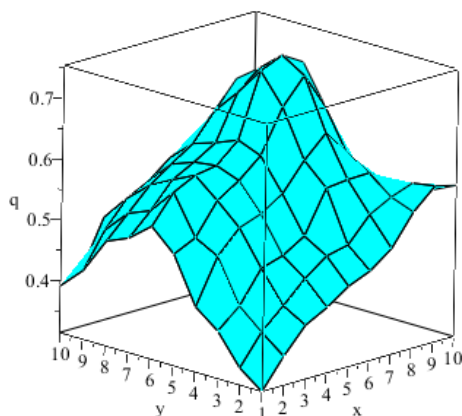


Рисунок 3.5 – Распределение концентрации $\tilde{q}(P, t)$, $V_{x,0} = 10.8$ м/с, $V_{y,0} = 10.8$ м/с, $V_{z,0} = 10.8$ м/с, $K_{x,0} = 40$ м²/с, $K_{y,0} = 40$ м²/с, $K_{z,0} = 4$ м²/с, все значения нормированные, фиксированы координаты z и t ($l = 10$ и $j = 10$)

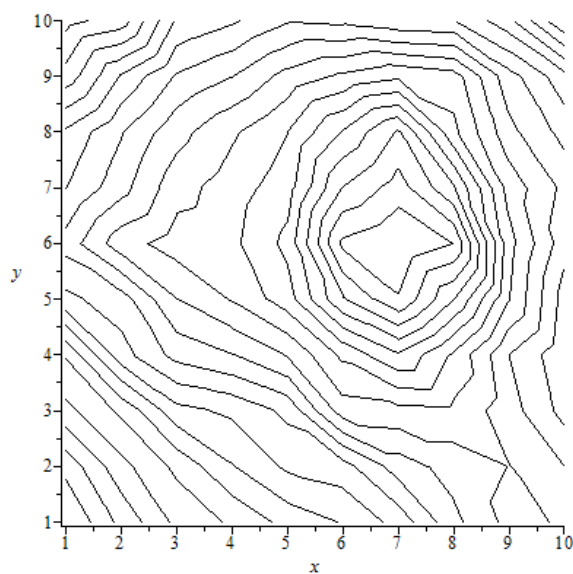


Рисунок 3.6 – Линии уровней, соответствующие распределению концентрации $\tilde{q}(P, t)$, $\tilde{q}_{\min}(P, t) = 0.314$ кг/м³, $\tilde{q}_{\max}(P, t) = 0.754$ кг/м³ (соответствует рис. 3.5)

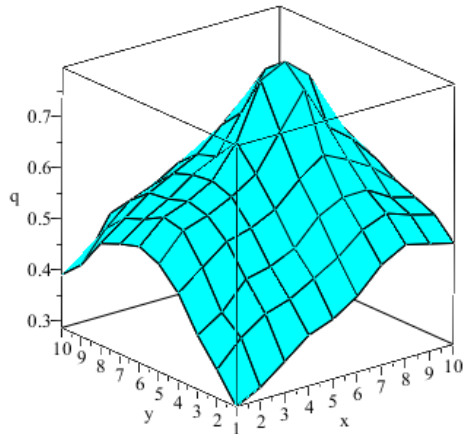


Рисунок 3.7 – Распределение концентрации $\tilde{q}(P, t)$, $V_{x,0} = 13.5$ м/с , $V_{y,0} = 13.5$ м/с , $V_{z,0} = 13.5$ м/с , $K_{x,0} = 25$ м²/с , $K_{y,0} = 25$ м²/с , $K_{z,0} = 5$ м²/с , все значения нормированные, фиксированы координаты z и t ($l = 10$ и $j = 10$)

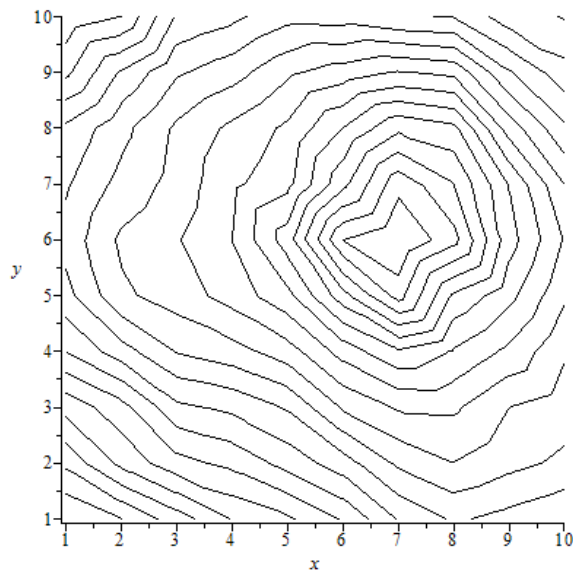


Рисунок 3.8 – Линии уровней, соответствующие распределению концентрации $\tilde{q}(P, t)$, $\tilde{q}_{\min}(P, t) = 0.287$ кг/м³ , $\tilde{q}_{\max}(P, t) = 0.796$ кг/м³ (соответствует рис. 3.7)

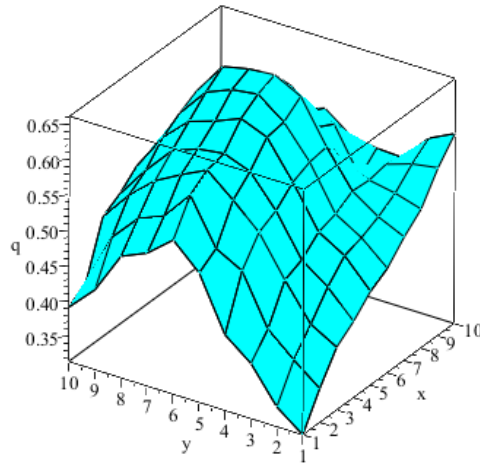


Рисунок 3.9 – Распределение концентрации $\tilde{q}(P, t)$, $V_{x,0} = 10.7$ м/с, $V_{y,0} = 10.7$ м/с, $V_{z,0} = 10.7$ м/с, $K_{x,0} = 150$ м²/с, $K_{y,0} = 150$ м²/с, $K_{z,0} = 15$ м²/с, все значения нормированные, фиксированы координаты z и t ($l = 10$ и $j = 10$)

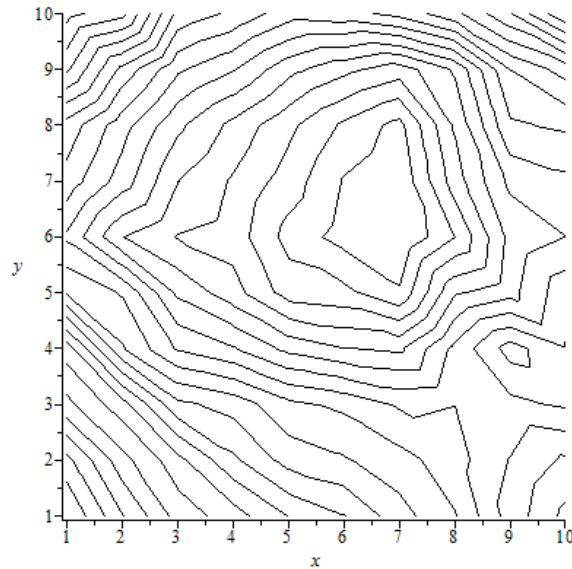


Рисунок 3.10 – Линии уровней, соответствующие распределению концентрации $\tilde{q}(P, t)$,
 $\tilde{q}_{\min}(P, t) = 0.309$ кг/м³, $\tilde{q}_{\max}(P, t) = 0.661$ кг/м³ (соответствует рис. 3.9)

При значениях исходных данных: $X = 980$ м, $T = 60$ с, $V_{x,0} = 16.4$ м/с, $V_{y,0} = 16.4$ м/с, $V_{z,0} = 16.4$ м/с, $K_{x,0} = 100$ м²/с, $K_{y,0} = 100$ м²/с, $K_{z,0} = 10$ м²/с, $\alpha = 0.013$ с⁻¹, $q_0 = 0.005$ кг/м³, что соответствует условиям состояния атмосферы: нейтральная стратификация, высота источника $Z = 500$ м, тяжелая примесь, слоисто-кучевая-дождевая облачность (Ns-As-Cb) (Приложение 2) с использованием разработанного программного

комплекса получено распределение поля концентрации примесей $\tilde{q}(P,t)$, показанное на рисунке 3.11, соответствующие ему линии уровней – на рисунке 3.12.

При значениях исходных данных: $X = 1200$ м, $T = 90$ с, $V_{x,0} = 13.2$ м/с, $V_{y,0} = 13.2$ м/с, $V_{z,0} = 13.2$ м/с, $K_{x,0} = 200$ м²/с, $K_{y,0} = 200$ м²/с, $K_{z,0} = 20$ м²/с, $\alpha = 0.013$ с⁻¹, $q_0 = 0.005$ кг/м³, что соответствует условиям состояния атмосферы: нейтральная стратификация, высота источника $Z = 500$ м, тяжелая примесь, мощно-кучевая облачность (CU-Cong) (Приложение 2) с использованием разработанного программного комплекса получено распределение поля концентрации примесей $\tilde{q}(P,t)$, представленное на рисунке 3.13 и соответствующие ему линии уровней – рисунок 3.14.

При значениях исходных данных: $X = 1200$ м, $T = 90$ с, $V_{x,0} = 13.7$ м/с, $V_{y,0} = 13.7$ м/с, $V_{z,0} = 13.7$ м/с, $K_{x,0} = 50$ м²/с, $K_{y,0} = 50$ м²/с, $K_{z,0} = 5$ м²/с, $\alpha = 0.003$ с⁻¹, $q_0 = 0.005$ кг/м³, что соответствует условиям состояния атмосферы: устойчивая стратификация, высота источника $Z = 500$ м, средняя примесь, слоисто-дождевая облачность среднего яруса (Ns-As-Ac) (Приложение 2) с использованием разработанного программного комплекса получено распределение поля концентрации примесей $\tilde{q}(P,t)$, показанное на рисунке 3.15, соответствующие ему линии уровней – на рисунке 3.16.

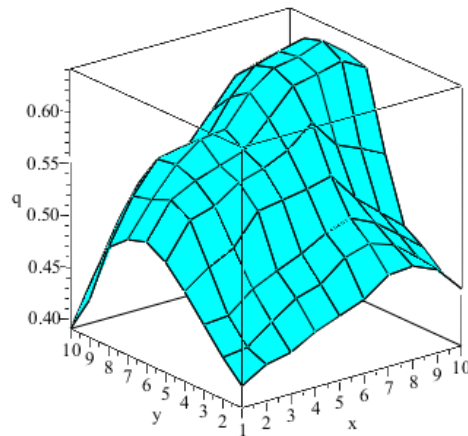


Рисунок 3.11 – Распределение концентрации $\tilde{q}(P,t)$, $V_{x,0} = 16.4$ м/с, $V_{y,0} = 16.4$ м/с, $V_{z,0} = 16.4$ м/с, $K_{x,0} = 100$ м²/с, $K_{y,0} = 100$ м²/с, $K_{z,0} = 10$ м²/с, все значения нормированные, фиксированы координаты z и t ($l = 10$ и $j = 10$)

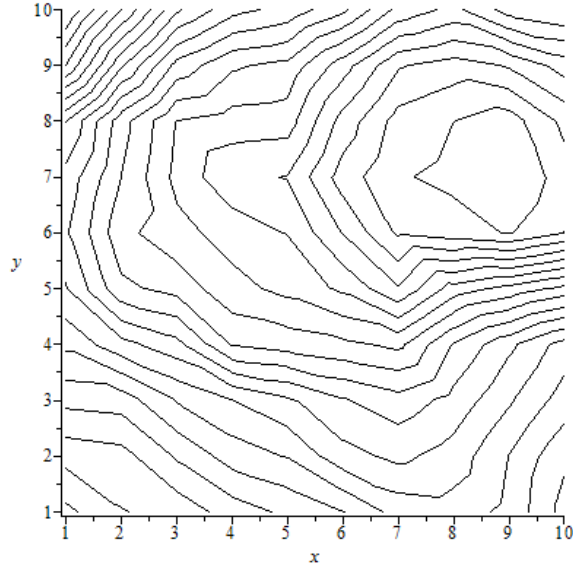


Рисунок 3.12 – Линии уровней, соответствующие распределению концентрации $\tilde{q}(P, t)$,
 $\tilde{q}_{\min}(P, t) = 0.391 \text{ кг/м}^3$, $\tilde{q}_{\max}(P, t) = 0.641 \text{ кг/м}^3$ (соответствует рис. 3.11)

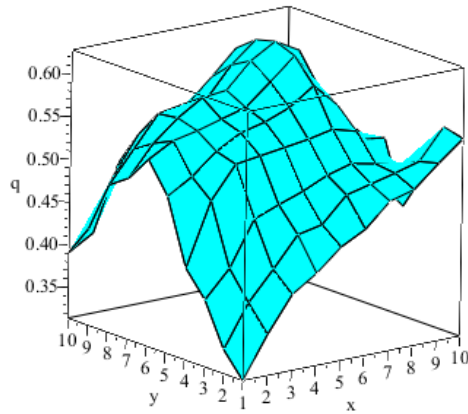


Рисунок 3.13 – Распределение концентрации $\tilde{q}(P, t)$, $V_{x,0} = 13.2 \text{ м/с}$, $V_{y,0} = 13.2 \text{ м/с}$,
 $V_{z,0} = 13.2 \text{ м/с}$, $K_{x,0} = 200 \text{ м}^2/\text{с}$, $K_{y,0} = 200 \text{ м}^2/\text{с}$, $K_{z,0} = 20 \text{ м}^2/\text{с}$, все значения нормированные,
 фиксированы координаты z и t ($l = 10$ и $j = 10$)

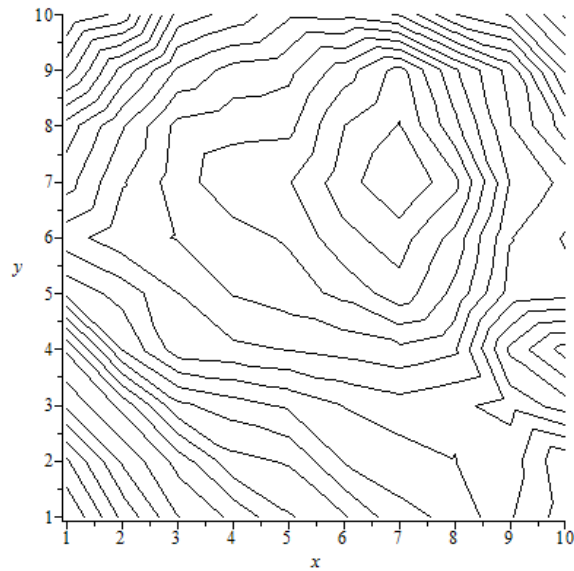


Рисунок 3.14 – Линии уровней, соответствующие распределению концентрации $\tilde{q}(P, t)$,
 $\tilde{q}_{\min}(P, t) = 0.314 \text{ кг/м}^3$, $\tilde{q}_{\max}(P, t) = 0.628 \text{ кг/м}^3$ (соответствует рис. 3.13)

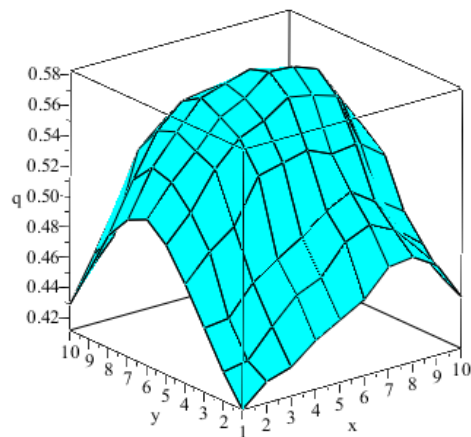


Рисунок 3.15 – Распределение концентрации $\tilde{q}(P, t)$, $V_{x,0} = 13.7 \text{ м/с}$, $V_{y,0} = 13.7 \text{ м/с}$,
 $V_{z,0} = 13.7 \text{ м/с}$, $K_{x,0} = 50 \text{ м}^2/\text{с}$, $K_{y,0} = 50 \text{ м}^2/\text{с}$, $K_{z,0} = 5 \text{ м}^2/\text{с}$, все значения нормированные, фиксированы координаты z и t ($l = 10$ и $j = 10$)

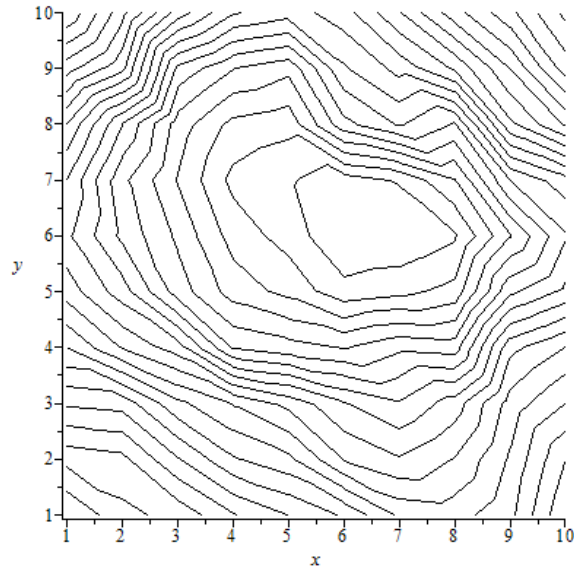


Рисунок 3.16 – Линии уровней, соответствующие распределению концентрации $\tilde{q}(P, t)$,

$$\tilde{q}_{\min}(P, t) = 0.412 \text{ кг/м}^3, \quad \tilde{q}_{\max}(P, t) = 0.583 \text{ кг/м}^3 \text{ (соответствует рис. 3.15)}$$

В таблице 3.2 представлены результаты вычисления минимального и максимального значений поля концентрации загрязняющих веществ, среднее значение в его пределах при различных значениях поля скорости ветра, коэффициента турбулентной диффузии и типа облачности атмосферы. Таким образом, получены данные о распространении загрязняющих веществ в атмосфере с учетом различных метеорологических параметров: высоты измерения, скорости ветра, турбулентности, типов стратификации, облачности, загрязняющей примеси. Анализируя полученные результаты, представленные в рисунках и таблицах, не трудно видеть, как на различных высотах с усилением скорости ветра процесс переноса ускоряется. С усилением турбулентности усиливается процесс размывания профиля $\tilde{q}(P, t)$ (рис. 3.5 – 3.15). С увеличением значения коэффициента турбулентности при том же значении скорости ветра (рис. 3.5, 3.9) уровень концентрации снижается: $\tilde{q}_{\max}(P, t) = 0.754 \text{ кг/м}^3$ - $\tilde{q}_{\max}(P, t) = 0.661 \text{ кг/м}^3$. При усилении скорости ветра (рис. 3.7) при достаточно высоком значении коэффициента турбулентности уровень концентрации несколько повышается за счет ускорения процесса переноса: $\tilde{q}_{\max}(P, t) = 0.796 \text{ кг/м}^3$, но при этом размывание профиля $\tilde{q}(P, t)$ усиливается.

Таблица 3.2 – Результаты вычислений значений концентрации загрязняющих веществ при различных значениях состояния атмосферы в пределах ее пограничного слоя

X (м)	Y (м)	Z (м)	T (с)	V (м/с)	K ($\text{м}^2/\text{с}$)	Тип примеси α (с^{-1})	Тип облачности	$[q_{\min}, q_{\max}]$ ($\text{кг}/\text{м}^3$)	q_{cp} ($\text{кг}/\text{м}^3$)
642	642	100	60	10.7	150	0.013	CU-Cong	[0.309, 0.661]	0.485
648	648	200	60	10.8	40	0.013	Ns-As-Ac	[0.314, 0.754]	0.534
1200	1200	200	90	13.2	200	0.013	CU-Cong	[0.314, 0.628]	0.471
810	810	500	60	13.5	25	0.013	Ns-As	[0.287, 0.796]	0.542
1200	1200	500	90	13.7	50	0.003	Ns-As-Ac	[0.412, 0.583]	0.498
980	980	500	60	16.4	100	0.013	Ns-As-Cb	[0.391, 0.641]	0.516
972	972	200	60	13.5	150	0.00013	Cb	[0.366, 0.674]	0.520
642	642	100	60	10.8	100	0.00013	Cb	[0.454, 0.616]	0.535
810	810	500	90	13.5	100	0.003	Cb	[0.498, 0.672]	0.585
810	810	500	60	16.4	250	0.003	Ns-As-Cb	[0.387, 0.521]	0.454

Примечание: X , Y , Z - пространственные переменные задачи, T - временная переменная задачи, V - скорость ветра, K - атмосферная турбулентность, α - вид осаждающейся примеси, $[q_{\min}, q_{\max}]$ - диапазон значений концентрации загрязняющей примеси, q_{cp} - среднее значение концентрации загрязняющей примеси на интервале. **Тип облачности:** Мощно-кучевая CU-Cong, Слоисто-дождевая среднего яруса Ns-As-Ac, Слоисто-дождевая Ns-As, Кучево-дождевая Cb, Слоистокучевая-дождевая Ns-As-Cb; **Тип примеси:** 0.013 (с^{-1}) - тяжелая, 0.003 (с^{-1}) - средняя, 0.00013 (с^{-1}) - легкая.

Выводы по главе 3

1. На основе конечно-разностного метода и численных методов решения СЛАУ разработан алгоритм тестового примера для параметризованной модели пространственной задачи переноса, на его основе в заключительной части главы выполнено тестирование алгоритмического и программного обеспечения. При построении тестового примера осуществлялся выбор исходных данных, выполнялась процедура нормирования всех переменных и распределений задачи, получена параметризованная модель пространственной задачи, проведена оценка значений параметров модели.

2. В рамках численного метода по координатного расщепления трехмерного параметризованного уравнения переноса с учетом уравнения неразрывности $div\vec{V} = 0$ и итерационно-рекурсивных методов и алгоритмов, построенных ранее для одномерной задачи, выполнено построение вычислительных алгоритмов пространственной задачи переноса. Здесь принимались следующие допущения: $\alpha(t) = \alpha = const$, $\beta_z(t) = \beta_z = const$, $\beta_x(t) = \beta_x = const$, $\beta_y(t) = \beta_y = const$, $\theta_z(t) = \theta_z = const$, $\theta_x(t) = \theta_x = const$, $\theta_y(t) = \theta_y = const$, $\xi_1(t) = \xi_1 = const$, $\xi_2(t) = \xi_2 = const$, $\xi_3(t) = \xi_3 = const$, что в целом позволило несколько упростить вычислительные схемы.

3. На основе разработанного программно-алгоритмического обеспечения и методики тестирования алгоритмов и программ выполнена постановка и проведен вычислительный эксперимент по исследованию влияния различных значений скорости ветра, турбулентности и других характеристик состояния пограничного слоя атмосферы на пространственно-временное распределение поля концентрации загрязняющих примесей на примере решения пространственной задачи переноса. Так, для различных значений скорости ветра, коэффициента турбулентной диффузии, типа облачности, вида загрязняющей примеси, характеризующих различные состояния пограничного слоя атмосферы, построены пространственно – временные распределения концентрации загрязняющих примесей и соответствующие им линии уровня.

Результаты, полученные в данной главе, опубликованы в работах диссертанта [35, 40, 42, 79, 83, 85].

Глава 4. Разработка системы компьютерного моделирования пространственно-временной изменчивости поля концентрации загрязнений в атмосфере на основе предложенных численных методов и программного обеспечения

Глава посвящена разработке системы компьютерного моделирования пространственно-временной изменчивости поля концентрации загрязнений в атмосфере на основе предложенных численных методов, подробно рассмотренных в предыдущих главах. Основная цель при этом заключается в том, чтобы отобразить взаимные связи между алгоритмами, составляющими целостную систему. Данная система может стать в дальнейшем ядром некоторой информационной системы, применяемой в задачах экологического мониторинга воздушной среды. В главе предлагаются концептуальные схемы такой системы, и затем на их основе выполняется построение самой системы компьютерного моделирования с помощью псевдоязыка программирования. Соответствующие этим алгоритмам программы представлены в приложениях.

4.1 Разработка информационно-логической схемы системы компьютерного моделирования пространственно-временной изменчивости поля концентрации загрязнений в атмосфере

Как отмечалось во введении, прогноз изменения экологических систем, подвергающихся воздействию антропогенных факторов, является важной задачей науки в настоящее время. Важная роль математического моделирования, проведения численных экспериментов и комплексный мониторинг в исследовании подобных систем обусловлены свойствами многомерности этих систем, неполной предсказуемости их поведения. Существующий опыт решения научных и практических задач природоохранного направления показывает, что математические модели и данные натурных исследований являются равноправными и дополняющими друг друга инструментами для изучения природных процессов. В результате сбора данных к исследователям попадает разнородная информация, с разных сторон характеризующая наблюдаемые явления. При этом возникает проблема обеспечения математических моделей исходными данными. В рамках этой проблемы в работе созданы математические модели исследуемых процессов, раз-

работаны соответствующие численные методы, на их основе построены вычислительные модели и алгоритмы, а также соответствующее программное обеспечение.

Перечисленные средства математического, вычислительного и программного обеспечения в совокупности представляют собой **информационно-вычислительную технологию моделирования атмосферных процессов**, в частности, процесса нестационарного диффузного переноса загрязняющих примесей в атмосфере, проблема создания и развития которой является актуальной, современной и относится к быстроразвивающимся в настоящее время информационно-вычислительным технологиям в науках об окружающей среде.

Реализация перечисленных задач основана на применении современных информационно-вычислительных технологий, а именно, разработке эффективных вычислительных алгоритмов с учетом современных достижений в области вычислительной математики, технологии распараллеливания, а также других высокопроизводительных технологий для организации вычислений в задачах моделирования атмосферных процессов. Совокупность разрабатываемых методов, моделей и алгоритмов, объединенных в систему, определяет технологию моделирования, позволяющую решать задачу совместного использования экологической информации и вычислительных моделей, способных воспринимать приближенные исходные данные.

Далее представлена разработка системы компьютерного моделирования пространственно-временной изменчивости поля концентрации загрязнений в атмосфере на основе численных методов, соответствующих тем вычислительным моделям, которые были построены выше. Все методы и модели объединяются в единую систему и предусматривают передачу данных из одного модуля в другие. Основные принципы построения подобных систем изложены в работах [21, 34, 44]. Система компьютерного моделирования представляет собой совокупность взаимосвязанных, полностью детализированных алгоритмов, ее организация выполнялась при использовании языка программирования Maple, построение драйверных модулей алгоритмической системы позволяет осуществлять последовательный переход от одного модуля к другому с передачей соответствующих данных, что обеспечивает их взаимосвязь и целостность системы. При этом система алгоритмов включает в себя головной драйверный алгоритм, драйверы подсистемы «Математические модели и вычислительные алгоритмы оценки концентрации загрязняющих примесей в турбулентной атмосфере» и алгоритмические модули общесистем-

ного назначения. Последние являются фактически инструментальными средствами, используемыми при формировании любого модуля системы. Алгоритмы модульной системы могут быть легко запрограммированы на любом языке высокого уровня.

Актуальность создания информационно – вычислительных технологий в задачах экологического мониторинга природных сред с последующей организацией системы компьютерного моделирования алгоритмов, обуславливается тем, что на его основе предложена концепция комплексного решения (вычислительная технология) задач переноса примесей в атмосфере в рамках некоторой информационно – измерительной системы, предположительно функционирующей на метеорологических станциях или в геофизических лабораториях. Данная система алгоритмов является ядром информационно – вычислительного обеспечения подобных систем и представляет собой комплекс взаимосвязанных модулей, каждый из которых есть алгоритмическая реализация того или иного вычислительного метода.

На рисунке 4.1 представлена структурная схема системы моделирования процессов переноса загрязняющих примесей, предлагаемая в данной работе, которая включает в себя совокупность блоков, обеспечивающих поступление исходных данных в вычислительные модели. При этом поступающие данные предварительно анализируются, выполняются процедуры нормирования распределений и параметризации вычислительных моделей. Ядром системы является блок моделирования, содержащий алгоритмы вычислительных моделей, построенных в работе. Результаты моделирования, полученные в этом блоке, подвергаются последующему анализу и могут поступать на вход других моделей, описывающих те или иные процессы, протекающие в атмосфере.

Остановимся подробно на каждом из блоков системы моделирования процессов переноса загрязняющих примесей в атмосфере. Как видно из рисунка 4.1, она включает в себя:

1. **«Блок исходных данных»**, который условно можно рассматривать как две взаимосвязанные части (рис. 4.2). Первая часть образована набором так называемых «тестовых» задач, генерирующих значения полей исходных данных (массивы исходных данных) и применяемых для последующего тестирования алгоритмов и программ, а также проведения численных экспериментов. В соответствии с этим в работе проведены построения тестовых задач для одномерного уравнения переноса, для трехмерного уравнения переноса и для качественных моделей. Вторая часть блока обеспечивает ана-

лиз и подготовку исходных данных к введению их в модели, а именно, реализует процедуры параметризации исходных моделей, нормирования всех величин и распределений, при необходимости, некоторых вычислений недостающих констант или переменных, а также процедуры аппроксимации экспериментальных данных и их производных. На схеме отражено движение информационных потоков между компонентами блока, все данные, генерируемые в «Блоке исходных данных» попадают в **«Блок моделирования»**.

2. **«Блок моделирования»**, являющийся ядром системы, поскольку он включает в себя все алгоритмы, разработанные в диссертации (рис. 4.3). Имеются в виду вычислительные схемы на основе конечных разностей для одномерной и пространственной моделей переноса; итерационные вычислительные схемы для одномерной и пространственной моделей переноса; вычислительные схемы для качественных моделей. Условно данный блок можно рассматривать как три взаимосвязанных блока – блок алгоритмизации вычислительных методов, блок программирования и блок результатов моделирования, новых данных (рис. 4.4). Все компоненты **«Блока моделирования»** взаимосвязаны и представляют собой комплексную систему алгоритмических модулей. При этом система остается открытой и может быть изменена или дополнена. На схеме отражено движение информационных потоков между компонентами блока, основное назначение которого – получение значений поля концентрации примесей, переносимых в приземном слое атмосферы.

3. В блок **«Результаты моделирования, новые данные»** (рис. 4.4) включены алгоритмы исследования рассматриваемых в работе вычислительных моделей на сходимость и устойчивость к погрешностям в исходных данных на основе разработанных в **«Блоке исходных данных»** тестовых задач. Также здесь отражено моделирование поля концентрации загрязняющих примесей при различных исходных данных, что сопровождается визуализацией полученных результатов.



Рисунок 4.1 – Структурная схема системы компьютерного моделирования процессов переноса загрязняющих примесей в атмосфере

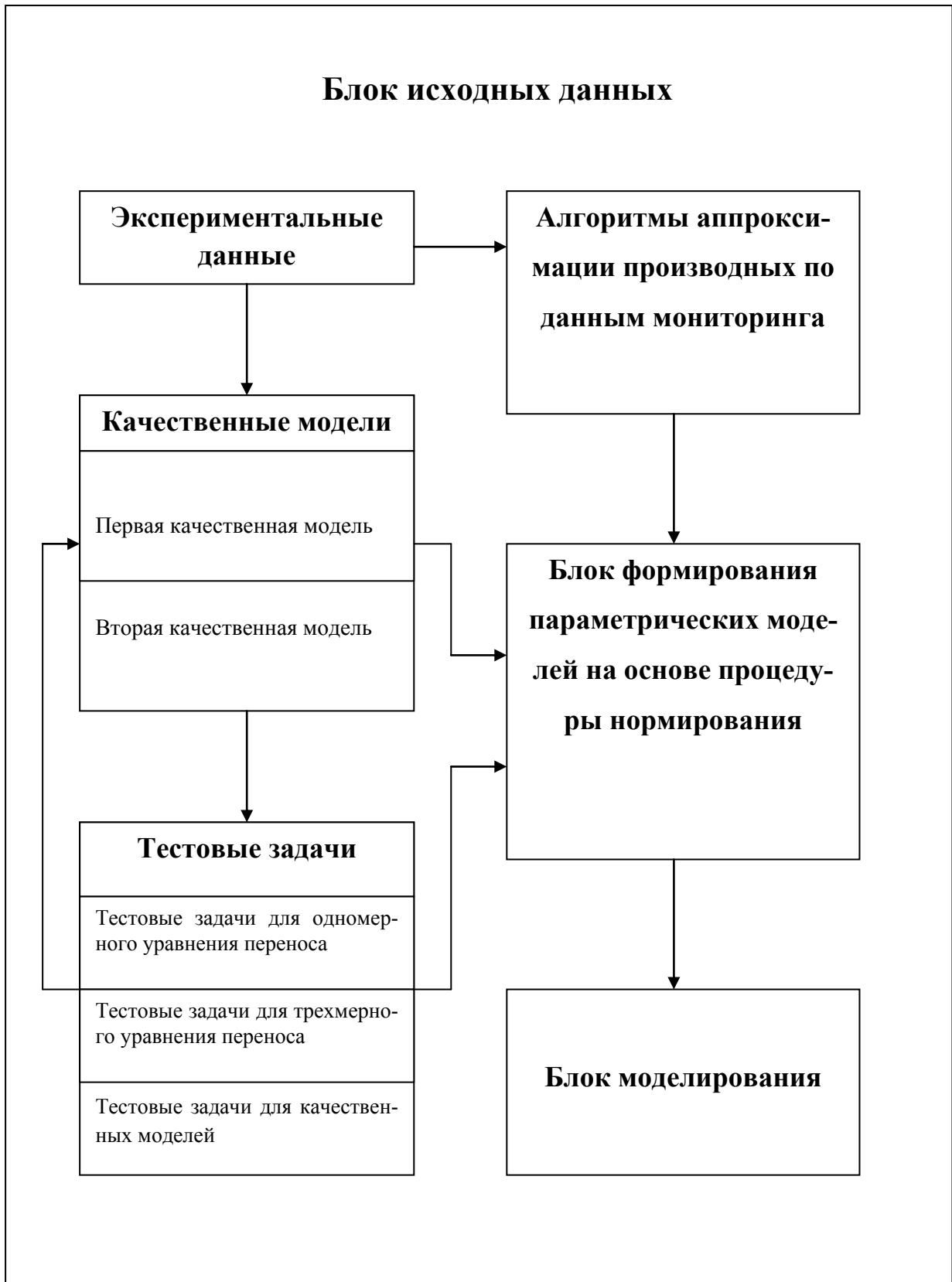


Рисунок 4.2 – Структурная схема «Блока исходных данных»

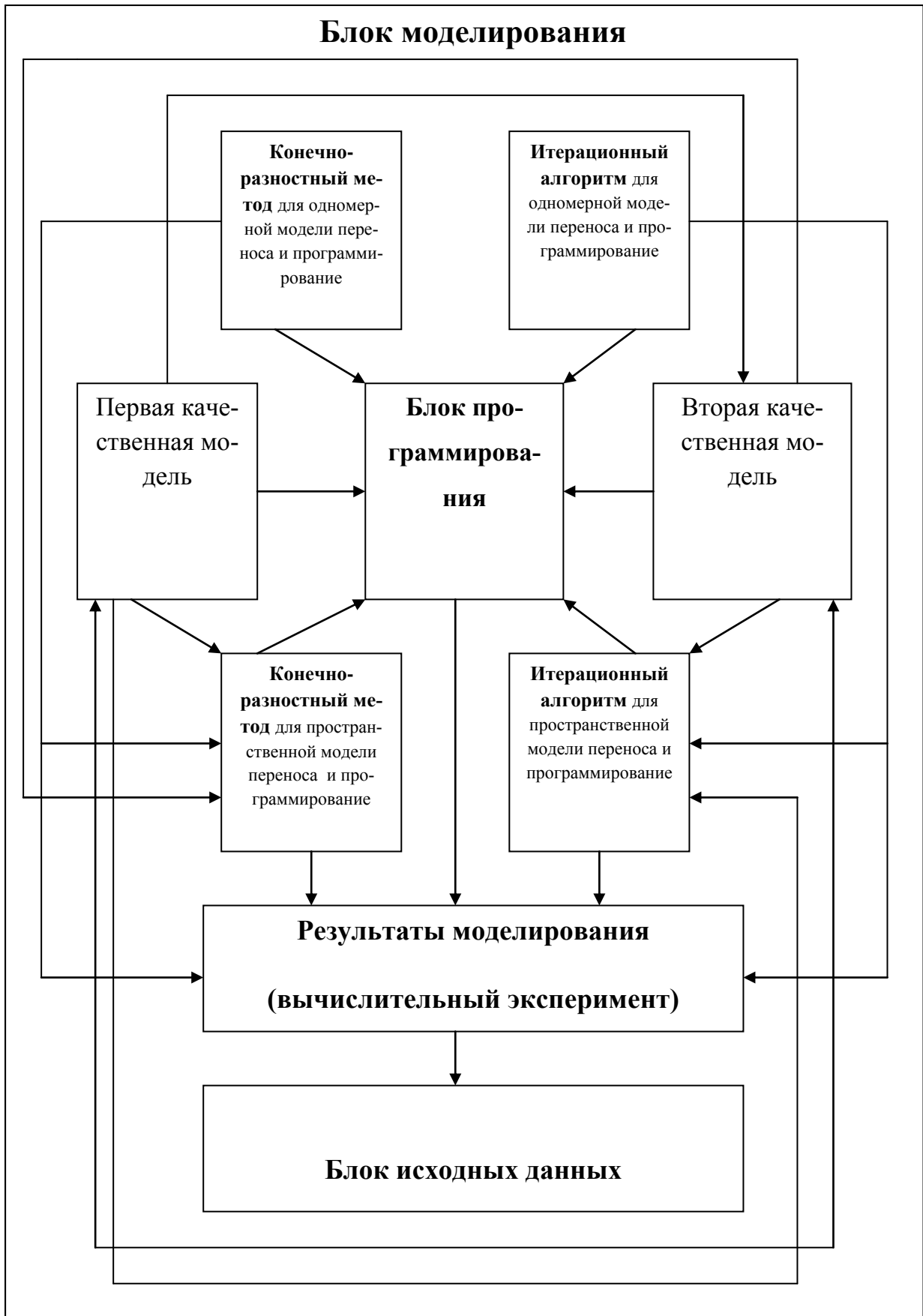


Рисунок 4.3 – Структурная схема «Блока моделирования»



Рисунок 4.4 – Структурная схема «Блока результатов моделирования, новых данных»

4.2 Разработка структуры системы компьютерного моделирования на основе модульного подхода

В пределах данной главы разрабатывается и описывается система компьютерного моделирования полностью детализированных алгоритмов решения задач нестационарного переноса загрязняющих примесей в атмосфере. Данная модульная система создается по аналогии с построенной и подробно описанной в [34], используются её основные принципы и подходы.

Под модульностью понимается то обстоятельство, что каждая явно выраженная функциональная компонента вычислительных методов представлена одним или более

отдельными модулями. Слово система отражает тот факт, что для некоторых стадий вычислительного процесса решения какого – либо из уравнений имеются несколько вариантов модулей, так что с помощью различных их комбинаций можно получать разнообразные методы. Кроме того, методы могут иметь несколько общих компонент. И, наконец, выражение «полностью детализированные алгоритмы» означает, что каждый модуль полностью определен, включая любые параметры типа допусков.

Приведем основные причины, по которым эти алгоритмы включены в данную работу: 1) дать полностью детализированное описание каждого рассмотренного в диссертации метода численного решения задач переноса. Алгоритмы в основном тексте предыдущих глав диссертации излагаются на таком уровне детализации, который является достаточным для основательного изучения предмета; 2) показать, что модульная конструкция алгоритмов может быть доведена до уровня программной реализации; 3) обеспечить методикой преподавания с использованием учебных проектов. Программная реализация одного или более методов из этой системы представляет собой превосходный способ закрепления материала, связанного, например, с изучением приближенных методов решения уравнений математической физики; 4) способствовать использованию структурированного и управляемого программного обеспечения в тестировании новых методов численного решения задач переноса и решения других, связанных с ними задач. Если готовые программы написаны в модульном стиле, как, например, рассматриваемые нами, то можно разработать и протестировать много новых методов, изменив только один или несколько модулей. Это обеспечивает пользователя простым управляемым средством тестирования; 5) помочь тем прикладникам, которым необходимо разработать свои собственные программы. Подчеркнем, что всякий раз, когда это возможно, следует пользоваться библиотеками вычислительного программного обеспечения. Однако из-за специфики того или иного класса прикладных задач иногда необходимо разрабатывать специализированные программы на базе либо копий существующих подпрограмм, либо их адаптированных вариантов. Эталоном в подобных случаях могут служить алгоритмы из данной модульной структуры.

Алгоритмы представлены в виде программ, написанных на языке программирования Maple 14. Организация данной системы алгоритмов отражает общую схему моделей, представленных во второй и третьей главах работы (рис. 4.1). Методы для всех типов моделей разбиваются на ряд отдельных функциональных компонент, и каждая ком-

понента может быть реализована одним или более модулями. В некоторых случаях для одной компоненты дается ряд вариантов, каждый из которых можно выбрать. Полный метод решения задач, соответствующих той или иной модели схемы (рис. 4.2), создается выбором и программированием алгоритмов для каждой из отдельных компонент, а затем объединением их с помощью драйверной программы.

Каждый алгоритм в том виде, как он был представлен в предыдущих главах диссертации, содержит указания на то, какие требуются компоненты, а также служит драйвером, в котором приведены основные шаги методов. Некоторые из этих шагов являются просто строками программы в драйверном алгоритме, тогда как другие представляют собой упомянутые выше компоненты, которые отвечают одному или более отдельным модулям. В п. 4.3 приведены отдельные компоненты и модули, используемые для их реализации. Раздел «Алгоритм» содержит программы драйверных алгоритмов, которые следует использовать для программной реализации всей системы алгоритмов. Для программной реализации одного или нескольких отдельно взятых методов, а не всей системы в целом, достаточно урезанного варианта драйвера. К каждому головному драйверному алгоритму добавляется примечание, которое задумано как совет пользователю этой системы. В нем могут обсуждаться вопросы выбора параметров, используемых в алгоритмах, например параметров, определяющих алгоритмический выбор. В готовой программной реализации эти параметры могли бы выбираться пользователем или задаваться определенными значениями в программе.

Структура каждого модуля имеет следующий вид:

- Назначение
- Наименование программы
- Входные параметры
- Выходные параметры
- Память
- Алгоритм
- Примечания

В разделе «Назначение» кратко излагается назначение модуля. В разделе «Наименование программы» дается ее символическое имя. В двух разделах, рассматривающих «параметры», приводятся имена, типы данных и дополнительная информация обо всех параметрах модуля. В разделе «Память» определяются размерности массивов. Раздел

«Алгоритм» содержит полностью детализированную программу модуля. Форма написания программ рассчитана на то, чтобы быть понятной для всех пользователей. В ее основе лежат управляющие структуры языка Maple, которые были выбраны из-за того, что их смысл очевиден из написания. Раздел «Примечания» включается на случай дополнительных комментариев, относящихся к модулю.

Драйверные модули системы формируются с помощью некоторых элементов так называемого псевдоязыка программирования [21, 34, 44]. Используются следующие управляющие структуры.

1) Оператор **IF-THEN**

1. **IF** (логическое условие) **THEN** оператор

Смысл оператора: если логическое условие истинно, то выполнить оператор и перейти к следующему оператору.

2) Оператор **IF-THEN-ELSE**

2. **IF** (логическое условие и) **THEN**

2Т. 1 первый оператор в THEN-блоке

.....

2Т. 5 последний оператор в THEN-блоке

ELSE (2)

2Е. 1 "первый оператор в ELSE-блоке

.....

2Е. 7 последний оператор в ELSE-блоке

ENDIF (2)

Смысл оператора: если логическое условие истинно, то выполнить операторы 2Т.1 – 5, иначе выполнить операторы 2Е.1 – 7; затем перейти к оператору, следующему за ENDIF.

2) Вызов подпрограммы выглядит так:

GALL имя модуля (список аргументов).

Имя модуля носит описательный характер. Номер модуля всегда дается как сопутствующий комментарий. Аргументы соответствуют в порядке их следования входным параметрам, входно-выходным параметрам и выходным параметрам вызываемого модуля.

3) оператор **END** имя модуля.

4.3 Программные модули алгоритмической системы компьютерного моделирования

Построенные в диссертационной работе алгоритмы и программные средства объединены в алгоритмическую систему, названную в работе «Математические модели и вычислительные алгоритмы оценки концентрации загрязняющих примесей в турбулентной атмосфере». Алгоритмическая система компьютерного моделирования представлена совокупностью взаимосвязанных драйверов, каждый из которых соответствует тому или иному блоку информационно-вычислительной системы мониторинга и прогноза экологического состояния воздушного бассейна, описание которой изложено выше. Основным (головным) является драйверный модуль GDM. Инициализация данного модуля позволяет последовательно активизировать подсистемы (их соответствующие драйверы), описание и назначение которых подробно излагается ниже. Каждый из этих драйверов в свою очередь подключает алгоритмические модули, соответствующие тем или иным вычислительным методам, выбор которых определяется значением соответствующих глобальных переменных типа global.

Головной драйверный алгоритм модульной системы GDM

Назначение: Обеспечивает взаимосвязь драйверных алгоритмов основных модулей системы и их последовательное выполнение.

Наименование программы: GDM (входные и выходные параметры).

Входные параметры: global1 – определяет выбор одного из двух базовых модулей системы MODUL1 – подсистема алгоритмов и программ, предназначенных для решения одномерной задачи переноса и MODUL2 – подсистема алгоритмов и программ, предназначенных для решения трехмерной задачи переноса.

Выходные параметры: выходные параметры каждого из модулей, описаны внутри них.

Память: описана внутри модулей

Алгоритм: (*Инициализирующая часть*)

IF (global1=1) THEN

CALL MODUL1 (input and output parameters);

END IF

IF (global1=3) THEN

```

CALL MODUL2 (input and output parameters);
END IF
END (GDM)

```

Драйвер модульной системы алгоритмов MODUL1 – «Подсистема алгоритмов решения одномерного уравнения переноса»

Назначение: Обеспечивает взаимосвязь драйверных алгоритмов решения одномерного уравнения переноса и их последовательное выполнение.

Наименование программы: MODUL1 (входные и выходные параметры).

Входные параметры: global2 – определяет выбор одного из четырех модулей системы MODUL1, а именно KRM_1 – алгоритм, реализующий конечно-разностный метод и метод прогонки решения одномерного уравнения переноса; ITERM_1 – итерационный алгоритм, реализующий метод интегральных уравнений решения одномерного уравнения переноса; KACHM_1 – алгоритм, соответствующий первой качественной модели решения одномерного уравнения переноса; KACHM_2 – алгоритм, соответствующий второй качественной модели решения одномерного уравнения переноса. Активизация программных модулей, соответствующих указанным алгоритмам, предваряется вызовом и активизацией программно-алгоритмического модуля VIDAN_1, соответствующего «Блоку исходных данных» для одномерной задачи переноса.

Выходные параметры: выходные параметры каждого из модулей, описаны внутри них.

Память: описана внутри модулей

Алгоритм: (*Инициализирующая часть*)

```

CALL VIDAN_1 (input and output parameters);
IF (global2=1) THEN
    CALL KRM_1 (input and output parameters);
END IF
IF (global2=2) THEN
    CALL ITERM_1 (input and output parameters);
END IF
IF (global2=3) THEN
    CALL KACHM_1 (input and output parameters);
END IF

```

```

IF (global2=4) THEN
  CALL KACHM_2 (input and output parameters);
END IF
END (MODUL1)

```

Драйвер модульной системы алгоритмов MODUL2 – «Подсистема алгоритмов решения трехмерного уравнения переноса»

Назначение: Обеспечивает взаимосвязь драйверных алгоритмов решения трехмерного уравнения переноса и их последовательное выполнение.

Наименование программы: MODUL2 (входные и выходные параметры).

Входные параметры: global3 – определяет выбор одного из четырех модулей системы MODUL2, а именно KRM_3 – алгоритм, реализующий конечно-разностный метод и метод прогонки решения трехмерного уравнения переноса в рамках метода по координатного расщепления; ITERM_3 – итерационный алгоритм, реализующий метод интегральных уравнений решения трехмерного уравнения переноса в рамках метода по координатного расщепления; KACHM_3_1 – алгоритм, соответствующий первой качественной модели решения трехмерного уравнения переноса, реализуемый в рамках метода по координатного расщепления; KACHM_3_2 – алгоритм, соответствующий второй качественной модели решения трехмерного уравнения переноса, реализуемый в рамках метода по координатного расщепления. Активизация программных модулей, соответствующих указанным алгоритмам, предваряется вызовом и активизацией программно-алгоритмического модуля VIDAN_3, соответствующего «Блоку исходных данных» для трехмерной задачи переноса.

Выходные параметры: выходные параметры каждого из модулей, описаны внутри них.

Память: описана внутри модулей

Алгоритм: (*Инициализирующая часть*)

```

CALL VIDAN_3 (input and output parameters);
IF (global3=1) THEN
  CALL KRM_3 (input and output parameters);
END IF
IF (global3=2) THEN

```

```

CALL ITERM_3 (input and output parameters);
END IF
IF (global3=3) THEN
CALL KACHM_3_1 (input and output parameters);
END IF
IF (global3=4) THEN
CALL KACHM_3_2 (input and output parameters);
END IF
END (MODUL2)

```

Модуль VIDAN_1, вызываемый в драйвере MODUL1

Назначение: Генерация массивов и переменных исходных данных для одномерной задачи в модели диффузионного переноса субстанции.

Наименование программы: VIDAN_1 (входные и выходные параметры).

Входные параметры: x – пространственная переменная, t – временная переменная; $V(x,t)$ – поле скорости ветра для одномерной задачи переноса, $K(x,t)$ – поле коэффициента турбулентной диффузии для одномерной задачи переноса, $q_T(x,t)$ – поле концентрации примесей (точное решение) для одномерной задачи переноса; V_0 , K_0 , q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); α , β , θ , ξ – безразмерные параметры модели для одномерной задачи переноса; X , T – величины, связанные с выбором интервалов $[0, X]$ и $[0, T]$ для одномерной задачи переноса.

Выходные параметры: Размерность задачи $m \times n$; нормировочные коэффициенты V_{\max} , K_{\max} , q_{\max} , S_{\max} ; массивы, соответствующие пространственным $\{\hat{x}_i\}$, $i = \overline{0, m}$ и временным $\{\hat{t}_j\}$, $j = \overline{0, n}$ переменным; массивы значений скорости ветра $\{\hat{V}_{i,j}\}$, $i = \overline{0, m}$, $j = \overline{0, n}$, турбулентности $\{\hat{K}_{i,j}\}$, $i = \overline{0, m}$, $j = \overline{0, n}$, концентрации загрязняющих веществ $\{\hat{q}_{i,j}\}$, $i = \overline{0, m}$, $j = \overline{0, n}$, функции источника $\{\hat{S}_{i,j}\}$, $i = \overline{0, m}$, $j = \overline{0, n}$; массивы их первых и вторых производных.

Память: описана внутри модулей подсистемы алгоритмов MODUL1.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи приведена в п. 2.1; листинг соответствующей

программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.1.

Модуль KRM_1, вызываемый в драйвере MODUL1

Назначение: Численная реализация рекурсивного вычислительного алгоритма на основе конечно-разностного метода и метода прогонки в одномерной задаче расчета концентрации загрязняющих примесей $\tilde{q}(x,t)$, распространяющихся в приземном слое атмосферы.

Наименование программы: KRM_1 (входные и выходные параметры).

Входные параметры: x – пространственная переменная, t – временная переменная; Δx , Δt – значения, определяющие шаг сетки; $V(x,t)$ – поле скорости ветра для одномерной задачи переноса, $K(x,t)$ – поле коэффициента турбулентной диффузии для одномерной задачи переноса, $S(x,t)$ – поле источника загрязняющих примесей для одномерной задачи переноса, $q_T(x,t)$ – поле концентрации загрязняющих примесей (точное решение) для одномерной задачи переноса; V_0 , K_0 , q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); α , β , θ , ξ – безразмерные параметры модели для одномерной задачи переноса; X , T – величины, связанные с выбором интервалов $[0, X]$ и $[0, T]$ для одномерной задачи переноса.

Выходные параметры: $\tilde{q}(x,t)$ – поле концентрации загрязняющих примесей (приближенное решение) для одномерной задачи переноса.

Память: размерность массивов для точного $q_T(x,t)$ и приближенного $\tilde{q}(x,t)$ решений одномерной задачи переноса составила $m \times n = 16000$.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи, результаты вычислений приведены в п. 1.3 и п. 2.2; листинг соответствующей программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.2.

Модуль ITERM_1, вызываемый в драйвере MODUL1

Назначение: Численная реализация итерационного вычислительного алгоритма на основе «метода интегральных уравнений» в одномерной задаче расчета концентрации загрязняющих примесей $\tilde{q}(x,t)$, распространяющихся в приземном слое атмосферы.

Наименование программы: ITERM_1 (входные и выходные параметры).

Входные параметры: x – пространственная переменная, t – временная переменная; Δx , Δt – значения, определяющие шаг сетки; $V(x, t)$ – поле скорости ветра для одномерной задачи переноса, $K(x, t)$ – поле коэффициента турбулентной диффузии для одномерной задачи переноса, $S(x, t)$ – поле источника загрязняющих примесей для одномерной задачи переноса, $q_T(x, t)$ – поле концентрации загрязняющих примесей (точное решение) для одномерной задачи переноса; V_0 , K_0 , q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); α , β , θ , ξ – безразмерные параметры модели для одномерной задачи переноса; X , T – величины, связанные с выбором интервалов $[0, X]$ и $[0, T]$ для одномерной задачи переноса.

Выходные параметры: $\tilde{q}(x, t)$ – поле концентрации загрязняющих примесей (приближенное решение) для одномерной задачи переноса.

Память: размерность массивов для точного $q_T(x, t)$ и приближенного $\tilde{q}(x, t)$ решений одномерной задачи переноса составила $m \times n = 16000$.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи, результаты вычислений приведены в п. 1.3 и п.2.2; листинг соответствующей программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.3.

Модуль КАСНМ_1, вызываемый в драйвере MODUL1

Назначение: Численная реализация рекурсивного вычислительного алгоритма первой качественной модели одномерного уравнения переноса для расчета концентрации загрязняющих примесей $\tilde{q}(x, t)$, распространяющихся в приземном слое атмосферы.

Наименование программы: КАСНМ_1 (входные и выходные параметры).

Входные параметры: x – пространственная переменная, t – временная переменная; Δx , Δt – значения, определяющие шаг сетки; $V(x, t)$ – поле скорости ветра для одномерной задачи переноса, $K(x, t)$ – поле коэффициента турбулентной диффузии для одномерной задачи переноса, $S(x, t)$ – поле источника загрязняющих примесей для одномерной задачи переноса, $q_T(x, t)$ – поле концентрации загрязняющих примесей (точное решение) для одномерной задачи переноса; V_0 , K_0 , q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя

атмосферы); α , β , θ , ξ – безразмерные параметры модели для одномерной задачи переноса; λ , μ , η , r_1 , r_2 – параметры первой качественной модели; X , T – величины, связанные с выбором интервалов $[0, X]$ и $[0, T]$ для одномерной задачи переноса.

Выходные параметры: $\tilde{q}(x, t)$ – поле концентрации загрязняющих примесей (приближенное решение) для одномерной задачи переноса.

Память: размерность массивов для точного $q_T(x, t)$ и приближенного $\tilde{q}(x, t)$ решений одномерной задачи переноса составила $m \times n = 1000$.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи, результаты вычислений приведены в п. 2.4 и п. 2.5; листинг соответствующей программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.4.

Модуль КАСНМ_2, вызываемый в драйвере MODUL1

Назначение: Численная реализация итерационного вычислительного алгоритма второй качественной модели одномерного уравнения переноса для расчета концентрации загрязняющих примесей $\tilde{q}(x, t)$, распространяющихся в приземном слое атмосферы.

Наименование программы: КАСНМ_2 (входные и выходные параметры).

Входные параметры: x – пространственная переменная, t – временная переменная; Δx , Δt – значения, определяющие шаг сетки; $V(x, t)$ – поле скорости ветра для одномерной задачи переноса, $K(x, t)$ – поле коэффициента турбулентной диффузии для одномерной задачи переноса, $S(x, t)$ – поле источника загрязняющих примесей для одномерной задачи переноса, $q_T(x, t)$ – поле концентрации загрязняющих примесей (точное решение) для одномерной задачи переноса; V_0 , K_0 , q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); α , β , θ , ξ – безразмерные параметры модели для одномерной задачи переноса; λ – параметр сетки второй качественной модели; X , T – величины, связанные с выбором интервалов $[0, X]$ и $[0, T]$ для одномерной задачи переноса.

Выходные параметры: $\tilde{q}(x, t)$ – поле концентрации загрязняющих примесей (приближенное решение) для одномерной задачи переноса.

Память: размерность массивов для точного $q_T(x, t)$ и приближенного $\tilde{q}(x, t)$ решенной одномерной задачи переноса составила $m \times n = 2000$.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи, результаты вычислений приведены в п. 2.6 и п. 2.7; листинг соответствующей программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.5.

Модуль VIDAN_3, вызываемый в драйвере MODUL2

Назначение: Генерация массивов и переменных исходных данных для пространственной задачи в модели диффузионного переноса субстанции.

Наименование программы: VIDAN_3 (входные и выходные параметры).

Входные параметры: x, y, z – пространственные переменные, t – временная переменная; $V(P, t)$ – поле скорости ветра для пространственной задачи переноса, $K(P, t)$ – поле коэффициента турбулентной диффузии для пространственной задачи переноса, $q_T(P, t)$ – поле концентрации примесей (точное решение) для пространственной задачи переноса; V_0, K_0, q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); $\alpha, \beta, \theta, \xi$ – безразмерные параметры модели для пространственной задачи переноса; X, Y, Z, T – величины, связанные с выбором интервалов $[0, X], [0, Y], [0, Z]$ и $[0, T]$ для пространственной задачи переноса.

Выходные параметры: Размерность задачи $m \times m \times m \times n$; нормировочные коэффициенты $V_{\max}, K_{\max}, q_{\max}, S_{\max}$; массивы, соответствующие пространственным $\{\hat{x}_i\}$, $i = \overline{0, m}$, $\{\hat{y}_k\}$, $k = \overline{0, m}$, $\{\hat{z}_l\}$, $l = \overline{0, m}$, и временным переменным $\{\hat{t}_j\}$, $j = \overline{0, n}$; массивы значений скорости ветра $\{\hat{V}_{i,k,l,j}\}$, $i = \overline{0, m}$, $k = \overline{0, m}$, $l = \overline{0, m}$, $j = \overline{0, n}$, турбулентности $\{\hat{K}_{i,k,l,j}\}$, $i = \overline{0, m}$, $k = \overline{0, m}$, $l = \overline{0, m}$, $j = \overline{0, n}$, концентрации загрязняющих веществ $\{\hat{q}_{i,k,l,j}\}$, $i = \overline{0, m}$, $k = \overline{0, m}$, $l = \overline{0, m}$, $j = \overline{0, n}$, функции источника $\{\hat{S}_{i,k,l,j}\}$, $i = \overline{0, m}$, $k = \overline{0, m}$, $l = \overline{0, m}$, $j = \overline{0, n}$; массивы их первых и вторых производных.

Память: описана внутри модулей подсистемы алгоритмов MODUL2.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи, результаты вычислений приведены в п. 3.1;

листинг соответствующей программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.6.

Модуль KRM_3, вызываемый в драйвере MODUL2

Назначение: Реализация рекурсивного вычислительного алгоритма на основе конечно-разностного метода решения трехмерного уравнения переноса в рамках метода покоординатного расщепления для расчета поля концентрации загрязняющих примесей $\tilde{q}(P,t)$.

Наименование программы: KRM_3 (входные и выходные параметры).

Входные параметры: x, y, z – пространственные переменные, t – временная переменная; $\Delta x, \Delta y, \Delta z, \Delta t$ – значения, определяющие шаг сетки; $V(P,t)$ – поле скорости ветра для пространственной задачи переноса, $K(P,t)$ – поле коэффициента турбулентной диффузии для пространственной задачи переноса, $S(P,t)$ – поле источника загрязняющих примесей для пространственной задачи переноса, $q_T(P,t)$ – поле концентрации примесей (точное решение) для пространственной задачи переноса; V_0, K_0, q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); $\alpha, \beta, \theta, \xi$ – безразмерные параметры модели для пространственной задачи переноса; X, Y, Z, T – величины, связанные с выбором интервалов $[0, X], [0, Y], [0, Z]$ и $[0, T]$ для пространственной задачи переноса.

Выходные параметры: $\tilde{q}(P,t)$ – поле концентрации загрязняющих примесей (приближенное решение) для пространственной задачи переноса.

Память: размерность массивов для точного $q_T(P,t)$ и приближенного $\tilde{q}(P,t)$ решений пространственной задачи переноса составила $m \times m \times m \times n = 10000$.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи, результаты вычислений приведены в п. 3.2; листинг соответствующей программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.7.

Модуль INERM_3, вызываемый в драйвере MODUL2

Назначение: Реализация рекурсивно - итерационного вычислительного алгоритма решения трехмерного уравнения переноса в рамках метода покоординатного расщепления для расчета поля концентрации загрязняющих примесей $\tilde{q}(P,t)$.

Наименование программы: INERM_3 (входные и выходные параметры).

Входные параметры: x, y, z – пространственные переменные, t – временная переменная; $\Delta x, \Delta y, \Delta z, \Delta t$ – значения, определяющие шаг сетки; $V(P, t)$ – поле скорости ветра для пространственной задачи переноса, $K(P, t)$ – поле коэффициента турбулентной диффузии для пространственной задачи переноса, $S(P, t)$ – поле источника загрязняющих примесей для пространственной задачи переноса, $q_T(P, t)$ – поле концентрации примесей (точное решение) для пространственной задачи переноса; V_0, K_0, q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); $\alpha, \beta, \theta, \xi$ – безразмерные параметры модели для пространственной задачи переноса; X, Y, Z, T – величины, связанные с выбором интервалов $[0, X], [0, Y], [0, Z]$ и $[0, T]$ для пространственной задачи переноса.

Выходные параметры: $\tilde{q}(P, t)$ – поле концентрации загрязняющих примесей (приближенное решение) для пространственной задачи переноса.

Память: размерность массивов для точного $q_T(P, t)$ и приближенного $\tilde{q}(P, t)$ решений пространственной задачи переноса составила $m \times m \times m \times n = 10000$.

Примечание: разработка соответствующего вычислительного алгоритма с описанием всех переменных и массивов задачи, результаты вычислений приведены в п. 3.3; листинг соответствующей программы, выполненной на языке программирования Maple 14, расположен в Приложении 3.8.

Модуль КАСНМ_3_1, вызываемый в драйвере MODUL2

Назначение: Численная реализация рекурсивного вычислительного алгоритма решения трехмерного уравнения переноса в рамках метода покоординатного расщепления для расчета поля концентрации загрязняющих примесей $\tilde{q}(P, t)$ на основе первой качественной модели одномерного уравнения переноса.

Наименование программы: КАСНМ_3_1 (входные и выходные параметры).

Входные параметры: x, y, z – пространственные переменные, t – временная переменная; $\Delta x, \Delta y, \Delta z, \Delta t$ – значения, определяющие шаг сетки; $V(P, t)$ – поле скорости ветра для пространственной задачи переноса, $K(P, t)$ – поле коэффициента турбулентной диффузии для пространственной задачи переноса, $S(P, t)$ – поле источника загрязняющих примесей для пространственной задачи переноса, $q_T(P, t)$ – поле концентрации примесей (точное решение) для пространственной задачи переноса; V_0, K_0, q_0 – задан-

ные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); α , β , θ , ξ – безразмерные параметры модели для пространственной задачи переноса; X , Y , Z , T – величины, связанные с выбором интервалов $[0, X]$, $[0, Y]$, $[0, Z]$ и $[0, T]$ для пространственной задачи переноса.

Выходные параметры: $\tilde{q}(P, t)$ – поле концентрации загрязняющих примесей (приближенное решение) для пространственной задачи переноса.

Примечание: требуется разработка соответствующего вычислительного алгоритма и его программная реализация.

Модуль КАСНМ_3_2, вызываемый в драйвере MODUL2

Назначение: Численная реализация рекурсивно - итерационного вычислительного алгоритма решения трехмерного уравнения переноса в рамках метода покоординатного расщепления для расчета поля концентрации загрязняющих примесей $\tilde{q}(P, t)$ на основе второй качественной модели одномерного уравнения переноса.

Наименование программы: КАСНМ_3_2 (входные и выходные параметры).

Входные параметры: x , y , z – пространственные переменные, t – временная переменная; Δx , Δy , Δz , Δt – значения, определяющие шаг сетки; $V(P, t)$ – поле скорости ветра для пространственной задачи переноса, $K(P, t)$ – поле коэффициента турбулентной диффузии для пространственной задачи переноса, $S(P, t)$ – поле источника загрязняющих примесей для пространственной задачи переноса, $q_T(P, t)$ – поле концентрации примесей (точное решение) для пространственной задачи переноса; V_0 , K_0 , q_0 – заданные константы (усредненные значения соответствующих физических характеристик пограничного слоя атмосферы); α , β , θ , ξ – безразмерные параметры модели для пространственной задачи переноса; X , Y , Z , T – величины, связанные с выбором интервалов $[0, X]$, $[0, Y]$, $[0, Z]$ и $[0, T]$ для пространственной задачи переноса.

Выходные параметры: $\tilde{q}(P, t)$ – поле концентрации загрязняющих примесей (приближенное решение) для пространственной задачи переноса.

Примечание: требуется разработка соответствующего вычислительного алгоритма и его программная реализация.

Выводы по главе 4

1. Предложена концепция построения системы информационно-вычислительного обеспечения модели диффузионного переноса субстанции, протекающего в пограничном слое атмосферы. Основная цель заключается в том, чтобы отобразить взаимные связи между вычислительными алгоритмами, которые построены в работе и составляют целостную систему, определяющую технологию моделирования и позволяющую решать задачу совместного использования экологической информации и вычислительных моделей, способных воспринимать приближенные исходные данные.

2. Данную систему можно в дальнейшем рассматривать как ядро некоторой информационной – измерительной системы, предположительно функционирующей на метеорологических станциях или в геофизических лабораториях. В главе подробно изложены назначение и принципы организации алгоритмической системы, структура каждого модуля.

3. Выполнено построение алгоритмической системы компьютерного моделирования с помощью псевдоязыка программирования. Соответствующие этим алгоритмам программы представлены в приложении 3.

Результаты, полученные в данной главе, опубликованы в работах диссертанта [74, 75, 81].

Заключение

Работа посвящена построению и численному исследованию математических моделей нестационарного переноса загрязняющих примесей в турбулентной атмосфере.

Основные научные результаты диссертационной работы:

1. Для параметризованного уравнения переноса загрязняющих примесей в атмосфере построены две качественные расчетно-аналитические модели оценки концентрации примесей, первая на основе фундаментальной системы решений однородного дифференциального уравнения второго порядка с постоянными коэффициентами; вторая – на основе представления искомого решения интегральным уравнением Вольтерра второго рода и метода последовательных приближений. Это позволяет выполнять качественную оценку значений параметров в задачах математического моделирования явления переноса в турбулентных средах. Построены итерационно-рекурсивные алгоритмы для первой и второй качественных расчетно-аналитических моделей соответственно;

2. Разработана и программно реализована методика проведения вычислительного эксперимента, на основе которой в работе проведены численные исследования расчетно-аналитических моделей;

3. На основе численного метода покоординатного расщепления трехмерного параметризованного уравнения переноса и итерационно-рекурсивных методов и алгоритмов, разработанных первоначально для одномерной задачи, выполнено построение вычислительных алгоритмов для решения пространственной задачи переноса;

4. Проведен вычислительный эксперимент по исследованию свойств рекурсивно-итерационных алгоритмов решения одномерной задачи переноса, а именно сходимости и устойчивости к погрешностям в исходных данных; по исследованию влияния различных значений скорости ветра, турбулентности и других характеристик состояния пограничного слоя атмосферы на пространственно-временное распределение поля концентрации загрязняющих примесей на примере решения пространственной задачи переноса;

5. Предложена концептуальная схема информационно-вычислительного обеспечения задач переноса загрязняющих примесей в атмосфере и на ее основе разработана программно-алгоритмическая система компьютерного моделирования.

Литература

1. **Алоян, А.Е.** Математическое моделирование взаимодействия газовых примесей и аэрозолей в атмосферных дисперсных системах / А.Е. Алоян // Вычислительная математика и математическое моделирование: труды международной конференции. – т 1. – 2000. – С. 214 – 230.
2. **Алпатов, Ю.Н.** Синтез системы управления состоянием воздушного бассейна в промышленных зонах городов / Ю.Н. Алпатов, Ю.А. Михайлов // Математическое моделирование, численные методы и комплексы программ. Межвузовский тематический сборник трудов. – СПб. СПбГАСУ. – 1996. – С. 26 – 34.
3. **Бабешко, В.А.** К проблеме оценки выбросов загрязняющих веществ источниками различных типов / В.А. Бабешко, И.Б. Глацкой, М.В. Зарецкая, Е.В. Кособуцкая // Доклады Академии наук. – т. 342. – 1995. – № 6.
4. **Бабешко, В.А.** Математика и проблема безопасной эвакуации при авариях радиационной и токсической природы / В.А. Бабешко // Соровский образовательный журнал. – № 7. – 1997. – С. 116 – 120.
5. **Балов, А.В.** Интегральные методы исследования слабо выраженных крупномасштабных атмосферных и ионосферных облачных образований / А.В. Балов, Ю.П. Вербин, Л.С. Ивлев, Г.Н. Крылов // Естественные и антропогенные аэрозоли: 3-я Международная конференция. – СПб. – 2001. – С.551 – 558.
6. **Белов, П.Н.** Траекторная модель переноса загрязнений / П.Н. Белов, З.Л. Карлова // Метеорология и гидрология. – 1990. – № 12. – С. 67 – 74.
7. **Белоцерковский, О.М.** Турбулентность: новые подходы / О.М. Белоцерковский, А.М. Опарин, В.М. Чечеткин – М.: Наука, 2003. – 286 с.
8. **Белоцерковский, О.М.** Численный эксперимент в турбулентности: от порядка к хаосу / О.М. Белоцерковский, А.М. Опарин. – М.: Наука. – 2001. – 223 с.
9. **Вербин, Ю.П.** Дистанционный мониторинг слаборадиоактивных выбросов и атмосферных неоднородностей – предвестников землетрясений и вулканических извержений / Ю.П. Вербин, Л.С. Ивлев, Г.Н. Крылов // Фундаментальные исследования в технических университетах. Национальная безопасность. – СПб. – 2000. – С. 146 – 147.
10. **Воеводин, В.В.** Математические модели и методы в параллельных процессах / В.В. Воеводин. – М.: Наука, 1986. – 296 с.

11. **Воеводин, В.В.** Отображение проблем вычислительной математики на архитектуру вычислительных систем / В.В. Воеводин // Вычислительная математика и математическое моделирование. Международная конференция. – т. 1. – М.: МГУ. – 2000. – С. 242 – 255.
12. **Горстко, А.Б.** Введение в моделирование эколого-экономических систем / А.Б. Горстко, Г.А. Угольницкий. – Ростов – на – Дону: Изд-во РГУ, 1990. – 112 с.
13. **Демидович, Б.П.** Дифференциальные уравнения / Б.П. Демидович, В.П. Моденов. – СПб: Лань, 2008. – 288 с.
14. **Дударов, С.П.** Методы идентификации промышленных источников аварийных выбросов / С.П. Дударов, Т.В. Савицкая // Сборник трудов МНК ММТТ14. – т. 1. – Смоленск: СФМЭИ. – 2001.
15. **Дымников, В.П.** Основы математической теории климата / В.П. Дымников, А.Н. Филатов. – М.: ВИНТИ, 1994. – 252 с.
16. **Захаров, В.М.** Лидары и исследование климата / В.М. Захаров, О.К. Костко, С.С. Хмельцов. – Л: Гидрометеиздат, 1990. – 320 с.
17. **Израэль, Ю.А.** Экология и контроль состояния природной среды / Ю.А. Израэль. – М.: Гидрометеиздат, 1984. – 560 с.
18. **Кабанов, М.В.** Региональный мониторинг атмосферы. Научно-методические основы: монография / М.В. Кабанов: под общ. Ред. В.Е. Зуева. – Томск: Спектр, 1997. – Ч.1. – 211 с.
19. **Клочкова, Л.В.** Развитие методов математического прогнозирования качества окружающей среды на длительный период времени / Л.В. Клочкова, Ю.А. Повещенко, В.Ф. Тишкин // Современные проблемы математического моделирования: сборник трудов XIV молодежной конференции – школы с международным участием. – Ростов – на – Дону, изд-во ЮФУ. – 2011. – С. 147 – 162.
20. **Кожевникова, М.Ф.** Идентификация источников загрязнения: вычислительные методы / М.Ф. Кожевникова. – Харьков: Национальный научный центр «Харьковский физико – технический институт», 2011. – 156 с.
21. **Коханер, Д.** Численные методы и программное обеспечение / Д. Коханер, К. Моулер, С. Нэш. – М., 1998. – 328 с.
22. **Лайтхман, Д.Л.** Физика пограничного слоя атмосферы / Д.Л. Лайтхман. – Л., 1970. – 366 с.

23. **Марчук, Г.И.** Математические модели геофизической гидродинамики и численные методы их реализации / Г.И. Марчук, В.П. Дымников, В.Б Залесный. – Л.: Гидрометеиздат, 1987. – 296 с.
24. **Марчук, Г.И.** Математическое моделирование в проблеме окружающей среды / Г.И. Марчук. – М.: Наука, 1982. – 320 с.
25. **Марчук, Г.И.** Методы вычислительной математики / Г.И. Марчук. – М.: Наука, 1977. – 456 с.
26. **Марчук, Г.И.** Методы расщепления / Г.И. Марчук. – М.: Наука, 1988. – 262 с.
27. **Марчук, Г.И.** Проблемы вычислительной техники и фундаментальные исследования / Г.И. Марчук, В.Е. Котов // Автом. и вычисл. техн. 1979. – № 2. – С. 3 – 14.
28. **Марчук, Г.И.** Сопряженные уравнения и анализ сложных систем / Г.И. Марчук. – М.: Наука, 1992. – 336 с.
29. **Марчук, Г.И.** Численное решение задач динамики атмосферы и океана / Г.И. Марчук. – Л.: Гидрометеиздат, 1974. – 303 с.
30. **Марчук, Г.И.** Численные методы в прогнозе погоды / Г.И. Марчук. – Л.: Гидрометеиздат, 1967. – 356 с.
31. **Марчук, Г.И.,** Математическое моделирование общей циркуляции атмосферы и океана / Г.И. Марчук, В.П. Дымников, В.Б Залесный, В.Н. Лыкосов, В.Я. Галин. – Л.: Гидрометеиздат, 1984. – 320 с.
32. **Матвеев, Л.Г.** Курс общей метеорологии. Физика атмосферы / Л.Г. Матвеев. – Л.: Гидрометеиздат, 1984. – 752 с.
33. **Методика** расчета концентраций в атмосферном воздухе вредных веществ, содержащихся в выбросах предприятий: Госкомгидромет: ОНД-56 / под ред. Н.К. Берлянд и др. – Ленинград: Гидрометеиздат. – 1987. – 93 с.
34. **Наац, В.И.** Вычислительные методы и модели нестационарного диффузного переноса примесей в задачах контроля и прогноза экологического состояния атмосферы: дис. ... д – ра физ.-мат. наук: 05.13.18 / Наац Виктория Игоревна. - Ставрополь, 2005. – 339 с.
35. **Наац, В.И.** Вычислительный алгоритм в модели переноса примесей в турбулентной атмосфере на основе нестационарного трехмерного уравнения / В.И. Наац, Е.П. Ярцева // Моделирование и анализ информационных систем: труды международ-

ной научной конференции, посвященной 35-летию математического факультета и 25-летию факультета информатики и вычислительной техники Ярославского государственного университета им П.Г. Демидова. – Ярославль, ЯрГУ. – 2012. – С. 143 – 145.

36. Наац, В.И. Итерационные методы численного решения задач переноса на основе интегральных уравнений / В.И. Наац, Н.И. Каргин // Известия вузов. Северо-Кавказский регион. Естественные науки. – Приложение 3'04. – Ростов – на – Дону. – 2004. – С. 3 – 16.

37. Наац, В.И. Качественная модель оценки концентрации аэрозольных примесей в атмосфере, основанная на интегральном представлении решения уравнения турбулентной диффузии / В.И. Наац, Е.П. Ярцева // Известия высших учебных заведений. Северо – Кавказский регион. Естественные науки. – 2012. – № 1. – С. 38 – 43.

38. Наац, В.И. Математические модели и численные методы в задачах экологического мониторинга атмосферы / В.И. Наац, И.Э. Наац. – М.: ФИЗМАТЛИТ, 2010. – 328 с.

39. Наац, В.И. Метод интегральных уравнений в задачах переноса / В.И. Наац, И.Э. Наац // Сборник научных трудов. Серия «Физико-химическая». Выпуск 6. – Ставрополь. – 2002. – С. 99 – 101.

40. Наац, В.И. Метод интегральных уравнений в схеме покоординатного расщепления уравнения переноса аэрозолей в атмосфере / В.И. Наац, Е.П. Ярцева // Актуальные задачи математического моделирования и информационных технологий: материалы международной научно - практической конференции. – Сочи. – 2012. – С. 80 – 81.

41. Наац, В.И. Построение и исследование вычислительного алгоритма для параметризованного нестационарного уравнения переноса примесей в атмосфере / В.И. Наац, Е.П. Ярцева // Современные достижения в науке и образовании: математика и информатика: материалы международной научно-практической конференции. – Архангельск. – 2010. – С. 561 – 562.

42. Наац, В.И. Построение рекурсивно-итерационного алгоритма для трехмерного нестационарного уравнения переноса примесей в атмосфере / В.И. Наац, Е.П. Ярцева // Вестник Северо – Кавказского федерального университета. Научный журнал. – 2014. – № 2 (41). – С. 9 – 14.

43. Наац, В.И. Разработка качественной модели и итерационного алгоритма для оценки концентрации загрязняющих примесей в атмосфере / В.И. Наац, Е.П. Ярцева

// Вестник Северо – Кавказского федерального университета. Научный журнал. – 2013. – № 1 (34). – С. 15 – 21.

44. Наац, В.И. Система информационно - вычислительного обеспечения задач экологического мониторинга / В.И. Наац // Информационные технологии в науке, проектировании и производстве: XV Всероссийская научно – техническая конференция. – Нижний Новгород. – 2005. – С. 5 – 6.

45. Наац, В.И. Численное исследование рекурсивных и итерационных алгоритмов в задаче моделирования переноса аэрозолей в атмосфере / В.И. Наац, Е.П. Ярцева // Вестник Ставропольского государственного университета. Научный журнал. – 2011. – Выпуск 75 [4]. – С. 44 – 50.

46. Наац, И.Э. Изучение влияния выбора базисных функций на ошибку при решении задач влагопереноса / И.Э. Наац, В.С. Кирилов // Математическое моделирование в научных исследованиях: Всероссийская научная конференция. – Ставрополь. – 2000. – С.69 – 73.

47. Наац, И.Э. Математическое моделирование динамики пограничного слоя в задачах мониторинга окружающей среды / И.Э. Наац, Е.А. Семенчин. – Ставрополь: СГПУ, 1995. – 196 с.

48. Наац, И.Э. Метод обратной задачи в восстановлении характеристик светорассеяния дисперсными средами / И.Э. Наац, В.Д. Бумцев // ДАИ. Серия «Геофизика». – 1988. – т.303. – № 3. – С. 583 – 585.

49. Наац, И.Э. Обратные задачи оптики атмосферы / И.Э. Наац, В.Е. Зувев. – Л.: Гидрометеиздат. – 1990. – 270 с.

50. Наац, И.Э. Обратные задачи светорассеяния аэрозольными системами, взаимодействующими с физическими полями / И.Э. Наац // Оптика атмосферы. АН СССР. – 1989. - т.2. – № 10. – С. 1107 – 1112.

51. Наац, И.Э. Оптические методы в исследовании динамики пограничного слоя атмосферы / И.Э. Наац // Оптика атмосферы. АН СССР. – 1989. – т.2. – № 8. – С. 843 – 850.

52. Новиков, Ю.В. Охрана окружающей среды / Ю.В. Новиков. – М.: Высшая школа, 1999. – 287 с.

53. Пененко, В.В. Вариационные принципы и оптимизация во взаимосвязанных задачах экологии и климата / В.В. Пененко // Вычислительная математика и мате-

матическое моделирование: труды международной конференции. – т.1. – Москва. – 2000. – С.135 – 148.

54. Пененко, В.В. Методы численного моделирования атмосферных процессов / В.В. Пененко. – Л.: Гидрометеиздат, 1981. – 352 с.

55. Пененко, В.В. Модели и методы для задач охраны окружающей среды / В.В. Пененко, А.Е. Алоян. – М.: Наука, 1985. – 256 с.

56. Пененко, В.В. Подготовка данных для экологических исследований с использованием Reanalysis / В.В. Пененко, Е.А. Цветова // Оптика атмосферы и океана. – 1999. – т. 12. – № 5. – С. 463 – 465.

57. Региональный мониторинг атмосферы. Ч.2. Новые приборы и методики измерений / Под ред. М.В.Кабанова. – Изд.СО РАН. – Томск. – 1997. – 294 с.

58. Рыбакова, И. В. Численная модель дальнего и трансграничного переноса загрязнений в атмосфере / И.В. Рыбакова, П.И. Свиренко // Реферативный журнал по экологии, 1198.

59. Рыскаленко, Р.А. Разработка и исследование вычислительных моделей поля скорости ветра в атмосфере применительно к задачам экологического мониторинга: дис. ... канд. физ. – мат. наук: 05.13.18 / Рыскаленко Роман Андреевич. – Ставрополь, 2008. – 196 с.

60. Самарский, А.А. Численные методы решения задач конвекции – диффузии / А.А. Самарский, П.Н. Вабищевич. – Изд – во: Едиториал УРСС, 2009. – 248 с.

61. Самарский, А.А. Численные методы: учеб. пособие для вузов / А.А. Самарский, А.В. Гулин. – М.: Наука, гл. ред. физ.-мат. лит., 1989. – 432 с.

62. Семенчин, Е.А. Математические методы и модели в проблеме распространения примесей в температурно-стратифицированной атмосфере: дис. ...д - ра физ. – мат. наук: 05.13.18 / Семенчин Евгений Андреевич. – Ставрополь, 1997.

63. Семенчин, Е.А. Математическое моделирование нестационарного переноса примеси в пограничном слое атмосферы: монография / Е.А. Семенчин, В.И. Наац, И.Э. Наац. – М.: Изд-во физ. – мат. литературы, 2003. – 291 с.

64. Семенчин, Е.А. Стохастические методы решения обратных задач в математической модели атмосферной диффузии: монография / Е.А. Семенчин, М.В. Кузякина. – Москва: Физматлит, 2012. – 173 с.

- 65. Сухинов, А.И.** Двумерные схемы расщепления и некоторые их приложения: монография / А.И. Сухинов. – Москва: МГУ, 2005. – 408 с.
- 66. Сухинов, А.И.** Математическое моделирование движения многокомпонентной воздушной среды и транспорта / А.И. Сухинов, А.Е. Чистяков, Д.С. Хачунц // Известия Южного федерального университета. Актуальные задачи математического моделирования. – Ростов. – 2011. – С. 73 – 78.
- 67. Сухинов, А.И.** Построение и исследование дискретной математической модели расчета прибрежных волновых процессов / А.И. Сухинов, А.Е. Чистяков, Е.А. Проценко // Известия Южного федерального университета. Актуальные задачи математического моделирования. – Ростов. – 2011. – С. 22 – 32.
- 68. Угольницкий, Г.А.** Информационно – аналитическая система управления эколого-экономическими объектами / Г.А. Угольницкий, А.Б. Усов // Известия РАН. Теория и системы управления. – 2008. – № 2. – С. 168 – 176.
- 69. Угольницкий, Г.А.** Технологии разработки моделей эколого-экономических систем / Г.А. Угольницкий. – Ростов, 1995. – 33 с.
- 70. Хокни, Р.** Численное моделирование методом частиц / Р. Хокни, Дж. Иствуд. – М.: Мир, 1987. – 640 с.
- 71. Четвертушкин, Б.Н.** Применение высокопроизводительных многопроцессорных вычислений в газовой динамике / Б.Н. Четвертушкин, В.Ф. Тишкин // Математическое моделирование: проблемы и результаты. – Москва. – 2003. – С. 3 – 11.
- 72. Штырева, Н.В.** Численное моделирование дальнего переноса загрязняющих веществ в атмосфере / Н.В. Штырева // Реферативный журнал по экологии, 1197.
- 73. Янковская, Л.К.** Статистические модели и методы исследования переноса загрязнений в приземном слое атмосферы: дис. ... канд. физ. – мат. наук: 05.13.18 / Янковская Лариса Константиновна. – Ставрополь, 2002. – 170 с.
- 74. Ярцева Е.П.** Качественные модели и вычислительные алгоритмы оценки концентрации загрязняющих примесей в турбулентной атмосфере. Свидетельство о государственной регистрации программы для ЭВМ № 2013614085. Зарегистрировано в Реестре программ для ЭВМ 23 апреля 2013.
- 75. Ярцева Е.П.** Программа для реализации рекурсивных и итерационных вычислительных алгоритмов для пространственной задачи переноса загрязняющих примесей в турбулентной атмосфере. Свидетельство о государственной регистрации програм-

мы для ЭВМ № 2014661580. Зарегистрировано в Реестре программ для ЭВМ 31 октября 2014.

76. Ярцева, Е.П. Качественная модель оценки концентрации аэрозольных примесей в атмосфере на основе итерационного метода / Е.П. Ярцева // Наука и устойчивое развитие: материалы VII Всероссийской научно-практической конференции молодых ученых. – Нальчик. – 2013. – С. 176 – 178.

77. Ярцева, Е.П. Метод качественной оценки значений параметров интегральных моделей в задачах моделирования переноса субстанции в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы 56 научно-методической конференции преподавателей и студентов Ставропольского государственного университета. – Ставрополь. – 2011. – ч. 2. – С. 144 – 147.

78. Ярцева, Е.П. Оценка концентрации загрязняющих примесей в пограничном слое атмосферы на основе качественной модели / Е.П. Ярцева, В.И. Нац // Аналитические и численные методы моделирования естественнонаучных и социальных проблем: сборник статей VII Международной научно-технической конференции. – Пенза. – 2012. – С. 54 – 57.

79. Ярцева, Е.П. Построение и численные исследования трехмерной вычислительной модели переноса загрязняющих примесей в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы 57 научно-методической конференции преподавателей и студентов Ставропольского государственного университета. – Ставрополь. – 2012. – ч. 1. – С. 243 – 246.

80. Ярцева, Е.П. Построение итерационного вычислительного алгоритма качественной оценки концентрации загрязняющих примесей в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы ежегодной научно – практической конференции Северо – Кавказского федерального университета «Университетская наука - региону». – Ставрополь. – 2013. – С. 113 – 116.

81. Ярцева, Е.П. Построение модульной системы «Математические модели и вычислительные алгоритмы оценки концентрации загрязняющих примесей в турбулентной атмосфере» / Е.П. Ярцева // Наука и образование XXI века: сборник статей Международной научно – практической конференции. – Уфа. – 2014. – Часть 2. – С. 5 – 7.

- 82. Ярцева, Е.П.** Рекурсивная вычислительная схема решения уравнения переноса примесей в турбулентной атмосфере / Е.П. Ярцева // Ломоносов – 2011: материалы XVIII Международной конференции студентов, аспирантов и молодых ученых. – М.: МГУ. – 2011. – 1 с.
- 83. Ярцева, Е.П.** Численные исследования алгоритма решения нестационарного уравнения переноса загрязняющих веществ в атмосфере / Е.П. Ярцева // Университетская наука – региону: материалы 55 научно-методической конференции преподавателей и студентов Ставропольского государственного университета. – Ставрополь. – 2010. – ч. 1. – С. 112 – 116.
- 84. Ярцева, Е.П.** Численные исследования алгоритма решения уравнения переноса субстанции в турбулентной атмосфере / Е.П. Ярцева // Ломоносов – 2010: материалы XVII Международной конференции студентов, аспирантов и молодых ученых. – М.: МГУ. – 2010. – 1 с.
- 85. Ярцева, Е.П.** Численные исследования вычислительного алгоритма в задаче переноса примесей в турбулентной атмосфере на основе нестационарного трехмерного уравнения / Е.П. Ярцева, В.И. Наац // Актуальные научные вопросы: реальность и перспективы: материалы Международной заочной научно – практической конференции. – Тамбов. – 2012. – С. 165 – 166.
- 86. Ярцева, Е.П.,** Итерационный метод численного решения уравнения переноса примесей в атмосфере / Е.П. Ярцева, В.И. Наац // Актуальные задачи математического моделирования и информационных технологий: материалы VI Всероссийской открытой научно – практической конференции. – Сочи. – 2010. – С. 178 – 180.
- 87. Ясенский, А.Н.** Оптимизация пространственной структуры сети наблюдений при контроле загрязнений / А.Н. Ясенский // Труды ГГО. – 1987. – Вып. 492.
- 88. Aloyan, A.E.** Numerical modeling of lindane transport in the Northern Hemisphere / A.E. Aloyan, V.O. Arutyunyan. – MSC-E Rep. – 1997. – 37 с.
- 89. Aloyan, A.E.** Transport of coagulating aerosol in the atmosphere / A.E. Aloyan, V.O. Arutyunyan, A.A. Lushnikov, V.A. Zagainov // J. Aeros. Sci. – 1997. – Vol. 28. – №1. – P. 67 – 85.
- 90. Hopke, P.K.** Recent developments in receptor modeling / P.K. Hopke // Journal of Chemometrics. – 2003. – v. 17. – p. 225 – 265.

- 91. Hopke, P.K.** Receptor Modeling for Air Quality Management / P.K. Hopke // Elsevier: Amsterdam, 1991.
- 92. Penenko, V.V.** Methodology of inverse modeling for the problems of climate changes and environmental protection / V.V. Penenko // Advanced mathematics: computations and applications. – 1995. – Novosibirsk: NCC Publisher. – P. 358 – 367.
- 93. Voevodin, V.V.** Mathematical foundation of parallel computing / V.V. Voevodin. – World Scientific Publishing Co., Series in Computer Science, 1992. – 343 p.
- 94. Yartseva, E.P.** Building a Computational Algorithm to Model Pollutant in the Turbulent Atmosphere on the Basis of Unsteady Three-Dimensional Equation / E.P. Yartseva // International Multidisciplinary Journal European Researcher. – 2013. – № 5. – P. 1215 – 1219.
- 95. Zhiyu, Shen.** An empirical study of FORTRAN programs for parallelizing compilers / Zhiyu Shen, Zhiyuan Li, Pen-Chung Yew // IEEE Trans, on Parallel and Distributed Systems. –1990. – P. 350 – 364.

Приложение 1. Свидетельства о регистрации программ для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2013614085

Качественные модели и вычислительные алгоритмы
оценки концентрации загрязняющих примесей
в турбулентной атмосфере

Правообладатель(ли): *Ярцева Елена Павловна (RU)*Автор(ы): *Ярцева Елена Павловна (RU)*

Заявка № 2013611516

Дата поступления 26 февраля 2013 г.

Зарегистрировано в Реестре программ для ЭВМ
23 апреля 2013 г.Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014661580

Программа для реализации рекурсивных и итерационных вычислительных алгоритмов для пространственной задачи переноса загрязняющих примесей в турбулентной атмосфере

Правообладатель: **Ярцева Елена Павловна (RU)**

Автор: **Ярцева Елена Павловна (RU)**

Заявка № **2014617073**

Дата поступления **18 июля 2014 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **31 октября 2014 г.**

Врио руководителя Федеральной службы
по интеллектуальной собственности

Л.Л. Курий



Приложение 2. Таблицы значений метеорологических параметров, взятых из научных публикаций

Таблица 1 – Сравнение средних профилей ветра, полученных экспериментальным путем, в стратифицированном пограничном слое с расчетными значениями по схеме Лайхтмана [22]

Z м	u(z) при нейтральной стратификации, м/с			u(z) при неустойчивой стратификации, м/с			u(z) при устойчивой стратификации, м/с		
	Эксперимент	модель (а)	модель (б)	Эксперимент	модель (а)	модель (в)	Эксперимент	модель (а)	модель (в)
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,25	3,00	2,98	2,98	3,60	0,10	0,02	1,00	0,02	0,03
0,50	3,70	3,76	3,76	4,40	0,13	0,05	1,20	0,05	0,06
1,00	4,50	4,55	4,55	5,30	0,15	0,09	1,60	0,10	0,12
2,00	5,40	5,33	5,33	6,20	0,18	0,19	2,00	0,19	0,24
4,00	6,20	6,11	6,11	6,90	0,20	0,38	2,40	0,39	0,48
8,60	7,00	6,98	6,98	7,90	0,23	0,81	3,10	0,83	1,04
10,00	7,20	7,15	7,15	8,00	0,24	0,94	3,30	0,97	1,21
100,00	10,80	9,76	9,76	10,80	0,33	9,42	10,70	9,69	12,08
130,00	11,60	10,05	10,06	11,40	0,34	12,24	10,90	12,60	15,71
200,00	13,50	10,54	10,55	12,10	0,35	18,83	11,70	19,39	24,17
430,00	16,20	11,41	11,42	12,90	0,38	40,50	13,20	41,69	51,97
500,00	16,40	11,58	11,60	13,20	0,39	47,09	13,70	48,47	60,43
930,00	17,90	12,28	12,32	15,20	0,41	87,59	12,40	90,16	112,40
1430,00	18,40	12,77	12,82	17,00	0,43	134,68	11,70	138,64	172,83
1930,00	18,60	13,11	13,18	18,20	0,44	181,77	11,80	187,12	233,26
2930,00	18,40	13,58	13,70	18,90	0,45	275,95	11,90	284,07	354,13

Таблица 2 – Диапазон вероятных значений коэффициента турбулентной диффузии в зависимости от типа облачных систем при скорости ветра $V = (5-30)$ м/с [22, 63]

Тип облачной системы	Значение коэффициента турбулентной диффузии K (m^2/c)
Мощно-кучевая облачность CU-Cong	50 – 200
Слоисто-кучевая-дождевая Ns-As-Cb	50 – 500
Кучево-дождевая Cb	50 – 1500
Слоисто-дождевая облачность среднего яруса Ns-As-As	25 – 40
Слоисто-дождевая облачность Ns-As	2 – 25

Таблица 3 – Зависимость коэффициента α и скорости осаждения частиц на высоте $H=100$ м от размеров частиц [34, 73]

Вид осаждающейся примеси	Размеры частиц r_n (в мкм)	Скорость осаждения частиц w (в см/с)	Коэффициент α (в c^{-1})
легкая	1	0.013	$0,013 \cdot 10^{-4}$
	5	0.3	$0,3 \cdot 10^{-4}$
	10	1.3	$1,3 \cdot 10^{-4}$
средняя	50	30	$30 \cdot 10^{-4}$
тяжелая	100	130	$130 \cdot 10^{-4}$

Таблица 4 – Аномально неблагоприятные метеоусловия для основных групп источников выбросов в атмосферу [22, 63]

Характеристика выбросов	Термическая стратификация нижнего слоя атмосферы	Скорость ветра (м/с) на уровне		Вид инверсии, ее высота над источником
		флюгера	выбросов	
Горячие высокие	Неустойчивая	3-7 Штиль	5-10	Приподнятая (100-300 м)
Холодные высокие	-	1-2 Штиль	2-4	Приподнятая (100-300 м)
Низкие	Устойчивая	Штиль	Штиль	Приземная

Таблица 5 – Максимальные значения относительных концентраций сернистого газа в районе ПК при различных сочетаниях направления и скорости ветра [22, 63]

Скорость ветра, м/с	Направление ветра, румб							
	С	СВ	В	ЮВ	Ю	ЮЗ	З	СЗ
0,5	0,30	0,28	0,36	0,24	0,26	0,40	0,28	0,32
2	0,48	0,70	0,70	0,62	0,50	0,74	1,00	0,70
4	0,56	0,64	0,80	0,64	0,58	0,62	0,97	0,60
8	0,42	0,48	0,54	0,46	0,44	0,46	0,64	0,44

Таблица 6 – Значения фоновых концентраций загрязняющих веществ в атмосферном воздухе г. Ставрополя [34, 73]

Контрольная точка	Загрязняющие вещества		Фоновые концентрации				
			0 - 2 м/с	при скорости ветра 3 и более м/с			
	код	наименование		С	В	Ю	З
Стационарный пункт наблюдения	301	Азота диоксид	0.085	0.066	0.065	0.054	0.052
	330	Серы диоксид	0.004	0.002	0.003	0.003	0.003
	333	Сероводород	0.001	-	0.001	0.001	0.001
	337	Углерода оксид	2.376	1.743	1.656	1.581	1.576

Таблица 7 – Предельно допустимые концентрации в атмосферном воздухе населенных мест

Вещество	ПДК, мг/м ³		Класс опасности вещества
	максимальная разовая	средняя суточная	
Азота диоксид	0.085	0.04	2
Серы диоксид	0.5	0.05	3
Углерода оксид	5.0	3.0	4
Пыль (взвешенные вещества)	0.5	0.15	3
Аммиак	0.2	0.04	4
Кислота серная	0.3	0.1	2
Фенол	0.01	0.003	2
Ртуть металлическая	-	0.0003	1

Таблица 8 – Зависимость количества стационарных постов от численности населения [34, 73]

Численность населения, тыс. чел.	Количество постов
<50	1
50 – 100	2
100 – 200	3
200 – 500	3 – 5
500 – 1000	5 – 10
1000 – 2000	10 – 15
>2000	15 – 20

Приложение 3. Листинг программ

Приложение 3.1

```

restart;
# Тестовый пример для одномерной задачи переноса
V(x, t) := (1 + av·sin(bv·x + cv))·(1 + dv·sin(rv·t + gv)) :
K(x, t) := (1 + ak·sin(bk·x + ck))·(1 + dk·cos(rk·t + gk)) :
q(x, t) := (1 + aq·cos(bq·x + cq))·(1 + dq·cos(rq·t + gq)) :
P1 := diff(q(x, t), t) : P2 := q(x, t) : P3 := diff(V(x, t)·q(x, t), x) :
P4 := diff(K(x, t)·diff(q(x, t), x), x) :
α := 0.1 : β := 0.1 : θ := 0.001 : ξ := 1 : T := 10 : Smax := 0.0001 : qmax := 0.001 :
Q(x, t) := P1 + α·P2 + β·P3 - θ·P4 : S(x, t) :=  $\frac{Q(x, t) \cdot qmax}{Smax \cdot T}$  :
# Коэффициенты для скорости ветра
av := 0.8 : bv := 7 : cv := 1.5 : dv := 0.75 : rv := 2 : gv := 1.5 :
# Коэффициенты для турбулентной диффузии
ak := 0.75 : bk := 7 : ck := 3 : dk := 0.75 : rk := 7 : gk := 3 :
# Коэффициенты для концентрации ЗВ
aq := 0.075 : bq := -5 : cq := 5 : dq := -1 : rq := 2.75 : gq := 0.1 :
Q(x, t) := -(1 + aq cos(bq x + cq)) dq sin(rq t + gq) rq + α·((1 + aq cos(bq x + cq)) (1 + dq cos(rq t + gq))) + β·(av cos(bv x + cv) bv (1 + dv sin(rv t + gv)) (1 + aq cos(bq x + cq)) (1 + dq cos(rq t + gq)) - (1 + av sin(bv x + cv)) (1 + dv sin(rv t + gv)) aq sin(bq x + cq) bq (1 + dq cos(rq t + gq))) - θ·(-ak cos(bk x + ck) bk (1 + dk cos(rk t + gk)) aq sin(bq x + cq) bq (1 + dq cos(rq t + gq)) - (1 + ak sin(bk x + ck)) (1 + dk cos(rk t + gk)) aq cos(bq x + cq) bq² (1 + dq cos(rq t + gq))) :
S(x, t) :=  $\frac{qmax}{Smax \cdot T}$  · (2.75 (1 + 0.075 cos(5 x - 5)) sin(2.75 t + 0.1) + 0.1 (1 + 0.075 cos(5 x - 5)) (1 - cos(2.75 t + 0.1)) + 0.56 cos(7 x + 1.5) (1 + 0.75 sin(2 t + 1.5)) (1 + 0.075 cos(5 x - 5)) (1 - cos(2.75 t + 0.1)) - 0.0375 (1 + 0.8 sin(7 x + 1.5)) (1 + 0.75 sin(2 t + 1.5)) sin(5 x - 5) (1 - cos(2.75 t + 0.1)) + 0.00196875 cos(7 x + 3) (1 + 0.75 cos(7 t + 3)) sin(5 x - 5) (1 - cos(2.75 t + 0.1)) + 0.001875 (1 + 0.75 sin(7 x + 3)) (1 + 0.75 cos(7 t + 3)) cos(5 x - 5) (1 - cos(2.75 t + 0.1))) :
with (plots) :
plot3d(P1(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "P1"]) :
plot3d(α·P2(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "α·P2"]) :
plot3d(β·P3(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "β·P3"]) :
plot3d(θ·P4(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "θ·P4"]) :
plot3d(S(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "S"]) :
plot3d(V(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "V"]) :
plot3d(K(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "K"]) :
plot3d(q(x, t), x=0 ..1, t=0 ..1, color=navy, labels=["x", "t", "q"]) :

```

```

restart;
with(linalg):
#Одномерное уравнение - конечно-разностный метод
#Задание интервалов для x и t, задание сетки
m := 20 : a := 0 : b := 1 :
Δx := evalf((b-a)/m) : λ := evalf(1/2) : Δt := evalf((Δx)^2·λ) : nt := evalf(1/Δt) : n := 800 : x := array(0..m) : t := array(0..n) :

for i from 0 to m do x[i] := i·Δx, end do:
for j from 0 to n do t[j] := j·Δt, end do:
#Задание полей скорости ветра, коэф. турбулентной диффузии, концентрации загрязняющих веществ
av := 0.8 : bv := 7 : cv := 1.5 : dv := 0.75 : rv := 2 : gv := 1.5 :
V := proc(u, w) (1 + av·sin(bv·u + cv))·(1 + dv·sin(rv·w + gv)); end proc:
bk := 7 : ck := 3 : dk := 0.75 : rk := 7 : gk := 3 : ak := 0.75 :
K := proc(u, w) (1 + ak·sin(bk·u + ck))·(1 + dk·cos(rk·w + gk)); end proc:
bq := -5 : aq := 0.075 : cq := 5 : dq := -1 : rq := 2.75 : gq := 0.1 :
q := proc(u, w) (1 + aq·cos(bq·u + cq))·(1 + dq·cos(rq·w + gq)); end proc:
α := 0.1 : β := 0.1 : θ := 0.001 : T := 10 : V0 := 10 : K0 := 100 :
Q := proc(u, w) diff(q(u, w), w) + α·q(u, w) + β·diff(V(u, w)·q(u, w), u) -
θ·diff(K(u, w)·diff(q(u, w), u), u); end proc:
S0 := 0.0001 : q0 := 0.001 :
#Задание массивов исходных данных
v := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
v[i, j] := V(x[i], t[j]);
end do;
end do;
Vmax := V0·max(seq(seq(v[i, j], j=0..n), i=0..m)) :
for i from 0 by 4 to m do
for j from 0 by 160 to n do
#print("v[" , i, j, "]=", v[i, j]);
end do;
end do;
kf := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
kf[i, j] := K(x[i], t[j]);
end do;
end do;
Kmax := K0·max(seq(seq(kf[i, j], j=0..n), i=0..m)) :
for i from 0 by 4 to m do
for j from 0 by 160 to n do
#print("kf[" , i, j, "]=", kf[i, j]);
end do;
end do;
qtoch := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
qtoch[i, j] := q(x[i], t[j]);
end do;
end do;
qmax := q0·max(seq(seq(qtoch[i, j], j=0..n), i=0..m)) :
for i from 0 by 2 to m do
for j from 0 by 80 to n do
#print("qtoch[" , i, j, "]=", qtoch[i, j]);
end do;
end do;
s := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
s[i, j] := evalf(subs(u=x[i], w=t[j], Q(u, w)));
end do;
end do;
for i from 0 by 4 to m do
for j from 0 by 160 to n do
#print("s[" , i, j, "]=", s[i, j]);
end do;
end do;
ξ := max(seq(seq(s[i, j], j=0..n), i=0..m)) :
X := Kmax·β/θ : T := Kmax·β^2/Vmax^2 : Smax := qmax/T·ξ :
#Блок моделирования
k := m-1 : q := array(0..m, 0..n) :
for i from 0 to m do
q[i, 0] := qtoch[i, 0];
end do;
σ := array(0..n) : sl := 0 : σ[0] := 0 : ε := 0.1 : j := 0 :
while σ[j] < ε and j < n do
pogr := 0 :
Matrsist := array(1..k, 1..k) :
for ll from 1 to k do

```

```

for l2 from 1 to k do
  Matrsis[l1, l2] := 0;
end do;
end do;
for l from 1 to k do
  if l - 1 > 0 then
    Matrsis[l - 1, l] := -  $\frac{\theta \cdot \lambda}{2}$  · kf[l, j] :
  end if;
  Matrsis[l, l] := 1 +  $\lambda \cdot \theta \cdot kf[l, j]$  :
  if l + 1 ≤ k then
    Matrsis[l + 1, l] := -  $\frac{\theta \cdot \lambda}{2}$  · kf[l, j] :
  end if;
end do;
fsist := array(1 ..k) :
for i from 1 to k do
  somnog1 := 0.5 ·  $\lambda \cdot \Delta x \cdot \beta \cdot v[i, j]$  - 0.5 ·  $\lambda \cdot \Delta x \cdot \theta \cdot (kf[i + 1, j] - kf[i - 1, j])$  + 0.5 ·  $\lambda \cdot \theta \cdot kf[i, j]$ ;
  somnog2 := 1 -  $\alpha \cdot \Delta x$  - 0.5 ·  $\lambda \cdot \Delta x \cdot \beta \cdot (v[i + 1, j] - v[i - 1, j])$  -  $\lambda \cdot \theta \cdot kf[i, j]$ ;
  somnog3 := -0.5 ·  $\lambda \cdot \Delta x \cdot \beta \cdot v[i, j]$  + 0.5 ·  $\lambda \cdot \Delta x \cdot \theta \cdot (kf[i + 1, j] - kf[i - 1, j])$  + 0.5 ·  $\lambda \cdot \theta \cdot kf[i, j]$ ;
  slag := s[i, j] ·  $\Delta x$ ;
  fsist[i] := q[i - 1, j] · somnog1 + q[i, j] · somnog2 + q[i + 1, j] · somnog3 + slag;
end do;
y := linsolve(Matrsist, fsist) :
multiply(Matrsist, y) :
q[0, j + 1] := qtoch[0, j + 1] :
for i from 1 to k do
  q[i, j + 1] := y[i] :
end do;
q[m, j + 1] := qtoch[m, j + 1] :
for i from 1 to k do
  #print(q[i, j + 1], " ", qtoch[i, j + 1]);
  pogr := pogr + abs(q[i, j + 1] - qtoch[i, j + 1]) :
end do;
 $\sigma[j + 1] := \frac{pogr}{m - 1}$  :
#print("σj=",  $\sigma[j + 1]$ ) :
s1 := s1 +  $\sigma[j + 1]$ ;
j := j + 1 :
end do;
jend := j :
for i from 0 by 2 to m do
  for j from 0 by 80 to jend do
    #print("q[" , i, j, "]=", q[i, j]);
  end do;
end do;
#print("jend=", j, " sigma=",  $\sigma[jend - 1]$ );  $\alpha t := \frac{s1}{j}$  : print("sigmaitog=",  $\alpha t$ );
eps1 := 0.001 : n := jend :
slc := array(0 ..m, 0 ..n) : Sp := array(0 ..m, 0 ..n) :
for i from 0 to m do
  for j from 0 to n do
    slc[i, j] := evalf(-1 + 2 ·  $\frac{rand()}{1000000000000}$ , 2);
    Sp[i, j] := s[i, j] · (1 + slc[i, j] · eps1);
  end do;
end do;
#Защумление источника S(x, t)
k := m - 1 : qprib := array(0 ..m, 0 ..n) :
for i from 0 to m do
  qprib[i, 0] := qtoch[i, 0];
end do;
 $\sigma := array(0 ..n) : s1 := 0 : \sigma[0] := 0 : \epsilon := 0.1 : j := 0$  :
while  $\sigma[j] < \epsilon$  and j < n do
  pogr := 0 : Matrsist := array(1 ..k, 1 ..k) :
  for l1 from 1 to k do
    for l2 from 1 to k do
      Matrsis[l1, l2] := 0;
    end do;
  end do;
  for l from 1 to k do
    if l - 1 > 0 then
      Matrsis[l - 1, l] := -  $\frac{\theta \cdot \lambda}{2}$  · kf[l, j] :
    end if;
    Matrsis[l, l] := 1 +  $\lambda \cdot \theta \cdot kf[l, j]$  :
    if l + 1 ≤ k then
      Matrsis[l + 1, l] := -  $\frac{\theta \cdot \lambda}{2}$  · kf[l, j] :
    end if;
  end do;

```



```

end if;
end do;
fsist := array(1..k);
for i from 1 to k do
  somnog1 := 0.5·λ·Δx·β·v[i,j] - 0.5·λ·Δx·θ·(kf[i+1,j] - kf[i-1,j]) + 0.5·λ·θ·kf[i,j];
  somnog2 := 1 - α·Δt - 0.5·λ·Δx·β·(v[i+1,j] - v[i-1,j]) - λ·θ·kf[i,j];
  somnog3 := -0.5·λ·Δx·β·v[i,j] + 0.5·λ·Δx·θ·(kf[i+1,j] - kf[i-1,j]) + 0.5·λ·θ·kf[i,j];
  slag := Sp[i,j]·Δx;
  fsist[i] := qprib[i-1,j]·somnog1 + qprib[i,j]·somnog2 + qprib[i+1,j]·somnog3 + slag;
end do;
y := linsolve(Matrsist, fsist) : multiply(Matrsist, y) : qprib[0,j+1] := qtoch[0,j+1] :
for i from 1 to k do
  qprib[i,j+1] := y[i] :
end do;
qprib[m,j+1] := qtoch[m,j+1] :
for i from 1 to k do
  #print(qprib[i,j+1], " ", qtoch[i,j+1]);
  pogr := pogr + abs(qprib[i,j+1] - qtoch[i,j+1]) :
end do;
σ[j+1] :=  $\frac{pogr}{m-1}$  :
#print("σj=", σ[j+1]) :
sl := sl + σ[j+1]; j := j+1 :
end do;
#print("jend=", j, " sigma=", σ[jend-1]); αt :=  $\frac{sl}{j}$  : print("sigmaitog=", αt);
δ := sqrt(sum(sum((Sp[i1,j1] - s[i1,j1])2, i1=0..m), j1=0..n)) :
αq := sqrt(sum(sum((qprib[i1,j1] - qtoch[i1,j1])2, i1=0..m), j1=0..n)) :
δotn := sqrt( $\frac{\text{sum}(\text{sum}((Sp[i1,j1] - s[i1,j1])^2, i1=0..m), j1=0..n)}{\text{sum}(\text{sum}(s[i1,j1]^2, i1=0..m), j1=0..n)}$ ) :
αqotn := sqrt( $\frac{\text{sum}(\text{sum}((qprib[i1,j1] - qtoch[i1,j1])^2, i1=0..m), j1=0..n)}{\text{sum}(\text{sum}(qtoch[i1,j1]^2, i1=0..m), j1=0..n)}$ ) :

η :=  $\frac{\delta}{\alpha_q}$  : ηl :=  $\frac{\delta_{otn}}{\alpha_{qotn}}$  :
A := matrix(m-1, n) :
for j from 1 to n do
  for i from 1 to m-1 do
    A[i,j] := qtoch[i,j];
  end do;
end do;
with(plots) : with(LinearAlgebra) :
#matrixplot(A, style=patch, axes=boxed, labels=["x", "t", "qtoch"]);
B := matrix(m-1, n) :
for j from 1 to n do
  for i from 1 to m-1 do
    B[i,j] := qprib[i,j];
  end do;
end do;
#matrixplot(B, style=patch, axes=boxed, labels=["x", "t", "q"]);

```

```

restart,
with(linalg):
#Одномерное уравнение - метод интегральных уравнений
#Задание интервалов для x и t, задание сетки
m := 10 : a := 0 : b := 1 :

$$\Delta x := \text{evalf}\left(\frac{(b-a)}{m}\right) : \lambda := \text{evalf}\left(\frac{1}{2}\right) : \Delta t := \text{evalf}\left((\Delta x)^2 \cdot \lambda\right) : nt := \text{evalf}\left(\frac{1}{\Delta t}\right) :$$

n := 200 : x := array(0 ..m) : t := array(0 ..n) :
for k from 0 to m do x[k] := k· $\Delta x$ ; end do:
for j from 0 to n do t[j] := j· $\Delta t$ ; end do:
#Задание полей скорости ветра, коэф. турбулентной диффузии, концентрации загрязняющих веществ
av := 0.8 : bv := 7 : cv := 1.5 : dv := 0.75 : rv := 2 : gv := 1.5 :
V := proc(u, w) (1 + av·sin(bv·u + cv))·(1 + dv·sin(rv·w + gv)); end proc:
bk := 7 : ck := 3 : dk := 0.75 : rk := 7 : gk := 3 : ak := 0.75 :
K := proc(u, w) (1 + ak·sin(bk·u + ck))·(1 + dk·cos(rk·w + gk)); end proc:
bq := -0.1 : aq := 2 : cq := 1 : dq := -1 : rq := 2.5 : gq := 0.1 :
q := proc(u, w) (1 + aq·cos(bq·u + cq))·(1 + dq·cos(rq·w + gq)); end proc:
 $\alpha := 0.1 : \beta := 0.1 : \theta := 0.001 : T := 10 : V0 := 10 : K0 := 100 :$ 
Q := proc(u, w) diff(q(u, w), w) +  $\alpha$ ·q(u, w) +  $\beta$ ·diff(V(u, w)·q(u, w), u) -
 $\theta$ ·diff(K(u, w)·diff(q(u, w), u), u); end proc:
S0 := 0.0001 : q0 := 0.001 :
# Задание массивов исходных данных
v := array(0 ..m, 0 ..n) :
for k from 0 to m do
for j from 0 to n do
v[k, j] := V[x[k], t[j]];
end do;
end do;
Vmax := V0·max(seq(seq(v[k, j], j=0 ..n), k=0 ..m)) :
for i from 0 by 4 to m do
for j from 0 by 160 to n do
#print("v[" , k, j, "]=", v[k, j]);
end do;
end do;
kf := array(0 ..m, 0 ..n) :
for k from 0 to m do
for j from 0 to n do
kf[k, j] := K[x[k], t[j]];
end do;
end do;
Kmax := K0·max(seq(seq(kf[k, j], j=0 ..n), k=0 ..m)) :
for k from 0 by 4 to m do
for j from 0 by 160 to n do
#print("kf[" , k, j, "]=", kf[k, j]);
end do;
end do;
qtoch := array(0 ..m, 0 ..n) :
for k from 0 to m do
for j from 0 to n do
qtoch[k, j] := q[x[k], t[j]];
end do;
end do;
qmax := q0·max(seq(seq(qtoch[k, j], j=0 ..n), k=0 ..m)) :
s := array(0 ..m, 0 ..n) :
for k from 0 to m do
for j from 0 to n do
s[k, j] := evalf(subs(u=x[k], w=t[j], Q(u, w)));
end do;
end do;
for k from 0 by 4 to m do
for j from 0 by 160 to n do
#print("s[" , k, j, "]=", s[k, j]);
end do;
end do;
 $\xi := \max(\text{seq}(\text{seq}(s[k, j], j=0 ..n), k=0 ..m)) :$ 

$$X := \frac{Kmax}{Vmax} \cdot \frac{\beta}{\theta} : T := \frac{Kmax}{Vmax^2} \cdot \frac{\beta^2}{\theta} : Smax := \frac{qmax}{T \cdot \xi} :$$

# Метод последовательных приближений
 $\omega := \text{array}(0 ..n) : \omega[0] := \text{evalf}\left(\frac{1}{2}\right) : \omega[n] := \text{evalf}\left(\frac{1}{2}\right)$ 
for i from 1 to n - 1 do
 $\omega[i] := 1;$ 
end do:
# Номер итерации v
v := 0 :
# Вычисление ядра Ksv и функции  $\phi$ 
Ksv := array(0 ..m, 0 ..n, 0 ..n) :  $\phi := \text{array}(0 ..m, 0 ..n) :$ 
qv := array(0 ..m, 0 ..n + 1, 0 ..25) :  $\psi := \text{array}(0 ..m, 0 ..n, 0 ..25) : qm := \text{array}(0 ..m, 0 ..n) :$ 
for k from 1 to m - 1 do
for j from 1 to n do

```

```

s3 := 0; s4 := 0;
for i from 0 to j do
p1 :=  $\alpha \cdot \Delta x^2 + \beta \cdot \Delta x \cdot (v[k, i] - v[k-1, i])$ ; s3 := s3 +  $\omega[i] \cdot p1$ ;
end do;

s6 :=  $qtoch[k, 0] \cdot e^{(-\lambda \cdot s3)}$ ;
for i from 0 to j do
ss := 0;
for l from i to j do
ss := ss +  $\omega[l] \cdot \Delta x \cdot (\alpha \cdot \Delta x + \beta \cdot (v[k, l] - v[k-1, l]))$ ;
end do;

Ksv[k, i, j] :=  $e^{(-\lambda \cdot ss)}$ ;
s4 := s4 +  $\lambda \cdot \omega[i] \cdot s[k, i] \cdot Ksv[k, i, j] \cdot \Delta x^2$ ;
end do;
 $\varphi[k, j] := s6 + s4$ ;
 $qv[k, j, v] := \varphi[k, j]$ ;
#print("k=", k, "j=", j, "v=", v);
#print("qv=", qv[k, j, v]);
end do;
end do;
# Вычисление функции  $\psi$ 
s8 := 100;
 $\epsilon := 0.001$ ;
while s8 >  $\epsilon$  do
v := v + 1;
# Задание начальных условий
for k from 0 to m do
 $qv[k, 0, v-1] := qtoch[k, 0]$ ;
 $qm[k, 0] := qtoch[k, 0]$ ;
end do;
# Задание краевых условий
for j from 0 to n do
 $qv[0, j, v-1] := qtoch[0, j]$ ;
 $qm[0, j] := qtoch[0, j]$ ;
 $qv[m, j, v-1] := qtoch[m, j]$ ;
 $qm[m, j] := qtoch[m, j]$ ;
end do;
for k from 1 to m-1 do
for j from 1 to n do
s1 := 0;
for i from 0 to j do
 $p2 := qv[k, i, v-1] - qv[k-1, i, v-1]$ ;  $p3 := \beta \cdot \Delta x \cdot v[k, i]$ ;
 $p4 := \theta \cdot (kf[k, i] - kf[k-1, i])$ ;  $p5 := p2 \cdot (p3 - p4)$ ;
 $p6 := qv[k+1, i, v-1] - 2 \cdot qv[k, i, v-1] + qv[k-1, i, v-1]$ ;
 $p7 := \theta \cdot kf[k, i] \cdot p6$ ;  $\psi[k, i, v-1] := p5 - p7$ ;
 $s1 := s1 + \lambda \cdot \omega[i] \cdot Ksv[k, i, j] \cdot \psi[k, i, v-1]$ ;
end do;
 $qm[k, j] := \varphi[k, j] - s1$ ;
#print("k=", k, "j=", j, "qm[k, j]=", qm[k, j]);
end do;
end do;
for j from 1 to n do
for k from 1 to m-1 do
 $qv[k, j, v] := qm[k, j]$ ;
#print("qv[k, j, v]=", qv[k, j, v]);
end do;
end do;
s7 := 0;
for j from 1 to n do
for k from 1 to m-1 do
 $s7 := s7 + (abs(qv[k, j, v] - qv[k, j, v-1]))$ ;
#print("qv[v-1]=", qv[k, j, v-1], "qv[v]=", qv[k, j, v], "otkl=", s7);
end do;
end do;
 $s8 := \frac{s7}{n \cdot (m-1)}$ ;
#print("v=", v, "sredotkl=", s8);
end do;
s9 := 0; s11 := 0;
for k from 1 to m-1 do
for j from 1 by 20 to n do
 $s9 := s9 + abs(qtoch[k, j] - qv[k, j, v])$ ;
 $s11 := s11 + abs(qtoch[k, j])$ ;
end do;
end do;

```

```

s10 :=  $\frac{s9}{n \cdot (m-1)}$  : s12 :=  $\frac{s9}{s11}$  :
#print("v=", v, "sredotkl=", s10, "sotn=", s12);
eps1 := 0.125 : slc := array(0..m, 0..n) : Sp := array(0..m, 0..n) :
for k from 0 to m do
for j from 0 to n do
slc[k, j] := evalf(-1 + 2 *  $\frac{\text{rand}()}{1000000000000}$ , 2);
Sp[k, j] := s[k, j] * (1 + slc[k, j] * eps1);
end do;
end do;
# За шумление источника S(x,t)
# Метод последовательных приближений
omega := array(0..n) : omega[0] := evalf( $\frac{1}{2}$ ) : omega[n] := evalf( $\frac{1}{2}$ ) :
for i from 1 to n - 1 do
omega[i] := 1;
end do;
# Номер итерации v
v := 0;
# Вычисление ядра Ksv и функции phi
Ksv := array(0..m, 0..n, 0..n) : phi := array(0..m, 0..n) :
qvpribl := array(0..m, 0..n + 1, 0..25) : psi := array(0..m, 0..n, 0..25) : qm := array(0..m, 0..n) :
for k from 1 to m - 1 do
for j from 1 to n do
s3 := 0; s4 := 0;
for i from 0 to j do
p1 :=  $\alpha \cdot \Delta^2 + \beta \cdot \Delta \cdot (v[k, i] - v[k - 1, i])$ ; s3 := s3 + omega[i] * p1;
end do;
s6 := qtoch[k, 0] * e(-lambda * s3);
for i from 0 to j do
ss := 0;
for l from i to j do
ss := ss + omega[l] *  $\Delta \cdot (\alpha \cdot \Delta + \beta \cdot (v[k, l] - v[k - 1, l]))$ ;
end do;
Ksv[k, i, j] := e(-lambda * ss); s4 := s4 + lambda * omega[i] * Sp[k, i] * Ksv[k, i, j] *  $\Delta^2$ ;
end do;
phi[k, j] := s6 + s4; qvpribl[k, j, v] := phi[k, j];
#print("k=", k, "j=", j, "v=", v);
#print("qvpribl=", qvpribl[k, j, v]);
end do;
end do;
# Вычисление функции psi
s8 := 100;
epsilon := 0.001;
while s8 > epsilon do
v := v + 1;
# Задание начальных условий
for k from 0 to m do
qvpribl[k, 0, v - 1] := qtoch[k, 0];
qm[k, 0] := qtoch[k, 0];
end do;
# Задание краевых условий
for j from 0 to n do
qvpribl[0, j, v - 1] := qtoch[0, j];
qm[0, j] := qtoch[0, j];
qvpribl[m, j, v - 1] := qtoch[m, j];
qm[m, j] := qtoch[m, j];
end do;
for k from 1 to m - 1 do
for j from 1 to n do
s1 := 0;
for i from 0 to j do
p2 := qvpribl[k, i, v - 1] - qvpribl[k - 1, i, v - 1]; p3 := beta * Delta * v[k, i];
p4 := theta * (kf[k, i] - kf[k - 1, i]); p5 := p2 * (p3 - p4);
p6 := qvpribl[k + 1, i, v - 1] - 2 * qvpribl[k, i, v - 1] + qvpribl[k - 1, i, v - 1];
p7 := theta * kf[k, i] * p6; psi[k, i, v - 1] := p5 - p7;
s1 := s1 + lambda * omega[i] * Ksv[k, i, j] * psi[k, i, v - 1];
end do;
qm[k, j] := phi[k, j] - s1;
#print("k=", k, "j=", j, "qm[k, j]=", qm[k, j]);
end do;
end do;
for j from 1 to n do
for k from 1 to m - 1 do

```

```

qvprib[k, j, v] := qm[k, j];
#print("qvprib[k, j, v]=", qvprib[k, j, v]);
end do;
end do;
s7 := 0;
for j from 1 to n do
for k from 1 to m-1 do
s7 := s7 + (abs(qvprib[k, j, v] - qvprib[k, j, v-1]));
#print("qvprib[v-1]=", qvprib[k, j, v-1], "qvprib[v]=", qvprib[k, j, v], "otki=", s7);
end do;
end do;
s8 :=  $\frac{s7}{n \cdot (m-1)}$ ;
#print("v=", v, "sredotki=", s8);
end do;
for k from 1 to m-1 do
for j from 1 by 19 to n do
#print("qtoch[", k, j, "]=", qtoch[k, j], "qv[", k, j, "]=", qv[k, j, v]);
end do;
end do;
s9 := 0; s11 := 0;
for k from 1 to m-1 do
for j from 1 by 20 to n do
s9 := s9 + abs(qtoch[k, j] - qvprib[k, j, v]);
s11 := s11 + abs(qtoch[k, j]);
end do;
end do;
s10 :=  $\frac{s9}{n \cdot (m-1)}$ ; s12 :=  $\frac{s9}{s11}$ ;
#print("v=", v, "sredotki=", s10, "sotn=", s12);
 $\delta$  := sqrt(sum(sum((Sp[kl, jl] - s[kl, jl])2, kl=0..m), jl=0..n));
 $\delta q$  := sqrt(sum(sum((qm[kl, jl] - qtoch[kl, jl])2, kl=0..m), jl=0..n));
 $\delta$ otn := sqrt( $\frac{\text{sum}(\text{sum}((\text{Sp}[kl, jl] - s[kl, jl])^2, kl=0..m), jl=0..n)}{\text{sum}(\text{sum}(s[kl, jl]^2, kl=0..m), jl=0..n)}$ );
 $\delta$ qotn := sqrt( $\frac{\text{sum}(\text{sum}((qm[kl, jl] - qtoch[kl, jl])^2, kl=0..m), jl=0..n)}{\text{sum}(\text{sum}(qtoch[kl, jl]^2, kl=0..m), jl=0..n)}$ );
 $\eta$  :=  $\frac{\delta}{\delta q}$ ;  $\eta l$  :=  $\frac{\delta \text{otn}}{\delta q \text{otn}}$ ;
A := matrix(m-1, n);
for j from 1 to n do
for k from 1 to m-1 do
A[k, j] := qtoch[k, j];
end do;
end do;
with(plots):
with(LinearAlgebra):
#matrixplot(A, style=patch, axes=boxed, labels=["x", "t", "qtoch"]);
B := matrix(m-1, n);
for j from 1 to n do
for k from 1 to m-1 do
B[k, j] := qm[k, j];
end do;
end do;
#matrixplot(B, style=patch, axes=boxed, labels=["x", "t", "q"]);

```

```

restart,
with(linalg):
#Одномерное уравнение - первая качественная модель
#Задание интервалов для x и t, задание сетки
m := 15 : n := 20 :
a := 0 : b := 1 : c := 0 : d := 1 :

$$\Delta x := \text{evalf}\left(\frac{b-a}{m}\right) : \Delta t := \text{evalf}\left(\frac{d-c}{n}\right) :$$

x := array(0..m) : t := array(0..n) :
for i from 0 to m do x[i] := i· $\Delta x$ ; end do:
for j from 0 to n do t[j] := j· $\Delta t$ ; end do:
#Задание полей скорости ветра, коэффициента турбулентной диффузии, концентрации загрязняющих веществ
 $\alpha := 0.1 : \beta := 1 : \theta := 1 : V0 := 15 : K0 := 50 : q0 := 0.005 :$ 
 $aq := -0.2 : bq := 3 : cq := 1.5 : dq := 1 : rq := -1 : gq := 1.5 :$ 
V := proc(u, w) V0·1; end proc: K := proc(u, w) K0·1; end proc:
q := proc(u, w) q0·(1 + aq·cos(bq·u + cq))·(1 + dq·cos(rq·w + gq)); end proc:
Q := proc(u, w) diff(q(u, w), w) +  $\alpha$ ·q(u, w) +  $\beta$ ·diff(q(u, w), u) -  $\theta$ ·diff(diff(q(u, w), u), u); end proc:
#Задание массивов исходных данных - тестовый пример
v := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
v[i, j] := V(x[i], t[j]);
end do;
end do;
#print("v[" , i, j, "]=", v[i, j]);
Vmax := max(seq(seq(v[i, j], j=0..n), i=0..m)) :
for i from 0 to m do
for j from 0 to n do
vnorm[i, j] :=  $\frac{v[i, j]}{Vmax}$  ;
end do;
end do;
kf := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
kf[i, j] := K(x[i], t[j]);
end do;
end do;
#print("kf[" , i, j, "]=", kf[i, j]);
Kmax := max(seq(seq(kf[i, j], j=0..n), i=0..m)) :
for i from 0 to m do
for j from 0 to n do
kfnorm[i, j] :=  $\frac{kf[i, j]}{Kmax}$  ;
end do;
end do;
qtoch := array(0..m, 0..n) :
qtochnorm := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
qtoch[i, j] := q(x[i], t[j]);
end do;
end do;
#print("qtoch[" , i, j, "]=", qtoch[i, j]);
qmax := max(seq(seq(qtoch[i, j], j=0..n), i=0..m)) :
for i from 0 to m do
for j from 0 to n do
qtochnorm[i, j] :=  $\frac{qtoch[i, j]}{qmax}$  ;
end do;
end do;
s := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
s[i, j] := evalf(subs(u=x[i], w=t[j], Q(u, w)));
end do;
end do;
#print("s[" , i, j, "]=", s[i, j]);
Smax := max(seq(seq(s[i, j], j=0..n), i=0..m)) :

$$T := \frac{Kmax \cdot \beta^2}{Vmax^2 \cdot \theta} : X := \frac{Kmax \cdot \beta}{Vmax \cdot \theta} : \beta := \frac{Vmax \cdot T}{X} : \theta := \frac{Kmax \cdot T}{X^2} : \xi := \frac{Smax \cdot T}{qmax} :$$

snorm := array(0..m, 0..n) :
for i from 0 to m do
for j from 0 to n do
snorm[i, j] :=  $\frac{s[i, j]}{\xi}$  ;
end do;
end do;
#Блок моделирования

```

```

r1 :=  $\frac{\beta}{2 \cdot \theta} - \text{sqrt}\left(\frac{\beta^2}{4 \cdot \theta^2} + \frac{\alpha}{\theta}\right)$ ; r2 :=  $\frac{\beta}{2 \cdot \theta} + \text{sqrt}\left(\frac{\beta^2}{4 \cdot \theta^2} + \frac{\alpha}{\theta}\right)$ ;
λ :=  $\frac{1}{\theta} \cdot \frac{1}{r2 - r1}$ ; μ :=  $\frac{\lambda \cdot \Delta x}{\Delta x}$ ; η := λ · ξ · Δx;
ω := array(0..m): ω[0] := evalf( $\frac{1}{2}$ ): ω[m] := evalf( $\frac{1}{2}$ ):
for i from 1 to m-1 do
  ω[i] := 1;
end do;
#Задание начальных условий
q := array(0..m, 0..n):
for i from 0 to m do
  q[i, 0] := qtochnorm[i, 0];
end do;
#Задание краевых условий
for j from 0 to n do
  q[0, j] := qtochnorm[0, j];
  q[m, j] := qtochnorm[m, j];
end do;
Ksv := array(0..m, 0..m):
for i from 0 to m do
  for k from 0 to i do
    h1 :=  $e^{r2 \cdot (x[i] - x[k])}$ ; h2 :=  $e^{r1 \cdot (x[i] - x[k])}$ ;
    Ksv[i, k] :=  $e^{r2 \cdot (x[i] - x[k])} - e^{r1 \cdot (x[i] - x[k])}$ ;
    #print("h1=", h1, " h2=", h2, "Ksv[" , i, k, "], " , Ksv[i, k]);
  end do;
end do;
φ := array(0..m, 1..n):
for i from 0 to m do
  for j from 1 to n do
    s1 := 0;
    for k from 0 to i do
      if k=0 then w[k] := 0.5; end if;
      if k=i then w[k] := 0.5; end if;
      if k > 0 and k < i then w[k] := 1; end if;
      s1 := s1 + ω[k] · Ksv[i, k] · s[k, j];
    end do;
    φ[i, j] := q[0, j] - η · s1;
  end do;
end do;
for i from 1 to m-1 do
  for j from 1 to n do
    s2 := 0;
    s3 := 0;
    for k from 0 to i do
      if k=0 then w[k] := 0.5; end if;
      if k=i then w[k] := 0.5; end if;
      if k > 0 and k < i then w[k] := 1; end if;
      s2 := s2 + ω[k] · Ksv[i, k] · q[k, j];
      s3 := s3 + ω[k] · Ksv[i, k] · q[k, j-1];
    end do;
    q[i, j] := μ · s2 - μ · s3 + φ[i, j];
  end do;
end do;
for i from 0 to m do
  for j from 0 to n do
    #print("qtochnorm[" , i, j, "]=", qtochnorm[i, j], " q[" , i, j, "]=", q[i, j], " e=" , abs(qtochnorm[i, j] - q[i, j]));
  end do;
end do;
s := 0;
for i from 0 to m do
  for j from 0 to n do
    s := s + abs(qtochnorm[i, j] - q[i, j]);
  end do;
end do;
σ :=  $\frac{1}{(m+1) \cdot (n+1)}$  · s;
A := matrix(m-1, n):
for j from 1 to n do
  for i from 1 to m-1 do
    A[i, j] := qtochnorm[i, j];
  end do;
end do;
with(plots):
with(LinearAlgebra):
#matrixplot(A, style=patch, axes=boxed, labels=["x", "t", "qtoch"]);
B := matrix(m-1, n):
for j from 1 to n do
  for i from 1 to m-1 do

```

```

    B[i,j] := q[i,j];
end do;
end do;
#matrixplot(B, style = patch, axes = boxed, labels = ["x", "t", "q"]);
eps1 := 0.20;
slc := array(0..m, 0..n);
Sp := array(0..m, 0..n);
for i from 0 to m do
for j from 0 to n do
slc[i,j] := evalf(-1 + 2 *  $\frac{rand()}{1000000000000}$ , 2);
Sp[i,j] := snorm[i,j] * (1 + slc[i,j] * eps1);
end do;
end do;
#Защумление источника S(x,t)
#Блок моделирования
r1 :=  $\frac{\beta}{2 \cdot \theta} - \sqrt{\left(\frac{\beta^2}{4 \cdot \theta^2} + \frac{\alpha}{\theta}\right)}$ ; r2 :=  $\frac{\beta}{2 \cdot \theta} + \sqrt{\left(\frac{\beta^2}{4 \cdot \theta^2} + \frac{\alpha}{\theta}\right)}$ ;
 $\lambda := \frac{1}{\theta} \cdot \frac{1}{r2 - r1}$ ;  $\mu := \frac{\lambda \cdot \Delta x}{\Delta t}$ ;  $\eta := \lambda \cdot \xi \cdot \Delta x$ ;
 $\omega := array(0..m)$ ;  $\omega[0] := evalf\left(\frac{1}{2}\right)$ ;  $\omega[m] := evalf\left(\frac{1}{2}\right)$ ;
for i from 1 to m - 1 do
 $\omega[i] := 1$ ;
end do;
#Задание начальных условий
qpribl := array(0..m, 0..n);
for i from 0 to m do
qpribl[i, 0] := qtochnorm[i, 0];
end do;
#Задание краевых условий
for j from 0 to n do
qpribl[0, j] := qtochnorm[0, j];
qpribl[m, j] := qtochnorm[m, j];
end do;
Ksv := array(0..m, 0..m);
for i from 0 to m do
for k from 0 to i do
h1 :=  $e^{r2 \cdot (x[i] - x[k])}$ ; h2 :=  $e^{r1 \cdot (x[i] - x[k])}$ ;
Ksv[i, k] :=  $e^{r2 \cdot (x[i] - x[k])} \cdot e^{r1 \cdot (x[i] - x[k])}$ ;
#print("h1=", h1, " h2=", h2, "Ksv[" , i, k, "]=", Ksv[i, k]);
end do;
end do;
#print(Ksv);
 $\varphi := array(0..m, 1..n)$ ;
for i from 0 to m do
for j from 1 to n do
s1 := 0;
for k from 0 to i do
if k = 0 then w[k] := 0.5; end if;
if k = i then w[k] := 0.5; end if;
if k > 0 and k < i then w[k] := 1; end if;
s1 := s1 +  $\omega[k] \cdot Ksv[i, k] \cdot Sp[k, j]$ ;
end do;
 $\varphi[i, j] := qpribl[0, j] - \eta \cdot s1$ ;
end do;
end do;
for i from 1 to m - 1 do
for j from 1 to n do
s2 := 0;
s3 := 0;
for k from 0 to i do
if k = 0 then w[k] := 0.5; end if;
if k = i then w[k] := 0.5; end if;
if k > 0 and k < i then w[k] := 1; end if;
s2 := s2 +  $\omega[k] \cdot Ksv[i, k] \cdot qpribl[k, j]$ ;
s3 := s3 +  $\omega[k] \cdot Ksv[i, k] \cdot qpribl[k, j - 1]$ ;
end do;
qpribl[i, j] :=  $\mu \cdot s2 - \mu \cdot s3 + \varphi[i, j]$ ;
end do;
end do;
for i from 0 to m do
for j from 0 to n do
#print("qtochnorm[" , i, j, "]=", qtochnorm[i, j], "qpribl[" , i, j, "]=", qpribl[i, j], "e=", abs(qtochnorm[i, j] - qpribl[i, j]));
end do;
end do;
s := 0;
for i from 0 to m do
for j from 0 to n do

```



```

s := s + abs(qtochnorm[i,j] - qpribl[i,j]);
end do;
end do;

```

$$\sigma := \frac{1}{(m+1) \cdot (n+1)} \cdot s :$$

$$\mathcal{D} := \text{sqrt}(\text{sum}(\text{sum}(\text{Sp}[il,jl] - \text{snorm}[il,jl])^2, il=0..m), jl=0..n)) :$$

$$\hat{\alpha}_q := \text{sqrt}(\text{sum}(\text{sum}(\text{qpribl}[il,jl] - q[il,jl])^2, il=0..m), jl=0..n)) :$$

$$\mathcal{D}_{otn} := \text{sqrt}\left(\frac{\text{sum}(\text{sum}(\text{Sp}[il,jl] - \text{snorm}[il,jl])^2, il=0..m), jl=0..n)}{\text{sum}(\text{sum}(\text{snorm}[il,jl]^2, il=0..m), jl=0..n)}\right) :$$

$$\hat{\alpha}_{qotn} := \text{sqrt}\left(\frac{\text{sum}(\text{sum}(\text{qpribl}[il,jl] - q[il,jl])^2, il=0..m), jl=0..n)}{\text{sum}(\text{sum}(q[il,jl]^2, il=0..m), jl=0..n)}\right) :$$

$$\eta_l := \frac{\mathcal{D}}{\hat{\alpha}_q} : \eta_2 := \frac{\mathcal{D}_{otn}}{\hat{\alpha}_{qotn}} :$$

```

restart;
with(linalg):
#Вторая качественная модель - итерационная схема
#Задание интервалов для x и t, задание сетки
m1 := 10 : a1 := 0 : b1 := 1 :

$$\Delta x1 := \text{evalf}\left(\frac{b1 - a1}{m1}\right) : \lambda1 := \text{evalf}\left(\frac{1}{2}\right) : \Delta t1 := \text{evalf}\left((\Delta x1)^2 \cdot \lambda1\right) : nt := \text{evalf}\left(\frac{1}{\Delta t1}\right) :$$

n1 := 25 : x1 := array(0..m1) : t1 := array(0..n1) :
for kl from 0 to m1 do x1[kl] := kl· $\Delta x1$ ; end do:
for jl from 0 to n1 do t1[jl] := jl· $\Delta t1$ ; end do:
#Задание полей скорости ветра, коэф. турбулентной диффузии, концентрации загрязняющих веществ
V1 := 10 : K1 := 70 :
aql := 2 : bql := -0.1 : cql := 1 : dql := -1 : rql := 2.5 : gql := 0.1 :
ql := proc(u1, w1) (1 + aql·cos(bql·u1 + cql))·(1 + dql·cos(rql·w1 + gql)); end proc:
 $\alpha1 := 0.1 : \beta1 := 0.1 : \theta1 := 0.001 : T1 := 10 :$ 
Q1 := proc(u1, w1) diff(ql(u1, w1), w1) +  $\alpha1$ ·ql(u1, w1) +  $\beta1$ ·diff(V1·ql(u1, w1), u1) -  $\theta1$ ·diff(K1·diff(ql(u1, w1), u1), u1); end proc:
S01 := 0.0001 : q01 := 0.001 :
#`Задание массивов исходных данных
qtoch1 := array(0..m1, 0..n1) :
for kl from 0 to m1 do
for jl from 0 to n1 do
qtoch1[kl, jl] := ql(x1[kl], t1[jl]);
end do;
end do;
qmax1 := q01·max(seq(seq(qtoch1[kl, jl], jl=0..n1), kl=0..m1)) :
s1 := array(0..m1, 0..n1) :
for kl from 0 to m1 do
for jl from 0 to n1 do
s1[kl, jl] := evalf(subs(u1=x1[kl], w1=t1[jl], Q1(u1, w1)));
end do;
end do;
for kl from 0 by 2 to m1 do
for jl from 0 by 10 to n1 do
#print("s[" , kl, jl, "]=", s1[kl, jl]);
end do;
end do;
 $\xi1 := \max(\text{seq}(\text{seq}(s1[kl, jl], jl=0..n1), kl=0..m1)) :$ 
 $X1 := \frac{K1}{V1} \cdot \frac{\beta1}{\theta1} : T1 := \frac{K1}{V1^2} \cdot \frac{\beta1^2}{\theta1} : Smax1 := \frac{qmax1}{T1 \cdot \xi1} :$ 
#Метод последовательных приближений
 $\omega := \text{array}(0..n1) : \omega[0] := \text{evalf}\left(\frac{1}{2}\right) : \omega[n1] := \text{evalf}\left(\frac{1}{2}\right) :$ 
for il from 1 to n1 - 1 do
 $\omega[il] := 1;$ 
end do:
#Номер итерации v
v := 0 :
#Вычисление ядра Ksv и функции  $\phi$ 
Ksv1 := array(0..n1, 0..n1) :
 $\phi := \text{array}(0..m1, 0..n1) :$ 
qv := array(0..m1, 0..n1 + 1, 0..50) :
 $\psi := \text{array}(0..m1, 0..n1, 0..50) :$ 
qml := array(0..m1, 0..n1) :
for kl from 1 to m1 - 1 do
for jl from 1 to n1 do
s11 := 0 :
for il from 0 to jl do
Ksv1[il, jl] :=  $e^{(-\alpha1 \cdot (t1[jl] - t1[il]))}$ ;
p1 := qtoch1[kl, 0]· $e^{(-\alpha1 \cdot t1[jl])}$ ;
s11 := s11 +  $\omega[il]$ ·s1[kl, il]·Ksv1[il, jl]· $\Delta x1^2$ ;
end do:
 $\phi[kl, jl] := p1 + \lambda1 \cdot \xi1 \cdot s11;$ 
qV[kl, jl, v] :=  $\phi[kl, jl]$ ;
#print("k=", kl, "j=", jl, "v=", v1);
#print("qv=", qV[kl, jl, v]);
end do:
end do:
#Вычисление функции  $\psi$ 
ss := 100 :
 $\epsilon := 0.0001 :$ 
while ss >  $\epsilon$  do
v := v + 1 :
#`Задание начальных условий
for kl from 0 to m1 do
qV[kl, 0, v - 1] := qtoch1[kl, 0];
qml[kl, 0] := qtoch1[kl, 0];
end do:
#Задание краевых условий

```

```

for j1 from 0 to n1 do
qV[0,j1,v-1] := qtochl[0,j1];
qml[0,j1] := qtochl[0,j1];
qV[m1,j1,v-1] := qtochl[m1,j1];
qml[m1,j1] := qtochl[m1,j1];
end do:
for k1 from 1 to m1-1 do
for j1 from 1 to n1 do
s22 := 0:
for i1 from 0 to j1 do
p3 := qV[k1,i1,v-1] - qV[k1-1,i1,v-1];
p4 := qV[k1+1,i1,v-1] - 2·qV[k1,i1,v-1] + qV[k1-1,i1,v-1];
ψ[k1,i1,v-1] := β1·V1·Δx1·p3 - θ1·K1·p4;
s22 := s22 + ω[i1]·Ksv1[i1,j1]·ψ[k1,i1,v-1];
end do:
qml[k1,j1] := φ[k1,j1] - λ1·s22;
#print("k=",k1,"j=",j1,"qm[k,j]=",qml[k1,j1]);
end do:
end do:
for j1 from 1 to n1 do
for k1 from 1 to m1-1 do
qV[k1,j1,v] := qml[k1,j1];
#print("qV[" ,k1,j1,v, "]=",qV[k1,j1,v]);
end do:
end do:
s33 := 0:
for j1 from 1 to n1 do
for k1 from 1 to m1-1 do
s33 := s33 + (abs(qV[k1,j1,v] - qV[k1,j1,v-1]));
#print("qV[v-1]=",qV[k1,j1,v-1], "qV[v] =",qV[k1,j1,v], "otkl=",s33);
end do:
end do:
ss :=  $\frac{s33}{n1 \cdot (m1 - 1)}$ ;
#print("v=",v,"sredotkl=",ss);
end do:
s4 := 0: s5 := 0:
for k1 from 1 to m1-1 do
for j1 from 1 by 5 to n1 do
s4 := s4 + abs(qtochl[k1,j1] - qV[k1,j1,v]); s5 := s5 + abs(qtochl[k1,j1]);
end do:
end do:
s6 :=  $\frac{s4}{n1 \cdot (m1 - 1)}$  : s7 :=  $\frac{s4}{s5}$  :
#print("v=",v,"sredotkl=",s6,"sotn=",s7) :
A1 := matrix(m1-1,n1) :
for j1 from 1 to n1 do
for k1 from 1 to m1-1 do
A1[k1,j1] := qtochl[k1,j1];
end do:
end do:
with(plots) :
with(LinearAlgebra) :
#matrixplot(A1, style=patch, axes=boxed, labels=["x","t","qtoch"]);
B1 := matrix(m1-1,n1) :
for j1 from 1 to n1 do
for k1 from 1 to m1-1 do
B1[k1,j1] := qml[k1,j1];
end do:
end do:
#matrixplot(B1, style=patch, axes=boxed, labels=["x","t","q"]);

```

```

restart;
# Исходные данные для пространственной задачи переноса (тестовый пример)
X := 150 : Y := 150 : Z0 := 50 : Z := 100 : Tb := 60 :
V0Z := 7 : V0X := 5 : V0Y := 5 :
K0Z := 40 : K0X := 25 : K0Y := 25 :
alpha := 0.025 : q0 := 0.01 : S0 := 0.0005 :
# Вычисляемые параметры задачи
H := Z - Z0 :
beta := evalf( (V0Z*Tb) / H ) : theta := evalf( (K0Z*Tb) / H^2 ) : zeta := evalf( (Tb) / q0 ) :
fx := evalf( (V0X*Tb) / X ) : theta_x := evalf( (K0X*Tb) / X^2 ) : zeta_x := zeta :
fy := evalf( (V0Y*Tb) / Y ) : theta_y := evalf( (K0Y*Tb) / Y^2 ) : zeta_y := zeta :
#print(beta, theta, zeta);
# Размерность задачи
M := 5 : K := 5 : L := 5 : N := 7 :
# Массивы Xдан, Yдан, Zдан, Tдан
Xдан := array(0..M) : Yдан := array(0..K) : Zдан := array(0..L) : Tдан := array(0..N) :
DeltaXдан := evalf( X / M ) : DeltaYдан := evalf( Y / K ) : DeltaZдан := evalf( H / L ) : DeltaTдан := evalf( Tb / N ) :
for m from 0 to M do Xдан[m] := m * DeltaXдан; end do; #print(Xдан);
for k from 0 to K do Yдан[k] := k * DeltaYдан; end do; #print(Yдан);
for l from 0 to L do Zдан[l] := l * DeltaZдан; end do; #print(Zдан);
for j from 0 to N do Tдан[j] := j * DeltaTдан; end do; #print(Tдан);
# Нормирование пространственных и временной переменных задачи
x := array(0..M) : y := array(0..K) : z := array(0..L) : t := array(0..N) :
for m from 0 to M do x[m] := Xдан[m] / X : end do; #print(x);
for k from 0 to K do y[k] := Yдан[k] / Y : end do; #print(y);
for l from 0 to L do z[l] := Zдан[l] / H : end do; #print(z);
for j from 0 to N do t[j] := Tдан[j] / Tb : end do; #print(t);
DeltaX := evalf( 1 / M ) : DeltaY := evalf( 1 / K ) : DeltaZ := evalf( 1 / L ) : DeltaT := evalf( 1 / N ) :
# Формирование и нормирование массивов исходных данных задачи
v1 := array(0..M, 0..K, 0..L, 0..N) : vnorm1 := array(0..M, 0..K, 0..L, 0..N) :
kf1 := array(0..M, 0..K, 0..L, 0..N) : kfnorm1 := array(0..M, 0..K, 0..L, 0..N) :
qtoch1 := array(0..M, 0..K, 0..L, 0..N) : qtochnorm1 := array(0..M, 0..K, 0..L, 0..N) :
s1 := array(0..M, 0..K, 0..L, 0..N) : snorm1 := array(0..M, 0..K, 0..L, 0..N) :
v2 := array(0..M, 0..K, 0..L, 0..N) : vnorm2 := array(0..M, 0..K, 0..L, 0..N) :
kf2 := array(0..M, 0..K, 0..L, 0..N) : kfnorm2 := array(0..M, 0..K, 0..L, 0..N) :
s2 := array(0..M, 0..K, 0..L, 0..N) : snorm2 := array(0..M, 0..K, 0..L, 0..N) :
v3 := array(0..M, 0..K, 0..L, 0..N) : vnorm3 := array(0..M, 0..K, 0..L, 0..N) :
kf3 := array(0..M, 0..K, 0..L, 0..N) : kfnorm3 := array(0..M, 0..K, 0..L, 0..N) :
s3 := array(0..M, 0..K, 0..L, 0..N) : snorm3 := array(0..M, 0..K, 0..L, 0..N) :
for m from 0 to M do
for k from 0 to K do
for l from 0 to L do
for j from 0 to N do
# формирование массивов по оси Oz
v1n := (1 + 0.25 * sin(2 * x[m])) * (1 + 0.25 * sin(5 * y[k])) * (1 + 0.25 * cos(1.5 * z[l])) * (1 + 0.25 * cos(3 * t[j])) :
v1[m, k, l, j] := V0Z * v1n :
qn := (1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(z[l])) * (1 + sin(t[j])) :
qtoch1[m, k, l, j] := q0 * qn :
k1n := (1 + 0.25 * sin(5 * x[m])) * (1 + 0.25 * sin(1 * y[k])) * (1 + 0.25 * sin(3 * z[l])) * (1 + 0.25 * cos(2 * t[j])) :
kf1[m, k, l, j] := K0Z * k1n :
dqdt := ((1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(z[l]))) * cos(t[j]) :
dqz := ((1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(t[j]))) * cos(z[l]) :
dk1dz := (1 + 0.25 * sin(5 * x[m])) * (1 + 0.25 * sin(1 * y[k])) * (1 + 0.25 * cos(3 * z[l])) * 0.75 * cos(3 * z[l]) :
d2qdz := (1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(t[j])) * (-sin(z[l])) :
s1[m, k, l, j] := q0 / Tb * ( dqdt + alpha * Tb * qn + beta * v1n * dqz - theta * (dk1dz * dqz + k1n * d2qdz) ) :
# формирование массивов по оси Ox
v2n := (1 + 0.25 * sin(-2 * x[m])) * (1 + 0.25 * sin(5 * y[k])) * (1 + 0.25 * cos(-1.5 * z[l])) * (1 + 0.25 * cos(3 * t[j])) :
v2[m, k, l, j] := V0X * v2n :
k2n := (1 + 0.25 * sin(5 * x[m])) * (1 + 0.25 * sin(-1 * y[k])) * (1 + 0.25 * sin(3 * z[l])) * (1 + 0.25 * cos(-2 * t[j])) :
kf2[m, k, l, j] := K0X * k2n :
dqx := (cos(x[m])) * (1 + sin(y[k])) * (1 + sin(z[l])) * (1 + sin(t[j])) :
dk2dx := (1.25 * cos(5 * x[m])) * (1 + 0.25 * sin(-1 * y[k])) * (1 + 0.25 * sin(3 * z[l])) * (1 + 0.25 * cos(-2 * t[j])) :
d2qdx := (1 + sin(x[m])) * (-sin(y[k])) * (1 + sin(z[l])) * (1 + sin(t[j])) :
s2[m, k, l, j] := q0 / Tb * ( dqdt + beta * v2n * dqx - theta * (dk2dx * dqx + k2n * d2qdx) ) :
# формирование массивов по оси Oy
v3n := (1 + 0.25 * sin(-1 * x[m])) * (1 + 0.25 * sin(0.5 * y[k])) * (1 + 0.25 * cos(-2 * z[l])) * (1 + 0.25 * cos(1.5 * t[j])) :
v3[m, k, l, j] := V0Y * v3n :
k3n := (1 + 0.25 * sin(2.5 * x[m])) * (1 + 0.25 * sin(-0.1 * y[k])) * (1 + 0.25 * sin(2 * z[l])) * (1 + 0.25 * cos(-2 * t[j])) :

```

```

kf3[m, k, l, j] := KOY·k3n;
dqy := (1 + sin(x[m]))·(1 + sin(y[k]))·(cos(z[l]))·(1 + sin(t[j])) :
dk3dy := (1 + 0.25·sin(2.5·x[m]))·(1 + 0.25·sin(-0.1·y[k]))·(0.5·cos(2·z[l]))·(1 + 0.25·cos(-2·t[j])) :
d2qdy := (1 + sin(x[m]))·(1 + sin(y[k]))·(-sin(z[l]))·(1 + sin(t[j])) :
s3[m, k, l, j] :=  $\frac{q0}{Tb}$  · (dqdt +  $\beta y$ ·v3n·dqy -  $\Theta y$ ·(dk3dy·dqy + k3n·d2qdy)) :
end do;
end do;
end do;
end do;
Vmax1 := max(seq(seq(seq(seq(v1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Vmin1 := min(seq(seq(seq(seq(v1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Vmin1=", Vmin1, "Vmax1=", Vmax1);
Kmax1 := max(seq(seq(seq(seq(kf1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Kmin1 := min(seq(seq(seq(seq(kf1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Kmin1=", Kmin1, "Kmax1=", Kmax1);
qmax1 := max(seq(seq(seq(seq(qtoch1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
qmin1 := min(seq(seq(seq(seq(qtoch1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("qmin1=", qmin1, "qmax1=", qmax1);
Smax1 := max(seq(seq(seq(seq(s1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Smin1 := min(seq(seq(seq(seq(s1[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Smin1=", Smin1, "Smax1=", Smax1);
Vmax2 := max(seq(seq(seq(seq(v2[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Vmin2 := min(seq(seq(seq(seq(v2[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Vmin2=", Vmin2, "Vmax2=", Vmax2);
Kmax2 := max(seq(seq(seq(seq(kf2[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Kmin2 := min(seq(seq(seq(seq(kf2[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Kmin2=", Kmin2, "Kmax2=", Kmax2);
Smax2 := max(seq(seq(seq(seq(s2[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Smin2 := min(seq(seq(seq(seq(s2[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Smin2=", Smin2, "Smax2=", Smax2);
Vmax3 := max(seq(seq(seq(seq(v3[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Vmin3 := min(seq(seq(seq(seq(v3[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Vmin3=", Vmin3, "Vmax3=", Vmax3);
Kmax3 := max(seq(seq(seq(seq(kf3[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Kmin3 := min(seq(seq(seq(seq(kf3[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Kmin3=", Kmin3, "Kmax3=", Kmax3);
Smax3 := max(seq(seq(seq(seq(s3[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
Smin3 := min(seq(seq(seq(seq(s3[m, k, l, j], j=0 ..N), l=0 ..L), k=0 ..K), m=0 ..M)) :
#print("Smin3=", Smin3, "Smax3=", Smax3);
for m from 0 to M do
for k from 0 to K do
for l from 0 to L do
for j from 0 to N do
vnorm1[m, k, l, j] :=  $\frac{v1[m, k, l, j]}{Vmax1}$  : kfnorm1[m, k, l, j] :=  $\frac{kf1[m, k, l, j]}{Kmax1}$  :
qtochnorm1[m, k, l, j] :=  $\frac{qtoch1[m, k, l, j]}{qmax1}$  : snorm1[m, k, l, j] :=  $\frac{s1[m, k, l, j]}{Smax1}$  :
vnorm2[m, k, l, j] :=  $\frac{v2[m, k, l, j]}{Vmax2}$  : kfnorm2[m, k, l, j] :=  $\frac{kf2[m, k, l, j]}{Kmax2}$  : snorm2[m, k, l, j] :=  $\frac{s2[m, k, l, j]}{Smax2}$  :
vnorm3[m, k, l, j] :=  $\frac{v3[m, k, l, j]}{Vmax3}$  : kfnorm3[m, k, l, j] :=  $\frac{kf3[m, k, l, j]}{Kmax3}$  : snorm3[m, k, l, j] :=  $\frac{s3[m, k, l, j]}{Smax3}$  :
end do;
end do;
end do;
end do;
#конец тестового примера

```

```

restart;
# Трёхмерное уравнение переноса - конечно-разностная схема
V0Z := 5 : K0Z := 40 : q0 := 0.01 : Tb := 60 :
α := 0.02 : βx := 1.75 : θx := 0.03 :
V0X := 6 : K0X := 35 : βy := 1.75 : θy := 0.03 :
V0Y := 7 : K0Y := 45 : βz := 1.75 : θz := 0.03 :
m := 5 : n := 10 : a := 0 : b := 1 : c := 0 : d := 1 :
Δx := evalf( (b-a)/m ) : Δy := evalf( (b-a)/m ) : Δz := evalf( (b-a)/m ) :
λ1 := 0.5 : λ2 := 0.5 : λ3 := 0.5 : Δ := evalf( λ1 · Δx² ) :
x := array(0..m) : y := array(0..m) : z := array(0..m) : t := array(0..n) :
for i from 0 to m do x[i] := i · Δx; end do;
for k from 0 to m do y[k] := k · Δy; end do;
for l from 0 to m do z[l] := l · Δz; end do;
for j from 0 to n do t[j] := j · Δt; end do;
# Коэффициенты для функций исходных данных
av1 := 2 : bv1 := 5 : cv1 := 1.5 : dv1 := 3 : ak1 := 5 : bk1 := 1 : ck1 := 3 : dk1 := 2 :
av2 := -2 : bv2 := 5 : cv2 := -1.5 : dv2 := 3 : ak2 := 5 : bk2 := -1 : ck2 := 3 : dk2 := -2 :
av3 := -1 : bv3 := 0.5 : cv3 := -2 : dv3 := 1.5 : ak3 := 2.5 : bk3 := -0.1 : ck3 := 2 : dk3 := -2 :
# Задание полей скорости ветра, турбулентной диффузии, концентрации загрязняющих веществ
V1 := proc(u, v, w, r) V0Z · (1 + 0.25 · sin(av1 · u)) · (1 + 0.25 · sin(bv1 · v)) · (1 + 0.25 · cos(cv1 · w)) · (1 + 0.25 · cos(dv1 · r)); end proc;
K1 := proc(u, v, w, r) K0Z · (1 + 0.25 · sin(ak1 · u)) · (1 + 0.25 · sin(bk1 · v)) · (1 + 0.25 · sin(ck1 · w)) · (1 + 0.25 · cos(dk1 · r)); end proc;
q1 := proc(u, v, w, r) q0 · (1 + sin(u)) · (1 + sin(v)) · (1 + sin(w)) · (1 + sin(r)); end proc;
Δ := proc(u, v, w, r) diff(q1(u, v, w, r), r) + α · Tb · q1(u, v, w, r) + βx · V1(u, v, w, r) · diff(q1(u, v, w, r), w) - θx · diff(K1(u, v, w, r) · diff(q1(u, v, w, r), w), w);
end proc;
V2 := proc(u, v, w, r) V0X · (1 + 0.25 · sin(av2 · u)) · (1 + 0.25 · sin(bv2 · v)) · (1 + 0.25 · cos(cv2 · w)) · (1 + 0.25 · cos(dv2 · r)); end proc;
K2 := proc(u, v, w, r) K0X · (1 + 0.25 · sin(ak2 · u)) · (1 + 0.25 · sin(bk2 · v)) · (1 + 0.25 · sin(ck2 · w)) · (1 + 0.25 · cos(dk2 · r)); end proc;
Q2 := proc(u, v, w, r) diff(q1(u, v, w, r), r) + βx · V2(u, v, w, r) · diff(q1(u, v, w, r), w) - θx · diff(K2(u, v, w, r) · diff(q1(u, v, w, r), w), w); end proc;
V3 := proc(u, v, w, r) V0Y · (1 + 0.25 · sin(av3 · u)) · (1 + 0.25 · sin(bv3 · v)) · (1 + 0.25 · cos(cv3 · w)) · (1 + 0.25 · cos(dv3 · r)); end proc;
K3 := proc(u, v, w, r) K0Y · (1 + 0.25 · sin(ak3 · u)) · (1 + 0.25 · sin(bk3 · v)) · (1 + 0.25 · sin(ck3 · w)) · (1 + 0.25 · cos(dk3 · r)); end proc;
Q3 := proc(u, v, w, r) diff(q1(u, v, w, r), r) + βy · V3(u, v, w, r) · diff(q1(u, v, w, r), w) - θy · diff(K3(u, v, w, r) · diff(q1(u, v, w, r), w), w); end proc;
v1 := array(0..m, 0..m, 0..m, 0..n) : vnorm1 := array(0..m, 0..m, 0..m, 0..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
v1[i, k, l, j] := V1(x[i], y[k], z[l], t[j]);
end do;
end do;
end do;
end do;
Vmax1 := max(seq(seq(seq(seq(v1[i, k, l, j], j = 0..n), l = 0..m), k = 0..m), i = 0..m)) :
Vmin1 := min(seq(seq(seq(seq(v1[i, k, l, j], j = 0..n), l = 0..m), k = 0..m), i = 0..m)) :
#print("Vmin1=", Vmin1, "Vmax1=", Vmax1);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
vnorm1[i, k, l, j] := v1[i, k, l, j] / Vmax1;
end do;
end do;
end do;
end do;
kfl := array(0..m, 0..m, 0..m, 0..n) : kfnorm1 := array(0..m, 0..m, 0..m, 0..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
kfl[i, k, l, j] := K1(x[i], y[k], z[l], t[j]);
end do;
end do;
end do;
end do;
Kmax1 := max(seq(seq(seq(seq(kfl[i, k, l, j], j = 0..n), l = 0..m), k = 0..m), i = 0..m)) :
Kmin1 := min(seq(seq(seq(seq(kfl[i, k, l, j], j = 0..n), l = 0..m), k = 0..m), i = 0..m)) :
#print("Kmin1=", Kmin1, "Kmax1=", Kmax1);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
kfnorm1[i, k, l, j] := kfl[i, k, l, j] / Kmax1;
end do;
end do;
end do;
end do;
qtoch1 := array(0..m, 0..m, 0..m, 0..n) : qtochnorm1 := array(0..m, 0..m, 0..m, 0..n) :
for i from 0 to m do
for k from 0 to m do

```

```

for l from 0 to m do
for j from 0 to n do
  qtochI[i, k, l, j] := ql(x[i], y[k], z[l], t[j]);
end do;
end do;
end do;
end do;
qmaxI := max(seq(seq(seq(seq(qtochI[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
qminI := min(seq(seq(seq(seq(qtochI[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
#print("qminI=", qminI, "qmaxI=", qmaxI);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  qtochnormI[i, k, l, j] :=  $\frac{qtochI[i, k, l, j]}{qmaxI}$ ;
end do;
end do;
end do;
end do;
qmaxIn := max(seq(seq(seq(seq(qtochnormI[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
qminIn := min(seq(seq(seq(seq(qtochnormI[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
sI := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) : snormI := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  sI[i, k, l, j] := evalf(subs(u=x[i], v=y[k], w=z[l], r=t[j], QI(u, v, w, r)));
  sI[i, k, l, j] :=  $\frac{sI[i, k, l, j] \cdot qmaxI}{Tb}$ ;
end do;
end do;
end do;
end do;
SmaxI := max(seq(seq(seq(seq(sI[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
SminI := min(seq(seq(seq(seq(sI[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
#print("SminI=", SminI, "SmaxI=", SmaxI);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  snormI[i, k, l, j] :=  $\frac{sI[i, k, l, j]}{SmaxI}$ ;
end do;
end do;
end do;
end do;
T1 :=  $\frac{KmaxI \cdot \beta^2}{VmaxI^2 \cdot \epsilon}$  : X1 :=  $\frac{VmaxI \cdot Tb}{\beta}$  : E1 :=  $\frac{SmaxI \cdot Tb}{qmaxI}$  :
v2 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) : vnorm2 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  v2[i, k, l, j] := V2(x[i], y[k], z[l], t[j]);
end do;
end do;
end do;
end do;
Vmax2 := max(seq(seq(seq(seq(v2[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
Vmin2 := min(seq(seq(seq(seq(v2[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
#print("Vmin2=", Vmin2, "Vmax2=", Vmax2);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  vnorm2[i, k, l, j] :=  $\frac{v2[i, k, l, j]}{Vmax2}$ ;
end do;
end do;
end do;
end do;
kf2 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) : kfnorm2 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  kf2[i, k, l, j] := K2(x[i], y[k], z[l], t[j]);
end do;
end do;
end do;
end do;
Kmax2 := max(seq(seq(seq(seq(kf2[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :

```

```

Kmin2 := min(seq(seq(seq(seq(kf2[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
#print("Kmin2=", Kmin2, "Kmax2=", Kmax2);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  kfnorm2[i, k, l, j] :=  $\frac{kf2[i, k, l, j]}{Kmax2}$ ;
end do;
end do;
end do;
end do;
s2 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) : snorm2 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  s2[i, k, l, j] := evalf(subs(u=x[i], v=y[k], w=z[l], r=t[j], Q2(u, v, w, r)));
  s2[i, k, l, j] :=  $\frac{s2[i, k, l, j] \cdot qmax1}{Tb}$ ;
end do;
end do;
end do;
end do;
Smax2 := max(seq(seq(seq(seq(s2[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
Smin2 := min(seq(seq(seq(seq(s2[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
#print("Smin2=", Smin2, "Smax2=", Smax2);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  snorm2[i, k, l, j] :=  $\frac{s2[i, k, l, j]}{Smax2}$ ;
end do;
end do;
end do;
end do;
T2 :=  $\frac{Kmax2 \cdot \beta^2}{Vmax2^2 \cdot \alpha}$  : X2 :=  $\frac{Vmax2 \cdot Tb}{\beta}$  : E :=  $\frac{Smax2 \cdot Tb}{qmax1}$  :
v3 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) : vnorm3 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  v3[i, k, l, j] := V3(x[i], y[k], z[l], t[j]);
end do;
end do;
end do;
end do;
Vmax3 := max(seq(seq(seq(seq(v3[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
Vmin3 := min(seq(seq(seq(seq(v3[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
#print("Vmin3=", Vmin3, "Vmax3=", Vmax3);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  vnorm3[i, k, l, j] :=  $\frac{v3[i, k, l, j]}{Vmax3}$ ;
end do;
end do;
end do;
end do;
kf3 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) : kfnorm3 := array(0 ..m, 0 ..m, 0 ..m, 0 ..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  kf3[i, k, l, j] := K3(x[i], y[k], z[l], t[j]);
end do;
end do;
end do;
end do;
Kmax3 := max(seq(seq(seq(seq(kf3[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
Kmin3 := min(seq(seq(seq(seq(kf3[i, k, l, j], j=0 ..n), l=0 ..m), k=0 ..m), i=0 ..m)) :
#print("Kmin3=", Kmin3, "Kmax3=", Kmax3);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
  kfnorm3[i, k, l, j] :=  $\frac{kf3[i, k, l, j]}{Kmax3}$ ;
end do;
end do;
end do;
end do;

```



```

end do;
end do;
end do;
s3 := array(0..m, 0..m, 0..m, 0..n) : snorm3 := array(0..m, 0..m, 0..m, 0..n) :
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
s3[i, k, l, j] := evalf(subs(u=x[i], v=y[k], w=z[l], r=t[j], Q3(u, v, w, r)));
s3[i, k, l, j] :=  $\frac{s3[i, k, l, j] \cdot qmaxl}{Tb}$ ;
end do;
end do;
end do;
end do;
Smax3 := max(seq(seq(seq(seq(s3[i, k, l, j], j=0..n), l=0..m), k=0..m), i=0..m)) :
Smin3 := min(seq(seq(seq(seq(s3[i, k, l, j], j=0..n), l=0..m), k=0..m), i=0..m)) :
#print("Smin3=", Smin3, "Smax3=", Smax3);
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
for j from 0 to n do
snorm3[i, k, l, j] :=  $\frac{s3[i, k, l, j]}{Smax3}$ ;
end do;
end do;
end do;
end do;
T3 :=  $\frac{Kmax3 \cdot \beta^2}{Vmax3^2 \cdot \theta}$  : X3 :=  $\frac{Vmax3 \cdot Tb}{\beta}$  : Z :=  $\frac{Smax3 \cdot Tb}{qmaxl}$  :
with(linalg) :
q1 := array(0..m, 0..m, 0..m, 0..n) : q2 := array(0..m, 0..m, 0..m, 0..n) :
q3 := array(0..m, 0..m, 0..m, 0..n) : qmod := array(0..m, 0..m, 0..m, 0..n) :
Matrsist1 := array(1..m-1, 1..m-1) : Matrsist2 := array(1..m-1, 1..m-1) :
Matrsist3 := array(1..m-1, 1..m-1) : fsist1 := array(1..m-1) :
fsist2 := array(1..m-1) : fsist3 := array(1..m-1) :
# Задание начальных условий
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
q1[i, k, l, 0] := qtochnorm1[i, k, l, 0];
#print("q1[" , i, k, l, "0]=", q1[i, k, l, 0]);
end do;
end do;
end do;
ap1 := array(1..m-1) : bp1 := array(1..m-1) : ytek1 := array(1..m-1) :
ap2 := array(1..m-1) : bp2 := array(1..m-1) : ytek2 := array(1..m-1) :
ap3 := array(1..m-1) : bp3 := array(1..m-1) : ytek3 := array(1..m-1) :
for j from 0 to n-1 do
# Определение коэффициентов  $a_2, b_2, A1, B1, C1, D1, \lambda$ 
for gl from 1 to m-1 do
for g2 from 1 to m-1 do
Matrsist1[gl, g2] := 0;
end do;
end do;
# Первая подзадача
for i from 0 to m do
for k from 0 to m do
for l from 0 to m do
if j=0 then
q1[i, k, l, 0] := qtochnorm1[i, k, l, 0];
else
q1[i, k, l, j] := q3[i, k, l, j] :
end if;
#print("q1[" , i, k, l, "0]=", q1[i, k, l, 0]);
end do;
end do;
end do;
for i from 1 to m-1 do
for k from 1 to m-1 do
q1[i, k, 0, j+1] := qtochnorm1[i, k, 0, j+1] :
q1[i, k, m, j+1] := qtochnorm1[i, k, m, j+1] :
for l from 1 to m-1 do
az :=  $\alpha \cdot Tb$  :
bz :=  $\beta \cdot vnorm1[i, k, l, j+1] - \frac{\theta}{\Delta} (kfnorm1[i, k, l, j+1] - kfnorm1[i, k, l-1, j+1])$  :
cz :=  $\theta \cdot kfnorm1[i, k, l, j+1]$ ; A1 :=  $bz \cdot \Delta + cz$ ; B1 :=  $az \cdot \Delta^2 + bz \cdot \Delta + 2 \cdot cz$ ;
C1 :=  $cz$ ; D1 :=  $\frac{Tb}{qmaxl} \cdot Smax1 \cdot snorm1[i, k, l, j+1] \cdot \Delta$ ;
if l=1 then
fsist1[l] :=  $-(D1 + q1[i, k, l, j]) - \lambda \cdot A1 \cdot q1[i, k, 0, j+1]$  :

```

```

Matrsist1[l, l] := -(1 + λ1·B1) : Matrsist1[l, l+1] := λ1·C1 :
apl[l] := -  $\frac{\text{Matrsist1}[l, l+1]}{\text{Matrsist1}[l, l]}$  : bpl[l] :=  $\frac{\text{fsist1}[l]}{\text{Matrsist1}[l, l]}$  :
end if
if l ≥ 2 and l ≤ m-2 then
Matrsist1[l, l-1] := λ1·A1 : Matrsist1[l, l] := -(1 + λ1·B1) :
Matrsist1[l, l+1] := λ1·C1 : fsist1[l] := -(D1 + q1[i, k, l, j]) :
apl[l] := -  $\frac{\text{Matrsist1}[l, l+1]}{\text{Matrsist1}[l, l-1] \cdot \text{apl}[l-1] + \text{Matrsist1}[l, l]}$  :
bpl[l] :=  $\frac{(\text{fsist1}[l] - \text{Matrsist1}[l, l-1] \cdot \text{bpl}[l-1])}{\text{Matrsist1}[l, l-1] \cdot \text{apl}[l-1] + \text{Matrsist1}[l, l]}$  :
end if
if l = m-1 then
fsist1[l] := -(D1 + q1[i, k, l, j]) - λ1·C1·q1[i, k, m, j+1] :
Matrsist1[l, l] := -(1 + λ1·B1) : Matrsist1[l, l-1] := λ1·A1 :
ytek1[l] :=  $\frac{(\text{fsist1}[l] - \text{Matrsist1}[l, l-1] \cdot \text{bpl}[l-1])}{\text{Matrsist1}[l, l-1] \cdot \text{apl}[l-1] + \text{Matrsist1}[l, l]}$  :
end if
end do:
for l from m-1 by -1 to 2 do
ytek1[l-1] := apl[l-1]·ytek1[l] + bpl[l-1] :
end do:
for l from 1 to m-1 do
q1[i, k, l, j+1] := ytek1[l] :
#print("q1[" , i, k, l, j+1, "]=", q1[i, k, l, j+1]);
end do:
end do:
end do:
# Вторая подзадача
# Условие перехода
for i from 1 to m-1 do
for k from 1 to m-1 do
for l from 1 to m-1 do
q2[i, k, l, j] := q1[i, k, l, j+1] :
end do:
end do:
end do:
for k from 1 to m-1 do
for l from 1 to m-1 do
q2[0, k, l, j+1] := qtochnorm1[0, k, l, j+1] :
q2[m, k, l, j+1] := qtochnorm1[m, k, l, j+1] :
end do:
end do:
# Определение коэффициентов b_x, c_x, A2, B2, C2, D2, λ2
for g1 from 1 to m-1 do
for g2 from 1 to m-1 do
Matrsist2[g1, g2] := 0;
end do:
end do:
for k from 1 to m-1 do
for l from 1 to m-1 do
for i from 1 to m-1 do
bx := βx·vnorm2[i, k, l, j+1] -  $\frac{\theta x}{\Delta x} (kfnorm2[i, k, l, j+1] - kfnorm2[i-1, k, l, j+1])$  :
cx := θx·kfnorm2[i, k, l, j+1]; A2 := bx·Δx + cx; B2 := bx·Δx + 2·cx;
C2 := cx; D2 :=  $\frac{Tb}{qmax1} \cdot Smax2 \cdot snorm2[i, k, l, j+1] \cdot \Delta x$ ;
if i = 1 then
fsist2[i] := -(D2 + q2[i, k, l, j]) - λ2·A2·q2[0, k, l, j+1] :
Matrsist2[i, i] := -(1 + λ2·B2) : Matrsist2[i, i+1] := λ2·C2 :
ap2[i] := -  $\frac{\text{Matrsist2}[i, i+1]}{\text{Matrsist2}[i, i]}$  : bp2[i] :=  $\frac{\text{fsist2}[i]}{\text{Matrsist2}[i, i]}$  :
end if
if i ≥ 2 and i ≤ m-2 then
Matrsist2[i, i-1] := λ2·A2 : Matrsist2[i, i] := -(1 + λ2·B2) :
Matrsist2[i, i+1] := λ2·C2 : fsist2[i] := -(D2 + q2[i, k, l, j]) :
ap2[i] := -  $\frac{\text{Matrsist2}[i, i+1]}{\text{Matrsist2}[i, i-1] \cdot \text{ap2}[i-1] + \text{Matrsist2}[i, i]}$  :
bp2[i] :=  $\frac{(\text{fsist2}[i] - \text{Matrsist2}[i, i-1] \cdot \text{bp2}[i-1])}{\text{Matrsist2}[i, i-1] \cdot \text{ap2}[i-1] + \text{Matrsist2}[i, i]}$  :
end if
if i = m-1 then
fsist2[i] := -(D2 + q2[i, k, l, j]) - λ2·C2·q2[m, k, l, j+1] :
Matrsist2[i, i] := -(1 + λ2·B2) : Matrsist2[i, i-1] := λ2·A2 :
ytek2[i] :=  $\frac{(\text{fsist2}[i] - \text{Matrsist2}[i, i-1] \cdot \text{bp2}[i-1])}{\text{Matrsist2}[i, i-1] \cdot \text{ap2}[i-1] + \text{Matrsist2}[i, i]}$  :
end if

```

```

end do;
for i from m-1 by -1 to 2 do
ytek2[i-1] := ap2[i-1]·ytek2[i] + bp2[i-1];
end do;
for i from 1 to m-1 do
q2[i, k, l, j+1] := ytek2[i];
#print("q2[" , i, k, l, j+1, "]=", q2[i, k, l, j+1]);
end do;
end do;
end do;
# Третья подзадача
# Условие перехода
for i from 1 to m-1 do
for k from 1 to m-1 do
for l from 1 to m-1 do
q3[i, k, l, j] := q2[i, k, l, j+1];
end do;
end do;
end do;
for i from 1 to m-1 do
for l from 1 to m-1 do
q3[i, 0, l, j+1] := qtochnorml[i, 0, l, j+1];
q3[i, m, l, j+1] := qtochnorml[i, m, l, j+1];
end do;
end do;
# Определение коэффициентов by, cy, A3, B3, C3, D3, λ3
for g1 from 1 to m-1 do
for g2 from 1 to m-1 do
Matrsist3[g1, g2] := 0;
end do;
end do;
for i from 1 to m-1 do
for l from 1 to m-1 do
for k from 1 to m-1 do
byk := β·vnorm3[i, k, l, j+1] -  $\frac{\Theta_y}{\Delta_y}$  (kfnorm3[i, k, l, j+1] - kfnorm2[i, k-1, l, j+1]);
cy := Θy·kfnorm3[i, k, l, j+1]; A3 := byk·Δy + cy; B3 := byk·Δy + 2·cy;
C3 := cy; D3 :=  $\frac{Tb}{qmaxl}$  ·Smax3·snorm3[i, k, l, j+1]·Δx;
if k=1 then
fsist3[k] := -(D3 + q3[i, k, l, j]) - λ3·A3·q3[i, 0, l, j+1];
Matrsist3[k, k] := -(1 + λ3·B3); Matrsist3[k, k+1] := λ3·C3;
ap3[k] := - $\frac{Matrsist3[k, k+1]}{Matrsist3[k, k]}$ ; bp3[k] :=  $\frac{fsist3[k]}{Matrsist3[k, k]}$ ;
end if;
if k ≥ 2 and k ≤ m-2 then
Matrsist3[k, k-1] := λ3·A3; Matrsist3[k, k] := -(1 + λ3·B3);
Matrsist3[k, k+1] := λ3·C3; fsist3[k] := -(D3 + q3[i, k, l, j]);
ap3[k] := - $\frac{Matrsist3[k, k+1]}{Matrsist3[k, k-1]·ap3[k-1] + Matrsist3[k, k]}$ ;
bp3[k] :=  $\frac{(fsist3[k] - Matrsist3[k, k-1]·bp3[k-1])}{Matrsist3[k, k-1]·ap3[k-1] + Matrsist3[k, k]}$ ;
end if;
if k=m-1 then
fsist3[k] := -(D3 + q3[i, k, l, j]) - λ3·C3·q3[i, m, l, j+1];
Matrsist3[k, k] := -(1 + λ3·B3); Matrsist3[k, k-1] := λ3·A3;
ytek3[k] :=  $\frac{(fsist3[k] - Matrsist3[k, k-1]·bp3[k-1])}{Matrsist3[k, k-1]·ap3[k-1] + Matrsist3[k, k]}$ ;
end if;
end do;
for k from m-1 by -1 to 2 do
ytek3[k-1] := ap3[k-1]·ytek3[k] + bp3[k-1];
end do;
for k from 1 to m-1 do
q3[i, k, l, j+1] := ytek3[k];
#print("q3[" , i, k, l, j+1, "]=", q3[i, k, l, j+1]);
end do;
end do;
end do;
ssm := 0; ssm1 := 0;
for i from 1 to m-1 do
for k from 1 to m-1 do
for l from 1 to m-1 do
qmod[i, k, l, j+1] := q3[i, k, l, j+1];
ssm := ssm + (qmod[i, k, l, j+1] - qtochnorml[i, k, l, j+1])2;
ssm1 := ssm1 + (qtochnorml[i, k, l, j+1])2;
if j+1=10 then
#print("qtochnorml[" , i, k, l, j+1, "]=", qtochnorml[i, k, l, j+1], "qmod[" , i, k, l, j+1, "]=", qmod[i, k, l, j+1]);
#print("qtochnorml[" , i, k, l, j+1, "]=", qmod[i, k, l, j+1]);
end if;
end do;
end do;
end do;
sigotn := sqrt( $\frac{ssm}{ssm1}$ );
#print("j=", j, "sigotn=", sigotn);
end do;

```

```

restart;
#Метод интегральных уравнений в вычислительной схеме по координатного расщепления уравнения переноса
# Исходные данные (тестовый пример)
X := 150 : Y := 150 : ZO := 50 : Z := 100 : Tb := 60 :
VOZ := 7 : VOX := 5 : VOY := 5 :
K0Z := 40 : K0X := 25 : K0Y := 25 :
alpha := 0.025 : q0 := 0.01 : S0 := 0.0005 :
#Вычисляемые параметры задачи
H := Z - ZO :
fx := evalf( (VOZ*Tb) / H ) : theta := evalf( (K0Z*Tb) / H^2 ) : zeta := evalf( (Tb) / q0 ) :
fy := evalf( (VOX*Tb) / X ) : theta := evalf( (K0X*Tb) / X^2 ) : zeta := zeta :
fy := evalf( (VOY*Tb) / Y ) : theta := evalf( (K0Y*Tb) / Y^2 ) : zeta := zeta :
#print(fx, theta, zeta);
#Размерность задачи
M := 5 : K := 5 : L := 5 : N := 7 :
# Массивы Xdan, Ydan, Zdan, Tdan
Xdan := array(0..M) : Ydan := array(0..K) : Zdan := array(0..L) : Tdan := array(0..N) :
deltaXdan := evalf( X / M ) : deltaYdan := evalf( Y / K ) : deltaZdan := evalf( H / L ) : deltaTdan := evalf( Tb / N ) :
for m from 0 to M do Xdan[m] := m * deltaXdan, end do : #print(Xdan) :
for k from 0 to K do Ydan[k] := k * deltaYdan, end do : #print(Ydan) :
for l from 0 to L do Zdan[l] := l * deltaZdan, end do : #print(Zdan) :
for j from 0 to N do Tdan[j] := j * deltaTdan, end do : #print(Tdan) :
#Нормирование пространственных и временной переменных задачи
x := array(0..M) : y := array(0..K) : z := array(0..L) : t := array(0..N) :
for m from 0 to M do x[m] := X / Xdan[m] : end do : #print(x) :
for k from 0 to K do y[k] := Y / Ydan[k] : end do : #print(y) :
for l from 0 to L do z[l] := H / Zdan[l] : end do : #print(z) :
for j from 0 to N do t[j] := Tb / Tdan[j] : end do : #print(t) :
deltaX := evalf( 1 / M ) : deltaY := evalf( 1 / K ) : deltaZ := evalf( 1 / L ) : deltaT := evalf( 1 / N ) :
#Формирование и нормирование массивов исходных данных задачи
v1 := array(0..M, 0..K, 0..L, 0..N) : vnorm1 := array(0..M, 0..K, 0..L, 0..N) :
kf1 := array(0..M, 0..K, 0..L, 0..N) : kfnorm1 := array(0..M, 0..K, 0..L, 0..N) :
qtochl := array(0..M, 0..K, 0..L, 0..N) : qtochnorm1 := array(0..M, 0..K, 0..L, 0..N) :
s1 := array(0..M, 0..K, 0..L, 0..N) : snorm1 := array(0..M, 0..K, 0..L, 0..N) :
v2 := array(0..M, 0..K, 0..L, 0..N) : vnorm2 := array(0..M, 0..K, 0..L, 0..N) :
kf2 := array(0..M, 0..K, 0..L, 0..N) : kfnorm2 := array(0..M, 0..K, 0..L, 0..N) :
s2 := array(0..M, 0..K, 0..L, 0..N) : snorm2 := array(0..M, 0..K, 0..L, 0..N) :
v3 := array(0..M, 0..K, 0..L, 0..N) : vnorm3 := array(0..M, 0..K, 0..L, 0..N) :
kf3 := array(0..M, 0..K, 0..L, 0..N) : kfnorm3 := array(0..M, 0..K, 0..L, 0..N) :
s3 := array(0..M, 0..K, 0..L, 0..N) : snorm3 := array(0..M, 0..K, 0..L, 0..N) :
for m from 0 to M do
for k from 0 to K do
for l from 0 to L do
for j from 0 to N do
# формирование массивов по оси Oz
v1n := (1 + 0.25 * sin(2 * x[m])) * (1 + 0.25 * sin(5 * y[k])) * (1 + 0.25 * cos(1.5 * z[l])) * (1 + 0.25 * cos(3 * t[j])) :
v1[m, k, l, j] := VOZ * v1n :
qn := (1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(z[l])) * (1 + sin(t[j])) :
qtochl[m, k, l, j] := q0 * qn :
k1n := (1 + 0.25 * sin(5 * x[m])) * (1 + 0.25 * sin(1 * y[k])) * (1 + 0.25 * sin(3 * z[l])) * (1 + 0.25 * cos(2 * t[j])) :
kf1[m, k, l, j] := K0Z * k1n :
dqdt := ((1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(z[l]))) * (cos(t[j])) :
dqz := ((1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(t[j]))) * cos(z[l]) :
dk1dz := (1 + 0.25 * sin(5 * x[m])) * (1 + 0.25 * sin(1 * y[k])) * (1 + 0.25 * cos(3 * t[j])) * 0.75 * cos(3 * z[l]) :
d2qdz := (1 + sin(x[m])) * (1 + sin(y[k])) * (1 + sin(t[j])) * (-sin(z[l])) :
s1[m, k, l, j] := q0 / Tb * (dqdt + alpha * Tb * qn + fx * v1n * dqz - theta * (dk1dz * dqz + k1n * d2qdz)) :
# формирование массивов по оси Ox
v2n := (1 + 0.25 * sin(-2 * x[m])) * (1 + 0.25 * sin(5 * y[k])) * (1 + 0.25 * cos(-1.5 * z[l])) * (1 + 0.25 * cos(3 * t[j])) :
v2[m, k, l, j] := VOX * v2n :
k2n := (1 + 0.25 * sin(5 * x[m])) * (1 + 0.25 * sin(-1 * y[k])) * (1 + 0.25 * sin(3 * z[l])) * (1 + 0.25 * cos(-2 * t[j])) :
kf2[m, k, l, j] := K0X * k2n :
dqx := (cos(x[m])) * (1 + sin(y[k])) * (1 + sin(z[l])) * (1 + sin(t[j])) :
dk2dx := (1.25 * cos(5 * x[m])) * (1 + 0.25 * sin(-1 * y[k])) * (1 + 0.25 * sin(3 * z[l])) * (1 + 0.25 * cos(-2 * t[j])) :
d2qdx := (1 + sin(x[m])) * (-sin(y[k])) * (1 + sin(z[l])) * (1 + sin(t[j])) :
s2[m, k, l, j] := q0 / Tb * (dqdt + fx * v2n * dqx - theta * (dk2dx * dqx + k2n * d2qdx)) :
# формирование массивов по оси Oy
v3n := (1 + 0.25 * sin(-1 * x[m])) * (1 + 0.25 * sin(0.5 * y[k])) * (1 + 0.25 * cos(-2 * z[l])) * (1 + 0.25 * cos(1.5 * t[j])) :

```

```

v3[m, k, l, j] := V0Y·v3n :
k3n := (1 + 0.25·sin(2.5·x[m]))·(1 + 0.25·sin(-0.1·y[k]))·(1 + 0.25·sin(2·z[l]))·(1 + 0.25·cos(-2·t[j])) :
kf3[m, k, l, j] := K0Y·k3n :
dqy := (1 + sin(x[m]))·(1 + sin(y[k]))·(cos(z[l]))·(1 + sin(t[j])) :
dk3dy := (1 + 0.25·sin(2.5·x[m]))·(1 + 0.25·sin(-0.1·y[k]))·(0.5·cos(2·z[l]))·(1 + 0.25·cos(-2·t[j])) :
d2qdy := (1 + sin(x[m]))·(1 + sin(y[k]))·(-sin(z[l]))·(1 + sin(t[j])) :
s3[m, k, l, j] :=  $\frac{q0}{Tb}$  · (dqdt + β·v3n·dqy - θ·(dk3dy·dqy + k3n·d2qdy)) :

end do;
end do;
end do;
end do;
Vmax1 := max(seq(seq(seq(seq(v1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Vmin1 := min(seq(seq(seq(seq(v1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Vmin1=", Vmin1, "Vmax1=", Vmax1);
Kmax1 := max(seq(seq(seq(seq(kf1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Kmin1 := min(seq(seq(seq(seq(kf1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Kmin1=", Kmin1, "Kmax1=", Kmax1);
qmax1 := max(seq(seq(seq(seq(qtoch1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
qmin1 := min(seq(seq(seq(seq(qtoch1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("qmin1=", qmin1, "qmax1=", qmax1);
Smax1 := max(seq(seq(seq(seq(s1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Smin1 := min(seq(seq(seq(seq(s1[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Smin1=", Smin1, "Smax1=", Smax1);
Vmax2 := max(seq(seq(seq(seq(v2[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Vmin2 := min(seq(seq(seq(seq(v2[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Vmin2=", Vmin2, "Vmax2=", Vmax2);
Kmax2 := max(seq(seq(seq(seq(kf2[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Kmin2 := min(seq(seq(seq(seq(kf2[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Kmin2=", Kmin2, "Kmax2=", Kmax2);
Smax2 := max(seq(seq(seq(seq(s2[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Smin2 := min(seq(seq(seq(seq(s2[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Smin2=", Smin2, "Smax2=", Smax2);
Vmax3 := max(seq(seq(seq(seq(v3[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Vmin3 := min(seq(seq(seq(seq(v3[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Vmin3=", Vmin3, "Vmax3=", Vmax3);
Kmax3 := max(seq(seq(seq(seq(kf3[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Kmin3 := min(seq(seq(seq(seq(kf3[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Kmin3=", Kmin3, "Kmax3=", Kmax3);
Smax3 := max(seq(seq(seq(seq(s3[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
Smin3 := min(seq(seq(seq(seq(s3[m, k, l, j], j=0..N), l=0..L), k=0..K), m=0..M)) :
#print("Smin3=", Smin3, "Smax3=", Smax3);
for m from 0 to M do
for k from 0 to K do
for l from 0 to L do
for j from 0 to N do
vnorm1[m, k, l, j] :=  $\frac{v1[m, k, l, j]}{Vmax1}$  : kfnorm1[m, k, l, j] :=  $\frac{kf1[m, k, l, j]}{Kmax1}$  :
qtochnorm1[m, k, l, j] :=  $\frac{qtoch1[m, k, l, j]}{qmax1}$  : snorm1[m, k, l, j] :=  $\frac{s1[m, k, l, j]}{Smax1}$  :
vnorm2[m, k, l, j] :=  $\frac{v2[m, k, l, j]}{Vmax2}$  : kfnorm2[m, k, l, j] :=  $\frac{kf2[m, k, l, j]}{Kmax2}$  : snorm2[m, k, l, j] :=  $\frac{s2[m, k, l, j]}{Smax2}$  :
vnorm3[m, k, l, j] :=  $\frac{v3[m, k, l, j]}{Vmax3}$  : kfnorm3[m, k, l, j] :=  $\frac{kf3[m, k, l, j]}{Kmax3}$  : snorm3[m, k, l, j] :=  $\frac{s3[m, k, l, j]}{Smax3}$  :
end do;
end do;
end do;
end do;
#конец тестового примера
#Метод последовательных приближений
ω := array(0..N) : ω[0] := evalf( $\frac{1}{2}$ ) : ω[N] := evalf( $\frac{1}{2}$ ) : for i from 1 to N-1 do ω[i] := 1 : end do : #print(ω);
Ky := array(0..N, 0..N) : φ1 := array(0..M, 0..K, 0..L, 0..N) :
qv1 := array(0..M, 0..K, 0..L, 0..N, 0..20) : ψ1 := array(0..M, 0..K, 0..L, 0..N, 0..20) :
φ2 := array(0..M, 0..K, 0..L, 0..N) : qv2 := array(0..M, 0..K, 0..L, 0..N, 0..20) : ψ2 := array(0..M, 0..K, 0..L, 0..N, 0..20) :
φ3 := array(0..M, 0..K, 0..L, 0..N) : qv3 := array(0..M, 0..K, 0..L, 0..N, 0..20) :
ψ3 := array(0..M, 0..K, 0..L, 0..N, 0..20) : qit := array(0..M, 0..K, 0..L, 0..N) :
q1 := array(0..M, 0..K, 0..L, 0..N) : q2 := array(0..M, 0..K, 0..L, 0..N) : q3 := array(0..M, 0..K, 0..L, 0..N) :
# Формирование начальных условий
for m from 0 to M do
for k from 0 to K do
for l from 0 to L do
q1[m, k, l, 0] := qtochnorm1[m, k, l, 0];
end do;
end do;
end do;
α := α·Tb : Ntek := N : h :=  $\frac{A}{Ntek}$  :
for j from 0 to N-1 do
#Первая подзадача
for i from 0 to Ntek do

```

```

for r from 0 to Ntek do
  Kv[i, r] := 0 :
end do:
end do:
# Вычисление ядра Kv
for i from 0 to Ntek do
  for r from 0 to i do
    Kv[i, r] := exp(-α·h(i-r)) :
  end do:
end do:
#print(Kv);
# краевые условия
for m from 0 to M do
  for k from 0 to K do
    ql[m, k, 0, j] := qtochnorml[m, k, 0, j] : ql[m, k, L, j] := qtochnorml[m, k, L, j] :
  end do:
end do:
# итерационный алгоритм
for i from 0 to Ntek do
  sum1 := 0 :
  for r from 0 to i do
    for m from 0 to M do
      for k from 0 to K do
        for l from 0 to L do
          sl[m, k, l, r] := sl[m, k, l, j] + (sl[m, k, l, j+1] - sl[m, k, l, j])·h·r :
          sum1 := sum1 + ω[r]·ξ·sl[m, k, l, r]·Kv[i, r]·h :
        end do:
      end do:
    end do:
  end do:
  φl[m, k, l, i] := ql[m, k, l, 0]·exp(-α·h·i) + sum1 :
  qvl[m, k, l, i, 0] := φl[m, k, l, i] :
end do:
#print(φl);
v := 0 : rdiml := (M+1)·(K+1)·(L-1) : ε := 0.1 : sigmal := 10 :
while sigmal > ε do
  v := v + 1 : rol := 0 :
  for m from 0 to M do
    for k from 0 to K do
      for l from 0 to L do
        qvl[m, k, l, j, v] := ql[m, k, l, j] :
      end do:
    end do:
  end do:
  for i from 0 to Ntek do
    for m from 0 to M do
      for k from 0 to K do
        qvl[m, k, 0, i, v] := ql[m, k, 0, j] + i·h·(ql[m, k, 0, j+1] - ql[m, k, 0, j]) :
        qvl[m, k, L, i, v] := ql[m, k, L, j] + i·h·(ql[m, k, L, j+1] - ql[m, k, L, j]) :
      end do:
    end do:
    sum2 := 0 :
    for r from 0 to i do
      for m from 0 to M do
        for k from 0 to K do
          for l from 1 to L-1 do
            vnorml[m, k, l, r] := vnorml[m, k, l, j] + (vnorml[m, k, l, j+1] - vnorml[m, k, l, j])·h·r :
            kfnorml[m, k, l, r] := kfnorml[m, k, l, j] + (kfnorml[m, k, l, j+1] - kfnorml[m, k, l, j])·h·r :
            ψl[m, k, l, r, v-1] := (βξ·vnorml[m, k, l, r] - εξ·(kfnorml[m, k, l, r] - kfnorml[m, k, l-1, r])) / Δξ
          end do:
        end do:
      end do:
      sum2 := sum2 + ω[r]·Kv[i, r]·ψl[m, k, l, r, v-1]·h :
    end do:
  end do:
  end do:
  end do:
  qvl[m, k, l, i, v] := φl[m, k, l, i] - sum2 :
  rol := rol + abs(qvl[m, k, l, i, v] - qvl[m, k, l, i, v-1]) :
end do:
sigmal := evalf(rol / rdiml) :
# print(v, sigmal);
end do:
# условия перехода между первой и второй подзадачей
for m from 0 to M do
  for k from 0 to K do

```

```

    for l from 0 to L do
      q1[m, k, l, j + 1] := qv1[m, k, l, Ntek, v] :
      q2[m, k, l, j] := q1[m, k, l, j + 1] :
    end do:
  end do:
end do:
# вторая подзадача
# итерационный алгоритм
for i from 0 to Ntek do
  sum2 := 0 :
  for r from 0 to i do
    for m from 0 to M do
      for k from 0 to K do
        for l from 0 to L do
          s2[m, k, l, r] := s2[m, k, l, j] + (s2[m, k, l, j + 1] - s2[m, k, l, j]) · h · r :
          sum2 := sum2 + ω[r] · ξ · s2[m, k, l, r] · h :
        end do:
      end do:
    end do:
  end do:
  φ2[m, k, l, i] := q2[m, k, l, j] + sum2 :
  qv2[m, k, l, i, 0] := φ2[m, k, l, i] :
end do:
#print(φ2);
v := 0 : rdim2 := (M - 1) · (K + 1) · (L + 1) : ε := 0.1 : sigma2 := 10 :
while sigma2 > ε do
  v := v + 1 : ro2 := 0 :
  for m from 0 to M do
    for k from 0 to K do
      for l from 0 to L do
        qv2[m, k, l, j, v] := q2[m, k, l, j] :
      end do:
    end do:
  end do:
  for i from 0 to Ntek do
    for l from 0 to L do
      for k from 0 to K do
        qv2[0, k, l, i, v] := q2[0, k, l, i] : qv2[M, k, l, i, v] := q2[M, k, l, i] :
      end do:
    end do:
    sum2 := 0 :
    for r from 0 to i do
      for l from 0 to L do
        for k from 0 to K do
          for m from 1 to M - 1 do
            vnorm2[m, k, l, r] := vnorm2[m, k, l, j] + (vnorm2[m, k, l, j + 1] - vnorm2[m, k, l, j]) · h · r :
            kfnorm2[m, k, l, r] := kfnorm2[m, k, l, j] + (kfnorm2[m, k, l, j + 1] - kfnorm2[m, k, l, j]) · h · r :
            ψ2[m, k, l, r, v - 1] := (β · vnorm2[m, k, l, r] - α · (kfnorm2[m, k, l, r] - kfnorm2[m - 1, k, l, r])) / Δx
            · ( (qv2[m, k, l, r, v - 1] - qv2[m - 1, k, l, r, v - 1]) / Δx ) - α · kfnorm2[m, k, l, r]
            · ( (qv2[m + 1, k, l, r, v - 1] - 2 · qv2[m, k, l, r, v - 1] + qv2[m - 1, k, l, r, v - 1]) / Δx² ) :
          sum2 := sum2 + ω[r] · ψ2[m, k, l, r, v - 1] · h :
        end do:
      end do:
    end do:
  end do:
  qv2[m, k, l, i, v] := φ2[m, k, l, i] - sum2 :
  ro2 := ro2 + abs(qv2[m, k, l, i, v] - qv2[m, k, l, i, v - 1]) :
end do:
sigma2 := evalf( (ro2 / rdim2) ) :
#print(v, sigma2);
end do:
# условия перехода между второй и третьей подзадачей
for m from 0 to M do
  for k from 0 to K do
    for l from 0 to L do
      q2[m, k, l, j + 1] := qv2[m, k, l, Ntek, v] :
      q3[m, k, l, j] := q2[m, k, l, j + 1] :
    end do:
  end do:
end do:
# третья подзадача
# итерационный алгоритм
for i from 0 to Ntek do
  sum3 := 0 :
  for r from 0 to i do

```

```

for m from 0 to M do
  for l from 0 to L do
    for k from 0 to K do
      s3[m, k, l, r] := s3[m, k, l, j] + (s3[m, k, l, j + 1] - s3[m, k, l, j]) · h · r :
      sum3 := sum3 + ω[r] · ζ · s3[m, k, l, r] · h :
    end do:
  end do:
end do:
end do:
φ3[m, k, l, i] := q3[m, k, l, j] + sum3 :
qv3[m, k, l, i, 0] := φ3[m, k, l, i] :
end do:
#print(φ3);
v := 0 : rdim3 := (M + 1) · (K - 1) · (L + 1) : ε := 0.1 : sigma3 := 10 :
while sigma3 > ε do
  v := v + 1 : ro3 := 0 :
  for m from 0 to M do
    for k from 0 to K do
      for l from 0 to L do
        qv3[m, k, l, j, v] := q3[m, k, l, j] :
      end do:
    end do:
  end do:
  for i from 0 to Ntek do
    for l from 0 to L do
      for m from 0 to M do
        qv3[m, 0, l, i, v] := q3[m, 0, l, i] : qv3[m, K, l, i, v] := q3[m, K, l, i] :
      end do:
    end do:
    sum3 := 0 :
    for r from 0 to i do
      for l from 0 to L do
        for m from 0 to M do
          for k from 1 to K - 1 do
            vnorm3[m, k, l, r] := vnorm3[m, k, l, j] + (vnorm3[m, k, l, j + 1] - vnorm3[m, k, l, j]) · h · r :
            kfnorm3[m, k, l, r] := kfnorm3[m, k, l, j] + (kfnorm3[m, k, l, j + 1] - kfnorm3[m, k, l, j]) · h · r :
            ψβ[m, k, l, r, v - 1] := (β · vnorm3[m, k, l, r] - θ · (kfnorm3[m, k, l, r] - kfnorm3[m, k - 1, l, r])
            Δ)
            ( qv3[m, k, l, r, v - 1] - qv3[m, k - 1, l, r, v - 1] ) - θ · kfnorm3[m, k, l, r]
            Δ)
            ( qv3[m, k + 1, l, r, v - 1] - 2 qv3[m, k, l, r, v - 1] + qv3[m, k - 1, l, r, v - 1] ) :
            Δ2
            sum3 := sum3 + ω[r] · ψβ[m, k, l, r, v - 1] · h :
          end do:
        end do:
      end do:
    end do:
    qv3[m, k, l, i, v] := φ3[m, k, l, i] - sum3 :
    ro3 := ro3 + abs(qv3[m, k, l, i, v] - qv3[m, k, l, i, v - 1]) :
  end do:
  sigma3 := evalf( (ro3 / rdim3) ) :
  # print(v, sigma3);
end do:
# условия перехода между третьей и первой подзадачей
for m from 0 to M do
  for k from 0 to K do
    for l from 0 to L do
      q3[m, k, l, j + 1] := qv3[m, k, l, Ntek, v] :
      q1[m, k, l, j] := q3[m, k, l, j + 1] :
      qi[m, k, l, j + 1] := q3[m, k, l, j + 1] :
    end do:
  end do:
end do:
end do: #конец цикла по j
print(qit);

```