

Московский государственный университет  
им. М.В.Ломоносова  
Факультет вычислительной математики и кибернетики

Ю.В.Василевский

И.В.Капырин

**Практикум  
по современным вычислительным  
технологиям и основам  
математического моделирования**

**Аннотация**

В книге предлагаются краткий справочный материал и набор заданий, предназначенные для практического освоения базовых вычислительных технологий и основ математического моделирования. Рассматриваются технологические вопросы построения расчетных сеток, формирования и решения сеточных систем, а также методы численного решения двух модельных уравнений математической физики — стационарного уравнения диффузии и нестационарного уравнения конвекции-диффузии. Справочный материал охватывает ряд ключевых вычислительных технологий и основные методы дискретизации уравнений в частных производных — методы конечных разностей, конечных элементов, конечных объемов. Книга дополняет стандартные учебники по численным методам и рассчитана на студентов старших курсов и аспирантов. Она может служить учебным пособием к практикумам по численным методам и программированию.

Москва 2009 год

# Оглавление

<b>Введение</b>	<b>5</b>
<b>1 Современные вычислительные технологии</b>	<b>8</b>
1.1 Вычислительная технология и библиотеки подпрограмм	9
1.2 Организация обменов данными между приложениями	10
1.3 Задание 1: решение систем линейных уравнений	13
1.3.1 Краткий обзор технологий решения линейных систем	13
1.3.2 Постановка задачи	15
1.3.3 Содержание отчета	16
1.3.4 Варианты заданий практикума	17
1.4 Задание 2: построение расчетных сеток	19
1.4.1 Краткий обзор технологий построения триангуляций	19
1.4.2 Постановка задачи	20
1.4.3 Содержание отчета	21
1.4.4 Варианты заданий практикума	21
1.5 Задание 3: численное решение уравнения Пуассона	24
1.5.1 Краткий обзор технологических решений	24
1.5.2 Постановка задачи	25
1.5.3 Содержание отчета	26
1.5.4 Варианты заданий практикума	26
<b>2 Основы математического моделирования</b>	<b>29</b>
2.1 Стационарное уравнение диффузии	30
2.1.1 Формулировка задачи и расчетная сетка	30
2.1.2 Методы дискретизации уравнения диффузии	30
2.1.3 Задание 1: проверка сходимости численного метода на аналитическом решении	37

2.1.4 Задание 2: Изучение сходимости итерационных методов с переобуславливателями	40
2.1.5 Задание 3: Численное решение задачи о точечном источнике	40
2.1.6 Содержание отчета	41
2.1.7 Варианты заданий практикума	41
2.2 Нестационарное уравнение конвекции-диффузии	45
2.2.1 Формулировка задачи	45
2.2.2 Дискретизация по времени	46
2.2.3 Дискретизация по пространству	46
2.2.4 Численное интегрирование с помощью квадратурных формул Гаусса-Чебышева	53
2.2.5 Практические задания	54
2.2.6 Содержание отчета	56
2.2.7 Варианты заданий практикума	58

# Введение

Численное моделирование является неотъемлемой частью современного индустриального проектирования. За последние десятилетия численная модель стала основным инструментом инженерной деятельности, поскольку только модель может предсказать количественные характеристики проектируемого объекта, а также оценить критически важные величины (температура, напряжение, и т.п.) с любой глубиной детализации.

Численная модель представляет собой математическую модель, описывающую рассматриваемый объект в терминах уравнений и неравенств и реализующую эту модель вычислительную технологию. В современных индустриальных приложениях численное моделирование проводится, как правило, в рамках коммерческих программных продуктов, объединяющих набор математических моделей и вычислительных технологий в единый пользовательский интерфейс. Подобный подход очень удобен для моделирования процессов, описываемых реализованными моделями. Однако, если особенности конкретной задачи не вписываются в заданный набор моделей, или использование имеющихся моделей является неэффективным с точки зрения времени расчета и точности, необходимо создавать свою численную модель. Опыт создания численных моделей очень важен для понимания области применения коммерческих продуктов и степени доверия результатам, получаемым с их помощью.

Математическая модель формирует идейный фундамент численной модели на основе предположений о природе моделируемого явления и допустимых упрощений. Математическая модель использует уравнения математической физики для установления взаимозависимостей между неизвестными функциями, характеризующими объект. Уравнения могут быть дополнены неравенствами и другими ограничениями, связанными с моделью (область, граничные условия и т.д.). Математические вопросы корректности моделей (существо-

вание и единственность решения, а также его устойчивость) являются важными, однако часто нерешенными в используемых моделях. Существует множество примеров приближенного численного решения математических моделей, не исследованных математиками на предмет корректности. Наличие разумного численного решения часто используют как неформальное обоснование корректности модели. В связи с этим, вычислительная технология как процесс получения численного решения может оказывать влияние на формирование самой математической модели.

Вычислительная технология — это совокупность алгоритмов, структур данных, расчетных методик и программных реализаций математической модели на вычислительных системах. На практике вычислительная технология разрабатывается для получения численного решения в нужной форме, к заданному сроку и при известных ограничениях на доступ к машинным и человеческим ресурсам. Эффективность процесса разработки технологии неразрывно связана с производительностью труда разработчика. Общеизвестным средством повышения производительности является интенсивное использование готовых библиотек приложений, или базовых вычислительных технологий, имеющих универсальный характер.

Настоящее пособие представляет собой задачник по практикуму на ЭВМ, предназначенный для знакомства с некоторыми базовыми вычислительными технологиями и основами математического моделирования. Организация заданий и справочного материала заимствована из односеместрового практикума по современным вычислительным технологиям и двухсеместрового практикума на ЭВМ, проводящимися авторами пособия для студентов 3 и 4 курса факультета Вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова.

На первом этапе (первый семестр) студенты изучают уже реализованные вычислительные технологии с точки зрения организации программных решений и пользовательского интерфейса. Изучение технологий проводится в рамках решения индивидуальных задач, поэтому освоение нового сочетается с практическим применением. Задания охватывают основные этапы численного моделирования: построение сеток, дискретизацию уравнений, решение сеточных систем.

На втором этапе (второй и третий семестры) студенты решают модельные задачи математической физики путем разработки собственной численной модели и возможного использования базовых вычислительных технологий. При этом они используют знания и

опыт, приобретенные на первом этапе практикума. Изучение вычислительных технологий и построение численных моделей обеспечивают востребованный опыт математического моделирования.

С методической точки зрения, задания обеих частей курса организованы последовательно: каждое следующее задание опирается на опыт решения предыдущих заданий. В силу этого, необходимые навыки накапливаются и закрепляются постепенно. Кроме того, индивидуальные задания и система отчетности сформированы таким образом, чтобы студенты обогащали свои знания в рамках коллективного обсуждения (электронные презентации, печатные отчеты [4], анализ полученных результатов).

Авторы благодарны А.А.Данилову, К.Д.Никитину и Е.В.Чижонкову за полезные замечания.

## Глава 1

# Современные вычислительные технологии

## 1.1 Вычислительная технология и библиотеки подпрограмм

Разработка вычислительной технологии сводится, главным образом, к определению последовательности расчетных действий, приводящих к желаемому результату. Некоторые операции (или набор операций) часто встречаются в различных приложениях, поэтому их реализации в форме библиотек подпрограмм являются общедоступными.

Использование стандартных библиотек при создании вычислительной технологии дает ряд преимуществ. Во-первых, за счет уменьшения объема программирования и отладки сокращается время разработки и повышается производительность труда разработчика. Во-вторых, широкое использование стандартных библиотек обеспечивает переносимость технологии на разные типы компьютеров. В-третьих, использование библиотек для наиболее затратных алгоритмических блоков повышает эффективность программы, поскольку многие стандартные библиотеки учитывают традиционную архитектуру ЭВМ и адаптируются под новые вычислительные платформы их создателями. Наконец, программа с вызовами стандартных библиотек выглядит компактнее и прозрачнее для пользователя.

Библиотеки могут быть статическими и динамическими. Статическая библиотека представляет собой архив вида *libName.a*, содержащий набор объектных файлов. Этот архив используется на этапе сборки исполняемого файла. Динамическая библиотека — файл вида *libName.so*, подгружаемый в память компьютера во время работы исполняемого файла.

Часто используемыми при вычислениях библиотеками стандартных подпрограмм являются пакеты BLAS (basic linear algebra subprograms) и LAPACK (linear algebra package), доступные в Internet по адресу [www.netlib.org](http://www.netlib.org). Первая библиотека содержит основные операции с векторами и плотными матрицами. Вторая библиотека использует первую и реализует более сложные алгоритмы над плотными матрицами, такими, как обращение матриц или решение задач на собственные значения. Краткое описание возможностей этих библиотек можно найти по адресам [www.netlib.org/blas/blasqr.ps](http://www.netlib.org/blas/blasqr.ps), [www.netlib.org/lapack/lapackqref.ps](http://www.netlib.org/lapack/lapackqref.ps). Главными достоинствами библиотек BLAS и LAPACK являются их реализация на популярных языках программирования (FORTRAN, C), а также оптимизация под все архитектуры компьютеров.

Важность оптимизации под компьютерную архитектуру проил-

люстрируем на простом примере. Современные процессоры снабжены как минимум тремя типами памяти: кэшем первого уровня (L1), кэшем второго уровня (L2), и оперативной памятью. Влияние типа памяти на скорость счета можно исследовать с помощью простейшей программы оценки производительности процессора на примере вычисления скалярного произведения двух векторов длины  $N$ , доступной по адресу <http://www.inm.ras.ru/vtm/svt/checkmflops.c>. В зависимости от величины  $N$  производительность может отличаться в несколько раз. Если вектора помещаются в кэш первого уровня (например,  $N = 1000$ ), то обращение к памяти является чрезвычайно быстрым, и программа показывает производительность именно процессора. Если вектора не помещаются в кэш (например,  $N = 1000000$ ), то процессор вынужден простаивать в ожидании данных, поставляемых из оперативной памяти.

## 1.2 Организация обменов данными между приложениями

Как правило, библиотеки подпрограмм предназначены для использования в определенных приложениях или классах приложений, таких как решение систем линейных уравнений, или построение расчетной сетки. Исключением являются интегральные библиотеки, покрывающие все базовые этапы вычислительной технологии в рамках единого графического пользовательского интерфейса. Несмотря на быстроту организации вычислительной технологии в рамках интегральной библиотеки, ее использование сужает множество возможных приложений, расширить которое чрезвычайно сложно. Наиболее универсальным подходом является организация пользователем всей технологической цепочки вычислений, адаптированной под конкретную задачу и использующую внешние библиотеки приложений для определенных этапов расчета. При этом возникает техническая проблема организации обменов данными между библиотеками приложений.

Самым удобным способом организации обменов является использование библиотек, написанных на одном языке и использующих общий интерфейс базовых процедур (см., например, библиотеки BLAS и LAPACK). В этом случае пользователю достаточно написать главную программу на том же языке программирования.

В ряде случаев требуется использование библиотек, написанных на разных языках программирования. Некоторые библиотеки пред-

лагают интерфейсные процедуры для вызова приложений из программы на другом языке (см., например UMFPACK4.0 для вызова С-приложения из FORTRAN-программы или Ani2d/aniMBA для вызова FORTRAN-приложения из С-программы). Если подобных интерфейсных процедур библиотеками не предусматривается, главная программа пользователя неизбежно делает вызовы процедур, написанных на разных языках. Вообще говоря, универсальных правил создания такой программы, переносимой на все типы операционных систем и компиляторов, нет. Существуют правила, позволяющие работать с мультиязыковыми приложениями в рамках определенных операционных систем и компиляторов. Так, например, при вызове С- и FORTRAN-приложений из главной программы, транслируемых компиляторами Gnu-проекта (g77, g95) в операционных системах типа Linux, использовать языки C++, C и FORTRAN-77 можно напрямую, при этом необходимо учитывать декорации имен в объектных файлах разных языков, способы передачи аргументов функций, конвертацию типов данных в этих языках, а также тип целого для представления указателя.

В качестве иллюстрации разных декораций имен приведем представление одной и той же функции `int test(void)` в разных языках: `test (C)`, `test_ (FORTRAN)`, `_Z4testv (C++)`. В языках C и FORTRAN-77 для способа передачи аргументов функции обычно используется одно и то же соглашение. В языке C++ для методов класса дополнительно передается указатель на класс самым первым аргументом или с помощью регистра процессора.

Ряд программных приложений не предусматривает вызов соответствующих процедур из пользовательской программы. В этом случае приложение считывает данные из файла и записывает результат работы также в файл. При повторяемом (итерационном) вызове ручная организация обмена данными с таким приложением невозможна. Автоматизацию обмена данными через файл можно обеспечить за счет команд операционной системы. Так, например, в операционных системах типа Linux организация обмена между двумя программами `main1.exe` и `main2.exe` через файлы `out_main1`, `out_main2` может выглядеть следующим образом:

```
#!/bin/tcsh                                #!/bin/bash
@ i = 0                                     for ((i=0; i<10; i++))
while ($i != 10)                            do
    ./main1.exe                               ./main1.exe
    test -f out_main1                         [[ -f out_main1 ]] || break
    @ j = $?                                  ./main2.exe
    if ($j != 0) break                       [[ -f out_main2 ]] || break
    ./main2.exe                               done
    test -f out_main2
```

```
@ j = $?
if ($j != 0) break
@ i++
end
```

При этом предполагается, что программа `main1.exe` инициализирует вычисления в отсутствие файла `out_main2`, либо использует `out_main2` для выполнения итерационного подшага, программа `main2.exe` использует `out_main1` для выполнения итерационного подшага.

Запись в файл и считывание из файла могут быть заменены выдочей в стандартные выходной и входной потоки, соответственно, при этом обмен данными может быть организован через конвейер:

```
sh -c 'for ((i=0; i<9; i++)); do echo "./main1.exe | ./main2.exe |"; done;
echo "./main1.exe | ./main2.exe"'
```

## 1.3 Задание 1: решение систем линейных уравнений

Целью задания является освоение технологий решения систем линейных уравнений с квадратными невырожденными матрицами коэффициентов. Подобная задача возникает во многих вычислительных алгоритмах в качестве подзадачи.

### 1.3.1 Краткий обзор технологий решения линейных систем

Методы решения линейных систем разбиваются на два класса: прямые и итерационные методы. В прямых методах решение находится за конечное число действий точно, при условии, что вычисления проводятся в точной арифметике. В итерационных методах находится приближенное решение с желаемой точностью. Разумеется, реальные ошибки округления препятствуют поиску точного решения, однако их влияние, как правило, не является критичным с практической точки зрения. Если требование максимальной эффективности решения не является критическим для рассматриваемого приложения, удобно воспользоваться технологиями “черного ящика”, в которых пользователь должен предоставить минимальные данные о решаемой системе и не должен каким-либо образом настраивать процедуру решения. Рассмотрим несколько таких технологий.

Прямые методы основаны на треугольной факторизации [2, 10] квадратной невырожденной матрицы

$$A = LU,$$

при которой ищутся верхний и нижний треугольные множители  $U$  и  $L$ . Если факторизация выполнена, то дальнейшее решение системы с матрицей  $A$  находится посредством решения двух систем с треугольными матрицами:

$$Ax = y \rightarrow Lz = y, Ux = z.$$

На практике ищется треугольное разложение матрицы с переупорядоченными строками и столбцами. Переупорядочивание используется для устойчивости процедуры факторизации, а также для минимизации числа ненулевых элементов в треугольных множителях в случае факторизации разреженной матрицы. Матрица называется разреженной, если количество ненулевых элементов значительно

меньше количества нулевых. Для экономного хранения разреженных матриц разработан ряд форматов записи их элементов. Так, например, разреженный строчный формат CSR содержит три массива данных: 1) индексы столбцов ненулевых элементов, 2) значения ненулевых элементов, 3) указатели на начало каждой строки матрицы в предыдущих двух массивах. При этом предполагается, что ненулевые элементы матрицы записаны в массивах последовательно строка за строкой, начиная с первой строки.

Для матриц большого порядка процедура факторизации является гораздо более трудоемкой, чем процедура решения систем с треугольными матрицами. Кроме того, при решении многих систем с одной матрицей достаточно факторизовать матрицу только один раз. Поэтому большинство реализаций прямых методов включают независимые подпрограммы факторизации и последующего решения систем с треугольными матрицами.

Для решения систем с плотными матрицами удобно использовать библиотеку LAPACK. Для решения систем с разреженными матрицами можно использовать ряд библиотек (UMFPACK, MUMPS, WSMP, SuperLU, и т.д.).

Основным недостатком методов факторизации является их вычислительная сложность. Более эффективное решение линейных систем может быть получено итерационными методами. Для этого необходимо обеспечить высокую скорость итерационной сходимости к точному решению и низкую арифметическую цену каждой итерации. Современные итерационные методы [2, 10] являются методами на подпространствах Крылова и характеризуются автоматическим выбором итерационных параметров. Фактически, для решения системы с матрицей  $A$  достаточно предоставить процедуру умножения произвольного вектора на матрицу  $A$  и процедуру применения к вектору переобуславливателя  $B$ . Использование переобуславливателя не является обязательным в случае матриц  $A$  с малыми числами обусловленности, и становится почти неизбежным в противном случае, поскольку скорость итерационной сходимости для плохо обусловленных (жестких) систем оказывается очень низкой. Переобуславливатель сводит исходную жесткую систему к менее жесткой системе одним из двух преобразований:

$$Ax = y \rightarrow BAx = By, \quad \text{либо} \quad ABz = y, \quad x = Bz.$$

В первом случае говорят о левом переобуславливателе, во втором — о правом. Возможна комбинация левого и правого переобуславливателя.

С точки зрения максимальной скорости итерационной сходимости наилучшим переобуславливателем является обратная матрица,  $B = A^{-1}$ . В этом случае итерационные процессы на подпространствах Крылова сойдутся за одну или две итерации. Однако выбор  $B = A^{-1}$  не представляется практичным, поскольку формирование такого переобуславливателя является чрезвычайно трудоемкой процедурой по сравнению с исходной целью — решением линейной системы. С другой стороны, неудачный выбор переобуславливателя может даже ухудшить скорость сходимости, наблюдаемую при решении неперевословленных систем. Таким образом, выбор переобуславливателя должен быть основан на разумном компромиссе между эффективностью переобуславливателя и вычислительными затратами на его инициализацию и последующее применение.

Существует два общих подхода к формированию переобуславливателя. В первом подходе используется происхождение матрицы системы и ее специфические свойства (как, например, в многосеточных методах, в методах декомпозиции [2, 5]). Во втором подходе используются только элементы матрицы  $A$ , причем для разреженных матриц — только ненулевые элементы. Самым распространенным примером такого подхода является метод неполной факторизации для разреженных матриц [2, 10]. Суть метода заключается в замене вычисления точной факторизации на вычисление приближенной факторизации путем отбрасывания ряда действий. Правила отбрасывания формируют тот или иной метод неполной факторизации.

Таким образом, комбинация итерационного метода на подпространстве Крылова и какого-либо переобуславливателя порождает технологию решения систем линейных уравнений. Примером реализаций таких технологий могут служить библиотеки SPARSKIT, Api2d, и т.д. Аналогичные итерационные технологии могут быть применены и для нелинейных задач (см., например, решение частичной задачи на собственные значения с помощью библиотеки ARPACK).

### 1.3.2 Постановка задачи

Для выполнения задания требуется освоить ряд технологий черного ящика для решения систем линейных алгебраических уравнений. В рамках выполнения задания необходимо решить заданными способами наборы систем, предложенных преподавателем, и сделать вывод об арифметической сложности использованных процедур решения. Во всех системах правая часть задается формулой  $F_k = \sin(k)$ . В

случае итерационного решения начальное приближение выбирается нулевым, а начальная невязка должна быть уменьшена в  $10^8$  раз. Полученные данные должны быть занесены в таблицу следующего вида:

Порядок системы	x	x	x
Время инициализации	x	x	x
Время решения	x	x	x
Число итераций (итер.метод)	x	x	x

Время инициализации — время факторизации для прямых методов и время формирования переобуславливателя для итерационных методов.

В случае несвоевременной сдачи задания необходимо заполнить таблицу дополнительных расчетов с помощью библиотеки ARPACK:

Порядок матрицы	x	x	x
Поиск $\lambda_{\min}$			
Время инициализации	x	x	x
Время решения	x	x	x
Число итераций	x	x	x
Поиск $\lambda_{\max}$			
Время решения	x	x	x
Число итераций	x	x	x

где  $\lambda_{\min}$ ,  $\lambda_{\max}$  — минимальное и максимальное собственные значения исследуемых матриц.

### 1.3.3 Содержание отчета

Отчет по практикуму должен содержать:

1. Таблицы для заданных методов решения набора систем.
2. Вывод об асимптотической арифметической сложности процедуры инициализации метода решения и процедуры самого решения.
3. В случае несвоевременной сдачи задания: таблицу минимальных и максимальных собственных значений матриц, к которым применялся один из методов треугольной факторизации.

### 1.3.4 Варианты заданий практикума

Для формирования вариантов студенческих заданий вводятся следующие буквенные обозначения:

- А — метод факторизации плотных матриц (LAPACK)
- В — метод факторизации разреженных матриц (UMFPACK 4.0)
- С — метод сопряженных градиентов (Ani2d/aniILU)
- D — метод сопряженных градиентов (SPARSKIT2)
- E — метод бисопряженных градиентов (Ani2d/aniILU)
- F — метод бисопряженных градиентов (SPARSKIT2)
- G — метод минимальных невязок (SPARSKIT2)
- H — переобуславливатели ILU(0), ILU(1), ILU(2) (SPARSKIT2)
- I — переобуславливатели ILU0, ILU2 (Ani2d/aniILU)
- K — переобуславливатель ILUT (SPARSKIT2)

В случае I для метода ILU2 необходимо провести два исследования: фиксируется параметр  $\tau_1$  ( $\tau_1 = 10^{-1}$ ), меняется  $\tau_2$  (попробовать 3 разных значения:  $10^{-2}, 10^{-3}, 10^{-4}$ ); фиксируется параметр  $\tau_2$  ( $\tau_2 = 10^{-4}$ ), меняется  $\tau_1$  (также 3 значения:  $10^{-1}, 10^{-2}, 10^{-3}$ ). В случае K также необходимо провести два исследования: фиксируется параметр  $l_{fil}$  ( $l_{fil} = 50$ ), меняется  $droptol$  (попробовать 3 разных значения:  $10^{-1}, 10^{-2}, 10^{-3}$ ); фиксируется параметр  $droptol$  ( $droptol = 10^{-2}$ ), меняется  $l_{fil}$  (также 3 значения: 20, 40, 80).

В задании представлены наборы матриц двух типов. Первый тип содержит плотные матрицы порядка 250, 500, 1000, 2000. Элементы этих матриц имеют вид  $A_{ii} = 1, A_{ij} = 1/|i-j|$ . Второй тип содержит разреженные матрицы, представленные в CSR формате в файлах, доступных по адресу <http://www.inm.ras.ru/vtm/svt/matr.tgz>.

Каждому студенту даются для освоения две технологии решения систем и два набора матриц, представленных номером типа. Примеры заданий:

1. A1, GH2
2. B2, CI2

3. A2, EK2
4. B2, DH2
5. A1, FK2
6. B2, GK2
7. A2, CH2
8. B2, EI2
9. A1, EH2
10. B2, FH2
11. A2, GK2
12. B2, FI2

Дополнительно к выполненному практическому заданию студенту могут быть заданы следующие вопросы на понимание теории практикума:

1. Практические достоинства и недостатки прямых методов для плотных и разреженных матриц.
2. Область применимости итерационных методов CG, BCGstab, GMRES. Их практические достоинства и недостатки.
3. Разреженный строчный формат записи разреженных матриц.
4. Для чего нужен переобуславливатель и как он работает в итерационных методах?
5. Принципы генерации переобуславливателей класса ILU(k).

#### Интернет ресурсы:

<http://www.netlib.org/lapack/>

<http://www.cise.ufl.edu/research/sparse/umfpack>

<http://www-users.cs.umn.edu/~saad/software/SPARSKIT/sparskit.html>

<http://sourceforge.net/projects/ani2d/>

<http://www.caam.rice.edu/software/ARPACK>

## 1.4 Задание 2: построение расчетных сеток

Целью задания является освоение принципов задания областей и методов построения расчетных сеток. Рассматриваются технологии построения конформных треугольных сеток, имеющих максимально широкую область применения.

### 1.4.1 Краткий обзор технологий построения триангуляций

Важнейшим элементом численной модели является расчетная сетка. Конформные треугольные сетки позволяют достаточно точно аппроксимировать сколь угодно сложную геометрию расчетной области, поэтому такие сетки широко используются в приложениях.

Треугольная сетка называется конформной, если любые два треугольника сетки имеют либо одну общую вершину, либо одно общее ребро, либо не имеют общих точек. Любые частичные пересечения ячеек в конформной сетке запрещены. Треугольная сетка называется регулярной, если каждый треугольник сетки является небольшим искажением некоторого равностороннего треугольника. Другими словами, в регулярной триангуляции отношение диаметра треугольника и радиуса вписанной в него окружности не превышает фиксированной величины. Треугольная сетка называется квазиравномерной, если она регулярна и все треугольники имеют приблизительно один и тот же диаметр. Квазиравномерные сетки определяются параметром характерного размера  $h$ , а регулярные сетки со сгущениями или разрежениями определяются количеством треугольников, минимальным размером треугольника, степенью сгущения, и т.д.

Конформные треугольные сетки можно разбить по топологическим признакам на три класса — структурированные (с  $ij$ -структурой), иерархические локально измельченные и неструктурированные. По геометрическим признакам триангуляции подразделяют на квазиравномерные, регулярные и анизотропные. Методы и технологии построения сеток разных классов существенно отличаются.

Структурированные сетки строятся посредством отображения базовой прямоугольной сетки из единичного квадрата на расчетную область.

Иерархические сетки строятся иерархическим измельчением заданной сетки. Измельчение осуществляется путем многоуровневого

разбиения некоторых треугольников на два или четыре подтреугольника с последующим восстановлением конформности сетки. Если начальная триангуляция является регулярной, локально измельченная сетка также будет регулярной триангуляцией, которую в свою очередь можно локально разгрублять или измельчать. Структура таких сеток удобна для их динамического перестроения. Для построения таких сеток необходимо задать начальную конформную триангуляцию, а также указать правило выбора разбиваемых (разгрубляемых) треугольников на каждом уровне разбиения (разгрубления).

Неструктурированные регулярные триангуляции используются для дискретизации сложных областей и решений с сильными особенностями. В настоящее время применяется два метода построения неструктурированных триангуляций — метод Делоне и метод продвигаемого фронта. В обоих случаях область задается через свою границу, представленную либо в дискретном, либо в параметризованном виде. Для построения сеток с заданными свойствами необходимо определить в расчетной области скалярную функцию локального размера сетки  $h(x)$ .

Неструктурированные анизотропные триангуляции являются наиболее широким классом конформных треугольных сеток. Они позволяют эффективно приближать как изотропные, так и анизотропные особенности решений численных моделей. Анизотропные триангуляции часто определяются как триангуляции, квазиравномерные в некоторой тензорной метрике. В силу этого, для построения анизотропной триангуляции в расчетной области требуется задать тензорную функцию метрики. Существующие технологии способны произвести квазиравномерную в заданной метрике триангуляцию. При этом задание расчетной области сводится либо к заданию границы (параметризованной или дискретной), либо к заданию начальной сетки.

### 1.4.2 Постановка задачи

Для выполнения задания требуется построить двумерные локально измельчаемые и разгрубляемые динамические сетки (библиотека Ani2d/aniRCB), а также приобрести навыки в построении и адаптации неструктурированных сеток в двумерных областях с криволинейными границами (пакеты EMC2, FREEFEM, библиотеки Ani2d/aniAFT, Ani2d/aniMBA).

В случае несвоевременной сдачи задания необходимо выполнить дополнительное задание — построить конформную тетраэдральную

сетку в трехмерной области  $O \setminus K_1$ , где  $O$  — единичный шар с центром  $(0,0,0)$ ,  $K_1$  — октант  $x > 0, y > 0, z > 0$  (библиотека Ani3d/aniAFT).

### 1.4.3 Содержание отчета

Отчет по практикуму должен содержать:

1. Последовательность изображений динамически измельчаемых сеток.
2. Изображения построенных неструктурированных сеток.
3. В случае несвоевременной сдачи задания: изображение следа построенной тетраэдральной сетки на границе, а также один из центральных разрезов тетраэдральной сетки.

### 1.4.4 Варианты заданий практикума

Начнем с введения необходимых обозначений. Для последовательности иерархических локально измельченных сеток в единичном квадрате, сгущающихся к движущемуся источнику, зададим источник как

- $a$  — точка, чья траектория — окружность, вписанная в единичный квадрат
- $b$  — отрезок  $0.5 \leq x \leq 0.75, y = 0.5$ , вращающийся вокруг точки  $(0.5, 0.5)$
- $c$  — кривая  $y = 0.5 + 0.1 \sin(4\pi(x - t)), t \geq 0$
- $d$  — окружность с центром  $(0.5, 0.5)$  и радиусом  $t, t \geq 0$ .

Для построения неструктурированной регулярной триангуляции определим области

1.  $O \setminus ((K_1 \cup K_3) \setminus R)$ , где  $O$  — единичный круг с центром  $(0,0)$ ,  $K_1$  — квадрант  $x > 0, y > 0$ ,  $K_3$  — квадрант  $x < 0, y < 0$ ,  $R$  — ромб  $|x| + |y| \leq 0.1$ ,
2.  $(E_1 \cup E_2) \setminus R$ , где  $E_1$  — эллипс  $x^2 + 4y^2 < 1$ ,  $E_2$  — эллипс  $4x^2 + y^2 < 1$ ,  $R$  — ромб  $|x| + |y| \leq 0.1$ ,

и методы задания области

- $A$  — границей посредством меню в графической оболочке (EMC2)

- $B$  — границей посредством команд специализированной программы (FREEFEM)
- $C$  — границей через набор гладких параметризованных кусков (Ani2d/aniAFT)
- $D$  — границей через дискретизацию границы (Ani2d/aniAFT)
- $E$  — через грубую сетку с параметризованными ребрами (Ani2d/aniMBA).

Каждому студенту необходимо построить одну последовательность динамически измельчаемых сеток, сгущающихся к заданному типу источника, и одну неструктурированную сетку для области с криволинейными границами и способа задания этих границ. Примеры заданий:

1.  $a, A1$
2.  $b, B1$
3.  $c, C1$
4.  $d, D1$
5.  $a, E1$
6.  $b, A2$
7.  $c, B2$
8.  $d, C2$
9.  $a, D2$
10.  $b, E2$
11.  $c, D1$
12.  $d, E1$

Дополнительно к выполненному практическому заданию студенту могут быть заданы следующие вопросы на понимание используемого программного обеспечения:

1. Изменить степень сгущения в последовательности иерархических локально сгущающихся сеток.

2. Построить иерархические локально сгущающиеся сетки в области с криволинейными границами.
3. Построить неструктурированную сетку, локально сгущающуюся в заданной преподавателем зоне.
4. Перечислить и пояснить способы задания областей для сеточных генераторов.

**Интернет ресурсы:**

<http://sourceforge.net/projects/ani2d/>

<http://www-rocq.inria.fr/gamma/cdrom/www/emc2/eng.htm>

<http://www.freefem.org/ff++/index.htm>

<http://www.andrew.cmu.edu/user/sowen/mesh.htm>

<http://sourceforge.net/projects/ani3d/>

## 1.5 Задание 3: численное решение уравнения Пуассона

Целью задания является знакомство с аппроксимационными свойствами сеток и объемом вычислительной работы, затраченной на получение приближенного решения модельной краевой задачи с заданной точностью. Полученный в первых заданиях опыт применяется при выборе методов дискретизации и решения линейных систем, необходимых для формирования вычислительной технологии.

### 1.5.1 Краткий обзор технологических решений

Большинство математических моделей допускают только приближенное численное решение, которое достигается дискретизацией модели и, как правило, решением порождаемых систем линейных уравнений. Точность дискретной модели зависит от используемой сетки и метода аппроксимации уравнений математической модели, а также от точности решения возникающих линейных систем. Вычислительная работа, затраченная на получение приближенного решения, как правило, состоит из построения сетки, формирования сеточной системы и решения сформированной системы уравнений. В случае приближенного решения модельных задач с помощью уже разработанной вычислительной технологии, трудовые затраты пользователя не представляются значительными, поскольку они сводятся к изучению примеров использования технологии и возможной модификации этих примеров. В силу этого на выбор технологии влияет, в первую очередь, объем вычислительной работы для достижения заданной точности приближенного решения.

Вычислительные затраты на построение расчетной сетки зависят от типа сетки. Так, например, построение прямоугольной сетки в прямоугольной области или иерархической локально измельченной триангуляции в многоугольной области требуют пренебрежимо мало вычислений по сравнению с другими этапами вычислительной технологии. Построение неструктурированной регулярной триангуляции более затратно, однако возможно в любых областях. Построение неструктурированной анизотропной адаптивной триангуляции является, пожалуй, наиболее дорогой процедурой генерации расчетной сетки. Однако затраты на построение такой сетки могут окупиться за счет существенного уменьшения числа узлов и порядка сеточной системы, необходимых для достижения фиксированной точности приближения.

Количество расчетных действий по формированию системы сеточных уравнений, как правило, пропорционально количеству расчетных узлов и не является доминирующей частью вычислительной работы. Коэффициент пропорциональности зависит от типа сетки и метода аппроксимации математической модели. Используемые в задании методы аппроксимации либо общеизвестны (метод конечных разностей на прямоугольных сетках), либо являются составной частью изучаемой вычислительной технологии (метод конечных элементов на треугольных сетках). Краткое изложение этих методов можно найти во второй части настоящего пособия.

Арифметическая сложность решения системы сеточных уравнений зависит от метода нахождения решения и свойств матрицы системы, таких, как ее структура разреженности и ее спектральные свойства. Выбор метода решения зависит от многих факторов: доступность дополнительной информации о матрице системы, специальная структура матрицы, вычислительная сложность метода, пользовательские затраты (временные и денежные) на реализацию метода, и т.д. Среди методов решения, предлагаемых к использованию в рамках этого задания, есть как технологии черного ящика (прямой метод разреженной факторизации, итерационные методы неполной факторизации), так и специальные методы почти оптимальной асимптотики арифметической сложности (быстрый прямой метод  $O(N \log N)$ , итерационный многосеточный метод  $O(N \log \epsilon^{-1})$  для вычисления решения с погрешностью  $\epsilon$ ).

## 1.5.2 Постановка задачи

Для выполнения задания требуется двумя способами решить уравнение Пуассона приближенно с заданной точностью. Для этого необходимо построить сетку заданного преподавателем типа, сформировать систему сеточных уравнений заданным методом дискретизации, решить систему сеточных уравнений и изучить зависимость ошибки численного решения (0.01, 0.001, 0.0001 в непрерывной норме) от расчетного времени.

Рассматривается уравнение Пуассона в единичном квадрате

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x, y) \text{ в } \Omega = (0, 1)^2,$$

$$u = g \text{ на } \partial\Omega.$$

Правая часть  $f$  и граничное условие  $g$  таковы, что точное решение  $u = (|x - y| + 10^{-5})^{1.5}$ .

Полученные данные должны быть занесены в таблицу следующего вида:

Ошибка численного решения	0.01	0.001	0.0001
Время построения сетки	x	x	x
Время формирования сеточной системы	x	x	x
Время решения системы	x	x	x
Число итераций (итер.метод)	x	x	x

В случае итерационного решения начальное приближение выбирается нулевым, а начальная невязка должна быть уменьшена в  $10^8$  раз.

## 1.5.3 Содержание отчета

Отчет по практикуму должен содержать:

1. Две таблицы для двух заданных методов приближенного решения.
2. Вывод об асимптотической арифметической сложности (расчетного времени) каждого этапа поиска приближенного решения в зависимости от числа расчетных узлов.
3. Вывод об асимптотической арифметической сложности (расчетного времени) каждого этапа поиска приближенного решения в зависимости от точности численного решения.

## 1.5.4 Варианты заданий практикума

Для формирования вариантов студенческих заданий вводятся следующие буквенные обозначения:

- A — быстрый прямой метод для решения систем с сепарабельными матрицами (dc2d)
- B — прямой метод факторизации разреженных матриц (UMFPACK 4.0)
- C — итерационный метод сопряженных градиентов с переобуславливателем ILU2 (Ani2d/aniILU)
- D — итерационный метод сопряженных градиентов с переобуславливателем ILU(k) (SPARSKIT2)

- E — итерационный многосеточный метод (PLTMG)

Методы дискретизации и используемые сетки обозначаются следующим образом:

1. Метод конечных разностей на квадратной сетке
2. Метод конечных элементов  $P_1$  на адаптивной анизотропной неструктурированной триангуляции (Ani2d/aniMBA, Ani2d/aniFEM)
3. Метод конечных элементов  $P_1$  на адаптивной регулярной иерархической триангуляции (Ani2d/aniRCB, Ani2d/aniFEM)
4. Метод конечных элементов  $P_1$  на адаптивной анизотропной неструктурированной триангуляции (FREEFEM)
5. Метод конечных элементов  $P_1$  на адаптивной регулярной иерархической триангуляции (PLTMG)

Примеры заданий:

1. A1, B2
2. B1, C2
3. C1, D2
4. D1, B3
5. A1, C3
6. B1, D3
7. C1, B4
8. D1, E5
9. A1, C2
10. B1, D2
11. C1, B3
12. D1, C3

Дополнительно к выполненному практическому заданию студенту могут быть заданы следующие вопросы на понимание используемого программного обеспечения:

1. Пересчитать одну из таблиц для  $L_2(\Omega)$  нормы ошибки численного решения.
2. В случае построения анизотропной сетки, модифицировать программу для построения регулярной сетки, а также квазиравномерной сетки.
3. Идейная основа метода конечных разностей.
4. Идейная основа метода конечных элементов.

#### Интернет ресурсы:

<http://sourceforge.net/projects/ani2d/>

<http://www.cise.ufl.edu/research/sparse/umfpack>

<http://www-users.cs.umn.edu/~saad/software/SPARSKIT/sparskit.html>

<http://www.inm.ras.ru/vtm/svt/dc2d.tgz>

<http://www.freefem.org>

<http://www.scicomp.ucsd.edu/~reb/software.html>

# Глава 2

## Основы математического моделирования

### 2.1 Стационарное уравнение диффузии

#### 2.1.1 Формулировка задачи и расчетная сетка

Рассматривается стационарное уравнение диффузии

$$-\nabla \cdot \mathcal{D} \nabla C = f \quad (2.1)$$

в области  $\Omega \in \mathbb{R}^2$  с границей  $\partial\Omega$ .  $C$  — концентрация вещества,  $f = f(x, y)$  — функция источников или стоков. Граница  $\partial\Omega$  состоит из двух частей:  $\partial\Omega = \Gamma_D \cup \Gamma_N$ , на  $\Gamma_D$  заданы граничные условия типа Дирихле

$$C|_{\Gamma_D} = g_D, \quad (2.2)$$

а на  $\Gamma_N$  — граничные условия типа Неймана

$$-\mathcal{D} \frac{\partial C}{\partial \vec{n}} \Big|_{\Gamma_N} = g_N, \quad (2.3)$$

где  $\vec{n}$  — внешняя нормаль к границе области. В предлагаемых задачах область  $\Omega$  предполагается квадратной (к примеру, единичный квадрат  $(0; 1) \times (0; 1)$ ), а тензор диффузии — постоянным в расчетной области и диагональным:

$$\mathcal{D} = \begin{pmatrix} d_x & 0 \\ 0 & d_y \end{pmatrix} \quad (2.4)$$

В следующих параграфах описаны численные методы решения задачи (2.1),(2.2),(2.3). Для их формулировки введем в области  $\Omega$  равномерную квадратную сетку  $\varepsilon_h$ , содержащую вершины  $V_i^j$  с координатами  $(x_i, y_j)$ , где  $x_i = i \cdot h$ ,  $y_j = j \cdot h$ ,  $i, j = 0, \dots, N$ . Обозначим через  $h = N^{-1}$  шаг сетки.

#### 2.1.2 Методы дискретизации уравнения диффузии

##### Метод конечных разностей

Уравнение (2.1) с учетом вида тензора диффузии (2.4) можно представить в следующем виде:

$$-d_x \frac{\partial^2 C}{\partial x^2} - d_y \frac{\partial^2 C}{\partial y^2} = f. \quad (2.5)$$

Введем обозначение  $C_i^j$  — аппроксимация концентрации в точке  $(x_i, y_j)$ .

Построим с помощью метода конечных разностей (МКР) аппроксимацию вторых производных, входящих в (2.5). Для этого сначала запишем аппроксимацию первой производной по  $x$  в промежуточных точках  $x_{i-\frac{1}{2}} = x_i - h/2$  и  $x_{i+\frac{1}{2}} = x_i + h/2$  при фиксированном  $y_j$ :

$$\begin{aligned}\frac{\partial C}{\partial x}(x_{i-\frac{1}{2}}, y_j) &\approx \frac{C(x_i, y_j) - C(x_{i-1}, y_j)}{h} \approx \frac{C_i^j - C_{i-1}^j}{h}, \\ \frac{\partial C}{\partial x}(x_{i+\frac{1}{2}}, y_j) &\approx \frac{C(x_{i+1}, y_j) - C(x_i, y_j)}{h} \approx \frac{C_{i+1}^j - C_i^j}{h}.\end{aligned}$$

Используя последние выражения, строим аппроксимацию второй производной по  $x$  в центральной точке  $(x_i, y_j)$ :

$$\frac{\partial^2 C}{\partial x^2}(x_i, y_j) \approx \frac{C_{i+1}^j - 2C_i^j + C_{i-1}^j}{h^2}.$$

Аналогично получается аппроксимация второй производной по  $y$ :

$$\frac{\partial^2 C}{\partial y^2}(x_i, y_j) \approx \frac{C_i^{j+1} - 2C_i^j + C_i^{j-1}}{h^2}. \quad (2.6)$$

Заменяя производные в (2.5) их сеточными аппроксимациями, получаем дискретные уравнения:

$$\begin{aligned}-d_x \frac{C_{i+1}^j - 2C_i^j + C_{i-1}^j}{h^2} - d_y \frac{C_i^{j+1} - 2C_i^j + C_i^{j-1}}{h^2} = f(x_i, y_j), \\ i = 1, \dots, N-1, \quad j = 1, \dots, N-1.\end{aligned} \quad (2.7)$$

Требуется отдельная обработка точек, лежащих на границе  $\partial\Omega$ . В точках  $V_i^j \in \Gamma_D$  значения концентрации известны из граничных условий:  $C_i^j = g_D(x_i, y_j)$ . Эти известные значения переносятся в правую часть уравнений (2.7). Для точек  $V_i^j \in \Gamma_N$  предлагается использовать два варианта аппроксимации граничного условия типа Неймана. Пусть, к примеру,  $\Gamma_N$  — вертикальный отрезок, соответствующий участку границы  $x = x_N$ . Тогда первый вариант аппроксимации граничных условий в точке  $(x_N, y_j)$ , наиболее простой, первого порядка точности, выглядит следующим образом:

$$-d_x \frac{C_N^j - C_{N-1}^j}{h} = g_N(x_N, y_j). \quad (2.8)$$

Для симметризации получаемой общей линейной системы поделим обе части уравнения на  $h$ :

$$-d_x \frac{C_N^j - C_{N-1}^j}{h^2} = \frac{g_N(x_N, y_j)}{h}.$$

Построим более точную аппроксимацию граничного условия типа Неймана, пользуясь разложением в ряд Тейлора:

$$\begin{aligned}C(x_{N-1}, y_j) &= C(x_N, y_j) - h \frac{\partial C}{\partial x}(x_N, y_j) + \\ &+ \frac{h^2}{2} \frac{\partial^2 C}{\partial x^2}(x_N, y_j) + O(h^3).\end{aligned} \quad (2.9)$$

Используя исходное дифференциальное уравнение (2.5) и аппроксимацию второй производной (2.6), имеем:

$$\frac{\partial^2 C}{\partial x^2}(x_N, y_j) \approx -\frac{1}{d_x} \left( f(x_N, y_j) + d_y \frac{C_N^{j+1} - 2C_N^j + C_N^{j-1}}{h^2} \right). \quad (2.10)$$

Подставляя (2.10) в (2.9), и выражая производную, получаем:

$$\begin{aligned}\frac{\partial C}{\partial x}(x_N, y_j) &\approx \frac{C_N^j - C_{N-1}^j}{h} - \\ &- \frac{h}{2d_x} \left( f(x_N, y_j) + d_y \frac{C_N^{j+1} - 2C_N^j + C_N^{j-1}}{h^2} \right).\end{aligned}$$

Итоговое уравнение, соответствующее аппроксимации краевого условия (2.3) со вторым порядком точности, выглядит следующим образом:

$$\begin{aligned}-d_x \frac{C_{N-1}^j - C_N^j}{h} - d_y \frac{C_N^{j+1} - 2C_N^j + C_N^{j-1}}{2h} - \\ - \frac{h}{2} f(x_N, y_j) = g_N(x_N, y_j).\end{aligned} \quad (2.11)$$

Для симметризации получаемой общей линейной системы поделим обе части уравнения (2.11) на  $h$  и перенесем известную функцию источника в правую часть:

$$\begin{aligned}-d_x \frac{C_{N-1}^j - C_N^j}{h^2} - d_y \frac{C_N^{j+1} - 2C_N^j + C_N^{j-1}}{2h^2} = \\ = \frac{1}{2} f(x_N, y_j) + \frac{g_N(x_N, y_j)}{h}.\end{aligned} \quad (2.12)$$

Итак, полученная в результате дискретизации задачи (2.1), (2.2), (2.3) линейная система состоит из уравнений (2.7) для внутренних точек и уравнений (2.12) (или (2.8)) для точек на границе с условиями типа Неймана.

**Метод конечных элементов**

Будем искать приближенное решение задачи (2.1)  $C_h(x, y)$  в виде

$$C_h(x, y) = \sum_{i=0, j=0}^{i=N, j=N} C_i^j \phi_i^j(x, y).$$

Здесь  $\phi_i^j(x, y)$  — базисные кусочно-билинейные функции, принимающие значение 1 в точке  $V_i^j$  и 0 во всех остальных узлах сетки. Для узла сетки  $V_i^j$  носителем  $\phi_i^j(x, y)$  являются четыре ячейки сетки, окружающие вершину  $V_i^j$ . Ограничение базисной функции  $\phi_i^j(x, y)$  на ячейку сетки  $e_k^l = [x_k; x_{k+1}] \times [y_l; y_{l+1}]$  является билинейной функцией

$$\phi_i^j(x, y) \Big|_{e_k^l} = ax + by + cxy + d,$$

в которой коэффициенты  $a, b, c, d$  определяются значениями функции в вершинах ячейки  $e_k^l$ . К примеру, на единичном квадрате  $[0; 1] \times [0; 1]$  определены следующие четыре базисных билинейных функции:  $xy, (1-x)y, x(1-y), (1-x)(1-y)$ .

Для двух функций  $f(x, y), g(x, y) \in L_2(\Omega)$  определим  $(f, g) = \int_{\Omega} fg \, dx$  — скалярное произведение в  $L_2(\Omega)$ . Для построения конечно-элементной дискретизации умножим (2.1) на  $\phi_i^j$  и проинтегрируем по  $\Omega$ :

$$-(\nabla \cdot \mathcal{D}\nabla C, \phi_i^j) = (f, \phi_i^j), \\ i = 0, \dots, N, j = 0, \dots, N, \text{ т.ч. } (x_i, y_j) \notin \Gamma_D.$$

Заменим  $C(x, y)$  на  $C_h$  и запишем для  $C_h$  следующее представление, учитывающее наличие граничных условий типа Дирихле:

$$C_h = C_{h,0} + C_{h,\Gamma_D}, \quad \text{где} \quad (2.13) \\ C_{h,\Gamma_D} = \sum_{(x_i, y_j) \in \Gamma_D} C_{\Gamma_D, i}^j \phi_i^j = \sum_{(x_i, y_j) \in \Gamma_D} g_D(x_i, y_j) \phi_i^j, \\ C_{h,0} = \sum_{(x_i, y_j) \notin \Gamma_D} C_{0, i}^j \phi_i^j.$$

Применим формулу Гаусса-Остроградского для вычисления инте-

грала, стоящего в левой части уравнения:

$$-\int_{\partial\Omega} \mathcal{D}\nabla C_h \cdot \vec{n} \phi_i^j \, dl + (\mathcal{D}\nabla C_h, \nabla \phi_i^j) = (f, \phi_i^j) \\ \forall i, j : (x_i, y_j) \notin \Gamma_D.$$

Далее

$$-\int_{\Gamma_N} \mathcal{D}\nabla C_h \cdot \vec{n} \phi_i^j \, dl - \int_{\Gamma_D} \mathcal{D}\nabla C_h \cdot \vec{n} \phi_i^j \, dl + (\mathcal{D}\nabla C_h, \nabla \phi_i^j) = (f, \phi_i^j) \\ \forall i, j : (x_i, y_j) \notin \Gamma_D,$$

$$-\int_{\Gamma_N} \mathcal{D}\nabla C_h \cdot \vec{n} \phi_i^j \, dl - \int_{\Gamma_D} \mathcal{D}\nabla C_h \cdot \vec{n} \phi_i^j \, dl + (\mathcal{D}\nabla C_{h,0}, \nabla \phi_i^j) = \\ = (f, \phi_i^j) - (\mathcal{D}\nabla C_{h,\Gamma_D}, \nabla \phi_i^j) \quad \forall i, j : (x_i, y_j) \notin \Gamma_D.$$

На участке границы  $\Gamma_N$  выражение  $\int_{\Gamma_N} \mathcal{D}\nabla C_h \cdot \vec{n} \phi_i^j \, dl = - \int_{\Gamma_N} g_N \phi_i^j \, dl$  известно из граничных условий. На оставшейся части границы  $\int_{\Gamma_D} \mathcal{D}\nabla C_h \cdot \vec{n} \phi_i^j \, dl = 0$ , так как  $\phi_i^j \Big|_{\Gamma_D} = 0$  при таких  $i, j$ , что  $(x_i, y_j) \notin \Gamma_D$ . Переносим все известные в правую часть, получаем:

$$(\mathcal{D}\nabla C_{h,0}, \nabla \phi_i^j) = (f, \phi_i^j) - (\mathcal{D}\nabla C_{h,\Gamma_D}, \nabla \phi_i^j) - \int_{\Gamma_N} g_N \phi_i^j \, dl, \\ \forall i, j : (x_i, y_j) \notin \Gamma_D.$$

Подставляем в последнее уравнение выражения для  $C_{h,0}$  и  $C_{h,\Gamma_D}$  (2.13):

$$\sum_{k,l: (x_k, y_l) \notin \Gamma_D} C_{0,k}^l (\mathcal{D}\nabla \phi_k^l, \nabla \phi_i^j) = (f, \phi_i^j) - \\ - \sum_{k,l: (x_k, y_l) \in \Gamma_D} g_D(x_k, y_l) (\mathcal{D}\nabla \phi_k^l, \nabla \phi_i^j) - \int_{\Gamma_N} g_N \phi_i^j \, dl, \\ \forall i, j : (x_i, y_j) \notin \Gamma_D.$$

В итоге нами получена система уравнений с неизвестными  $C_{0k}^l$  размерности числа узлов сетки, не лежащих на границе  $\Gamma_D$ . Ее матрица является симметричной положительно определенной. Элементы матрицы и правой части вычисляются путем численного интегрирования по ячейкам сетки, входящим одновременно в носители базисных функций  $\phi_i^j$  и  $\phi_k^l$ .

### Метод конечных объемов

В методе конечных объемов (МКО), в отличие от двух первых методов, степени свободы ассоциированы не с узлами сетки, а с ячейками. Концентрация считается постоянной внутри каждой из ячеек сетки. Введем вспомогательную переменную  $\vec{u} = -\mathcal{D}\nabla C$  — плотность диффузионного потока. Перепишем (2.1) в виде системы:

$$\begin{aligned}\nabla \cdot \vec{u} &= f, \\ \vec{u} &= -\mathcal{D}\nabla C.\end{aligned}\quad (2.14)$$

Пусть  $e_k^l = [x_{k-1}; x_k] \times [y_{l-1}; y_l]$ ,  $k, l = 1, \dots, N$  — ячейки сетки. Проинтегрируем первое уравнение системы по ячейке  $e_i^j$ :

$$\int_{e_i^j} \nabla \cdot \vec{u} dx = \int_{e_i^j} f dx.$$

Применяем формулу Гаусса-Остроградского:

$$\begin{aligned}\int_{e_i^j} \nabla \cdot \vec{u} dx &= \int_{\partial e_i^j} \vec{u} \cdot \vec{n} dl = \\ &= (u_{i,i+1}^j - u_{i-1,i}^j)h + (u_i^{j,j+1} - u_i^{j-1,j})h = \int_{e_i^j} f dx,\end{aligned}\quad (2.15)$$

где  $u_{i,i+1}^j$  — аппроксимация плотности диффузионного потока от ячейки  $e_i^j$  к ячейке  $e_{i+1}^j$  через их общую грань в направлении нормали  $\vec{n}_{i,i+1}^j$ , внешней по отношению к  $e_i^j$ . Для определения  $u_{i,i+1}^j$  рассмотрим вспомогательную ячейку  $V = [x_i - \frac{h}{2}, x_i + \frac{h}{2}] \times [y_{j-1}, y_j]$ . Сначала построим на ней аппроксимацию плотности диффузионного потока  $\vec{u}_V$ , интегрируя (2.14) по объему  $V$ :

$$\int_V \mathcal{D}^{-1} \vec{u}_V dx = - \int_{\partial V} C \vec{n} dl \quad (2.16)$$

Положим  $u_{i,i+1}^j = \int_V \frac{\vec{u}_V \cdot \vec{n}}{h^2} dx$  и скалярно умножим последнее уравнение на нормаль  $\vec{n}_{i,i+1}^j$ :

$$\frac{h^2}{d_x} u_{i,i+1}^j + \int_{\partial V} C \vec{n} \cdot \vec{n}_{i,i+1} dl = 0.$$

Интегрируя по контуру, получаем:

$$\frac{h^2}{d_x} u_{i,i+1}^j + (C_{i+1}^j - C_i^j)h = 0.$$

В итоге имеем выражение для плотности диффузионного потока  $u_{i,i+1}^j$ :

$$u_{i,i+1}^j = -\frac{d_x(C_{i+1}^j - C_i^j)}{h}. \quad (2.17)$$

Аналогично определяются диффузионные потоки через остальные три ребра ячейки  $e_i^j$ .

В случае, когда ребро сетки  $\gamma$  находится на границе с условиями типа Неймана, плотность потока через него известна:

$$u_\gamma = \frac{1}{h} \int_\gamma g_N dl. \quad (2.18)$$

Если ребро  $\gamma$  ячейки  $e_k^l$  находится на границе типа Дирихле, то необходимо ввести на ней степень свободы  $C_\gamma = \frac{1}{h} \int_\gamma g_D dl$ , рав-

ную среднему значению на грани функции  $g_D$ . Допустим, для определенности, что ребро  $\gamma \in \Gamma_D$  ячейки  $e_N^j$  — отрезок, соединяющий точки  $(x_N, y_{j-1})$  и  $(x_N, y_j)$ . Построим вспомогательную ячейку  $V = [x_N - \frac{h}{2}, x_N] \times [y_{j-1}, y_j]$ . Интегрируя по  $V$  аналогично (2.16) и домножая на нормаль к  $\gamma$ , внешнюю по отношению к  $V$ , аппроксимируем диффузионный поток  $u_\gamma$  через  $\gamma$ :

$$u_\gamma = -\frac{d_x(C_\gamma - C_k^l)}{h/2}. \quad (2.19)$$

Аналогично строится аппроксимация диффузионного потока через другие ребра на границе с условиями типа Дирихле.

Итоговая система уравнений получается путем подстановки выражений для потоков (2.17), (2.18), (2.19) в уравнение (2.15) для каждой ячейки  $e_k^j$ , при этом известные потоки через границы с условиями Неймана, а также концентрации на границах с условиями Дирихле переносятся в правую часть. Например, уравнение:

$$-d_x(C_{i+1}^j - 2C_i^j + C_{i-1}^j) - d_y(C_i^{j+1} - 2C_i^j + C_i^{j-1}) = \int_{e_i^j} f dx$$

справедливо для любой  $ij$ -ой ячейки сетки, не имеющей граней на границе области.

### 2.1.3 Задание 1: проверка сходимости численного метода на аналитическом решении

В первом задании требуется запрограммировать один из методов дискретизации (МКР, МКЭ или МКО) уравнения диффузии. Для решения полученной линейной системы необходимо использовать метод сопряженных градиентов [10] с нулевым начальным вектором. Критерием остановки итерационного метода является падение евклидовой нормы невязки в  $10^6$  раз.

Изначально известно решение дифференциальной задачи (2.1), к примеру  $C = \sin(\pi x)\cos(\pi y)$ . Пользуясь этим решением, аналитически вычисляется правая часть и граничные условия в (2.1), которые затем используются при численном решении задачи. Тензор диффузии имеет следующий вид:

$$D = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix}.$$

Эксперименты проводятся для трех разных значений коэффициента  $\varepsilon$ . Результаты экспериментов записываются в таблицы вида 2.1. В первой колонке таблицы стоит шаг сетки, во второй и третьей — норма ошибки численного решения по отношению к аналитическому, в четвертой — общее число итераций метода сопряженных градиентов.

В отчете должны содержаться таблицы, соответствующие  $\varepsilon = 1, 10, 100$ . Для студентов, исследующих МКР, требуется дополнительно сделать одну таблицу для  $\varepsilon = 100$ , соответствующую аппроксимации краевых условий типа Неймана с первым порядком точности (остальные — со вторым порядком). В МКР и МКЭ для  $C_h$ -

Таблица 2.1: Результаты первого задания.

h	$C_h$ -норма	$L_{2,h}$ -норма	Число итераций
1/32	xx	xx	xx
1/64	xx	xx	xx
1/128	xx	xx	xx

нормы ошибки используется следующее выражение:

$$\|err\|_{C_h} = \max_{i=0,\dots,N, j=0,\dots,N} (C_i^j - C(x_i, y_j)),$$

где  $C(x_i, y_j)$  — значения аналитического решения в точках  $(x_i, y_j)$ . Для МКО используется несколько иное выражение:

$$\|err\|_{C_h} = \max_{i=1,\dots,N, j=1,\dots,N} (C_i^j - \bar{C}_i^j).$$

В последнем выражении  $\bar{C}_i^j$  означает среднее значение аналитического решения на ячейке сетки  $e_i^j$ , которое может быть вычислено с помощью кубатурной формулы, описанной далее. Для всех методов  $L_{2,h}$ -норма ошибки вычисляется по следующей формуле:

$$\begin{aligned} \|err\|_{L_{2,h}} &= \sqrt{\int_{\Omega} (C_h(x, y) - C(x, y))^2 dx} = \\ &= \sqrt{\sum_{k=1,\dots,N, l=1,\dots,N} \int_{e_k^l} (C_h(x, y) - C(x, y))^2 dx}. \end{aligned}$$

В МКР функция  $C_h(x, y)$  предполагается кусочно-билинейной, принимающей значения  $C_i^j$  в вершинах  $V_i^j$ ; в МКО — кусочно-постоянной, принимающей значения  $C_i^j$  на ячейках  $e_i^j$ . Для МКЭ дополнительных построений не требуется, так как решение  $C_h(x, y)$  лежит в пространстве кусочно-билинейных функций. Для вычисления интеграла в последней формуле необходимо использовать кубатурную формулу.

Опишем процесс интегрирования функции или нахождения ее среднего значения с помощью кубатурной формулы. Пусть  $f(x, y)$  — соответствующая функция, а  $T$  — треугольник, на котором требуется

вычислить ее интеграл  $I(f) = \int_T f(x, y) dT$ . Тогда можно вычислить приближенное значение  $I(f)$  в виде

$$S(f) = mes(T) \sum_{i=1}^m \omega_i f(\eta_i), \text{ где } \eta_i = (x_i, y_i) \in T. \quad (2.20)$$

Здесь  $mes(T)$ —площадь треугольника  $T$ , а  $\omega_i$  — веса кубатурной формулы,  $i = 1, \dots, m$ . Для интегрирования по треугольнику  $T$  обычно используются барицентрические координаты, где каждой точке присваиваются 3 координаты. Если вершины треугольника  $T$  заданы координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ , то для произвольной точки  $\eta$  ее декартовы координаты  $(x, y)$  и барицентрические  $(a_1, a_2, a_3)$  связаны соотношениями:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

Барицентрические координаты и веса для одной из кубатурных формул на треугольнике заданы в таблице 2.2. Кубатурная формула сим-

Таблица 2.2: Веса и точки для кубатурной формулы с  $m = 9$ .

$\omega_i$	$\eta_i = (a_1, a_2, a_3)$	кратность
0.20595 05047 60887	0.12494 95032 33232 0.43752 52483 83384 0.43752 52483 83384	3
0.06369 14142 86223	0.79711 26518 60071 0.16540 99273 89841 0.03747 74207 50088	6

метрична относительно пространственных координат, поэтому если встречается узел  $\eta_i = (a_1, a_2, a_3)$ , то обязательно встретятся и все его перестановки. Если все  $a_j (j = 1, \dots, 3)$  различны, то таких узлов в формуле 6; если два значения  $a_j$  совпадают, то таких узлов 3; если в формуле встречается точка  $(1/3, 1/3, 1/3)$ , то лишь один раз (см. столбец кратность). Общее количество слагаемых  $m$  в кубатурной формуле должно равняться суммарной кратности всех входящих в нее узлов.

При интегрировании по квадрату следует разбить квадрат на два треугольника и суммировать посчитанные на этих треугольниках ин-

тегралы. Для нахождения среднего значения  $\bar{f}_T$  функции  $f$  на треугольнике  $T$  нужно интеграл (2.20) разделить на площадь  $mes(T)$ :

$$\bar{f}_T = \sum_{i=1}^m \omega_i f(\eta_i).$$

### 2.1.4 Задание 2: Изучение сходимости итерационных методов с переобуславливателями

Из всех линейных систем, возникающих в первом задании, выбирается та, для которой сходимость метода сопряженных градиентов наиболее медленная. Требуется решить эту систему с помощью программного комплекса SPARSKIT, доступного по адресу <http://www-users.cs.umn.edu/~saad/software/SPARSKIT/sparskit.html>. В этом комплексе присутствует блок ITSOL, содержащий итерационные методы и переобуславливатели. Необходимо протестировать несколько итерационных методов с разными вариантами переобуславливания. Результаты записываются в таблицу следующего вида:

Переобуславливатель⇒	x	x	x
Итерационный метод⇓			
xx	xxx	xxx	xxx
xx	xxx	xxx	xxx

В ячейках таблицы x пишется название итерационного метода, xx — название переобуславливателя, xxx — число итераций метода и общее время решения системы. Критерий остановки итерационного метода — падение евклидовой нормы невязки в  $10^6$  раз.

### 2.1.5 Задание 3: Численное решение задачи о точечном источнике

Рассматривается задача (2.1) в области  $\tilde{\Omega} = [-\infty; \infty] \times [-\infty; \infty]$ . Правая часть  $f = k_1 \delta(x, y)$  — дельта-функция, то есть точечный источник, расположенный в точке  $(0, 0)$ . Тензор диффузии шаровой:  $\mathcal{D} = k_2 I$ , где  $I$  — единичная матрица.

Требуется сначала найти аналитическое решение этой задачи (см. [1]). Далее, бесконечная область  $\tilde{\Omega}$  заменяется на ограниченную  $\Omega = [-200; 200] \times [-200; 200]$ . На границах области накладываются граничные условия типа Дирихле путем использования явного выражения для аналитического решения. С помощью методов из двух

первых заданий находится численное решение на последовательности сеток  $32 \times 32$ ,  $64 \times 64$ ,  $128 \times 128$ . При измерении норм ошибки численного решения следует на некоторое расстояние отступить от точки  $(0, 0)$ , так как при приближении к ней аналитическое решение стремится к бесконечности. Для этого рекомендуется исключить из рассмотрения квадрат  $[-200/32; 200/32] \times [-200/32; 200/32]$ .

### 2.1.6 Содержание отчета

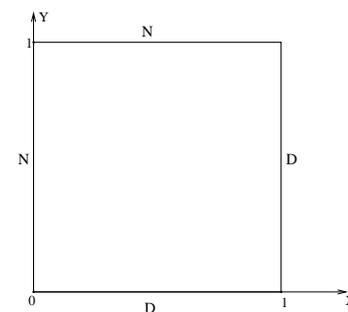
Отчет по практикуму должен содержать следующие пункты (желательно выполнение отчета в  $\text{\LaTeX}$ [4]):

1. Постановка задачи (уравнения, граничные условия).
2. Метод дискретизации уравнения, расчетные формулы.
3. Три таблицы для первого задания с  $\varepsilon = 1, 10, 100$  (для МКР дополнительно таблица с аппроксимацией 1-го порядка граничных условий типа Неймана и  $\varepsilon = 100$ ). Выводы об аппроксимационных свойствах вычислительных схем.
4. Одна таблица для второго задания.
5. График численного решения на самой мелкой сетке и таблица вида 2.1 норм ошибок для третьего задания.

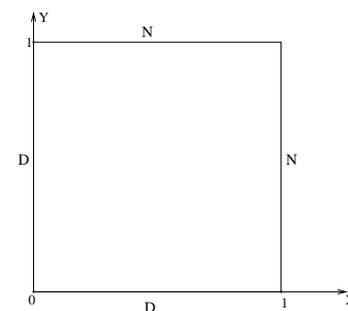
### 2.1.7 Варианты заданий практикума

Ниже формулируются 6 задач для решения с помощью описанных ранее методов дискретизации. Первые три задачи требуется решить в рамках первого задания практикума, три последние — в рамках третьего задания. На рисунках буквами  $D$  и  $N$  отмечены границы области (отрезки), на которых накладываются граничные условия типа Дирихле и Неймана соответственно.

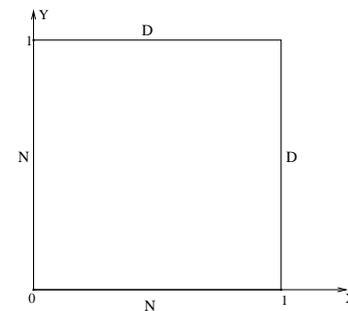
1.  $C = \sin(\pi x)\sin(\pi y)$ ,  $\Omega = (0; 1) \times (0; 1)$ . Граничные условия:



2.  $C = \sin(\pi x)\cos(\pi y)$ ,  $\Omega = (0; 1) \times (0; 1)$ . Граничные условия:



3.  $C = \cos(\pi x)\cos(\pi y)$ ,  $\Omega = (0; 1) \times (0; 1)$ . Граничные условия:



4.  $f = k_1\delta(x, y)$ ,  $D = k_2E$ ,  $k_1 = k_2 = 1$ .

5.  $f = k_1 \delta(x, y)$ ,  $\mathcal{D} = k_2 E$ ,  $k_1 = 2$ ,  $k_2 = 1$ .

6.  $f = k_1 \delta(x, y)$ ,  $\mathcal{D} = k_2 E$ ,  $k_1 = 1$ ,  $k_2 = 2$ .

Для формирования вариантов студенческих заданий вводятся следующие буквенные обозначения для используемых численных методов:

- A — метод конечных разностей
- B — метод конечных элементов
- C — метод конечных объемов
- D — метод сопряженных градиентов (CG)
- E — метод минимальных невязок (GMRES)
- F — метод бисопряженных градиентов (BiCGstab)
- G — переобуславливатели ILU(0), MILU(0), ILU(1)
- H — переобуславливатели ILU(2), ILU(3), ILU(4)
- I — переобуславливатель ILU<sub>t</sub>. Необходимо провести два исследования: фиксируется параметр  $lfil$ , меняется  $droptol$  (попробовать 3 разных значения, см. §1.3.4); фиксируется параметр  $droptol$ , меняется  $lfil$  (также 3 значения, см. §1.3.4).

Каждому студенту дается для программирования один из методов дискретизации, два итерационных метода и одна из групп переобуславливателей, одна задача с известным гладким аналитическим решением и одна задача с точечным источником. Примеры заданий:

1. ADEG14
2. ADFH25
3. AEFI36
4. BDEG14
5. BDFH25
6. BEFI36
7. CDEG14
8. CDFH25

## 9. CEFI36

Дополнительно к выполненному практическому заданию студенту могут быть заданы следующие вопросы на понимание теории практикума:

1. Метод конечных разностей.
2. Метод конечных элементов.
3. Метод конечных объемов.
4. Область применимости итерационных методов CG, BCGstab, GMRES. Их практические достоинства и недостатки.
5. Разреженный строчный формат записи разреженных матриц.
6. Для чего нужен переобуславливатель и как он работает в итерационных методах?
7. Принципы построения переобуславливателей класса ILU(k).
8. Какие шаблоны у матриц жесткости разных методов аппроксимации уравнения диффузии?

Для подготовки к ответам на вопросы рекомендуется использовать настоящее пособие и книгу Й.Саада [10].

## 2.2 Нестационарное уравнение конвекции-диффузии

### 2.2.1 Формулировка задачи

Рассматривается нестационарное уравнение конвекции-диффузии с бездивергентным полем конвективных потоков  $\vec{b}$  в обозначениях раздела 2.1:

$$\frac{\partial C}{\partial t} + \vec{b} \cdot \nabla C - \nabla \cdot \mathcal{D} \nabla C = f(x, y, t) \quad (2.21)$$

в области  $\Omega \in \mathbb{R}^2$  с границей  $\Gamma = \partial\Omega$ . Концентрация  $C = C(x, y, t)$  зависит как от пространственных переменных  $x, y$ , так и от времени  $t$ . Для простоты будем считать, что на всей границе  $\Gamma$  заданы граничные условия типа Дирихле:

$$C|_{\Gamma} = g_D(x, y, t). \quad (2.22)$$

Тензор диффузии, область  $\Omega$  и расчетная сетка такие же, как в §2.1.1. Поле конвективных потоков постоянное во времени и в пространстве, для простоты будем считать  $\vec{b} = (1, 0)$ . Система замыкается начальными условиями:

$$C(x, y, 0) = C^0(x, y) \text{ в } \Omega. \quad (2.23)$$

В следующих параграфах на основе методов дискретизации по пространству из раздела 2.1 построены схемы решения задачи (2.21), (2.22). Так как задача нестационарная, рассматриваются также три варианта дискретизации по времени.

При численном решении задач конвекции-диффузии с преобладающей конвекцией требуется регуляризация схем [6]. Соотношение между конвекцией и диффузией, определяющее необходимость в регуляризации, характеризуется сеточным числом Пекле  $Pe_h$ . Приведем один из способов определения  $Pe_h$ :

$$Pe_h = \frac{h \|\vec{b}\|}{\mathcal{D}_{\vec{b}}}, \quad (2.24)$$

где  $\mathcal{D}_{\vec{b}} = \frac{(\mathcal{D}\vec{b}, \vec{b})}{(\vec{b}, \vec{b})}$  — коэффициент диффузии в направлении конвективного потока. Для регуляризации схем используется противоположная аппроксимация конвективного члена в МКР или МКО и технология SUPG в МКЭ. При выполнении практикума необходимо проанализировать, при каком соотношении диффузии и конвекции нужна стабилизация и при каком — не нужна.

### 2.2.2 Дискретизация по времени

Обозначим с помощью  $L(C)$  пространственный оператор задачи (2.21):

$$L(C) = \vec{b} \cdot \nabla C - \nabla \cdot \mathcal{D} \nabla C.$$

Пусть  $C_h^n$  — численное решение задачи на  $n$ -ом шаге по времени,  $L_h(C_h)$  — соответствующий  $L(C)$  дискретный пространственный оператор,  $f_h^n$  — аппроксимация  $f$  в момент времени  $t_n = n\Delta t$ ,  $\Delta t$  — шаг схемы по времени. Используя эти обозначения, введем следующие три типа аппроксимации уравнения (2.21) по времени:

- Явная схема:

$$\frac{C_h^{n+1} - C_h^n}{\Delta t} + L_h(C_h^n) = f_h^n; \quad (2.25)$$

- Неявная схема:

$$\frac{C_h^{n+1} - C_h^n}{\Delta t} + L_h(C_h^{n+1}) = f_h^{n+1}; \quad (2.26)$$

- Схема Кранка-Николсон:

$$\frac{C_h^{n+1} - C_h^n}{\Delta t} + L_h \left( \frac{C_h^{n+1} + C_h^n}{2} \right) = f_h^{n+1/2}; \quad (2.27)$$

В качестве  $C_h^0$  во всех схемах берется некоторая аппроксимация  $C^0(x, y)$  на заданной сетке.

Студентам при выполнении практикума необходимо экспериментальным путем определить, при каких ограничениях на шаги по времени и по пространству та или иная схема дискретизации по времени устойчива.

### 2.2.3 Дискретизация по пространству

#### Метод конечных разностей

Обозначим  $C_{i,j}^n$  — значение сеточной функции  $C_h$  в точке  $(x_i, y_j)$  в момент времени  $t_n = n\Delta t$ . Тогда общий вид уравнений для явной схемы МКР без регуляризации следующий (с учетом того, что  $\vec{b} = (1, 0)$ ):

$$\begin{aligned} & \frac{C_{i,j}^{n+1} - C_{i,j}^n}{\Delta t} + \frac{C_{i+1,j}^n - C_{i-1,j}^n}{2h} - \\ & - d_x \frac{C_{i+1,j}^n - 2C_{i,j}^n + C_{i-1,j}^n}{h^2} - d_y \frac{C_{i,j+1}^n - 2C_{i,j}^n + C_{i,j-1}^n}{h^2} = f_{i,j}^n, \end{aligned} \quad (2.28)$$

где  $f_{i,j}^n = f(x_i, y_j, t_n)$ . Использование (2.28) возможно лишь при достаточно малом сеточном числе Пекле (2.24). При больших числах Пекле применяется противопотоковая аппроксимация конвективного члена:

$$-d_x \frac{C_{i,j}^{m+1} - C_{i,j}^n}{\Delta t} + \frac{C_{i,j}^n - C_{i-1,j}^n}{h} - d_x \frac{C_{i+1,j}^n - 2C_{i,j}^n + C_{i-1,j}^n}{h^2} - d_y \frac{C_{i,j+1}^n - 2C_{i,j}^n + C_{i,j-1}^n}{h^2} = f_{i,j}^n. \quad (2.29)$$

По отношению к (2.28) в (2.29) наилучшая аппроксимация конвективного члена наблюдается в точке  $(x_{i-1/2}, y_j)$ , смещенной относительно  $(x_i, y_j)$  на  $\frac{h}{2}$  выше по потоку. Перенос в (2.29) (аналогично и в (2.28)) все известные величины в правую часть, получаем явное выражение для  $C_{i,j}^{n+1}$  во внутренних точках расчетной области:

$$C_{i,j}^{n+1} = C_{i,j}^n - \Delta t \left( \frac{C_{i,j}^n - C_{i-1,j}^n}{h} - d_x \frac{C_{i+1,j}^n - 2C_{i,j}^n + C_{i-1,j}^n}{h^2} - d_y \frac{C_{i,j+1}^n - 2C_{i,j}^n + C_{i,j-1}^n}{h^2} \right) + \Delta t f_{i,j}^n \quad \forall i, j : (x_i, y_j) \notin \Gamma.$$

Для точек  $(x_i, y_j) \in \Gamma$  положим  $C_{i,j}^n = g_D(x_i, y_j, t_n)$ . Неявная схема отличается от явной (2.28) (или (2.29)) только заменой  $n$  на  $n+1$  в пространственном операторе и в правой части:

$$\frac{C_{i,j}^{m+1} - C_{i,j}^n}{\Delta t} + \frac{C_{i+1,j}^{m+1} - C_{i-1,j}^{m+1}}{2h} - d_x \frac{C_{i+1,j}^{m+1} - 2C_{i,j}^{m+1} + C_{i-1,j}^{m+1}}{h^2} - d_y \frac{C_{i,j+1}^{m+1} - 2C_{i,j}^{m+1} + C_{i,j-1}^{m+1}}{h^2} = f_{i,j}^{n+1}.$$

Схема Кранка-Николсон имеет следующий вид:

$$\frac{C_{i,j}^{m+1} - C_{i,j}^n}{\Delta t} + \frac{1}{2} \left( \frac{C_{i+1,j}^{m+1} - C_{i-1,j}^{m+1}}{2h} - d_x \frac{C_{i+1,j}^{m+1} - 2C_{i,j}^{m+1} + C_{i-1,j}^{m+1}}{h^2} - d_y \frac{C_{i,j+1}^{m+1} - 2C_{i,j}^{m+1} + C_{i,j-1}^{m+1}}{h^2} \right) + \frac{1}{2} \left( \frac{C_{i+1,j}^n - C_{i-1,j}^n}{2h} - d_x \frac{C_{i+1,j}^n - 2C_{i,j}^n + C_{i-1,j}^n}{h^2} - d_y \frac{C_{i,j+1}^n - 2C_{i,j}^n + C_{i,j-1}^n}{h^2} \right) = f_{i,j}^{n+\frac{1}{2}}.$$

Заметим, что матрицы линейных систем, возникающих в неявных схемах, являются несимметричными, поэтому решать их надо методом минимальных невязок или бисопряженных градиентов.

### Метод конечных элементов

Аналогично §2.1.2 будем искать аппроксимацию  $C_h(x, y, t)$  неизвестной функции  $C(x, y, t)$  в моменты времени  $t = n\Delta t$  ( $n = 1..K$ ) в виде

$$C_h(x, y, t) = \sum_{i=0, j=0}^{i=N, j=N} C_{i,j}^n \phi_i^j(x, y), \quad \text{где } \phi_i^j(x, y) \text{ — базисные кусочно-билинейные функции.}$$

Рассмотрим МКЭ-дискретизацию задачи (2.21)–(2.23) на примере явной схемы. Для двух функций  $f(x, y, t), g(x, y, t) \in L_2(\Omega, T)$  определим  $(f, g) = \int_{\Omega} f g d\Omega$  — скалярное произведение в  $L_2(\Omega)$  при фиксированном  $t$ . Перепишем (2.21) в проекционной форме:

$$\left( \frac{\partial C_h}{\partial t}, \phi_i^j \right) + \left( \vec{b} \cdot \nabla C_h, \phi_i^j \right) - \left( \nabla \cdot \mathcal{D} \nabla C_h, \phi_i^j \right) = \left( f, \phi_i^j \right), \quad \forall i, j : (x_i, y_j) \notin \Gamma. \quad (2.30)$$

Введем дискретизацию по времени  $C_h^n(x, y) = C_h(x, y, t_n)$  и построим явную аппроксимацию вида (2.25):

$$\left( \frac{C_h^{n+1} - C_h^n}{\Delta t}, \phi_i^j \right) + \left( \vec{b} \cdot \nabla C_h^n, \phi_i^j \right) - \left( \nabla \cdot \mathcal{D} \nabla C_h^n, \phi_i^j \right) = \left( f^n, \phi_i^j \right), \quad \forall i, j : (x_i, y_j) \notin \Gamma. \quad (2.31)$$

Здесь и далее  $f^n = f(x, y, t_n)$ .

Запишем для  $C_h^n(x, y) = C_h(x, y, t_n)$  следующее представление, учитывающее наличие граничных условий типа Дирихле:

$$C_h^n = C_0^n + C_{\Gamma}^n, \quad n = 1, \dots, K, \quad \text{где} \quad (2.32)$$

$$C_{\Gamma}^n = \sum_{(x_i, y_j) \in \Gamma} C_{\Gamma, i, j}^n \phi_i^j = \sum_{(x_i, y_j) \in \Gamma} g_D(x_i, y_j, n\Delta t) \phi_i^j,$$

$$C_0^n = \sum_{(x_i, y_j) \notin \Gamma} C_{0, i, j}^n \phi_i^j.$$

Применяя формулу Гаусса-Остроградского для вычисления интеграла  $(\nabla \cdot \mathcal{D} \nabla C_h, \phi_i^j)$ , и учитывая, что  $\phi_i^j(x, y)|_{\Gamma} = 0$  при  $(x_i, y_j) \notin \Gamma$ , преобразуем диффузионный член:

$$\left( \frac{C_h^{n+1} - C_h^n}{\Delta t}, \phi_i^j \right) + \left( \vec{b} \cdot \nabla C_h^n, \phi_i^j \right) + \left( \mathcal{D} \nabla C_h^n, \nabla \phi_i^j \right) = \left( f^n, \phi_i^j \right), \quad \forall i, j : (x_i, y_j) \notin \Gamma. \quad (2.33)$$

Переносим все известные члены в правую часть, получаем

$$\begin{aligned} \left( \frac{C_0^{n+1}}{\Delta t}, \phi_i^j \right) &= \left( \frac{C^n}{\Delta t}, \phi_i^j \right) - \left( \frac{C_{\Gamma}^{n+1}}{\Delta t}, \phi_i^j \right) - \left( \vec{b} \cdot \nabla C^n, \phi_i^j \right) - \\ &- \left( \mathcal{D}\nabla C^n, \nabla \phi_i^j \right) + \left( f^n, \phi_i^j \right) \quad \forall i, j : (x_i, y_j) \notin \Gamma. \end{aligned} \quad (2.34)$$

Подставим (2.32) в последнее выражение:

$$\begin{aligned} \sum_{k,l:(x_k,y_l) \notin \Gamma_D} C_{0k,l}^{n+1} \frac{(\phi_k^l, \phi_i^j)}{\Delta t} &= \sum_{k=0,l=0}^{k=N,l=N} C_{k,l}^n \frac{(\phi_k^l, \phi_i^j)}{\Delta t} - \\ &- \sum_{k,l:(x_k,y_l) \in \Gamma_D} C_{\Gamma k,l}^{n+1} \frac{(\phi_k^l, \phi_i^j)}{\Delta t} - \sum_{k=0,l=0}^{k=N,l=N} C_{k,l}^n \left( \vec{b} \cdot \nabla \phi_k^l, \phi_i^j \right) + \\ &+ \sum_{k=0,l=0}^{k=N,l=N} C_{k,l}^n \left( \mathcal{D}\nabla \phi_k^l, \nabla \phi_i^j \right) + \left( f^n, \phi_i^j \right), \quad \forall i, j : (x_i, y_j) \notin \Gamma. \end{aligned} \quad (2.35)$$

Таким образом, уравнения (2.35) полностью определяют решение  $C_{i,j}^n$  во внутренних точках  $\Omega$ , а в точках на границе решение определяется из граничных условий типа Дирихле:

$$C_{i,j}^n = g_D(x_i, y_j, t_n) \quad \text{для } i, j : (x_i, y_j) \in \Gamma.$$

Аналогичным образом строятся неявная схема и схема Кранка-Николсон для МКЭ, требуется выписать формулу типа (2.31) и повторить для нее шаги (2.32)-(2.35). Вывод расчетных формул для этих схем аппроксимации по времени предлагается студентам в качестве самостоятельного упражнения.

Для стабилизации решения при доминирующей конвекции (достаточно большом сеточном числе Пекле) применяется технология SUPG (Streamline Upwinding Petrov Galerkin) [7]. Она заключается в поэлементном добавлении в левую часть (2.30) скалярного произведения невязки исходного уравнения и  $\vec{b} \cdot \nabla \phi_i^j$  с коэффициентами  $\delta_e$ , определяемыми для каждой ячейки  $e$  сетки:

$$\begin{aligned} &\left( \frac{\partial C_h}{\partial t}, \phi_i^j \right) + \left( \vec{b} \cdot \nabla C_h, \phi_i^j \right) + \left( \mathcal{D}\nabla C_h, \nabla \phi_i^j \right) + \\ &+ \sum_{e \in \varepsilon_h} \delta_e \left( \frac{\partial C_h}{\partial t} + \vec{b} \cdot \nabla C_h - \nabla \mathcal{D}\nabla C_h - f, \vec{b} \cdot \nabla \phi_i^j \right)_e = \left( f, \phi_i^j \right), \end{aligned} \quad (2.36)$$

$$\forall i, j : (x_i, y_j) \notin \Gamma.$$

Следует заметить, что при подстановке в (2.36) вместо  $C_h$  решения дифференциальной задачи добавленный член обращается в ноль. Важно, что для линейных тестовых функций справедливо  $\nabla \mathcal{D}\nabla C_h = 0$ . При  $\delta_e = 0, \forall e \in \varepsilon_h$ , метод (2.36) превращается в простейшую дискретизацию  $P_1$ -МКЭ вида (2.30). Суть SUPG стабилизации состоит в поэлементном добавлении анизотропной искусственной диффузии в направлении потока в зависимости от величины локального сеточного числа Пекле. Возможны различные варианты выбора стабилизационных параметров  $\delta_e$ . В рамках практикума предлагается использовать следующее выражение:

$$\delta_e = \begin{cases} \frac{h}{2\|\vec{b}\|} \left( 1 - \frac{1}{Pe_e} \right), & \text{если } Pe_e \geq 1, \\ 0 & \text{в ином случае.} \end{cases}$$

Во всех схемах МКЭ независимо от наличия стабилизирующих членов требуется решать несимметричную систему линейных уравнений, поэтому рекомендуется применять методы минимальных невязок или бисопряженных градиентов.

### Метод конечных объемов

Дискретизация нестационарной задачи конвекции-диффузии методом конечных объемов отличается от описанной в §2.1.2 введением аппроксимаций производной по времени и конвективного потока. Степени свободы ассоциированы с ячейками сетки, аппроксимация концентрации  $C_h(x, y, t)$  считается постоянной внутри каждой из ячеек сетки. Для аппроксимации диффузионного члена используются те же рассуждения, что и в §2.1.2, на каждом шаге по времени.

Пусть  $e_k^l = [x_{k-1}; x_k] \times [y_{l-1}; y_l]$ ,  $k, l = 1, \dots, N$  — ячейки сетки. Проинтегрируем уравнение (2.21) по ячейке  $e_i^j$ :

$$\int_{e_i^j} \frac{\partial C}{\partial t} dx + \int_{e_i^j} \vec{b} \cdot \nabla C dx - \int_{e_i^j} \nabla \cdot \mathcal{D}\nabla C dx = \int_{e_i^j} f dx.$$

Введем дискретизацию по времени (на примере неявной схемы) и перейдем к аппроксимации  $C_h(x, y, t)$  решения  $C(x, y, t)$ :

$$\int_{e_i^j} \frac{C_h^{n+1} - C_h^n}{\Delta t} dx + \int_{e_i^j} \vec{b} \cdot \nabla C_h^{n+1} dx - \int_{e_i^j} \nabla \cdot \mathcal{D}\nabla C_h^{n+1} dx = \int_{e_i^j} f^{n+1} dx.$$

Так как  $C_h(x, y, t_n)|_{e_i^j} \equiv C_{i,j}^n$ , можем записать:

$$h^2 \frac{C_{i,j}^{n+1} - C_{i,j}^n}{\Delta t} + \int_{e_i^j} \vec{b} \cdot \nabla C_h^{n+1} dx - \int_{e_i^j} \nabla \cdot \mathcal{D} \nabla C_h^{n+1} dx = \int_{e_i^j} f^{n+1} dx.$$

Рассмотрим конвективный член  $\int_{e_i^j} \vec{b} \cdot \nabla C_h^{n+1} dx$ :

$$\int_{e_i^j} \vec{b} \cdot \nabla C_h^{n+1} dx = \vec{b} \cdot \int_{\partial e_i^j} C \vec{n} dl,$$

где  $\vec{n}$  — единичная внешняя нормаль к  $\partial e_i^j$ . Учитывая то, что  $\vec{b} = (1, 0)$ , а ребра сетки параллельны осям координат, заключаем, что

$$\vec{b} \cdot \int_{\partial e_i^j} C^{n+1} \vec{n} dl = -h C_{i-\frac{1}{2},j}^{n+1} + h C_{i+\frac{1}{2},j}^{n+1},$$

где  $C_{i-\frac{1}{2},j}^{n+1}$  и  $C_{i+\frac{1}{2},j}^{n+1}$  — некоторые средние концентрации соответственно на левом и правом вертикальных ребрах ячейки  $e_i^j$ . Функция  $C_h(x, y, t_{n+1})$  терпит разрыв на ребрах сетки, поэтому значения  $C_{i-\frac{1}{2},j}^{n+1}$  и  $C_{i+\frac{1}{2},j}^{n+1}$  требуют отдельного определения. Естественно взять в качестве концентрации на ребрах полусумму значений в соседних ячейках, граничащих по этому ребру:

$$C_{i-\frac{1}{2},j}^{n+1} = \frac{C_{i-1,j}^{n+1} + C_{i,j}^{n+1}}{2}, \quad C_{i+\frac{1}{2},j}^{n+1} = \frac{C_{i+1,j}^{n+1} + C_{i,j}^{n+1}}{2}.$$

В этом случае для конвективного члена получится схема центральных разностей:

$$\int_{e_i^j} \vec{b} \cdot \nabla C_h^{n+1} dx \approx h \frac{C_{i+1,j}^{n+1} - C_{i-1,j}^{n+1}}{2}. \quad (2.37)$$

Вышеописанная аппроксимация конвективного члена оказывается неустойчивой при достаточно больших числах Пекле. Для таких случаев необходима противопотоковая аппроксимация конвективного потока, то есть в качестве значений концентрации на ребрах необходимо брать значение концентрации с той ячейки, которая лежит

выше по потоку. В данном случае

$$C_{i-\frac{1}{2},j}^{n+1} = C_{i-1,j}^{n+1}, \quad C_{i+\frac{1}{2},j}^{n+1} = C_{i,j}^{n+1}.$$

Для конвективного члена имеем:

$$\int_{e_i^j} \vec{b} \cdot \nabla C_h^{n+1} dx \approx h (C_{i,j}^{n+1} - C_{i-1,j}^{n+1}). \quad (2.38)$$

Если одно из вертикальных ребер ячейки  $e_i^j$  лежит на границе типа Дирихле, то в случае аппроксимации конвекции центральными разностями (2.37) значения  $C_{i-\frac{1}{2},j}^{n+1}$  и  $C_{i+\frac{1}{2},j}^{n+1}$  вычисляются как средние от функции  $g_D(x, y, t_{n+1})$  по ребру (рекомендуется использовать формулу Симпсона). Например, пусть  $l_1 = \{x_0\} \times [y_{j-1}, y_j]$  и  $l_2 = \{x_N\} \times [y_{j-1}, y_j]$  — отрезки, лежащие на  $\Gamma$ , тогда:

$$C_{\frac{1}{2},j}^{n+1} = \frac{1}{h} \int_{l_1} g_D(x_0, y, t_{n+1}) dl, \quad C_{N+\frac{1}{2},j}^{n+1} = \frac{1}{h} \int_{l_2} g_D(x_N, y, t_{n+1}) dl.$$

Для противопотоковой аппроксимации конвекции (2.38), если конвективный поток через ребро направлен внутрь ячейки, то концентрация рассчитывается также как среднее от  $g_D(x, y, t_{n+1})$  по отрезку. В противном случае в качестве значения на ребре берется значение внутри ячейки. Это значит, что для тех же отрезков  $l_1$  и  $l_2$  выполнено:

$$C_{\frac{1}{2},j}^{n+1} = \frac{1}{h} \int_{l_1} g_D(x_0, y, t_{n+1}) dl, \quad C_{N+\frac{1}{2},j}^{n+1} = C_{N,j}^{n+1}.$$

Итоговая линейная система для неявной схемы центральных разностей формируется из уравнений вида (для внутренних ячеек):

$$h^2 \frac{C_{i,j}^{n+1} - C_{i,j}^n}{\Delta t} + h \frac{C_{i+1,j}^{n+1} - C_{i-1,j}^{n+1}}{2} - d_x (C_{i+1,j}^{n+1} - 2C_{i,j}^{n+1} + C_{i-1,j}^{n+1}) - d_y (C_{i,j+1}^{n+1} - 2C_{i,j}^{n+1} + C_{i,j-1}^{n+1}) = \int_{e_i^j} f^{n+1} dx. \quad (2.39)$$

Для схемы с противопотоковой аппроксимацией:

$$h^2 \frac{C_{i,j}^{n+1} - C_{i,j}^n}{\Delta t} + h (C_{i,j}^{n+1} - C_{i-1,j}^{n+1}) - d_x (C_{i+1,j}^{n+1} - 2C_{i,j}^{n+1} + C_{i-1,j}^{n+1}) - d_y (C_{i,j+1}^{n+1} - 2C_{i,j}^{n+1} + C_{i,j-1}^{n+1}) = \int_{e_i^j} f^{n+1} dx. \quad (2.40)$$

Рассмотренные схемы приводят к линейным системам с несимметричными матрицами, которые следует решать методом минимальных невязок или бисопряженных градиентов.

Вывод явной схемы и схемы Кранка-Николсон для МКО идентичен выводу схем (2.39),(2.40) и предлагается студентам в качестве самостоятельного упражнения.

### 2.2.4 Численное интегрирование с помощью квадратурных формул Гаусса-Чебышева

В некоторых задачах математической физики аналитические решения могут быть найдены путем применения прямых и обратных преобразований Лапласа, Фурье и др. Однако, в полученных решениях возникают неопределенные интегралы, не поддающиеся аналитическому вычислению, как, например, в решениях (2.43) и (2.44). В этом случае интеграл считается приблизительно с помощью квадратурных формул. Естественным выбором квадратурной формулы является квадратурная формула Гаусса [3], позволяющая при заданном количестве точек получить наиболее высокий алгебраический порядок точности формулы интегрирования. В практикуме будут использоваться формулы Гаусса-Чебышева, рекомендованные в [9] для интегрирования подобных решений.

Пусть необходимо вычислить следующий интеграл:

$$\int_0^t f(\tau) d\tau. \quad (2.41)$$

Сначала заменим переменную интегрирования  $\tau$ , меняющуюся в диапазоне от 0 до  $t$  на переменную  $\eta$ , меняющуюся в диапазоне от -1 до 1. Для этого используем линейное преобразование:

$$\tau = t \frac{\eta + 1}{2}.$$

Получаем

$$\int_0^t f(\tau) d\tau = \frac{t}{2} \int_{-1}^{+1} f\left(t \frac{\eta + 1}{2}\right) d\eta$$

Для некоторой функции  $g(\eta)$  квадратурная формула Гаусса-Чебышева

выглядит так:

$$\int_{-1}^{+1} (1 - \eta^2)^{-\frac{1}{2}} g(\eta) d\eta \approx \sum_{k=1}^n a_k g(\eta_k),$$

где  $\eta_k$  и  $a_k$  — узлы и веса интегрирования, определяемые формулами:

$$\eta_j = \cos\left(\frac{(2j-1)\pi}{2n}\right), \quad a_j = \frac{\pi}{n}, \quad j = 1, \dots, n,$$

$n$ —общее число этих узлов. Положим  $g(\eta) = (1 - \eta^2)^{\frac{1}{2}} f(\eta)$ , тогда интеграл (2.41) может быть приблизительно вычислен следующим образом:

$$\int_0^t f(\tau) d\tau = \frac{\pi t}{2n} \sum_{j=1}^n (1 - \eta_j^2)^{\frac{1}{2}} f\left(t \frac{\eta_j + 1}{2}\right). \quad (2.42)$$

### 2.2.5 Практические задания

Для моделирования студентам предлагаются два типа задач, основанных на уравнении (2.21):

1.  $\tilde{\Omega} = (0; +\infty) \times (-\infty; +\infty)$  — полуплоскость  $x > 0$ .

Граничные условия на границе  $x = 0$ :

$$C(0, y, t) = \begin{cases} c_0 & \text{при } |y| < a, \\ 0 & \text{при } |y| > a, \end{cases}$$

где  $a = 10$ . Начальные условия  $C(x, y, 0) \equiv 0$  в  $\tilde{\Omega}$ .

2.  $\tilde{\Omega} = (0; +\infty) \times (-\infty; +\infty)$  — полуплоскость  $x > 0$ .

Граничные условия на границе  $x = 0$ :

$$C(0, y, t) = \begin{cases} c_0 & \text{при } y < 0, \\ 0 & \text{при } y > 0. \end{cases}$$

Начальные условия  $C(x, y, 0) \equiv 0$  в  $\tilde{\Omega}$ .

Тензор диффузии для этих задач предполагается диагональным и анизотропным:

$$\mathcal{D} = \begin{pmatrix} d_x & 0 \\ 0 & d_y \end{pmatrix},$$

вектор конвективных потоков  $\vec{b} = (1; 0)$ ,  $c_0 = 1$ .

Известны аналитические решения [8] задач 1 и 2 для случая  $f(x, y, t) \equiv 0$ , т.е. нулевой правой части в уравнении (2.21):

$$1. C(x, y, t) = \frac{xc_0}{(16\pi d_x)^{\frac{1}{2}}} \int_0^t \tau^{-\frac{3}{2}} \left\{ \operatorname{erf} \left[ \frac{a+y}{(4d_y\tau)^{\frac{1}{2}}} \right] + \operatorname{erf} \left[ \frac{a-y}{(4d_y\tau)^{\frac{1}{2}}} \right] \right\} \cdot \exp \left[ - \left( \frac{x-\tau}{(4d_x\tau)^{\frac{1}{2}}} \right)^2 \right] d\tau. \quad (2.43)$$

$$2. C(x, y, t) = \frac{xc_0}{(16\pi d_x)^{\frac{1}{2}}} \int_0^t \tau^{-\frac{3}{2}} \operatorname{erfc} \left[ \frac{y}{(4d_y\tau)^{\frac{1}{2}}} \right] \cdot \exp \left[ - \left( \frac{x-\tau}{(4d_x\tau)^{\frac{1}{2}}} \right)^2 \right] d\tau, \quad (2.44)$$

Здесь  $\operatorname{erf}(x)$ ,  $\operatorname{erfc}(x)$  — Гауссовская функция ошибки и дополнительная функция ошибки соответственно. Дадим их определение:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad \operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt. \quad (2.45)$$

В обеих задачах требуется сначала заменить расчетную область на ограниченную,  $\Omega = [0; 200] \times [-100; 100]$ . В **первом задании** практикума студентам требуется приближенно решить эти задачи с помощью МКР, МКЭ или МКО. Для упрощения **первого задания** введем некоторые допущения, позволяющие с меньшими трудозатратами получить качественную оценку решения. Будем считать искусственно введенные границы достаточно удаленными и наложим на них нулевые граничные условия типа Дирихле. Если диффузия в направлении оси  $y$  достаточно мала, то условия на отрезках границы, параллельных оси  $x$ , будут мало влиять на зону вблизи прямой  $y = 0$ , решение в которой представляет наибольший интерес. При достаточно малом времени расчета  $T$  (к примеру,  $T < 50$ ) и значениях коэффициентов диффузии, используемых в настоящем задании, однородные условия на границе  $x = 200$  мало отличаются от реальных и не внесут ощутимой ошибки в решение. Тем самым, можно получить приблизительную картину распространения концентрации с течением времени.

Необходимо в рамках этого задания также провести исследование вопроса, при каком соотношении конвекции и диффузии требуется

регуляризация схем. Опыты должны проводиться на последовательности сеток  $200 \times 200$ ,  $400 \times 400$ ,  $800 \times 800$ . Коэффициент  $d_y = 10^{-3}$ , фиксирован. Коэффициент  $d_x$  в опытах меняется и принимает значения от  $10^{-4}$  до 1. На каждой сетке необходимо найти то его значение, ниже которого необходима регуляризация.

**Второе задание** практикума состоит в получении аппроксимации аналитических решений (2.43) и (2.44) с помощью квадратур Гаусса-Чебышева, а **третье задание** — в исследовании сходимости схем на негладких решениях задач 1 и 2, описанных выше. При этом граничные условия должны быть интерполяцией аналитического решения. Алгоритм аппроксимации аналитических решений (2.43), (2.44) описан в параграфе 2.2.4. Для нахождения значений аналитических решений по формулам (2.43), (2.44) интеграл по переменной  $\tau$  вычисляется с помощью (2.42). Интегралы в функциях ошибок (2.45) вычислять не требуется, функции ошибок обычно содержатся в общедоступных библиотеках для разных языков программирования.

## 2.2.6 Содержание отчета

Отчет студентов по практикуму состоит из письменного отчета, и показа примеров работы написанной программы преподавателю. Напомним, что практикум содержит 3 задания:

1. Программирование вычислительной схемы для решения задачи конвекции-диффузии. Решение задачи с разрывными граничными условиями типа Дирихле. Экспериментальный анализ устойчивости схемы в зависимости от шага по времени и от сеточного числа Пекле.
2. Вычисление аналитического решения с помощью квадратур Гаусса-Чебышева.
3. Моделирование задач с негладким аналитическим решением, граничные условия вычисляются с помощью квадратур Гаусса-Чебышева. Необходимо проанализировать сходимость схем в  $C$ - и  $L_2$ -нормах. Шаг по времени в опытах меняется пропорционально шагу по пространству.

Письменный отчет по практикуму должен содержать следующие пункты (желательно выполнение отчета в  $\text{\LaTeX}$ [4]):

1. Постановка задачи (уравнения, граничные условия).

2. Метод дискретизации уравнения, расчетные формулы.
3. Для первого задания (на всех графиках показать только область  $[0; 100] \times [-20; 20]$ ):
  - графики решения в момент времени  $T = 50$  для двух вариантов коэффициента диффузии  $d_x = 1$ ,  $d_x = 10^{-4}$  на сетках  $400 \times 400$  и  $800 \times 800$ . Мотивированные выводы об устойчивости схем по времени и по пространству (по сеточному числу Пекле);
  - графики решений с  $d_x = 1$  и  $d_x = 10^{-4}$  в моменты времени от 10 до 50 с шагом 10 на сетке  $800 \times 800$ .
4. График полученного аналитического решения для второго задания.
5. Для третьего задания — таблицы вида 2.1 для двух вариантов коэффициента диффузии в направлении оси X:  $d_x = 1$ ,  $d_x = 10^{-4}$ . Решение сравнивается с аналитическим в момент времени  $T = 50$ . В таблице также должен присутствовать столбец с шагом по времени.

Дополнительно к выполненному практическому заданию студенту могут быть заданы следующие вопросы на понимание теории практикума:

1. Что такое сеточное число Пекле, и что оно характеризует?
2. Опишите используемые в практикуме схемы дискретизации по времени, их достоинства и недостатки.
3. Опишите применение МКР для дискретизации уравнения конвекции-диффузии.
4. Опишите применение МКЭ для дискретизации уравнения конвекции-диффузии.
5. Опишите применение МКО для дискретизации уравнения конвекции-диффузии.
6. Какие условия определяют необходимость введения регуляризации в схемах?
7. Опишите алгоритм приближенного интегрирования функции с помощью кубатурных формул Гаусса-Чебышева.

### 2.2.7 Варианты заданий практикума

Каждому студенту дается для программирования одна из 9 схем дискретизации (3 по времени  $\times$  3 по пространству) и одна задача из (2.43), (2.44) для выполнения всех трех заданий. Введем следующие обозначения:

- A — метод конечных разностей
- B — метод конечных элементов
- C — метод конечных объемов
- D — явная схема
- E — неявная схема
- F — схема Кранка-Николсон
- 1 — задача (2.43)
- 2 — задача (2.44)

В таблице 2.3 приведены варианты комбинаций схем дискретизации и модельных задач.

Таблица 2.3: Примеры вариантов заданий.

AD1	AE1	AF1	AD2	AE2	AF2
BD1	BE1	BF1	BD2	BE2	BF2
CD1	CE1	CF1	CD2	CE2	CF2

# Литература

- [1] *Владимиров В.С.* Уравнения математической физики. — М.: Наука, 1981.
- [2] *Деммель Дж.* Вычислительная линейная алгебра. Теория и приложения. — М.: Мир, 2001.
- [3] *Лебедев В.И.* Функциональный анализ и вычислительная математика. — М.: Наука, 2002.
- [4] *Львовский С.М.* Набор и верстка в пакете L<sup>A</sup>T<sub>E</sub>X. — М.: Космосинформ, 1995.
- [5] *Ольшанский М.А., Василевский Ю.В.* Краткий курс по многосеточным методам и методам декомпозиции области. — М.: МАКС Пресс, 2007.
- [6] *Самарский А.А., Вабичевич П.Н.* Численные методы решения задач конвекции-диффузии. — М.: Эдиториал УРСС, 1999.
- [7] *Hughes T.J.R. and Brooks A.N.* A multidimensional upwind scheme with no crosswind diffusion //Finite Element Methods for Convection Dominated Flows. 1979. V. 34, P. 19–35.
- [8] *Leij F.J. and Dane J.H.* Analytical solutions of the one-dimensional advection equation and two- or three-dimensional dispersion equation //Water Resources Research 1990. V. 26, P. 1475–1482.
- [9] *Leij F.J., Dane J.H. and van Genuchten M.* Mathematical analysis of one-dimensional solute transport in a layered soil profile // Soil. Sci. Soc. Am. 1991. V. 55, P. 944–953.
- [10] *Saad Y.* Iterative methods for sparse linear systems, Second Edition. — Philadelphia, PA: SIAM, 2000.