

Вычислительная линейная алгебра с примерами на MATLAB

- *Теоретические основы
численных методов*
- *Прямые методы решения
систем линейных
алгебраических уравнений*
- *Итерационные методы
решения систем линейных
алгебраических уравнений*
- *Собственные значения
и векторы матриц*
- *Примеры на MATLAB*

В. И. Горбаченко

Вычислительная линейная алгебра с примерами на MATLAB

Рекомендовано Учебно-методическим объединением по образованию в области прикладной информатики в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности 080801 «Прикладная информатика (по областям)» и другим экономическим специальностям

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.06+512.64(075.8)
ББК 32.973.26–018.2/22.143я73
Г67

Горбаченко В. И.

Г67 Вычислительная линейная алгебра с примерами на MATLAB. — СПб.: БХВ-Петербург, 2011. — 320 с.: ил. — (Учебное пособие)

ISBN 978-5-9775-0725-7

Излагаются теоретические основы численных методов, включая теорию погрешностей, особенности машинной арифметики, корректность и обусловленность вычислительных задач; современные прямые и итерационные методы решения больших систем линейных алгебраических уравнений. Основное внимание уделено современным итерационным методам на основе подпространств Крылова. Рассмотрено решение частичной и полной проблемы собственных значений, в том числе для больших разреженных матриц. Для основных вычислительных методов приведены реализации с использованием программ, разработанных автором, а также соответствующие функции системы MATLAB.

Для студентов и преподавателей высших учебных заведений

УДК 681.3.06+512.64(075.8)
ББК 32.973.26–018.2/22.143я73

Рецензенты:

О. Э. Яремко, канд. физ.-мат. наук, проф., завкафедрой математического анализа Пензенского государственного педагогического университета им. В. Г. Белинского;

Кафедра информационно-вычислительных систем Пензенского государственного университета, завкафедрой д-р техн. наук, проф. Ю. Н. Косников.

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Юрий Якубович</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 26.07.11.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 25,8.

Тираж 1000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953 Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12.

Оглавление

Введение.....	1
Глава 1. Теоретические основы численных методов	5
1.1. Математическое моделирование и вычислительный эксперимент	5
1.2. Погрешности вычислений.....	7
1.2.1. Источники погрешностей вычислений	7
1.2.2. Приближенные числа. Абсолютная и относительная погрешность	9
1.2.3. Особенности машинной арифметики.....	11
1.2.4. Трансформированные погрешности арифметических операций	15
1.2.5. Трансформированные погрешности вычисления функций	18
1.3. Свойства вычислительных задач и алгоритмов.....	19
1.3.1. Корректность вычислительной задачи.....	19
1.3.2. Обусловленность вычислительной задачи.....	22
1.3.3. Требования, предъявляемые к численному методу.....	28
1.4. Вопросы и задания для самопроверки	29
Библиографический список к главе 1	30
Глава 2. Прямые методы решения систем линейных алгебраических уравнений	33
2.1. Системы линейных алгебраических уравнений. Матрицы и их свойства	33
2.2. Метод Гаусса.....	37
2.3. Метод прогонки	41
2.4. Метод LU-разложения.....	43
2.5. Метод Холецкого	45
2.6. Метод LDL ^T -разложения.....	48
2.7. Метод QR-разложения.....	50
2.7.1. Метод вращений.....	50
2.7.2. Метод отражений	57
2.7.3. Приведение матриц к форме Хессенберга.....	62
2.8. Вычисление определителей и обращение матриц.....	63
2.9. Оценка погрешностей решений, получаемых прямыми методами	65
2.10. Решение систем с прямоугольными матрицами	66
2.10.1. Постановка задачи наименьших квадратов. Нормальные уравнения	66

2.10.2. Использование QR-разложения для решения задачи наименьших квадратов	68
2.10.3. Использование сингулярного разложения	71
2.11. Реализация прямых методов в MATLAB	74
2.11.1. Некоторые функции матричных вычислений и реализации прямых методов в MATLAB	74
2.11.2. Хранение и обработка разреженных матриц	87
2.11.3. Примеры программ	98
2.12. Задания для лабораторных и самостоятельных работ	109
2.13. Вопросы и задания для самопроверки	116
Библиографический список к главе 2	117

Глава 3. Итерационные методы решения систем линейных алгебраических уравнений 119

3.1. Дискретизация задач математической физики и особенности решения систем алгебраических уравнений Equation Chapter 3 Section 1	119
3.2. Основные теоретические положения итерационных методов	125
3.3. Метод Рундсона	130
3.4. Методы простой итерации и Якоби	132
3.5. Методы Зейделя и последовательной верхней релаксации	135
3.5.1. Метод Зейделя	135
3.5.2. Метод последовательной верхней релаксации	136
3.6. Блочные и асинхронные итерационные методы	140
3.7. Методы спуска	142
3.8. Предобуславливатели	149
3.9. Методы подпространств Крылова	151
3.9.1. Краткие сведения из функционального анализа и линейной алгебры	151
3.9.2. Проекционные методы	158
3.9.3. Подпространства Крылова	163
3.9.4. Методы ортогонализации	164
3.9.5. Метод сопряженных градиентов	174
3.9.6. Методы подпространств Крылова для несимметричных и знаконеопределенных задач	184
3.10. Итерационные методы решения нормальных систем линейных алгебраических уравнений	197
3.11. Итоговые замечания	202
3.12. Реализация итерационных методов в MATLAB	203
3.12.1. Некоторые функции реализации итерационных методов в MATLAB	203
3.12.2. Примеры программ	212
3.13. Задания для лабораторных и самостоятельных работ	232
3.14. Вопросы и задания для самопроверки	234
Библиографический список к главе 3	235

Глава 4. Вычисление собственных значений и собственных векторов матриц	239
4.1. Собственные пары матриц и их свойства	239
4.2. Решение частичной проблемы собственных значений	245
й метод	245
4.2.2. Метод скалярных произведений	250
4.2.3. Метод обратных итераций. Обратные итерации со сдвигами	252
4.2.4. Градиентный метод решения частичной проблемы собственных значений	254
4.3. Решение полной проблемы собственных значений	255
4.3.1. QR-алгоритм решения полной проблемы собственных значений	255
4.3.2. Методы для симметричных задач на собственные значения	263
4.3.3. Использование QR-алгоритма для вычисления собственных векторов	265
4.4. Вычисление сингулярного разложения	267
4.4.1. Приведение матрицы к двухдиагональной форме	267
4.4.2. Сингулярное разложение двухдиагональной матрицы	270
4.5. Вычисление собственных значений больших разреженных матриц	275
4.5.1. Метод одновременных итераций	275
4.5.2. Метод Арнольди	278
4.5.3. Метод Ланцоша	281
4.6. Обобщенная задача на собственные значения	283
4.6.1. Основы теории	283
4.6.2. Решение обобщенной задачи на собственные значения	284
4.7. Вычисление собственных пар в MATLAB	288
4.7.1. Стандартные функции MATLAB	288
4.7.2. Примеры программ	296
4.8. Задания для лабораторных и самостоятельных работ	301
4.9. Вопросы и задания для самопроверки	302
Библиографический список к главе 4	303
Литература по вычислительной математике	305
Литература по по системе MATLAB	309
Предметный указатель	311

Введение

В основе большинства научных применений компьютеров лежит вычислительная линейная алгебра (матричные вычисления). Множество вычислительных задач сводится к решению систем линейных алгебраических уравнений и к вычислению собственных чисел и векторов матриц.

Целью данного пособия является обучение теоретическим основам и практическим навыкам решения вычислительных задач и проведения вычислительного эксперимента. Причем *упор делается на прикладные аспекты и на алгоритмическую сторону* численных методов.

Особенностями данного пособия являются:

- доступность изложения для студентов и специалистов, изучавших математику в объеме инженерных специальностей;
- особое внимание к теоретическим основам численных методов, включая теорию погрешностей, особенности машинной арифметики, корректность и обусловленность вычислительных задач;
- изложение современных прямых и итерационных методов решения больших систем линейных алгебраических уравнений, включая сингулярное разложение, обоснование преимуществ применения итерационных методов для решения больших разреженных систем, описание современных итерационных методов, прежде всего, методов на основе подпространств Крылова;
- подробное рассмотрение современных методов решения частичной и полной проблемы собственных значений, в том числе для больших разреженных матриц;
- детальное описание использования основных функций вычислительной линейной алгебры, реализованных в системе MATLAB, многим из которых в отечественной учебной литературе уделено мало внимания;
- наличие заданий для лабораторных занятий и самостоятельной работы, а также разработанной автором библиотеки учебных программ, что позволяет организовать эффективное изучение численных методов студентами с различным уровнем подготовки.

Данное пособие содержит следующие главы:

1. Теоретические основы численных методов.
2. Прямые методы решения систем линейных алгебраических уравнений.
3. Итерационные методы решения систем линейных алгебраических уравнений.
4. Вычисление собственных значений и собственных векторов матриц.

Каждая глава содержит список использованной литературы — это источники, которыми пользовался автор при написании данной главы, и специальная литература, более подробно освещающая те или иные главы. Кроме того, приведен общий список рекомендованной литературы по вычислительной математике. Списки литературы не претендуют на полноту, а лишь отражают вкусы и научные интересы автора. Из общего списка книг по вычислительной математике следует обратить внимание на книги Уоткинса Д., Амосова А. А. и др., Вержбицкого В. М.

В настоящее время на рынке доступно множество коммерческих систем компьютерной математики, наиболее известными из которых являются MATLAB, MathCAD, Mathematica, Maple. Из открытых систем компьютерной математики известна система профессионального уровня Scilab (<http://www.scilab.org/>), которая позиционируется в качестве аналога пакета MATLAB. Доступность и удобный интерфейс современных систем компьютерной математики создают у неопытного пользователя иллюзию легкости решения вычислительных задач и не критичное отношение к получаемым результатам. Однако грамотная постановка задачи, рациональный выбор метода решения и правильная интерпретация результатов требуют серьезного знания численных методов. Очень важной проблемой является оценка погрешности решения. Поэтому современные инструменты могут быть эффективно использованы и дать достоверные результаты только в руках профессионала. Большинство пользователей забывают, что системы компьютерной математики, являясь сложными программными продуктами, при всей тщательности тестирования фирмами-разработчиками, не гарантированы от ошибок¹.

Пособие ориентировано на использование системы MATLAB фирмы MathWorks Inc. (<http://www.mathworks.com/>), фактически ставшей мировым стандартом университетского образования по численным методам. Владение MATLAB сегодня является одним из показателей уровня профессиональной подготовки выпускников естественно-научных и технических факультетов университетов в области информационных технологий.

Название MATLAB является сокращением от Matrix Laboratory. В MATLAB основным типом данных является матрица, что делает эту систему особенно удобной для решения задач вычислительной линейной алгебры. Система MATLAB представляет собой высокоуровневый язык и интерактивную среду для разработки алгоритмов, численных расчетов, визуализации и анализа данных. Используя MATLAB, можно решать вычислительные задачи гораздо быстрее, чем с использованием традиционных языков программирования, например C++ или Pascal.

Каждая глава пособия (кроме первой) содержит описание реализации соответствующих методов в виде функций MATLAB. Кроме того, приводятся исходные тексты примеров программ на MATLAB, реализующих основные численные методы,

¹ Петров Ю. П. Обеспечение достоверности и надежности компьютерных расчетов — СПб.: БХВ-Петербург, 2008. — 160 с.

с минимальным использованием стандартных функций. Таким образом, использование пособия возможно на нескольких уровнях:

- ознакомление с основами численных методов без привязки к конкретному языку программирования;
- использование стандартных функций для решения вычислительных задач, что предъявляет минимальные требования к знаниям в области программирования, но требует грамотного выбора функций;
- разработка полноценных программ на языке MATLAB, что требует понимания алгоритмов решения и навыков программирования, но при этом позволяет реализовать любой алгоритм.

Пособие также содержит контрольные вопросы и задания для самостоятельной работы. Исходные тексты программ могут быть бесплатно высланы по электронной почте по запросу. Запрос можно сделать по адресу **gorvi_edu@mail.ru**. По этому же адресу автор ожидает предложений и замечаний по пособию.

Данное пособие не содержит описания системы MATLAB. Необходимую информацию можно получить из книг, приведенных в списке литературы по системе MATLAB. Обратите особое внимание на книги И. Е. Ануфриева, А. Б. Смирнова, Е. Н. Смирновой [2] и Б. Р. Ханта, Р. Л. Липсмана, Д. М. Розенберга [23]. Масса полезной информации содержится на сайте **http://matlab.exponenta.ru** Консультационного Центра MATLAB российской компании Softline (**http://www.softline.ru/**), официального дистрибьютора системы в России.

Глава 1

Теоретические основы численных методов

1.1. Математическое моделирование и вычислительный эксперимент

В современной науке и технике важную роль играет *математическое моделирование* [1—2]¹, заменяющее эксперименты с реальными объектами экспериментами с их математическими моделями. Возник даже термин "*вычислительный эксперимент*" [1—3]. Вычислительный эксперимент имеет ряд преимуществ по сравнению с натурным экспериментом:

- экономичность, так как не тратятся ресурсы реальной системы;
- возможность моделирования гипотетических, то есть не реализованных в природе объектов (прежде всего на разных этапах проектирования);
- доступность тестирования режимов, опасных или труднопроизводимых в натуре (критический режим ядерного реактора, работа системы противоракетной обороны, природные и техногенные катастрофы);
- возможность изменения масштаба времени;
- простота многоаспектного анализа;
- большая прогностическая сила вследствие возможности выявления общих закономерностей.

Следует отметить, что моделирование сложных объектов, например атомных, космических и многих других, требует проведения колоссальных объемов вычислений. Например, для решения многих прикладных задач аэродинамики и ядерной физики требуется выполнение более 10^{13} арифметических операций [4].

Для практических задач довольно редко удается найти аналитическое решение уравнений, составляющих математическую модель явления. Поэтому приходится применять численные методы. Сущность применения численных методов рассмотрим на схеме вычислительного эксперимента [3, 5—6], показанной на рис. 1.1.

¹ Ссылки в тексте относятся к библиографическому списку в конце главы. — *Ред.*

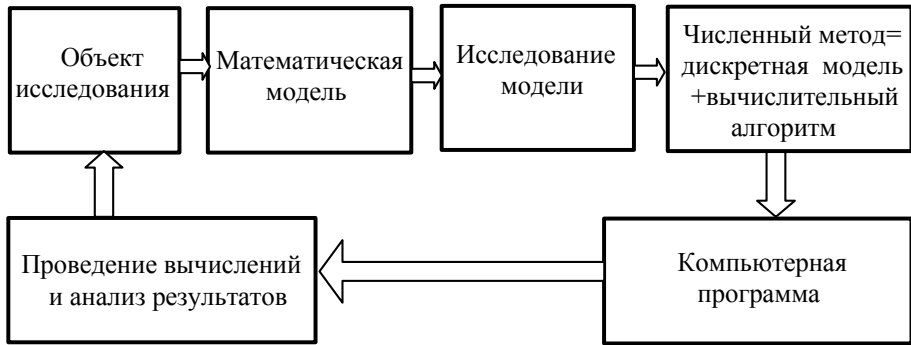


Рис. 1.1. Схема вычислительного эксперимента

Основу вычислительного эксперимента составляет триада: *модель — метод (алгоритм) — программа*. Сначала строится с некоторыми допущениями *математическая модель объекта*. Первоначально строится относительно простая, но достаточно полная с точки зрения экспериментальных данных модель. В ходе вычислительного эксперимента модель уточняется и дополняется. Поэтому можно говорить о *наборе математических моделей*, каждая из которых с различной точностью описывает объект или различные свойства объекта. Построение математических моделей выходит за рамки нашего курса. Более подробную информацию о построении математических моделей можно найти в книгах [2, 3].

После выбора или построения математической модели средствами прикладной математики проводится *предварительное качественное исследование модели*. Качественное исследование начинается с *приведения задачи к безразмерному виду* [6, 7]. Суть приведения задачи к безразмерному виду состоит в выделении характерных значений (величин) и масштабировании всех величин задачи. Приведение задачи к безразмерному виду сокращает общее число параметров математической модели, что упрощает анализ. Кроме того, в безразмерной задаче можно сравнивать параметры и упрощать задачу, отбрасывая малые параметры. Наконец, приведение к безразмерному виду уменьшает влияние ошибок округления при вычислениях на компьютере, так как все безразмерные величины изменяются примерно от -1 до $+1$. Приведение задачи к безразмерному виду представляет достаточно сложную задачу, так как основано на теории подобия [7] и требует знания особенностей процессов, протекающих в объекте исследования. В частности масштабы разных величин оказываются взаимосвязанными. Математическая модель часто оказывается очень сложной для качественного исследования. В этом случае для качественного исследования строят *модельные (упрощенные) задачи*. После приведения к безразмерному виду рассматриваются *существование и единственность* решения, *основные свойства решения*, влияние различных параметров задачи на решение, *устойчивость* решения относительно малых возмущений входных данных и другие вопросы. Качественное исследование модели зачастую представляет собой сложную самостоятельную задачу. Однако для многих прикладных задач, например для обыкновенных дифференциальных уравнений, качественный анализ уже проведен.

Затем необходимо решить систему уравнений, представляющую собой математическую модель объекта. Как уже говорилось, обычно приходится применять численные методы. Под *численным методом* понимается совокупность *дискретной модели*, реализуемой на компьютере, и *вычислительного алгоритма*, позволяющего решить дискретизированную задачу. Например, дискретной моделью вычисления определенного интеграла является вычисление суммы площадей прямоугольников, аппроксимирующих площадь криволинейной трапеции, являющейся геометрической интерпретацией задачи. Для реализации численного метода необходимо разработать *программу* на одном из языков программирования или применить готовый пакет прикладных программ. В настоящее время разработано несколько пакетов прикладных программ, или систем компьютерной математики, таких как MathCAD, MATLAB, Maple, Mathematica и других, которые позволяют решать большинство практически встречающихся задач. Однако грамотная постановка задачи, рациональный выбор метода решения, оценка погрешности и правильная интерпретация результатов требуют серьезных знаний численных методов. После отладки программы производятся вычисления на компьютере (обычно требуется провести много вариантов вычислений, для чего необходимо планировать вычислительный эксперимент) и анализ результатов. После получения результатов исследуется соответствие результатов вычислительного эксперимента процессу функционирования реального объекта (проверяется *адекватность модели*), и при необходимости уточняются компоненты схемы вычислительного эксперимента (рис. 1.1) до получения удовлетворительных результатов.

Можно выделить три основных типа вычислительного эксперимента [6]:

- *поисковый*, заключающийся в исследовании различных объектов и процессов посредством математических моделей;
- *оптимизационный*, направленный, например, на подбор оптимальных характеристик конструкций, технологических процессов;
- *диагностический*, когда по результатам натуральных экспериментов определяются свойства объектов или явлений (при этом решаются *обратные задачи*).

Одной и той же математической модели можно поставить в соответствие множество дискретных моделей и вычислительных алгоритмов, то есть численных методов. Для выбора подходящих численных методов надо знать их основные свойства. Рассмотрение свойств численных методов начнем с анализа погрешностей вычислений.

1.2. Погрешности вычислений

1.2.1. Источники погрешностей вычислений

Исследование реального объекта методом вычислительного эксперимента носит приближенный характер. На каждом этапе вычислительного эксперимента (рис. 1.1) возникают погрешности. Можно выделить 5 источников погрешностей [5, 8—11]:

- погрешность математической модели;
- погрешность дискретизации;

- трансформированная погрешность (погрешность искажения);
- методическая погрешность;
- погрешность округления.

Неустраняемая по отношению к численному методу *погрешность математической модели* обусловлена недостаточным знанием или чрезмерным упрощением моделируемых объектов и явлений. То есть причиной этого вида погрешности является неадекватность математической модели. Адекватность математической модели может быть оценена путем сравнения реального процесса и реализации на компьютере модели процесса. Это очень сложная задача, так как реализация модели на компьютере включает много составляющих погрешностей, исследование реального процесса также происходит с погрешностью. Оценка погрешности математической модели выходит за рамки данного курса.

Погрешность дискретизации — это погрешность от замены непрерывной математической модели дискретной моделью. Примером дискретизации является замена дифференциального уравнения системой алгебраических уравнений. Решение системы алгебраических уравнений отличается от решения дифференциального уравнения, что и составляет погрешность дискретизации. Обычно дискретная модель зависит от некоторых параметров, изменением которых теоретически можно понизить погрешность дискретизации. В рассмотренном примере, выбирая более мелкий шаг дискретизации (и увеличивая порядок системы алгебраических уравнений), можно уменьшить погрешность дискретизации.

Трансформированная погрешность (погрешность искажения) [8] — это погрешность, возникающая за счет *погрешности исходных данных*. Исходные данные, как правило, являются результатом измерений некоторых величин, естественно, эти измерения производятся с некоторой погрешностью. Кроме того, из-за ограниченной разрядности компьютеров возникает *погрешность представления исходных данных* (погрешность округления исходных данных). Многие задачи являются весьма чувствительными к погрешностям исходных данных и к погрешностям, возникающим в ходе реализации алгоритма. В этом случае говорят, что задача является *плохо обусловленной*. Для плохо обусловленной задачи погрешности исходных данных, даже при отсутствии погрешностей вычислений, могут приводить к значительным искажениям результата.

Методическая погрешность возникает из-за применения приближенных алгоритмов. Например, решая систему алгебраических уравнений итерационным методом, теоретически можно получить точное решение лишь при бесконечно большом числе итераций. Однако число итераций приходится ограничивать, что приводит к погрешности метода.

Погрешность округления возникает из-за того, что все вычисления выполняются с ограниченным числом значащих цифр, то есть производится округление чисел. Погрешности округления накапливаются и при плохой обусловленности задачи могут привести к большим отклонениям результата.

Как выбирать допустимые погрешности? Известны различные рекомендации. В [5] предлагается обеспечивать одинаковый порядок всех погрешностей. То есть

не следует очень точно решать задачу с неточными исходными данными. Другими словами, погрешность решения *всей* задачи (всей последовательности действий вычислительного эксперимента) должна быть соизмерима с погрешностью исходных данных.

В [12] рекомендуется обеспечивать методическую погрешность в 2—10 раз меньшую, чем погрешность модели, а погрешность округления — на порядок меньше погрешности метода.

1.2.2. Приближенные числа.

Абсолютная и относительная погрешность

Мерой точности вычислений являются погрешности. Пусть a — приближенное значение точного числа A . *Погрешностью*, или *ошибкой* Δa приближенного числа a называется разность

$$\Delta a = a - A.$$

В качестве меры погрешности используют абсолютную и относительную погрешность. *Абсолютная погрешность* приближенного числа a определяется формулой

$$\Delta a = |a - A|.$$

Так как точное значение числа A в большинстве случаев неизвестно, то используется оценка *предельной абсолютной погрешности* Δ_a , называемая также *границей абсолютной погрешности*:

$$\Delta a = |a - A| \leq \Delta_a.$$

Относительная погрешность определяется формулой

$$\delta a = \frac{|a - A|}{|A|} = \frac{\Delta a}{|A|}.$$

Относительная погрешность часто выражается в процентах.

Предельной относительной погрешностью δ_a приближенного числа a называется число, заведомо не меньшее относительной погрешности этого числа

$$\delta a \leq \delta_a.$$

Так как значение точного числа A неизвестно, то часто пользуются приближенными оценками предельных погрешностей

$$\delta_a \approx \frac{\Delta_a}{|a|}, \quad \Delta_a \approx |a| \delta_a.$$

Очень часто используется термин "точность решения". Хотя точность решения противоположна по смыслу погрешности решения, для измерения точности используются те же характеристики, что и для измерения погрешности. Когда говорят, что точность решения равна ε , то это означает, что принятая мера погрешности решения не превышает ε .

Абсолютная и относительная погрешности тесно связаны с понятием *верных значащих цифр*. Значащими цифрами числа называют все цифры в его записи, начиная

с первой ненулевой цифры слева. Например, число $0,000\underline{12900}$ имеет пять значащих цифр (значащие цифры подчеркнуты). Значащая цифра называется *верной*, если абсолютная погрешность числа не превышает вес разряда, соответствующего этой цифре. Значащая цифра называется *верной в узком смысле слова*, если абсолютная погрешность числа не превышает половины веса разряда, соответствующего этой цифре. Пусть, например, число равно $a = 9348$, абсолютная погрешность числа равна $\Delta a = 15$. Записывая число в виде

$$9348 = 9 \cdot 10^3 + 3 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0,$$

имеем $0,5 \cdot 10^1 < \Delta a < 0,5 \cdot 10^2$, следовательно, число имеет две верных в узком смысле значащих цифр (9 и 3).

Верная значащая цифра может не совпадать с соответствующей цифрой в записи точного числа. Например, $A = 1.000$, $a = 0.999$, $\Delta a = 0.001$, тогда у приближенного числа a все значащие цифры верны, но не совпадают с цифрами точного числа A .

Количество верных значащих цифр числа связано со значением его относительной погрешности [12]. Если десятичное приближенное число a содержит n верных значащих цифр, то для относительной погрешности справедлива оценка

$$\delta a \leq 10^{n-1} - 1^{-1} \approx 10^{-n+1}.$$

Чтобы число a содержало n верных значащих цифр, достаточно выполнения неравенства

$$\delta a \leq 10^n + 1^{-1} \approx 10^{-n}.$$

Поэтому, если необходимо вычислить приближенное число a с *точностью* 10^{-n} , то необходимо сохранить верной значащую цифру, стоящую в n -м разряде после десятичной запятой.

Тот факт, что число a является приближенным значением числа A с предельной абсолютной погрешностью Δ_a , записывают в виде

$$A = a \pm \Delta_a,$$

причем числа a и Δ_a записываются с одинаковым количеством цифр после запятой, например, $A = 2,347 \pm 0,002$ или $A = 2,347 \pm 2 \cdot 10^{-3}$.

Запись вида

$$A = a \pm 1 \pm \delta_a$$

означает, что число a является приближенным значением числа A с предельной относительной погрешностью δ_a . Предельные абсолютные и относительные погрешности принято записывать с одной или двумя значащими цифрами. Большая точность не имеет смысла, так как предельные погрешности представляют собой достаточно грубые оценки.

1.2.3. Особенности машинной арифметики

Одним из основных источников вычислительных погрешностей является приближенное представление чисел в компьютере, обусловленное конечностью разрядной сетки. При решении вычислительных задач обычно используют представление чисел в *форме с плавающей точкой (запятой)*. Число a в форме с плавающей точкой представляется в форме

$$a = \pm Mr^p,$$

где \pm — знак числа; r — основание системы счисления (как правило, $r = 2$); p — порядок числа a ; M — мантисса числа a , причем должно выполняться условие нормировки $r^{-1} \leq M < 1$, означающее, что в компьютере хранятся только значащие цифры.

Диапазон изменения чисел в компьютере ограничен. Для всех представимых в компьютере нормализованных чисел x (за исключением нуля) справедливо неравенство

$$0 < X_0 \leq |x| < X_\infty,$$

где X_0 — минимальное представимое в компьютере нормализованное число (*машинный ноль*) $X_0 = 2^{-p_{\max}+1}$; $p_{\max} = 2^{l+1} - 1$ — максимальное по абсолютной величине значение порядка; l — разрядность порядка; $X_\infty = M_{\max} 2^{p_{\max}} = 1 - 2^{-t} 2^{p_{\max}} \approx 2^{p_{\max}}$ — максимальное представимое в компьютере нормализованное число (*машинная бесконечность*); M_{\max} — максимальное по абсолютной величине значение мантиссы; t — разрядность мантиссы.

В большинстве языков программирования получение машинной бесконечности приводит к аварийному останову выполнения программы по *переполнению*. Все числа, по модулю меньшие X_0 , представляются как *ноль (исчезновение порядка)*.

Большинство современных компьютеров поддерживают *стандарт двоичной арифметики IEEE 754-1985 (ANSI 754) Binary floating-point arithmetic (1985 г.)*¹ [13]. Представление чисел с плавающей точкой в этом стандарте несколько отличается от рассмотренного "классического" представления. Под знак числа отводится один бит: 0 соответствует положительному числу, 1 — отрицательному. Мантисса нормализованных двоичных чисел удовлетворяет условию $1 \leq M < 2$, то есть мантисса нормализованного числа всегда содержит единицу в целой части. Так как целая часть всех нормализованных чисел равна единице, то эта единица не хранится, а хранится только дробная часть мантиссы. Число получается прибавлением единицы к дробной части. Сэкономленный разряд используется для хранения еще

¹ IEEE — The Institute of Electrical and Electronics Engineers, Inc. (произносится "ай-трипл-и"), Институт инженеров по электротехнике и радиоэлектронике, ИИЭР (США) — крупнейшая в мире организация (<http://www.ieee.org/>), объединяющая более 300 тыс. технических специалистов из 147 стран, ведущая организация по стандартизации, отвечающая также за сетевые стандарты.

одного двоичного разряда. Математически представление мантиссы, принятое в стандарте, эквивалентно "классическому" представлению, но увеличивает разрядность мантиссы и диапазон представления чисел. Порядок может быть как положительным, так и отрицательным. Чтобы не вводить бит знака порядка, используют смещенный порядок, прибавляя к порядку смещение, равное $2^{l-1} - 1$, где l — число разрядов, отведенное под смещенный порядок. Например, если под порядок отведен один байт, то смещение равно $2^7 - 1 = 127$. Если порядок равен $+4$, то смещенный порядок будет $4 + 127 = 131$. Если порядок равен -4 , то смещенный порядок будет $-4 + 127 = 123$. Число с нулевым порядком и нулевой мантиссой считается нулем. При этом формально различаются $+0$ и -0 .

Стандарт предусматривает два основных типа чисел с плавающей точкой: числа одинарной и двойной точности. В стандарте предусмотрены также расширенный одинарный (длиной 43 и более битов, редко используется) и расширенный двойной форматы (длиной 79 и более битов, обычно используется 80-битный формат Intel двойной точности).

Числа одинарной точности представляются 32 битами (4 байтами) в виде: 1 бит под знак, 8 бит под порядок, 23 бита под мантиссу. *Числа двойной точности* представляются 64 битами (8 байтами) в виде: 1 бит под знак, 11 бит под порядок, 52 бита под мантиссу.

IEEE-арифметика включает *субнормальные числа* — очень малые *ненормализованные* числа, расположенные между 0 и наименьшим по абсолютной величине нормализованным числом X_0 . При таких условиях нормализация чисел не производится. Субнормальные числа имеют нулевой смещенный порядок. Наличие субнормальных чисел приводит к тому, что малые числа не превращаются в машинный ноль. Например, равенство $x - y = 0$ возможно только при $x = y$.

IEEE-арифметика поддерживает специальные символы $\pm\infty$ (в MATLAB обозначается `inf`) и `NaN`. Символы $\pm\infty$ генерируются при переполнении. Бесконечность содержит единицы во всех разрядах смещенного порядка и нули в разрядах мантиссы. Правила для символа $\pm\infty$: $x/\pm\infty = 0$, $x/0 = \pm\infty$, $\infty + \infty = +\infty$ и т. д. Любая операция, результат которой (конечный или бесконечный) не определен корректно, генерирует символ `NaN` (Not a Number — не число). Например, символ `NaN` генерируется при операциях: $\infty - \infty$, $\frac{\infty}{\infty}$, $\frac{0}{0}$, $NaN \odot x$, где \odot — любая операция.

В случае если в результате арифметической операции получается неопределенность (`NaN`), машинная бесконечность, деление на ноль или машинный ноль, то процессор выставляет флаг исключительного состояния (`exception flag`). Большинство компиляторов по умолчанию не маскируют исключения с плавающей точкой. Это приводит к завершению программ при возникновении таких исключений. Программист должен маскировать и обрабатывать исключения с плавающей точкой самостоятельно, что может сильно усложнить программу. Система

MATLAB маскирует и обрабатывает исключительные ситуации, что существенно упрощает программирование.

В некоторых случаях оказывается недостаточно двойной точности стандарта IEEE 754-1985. Поэтому многие компиляторы поддерживают 80-битный формат Intel двойной точности (мантисса содержит 64 бита, порядок — 15 бит). Некоторые компиляторы программно поддерживают увеличенную разрядность вычислений (при этом существенно увеличивается время вычислений). В новой редакции стандарта IEEE 754-2008 предусмотрены числа учетверенной точности, мантисса которых содержит 112 бит, а порядок — 15 бит.

Для уменьшения погрешностей вычислений иногда используется *рациональная арифметика* — числа представляются в виде рациональных дробей и операции производятся как над обычными дробями. Рациональная арифметика реализована, например, в пакете Mathematica.

Существует также подход, известный как *интервальные вычисления* [14]. В интервальных вычислениях при выполнении каждой арифметической операции вычисляются границы ошибки. Однако алгоритмы интервальных вычислений оказываются сложными и медленными.

Найдем некоторые оценки точности представления чисел и выполнения арифметических операций. Для простоты используем простейшее представление чисел с плавающей точкой, то есть будем использовать условие нормировки мантиссы $2^{-1} \leq M < 1$. Полученные результаты справедливы и для представления чисел в стандарте IEEE 754, если учесть, что в стандарте разрядность мантиссы фактически больше на единицу физически выделенной под мантиссу разрядности.

Из-за конечной разрядной сетки в компьютере можно представить не все числа. Число a , не представимое в компьютере, подвергается округлению, то есть заменяется близким числом \tilde{a} , представимым в компьютере точно. Известно несколько способов округления числа до n значащих цифр. Простейший способ — *усечение*. При усечении отбрасываются все цифры, расположенные справа от n -й значащей цифры. Несколько более сложный способ, *округление по дополнению*, получил наибольшее распространение. В простейшем варианте этого способа анализируется первая отбрасываемая цифра. Если эта цифра равна 1 (в двоичной системе счисления), то к младшей сохраняемой цифре прибавляется единица.

Найдем границу *относительной погрешности представления числа с плавающей точкой* [5]. Допустим, что применяется простейшее округление — усечение. Система счисления — двоичная. Пусть надо записать число, представляющее бесконечную двоичную дробь

$$a = \pm 2^p \cdot \underbrace{\left(\frac{a_1}{2} + \frac{a_2}{2^2} + \dots + \frac{a_t}{2^t} + \frac{a_{t+1}}{2^{t+1}} + \dots \right)}_{\text{мантисса}},$$

где $a_j = \begin{cases} 0 \\ 1 \end{cases}$, $j = 1, 2, \dots$ — цифры мантиссы, а p — порядок.

Пусть под запись мантиссы отводится t двоичных разрядов. Отбрасывая лишние разряды, получим округленное число

$$\tilde{a} = \pm 2^p \left(\frac{a_1}{2} + \frac{a_2}{2^2} + \dots + \frac{a_t}{2^t} \right).$$

Абсолютная погрешность округления в этом случае равна

$$|\tilde{a} - a| = 2^p \left(\frac{a_{t+1}}{2^{t+1}} + \frac{a_{t+2}}{2^{t+2}} + \dots \right).$$

Наибольшая погрешность будет в случае $a_{t+1} = 1, a_{t+2} = 1, \dots$, тогда

$$|\tilde{a} - a| \leq 2^p \frac{1}{2^{t+1}} \underbrace{\left(1 + \frac{1}{2} + \frac{1}{2^2} + \dots \right)}_{=2} = 2^{p-t}.$$

Из условия нормировки для мантиссы всегда верно $a_1 = 1$, поэтому выполняется неравенство $M \geq 0,5$. Тогда $|a| \geq 2^p \cdot 2^{-1} = 2^{p-1}$, и относительную погрешность можно оценить следующим образом: $\delta a = \frac{|\tilde{a} - a|}{|a|} \leq 2^{-t+1}$. Практически применяют более точные методы округления [11], и для погрешности представления чисел справедлива оценка

$$\delta a = \frac{|\tilde{a} - a|}{|a|} \leq 2^{-t}, \quad (1.1)$$

то есть точность представления чисел определяется разрядностью мантиссы t .

Тогда приближенно представленное в компьютере число можно записать в виде $\tilde{a} = a \cdot 1 \pm \varepsilon$, где $\varepsilon = 2^{-t}$ — так называемый машинный эпсилон или машинная точность (часто обозначается `macheps`) — относительная погрешность представления чисел (1.1). В системе MATLAB переменная `eps` обозначает машинную точность, $\text{eps} = 2^{-52}$, что примерно равно $2.2204e-016$.

При вычислениях с плавающей точкой операция округления может потребоваться после выполнения любой арифметической операции. Так, умножение или деление двух чисел сводится к умножению или делению мантисс и к сложению или вычитанию порядков. Так как в общем случае количество разрядов мантисс произведений и частных больше допустимой разрядности мантиссы, то требуется округление мантиссы результата. При сложении или вычитании чисел с плавающей точкой операнды должны быть предварительно приведены к одному порядку. Это осуществляется сдвигом вправо мантиссы числа, имеющего меньший порядок, и увеличением в соответствующее число раз порядка этого числа. Сдвиг мантиссы вправо может привести к потере младших разрядов мантиссы, то есть появляется погрешность округления.

Обозначим округленное в системе с плавающей точкой число, соответствующее точному числу x , через `fl x` (от англ. *floating* — плавающий). Известно [5], что

выполнение каждой арифметической операции вносит относительную погрешность, не большую, чем погрешность представления чисел с плавающей точкой (1.1). Тогда можно записать

$$\text{fl } a \square b = a \square b (1 \pm \delta),$$

где \square — любая из арифметических операций, а $\delta \leq 2^{-l}$.

Таким образом, относительная погрешность арифметической операции над числами, представленными в форме с плавающей точкой, не превышает машинный эпсилон.

1.2.4. Трансформированные погрешности арифметических операций

Рассмотрим *трансформированные погрешности арифметических операций*. Будем считать, что арифметические операции проводятся над приближенными числами, ошибку арифметических операций учитывать не будем (эту ошибку легко учесть, прибавив ошибку округления соответствующей операции к вычисленной ошибке).

Рассмотрим сложение и вычитание приближенных чисел. Покажем, что абсолютная погрешность алгебраической суммы приближенных чисел не превосходит суммы абсолютных погрешностей слагаемых.

Действительно, пусть сумма точных чисел равна

$$S = A_1 + A_2 + \dots + A_n.$$

Абсолютная погрешность суммы равна

$$\begin{aligned} \Delta S &= \Delta (a_1 + a_2 + \dots + a_n) = |a_1 + a_2 + \dots + a_n - A_1 - A_2 - \dots - A_n| = \\ &= |a_1 - A_1 + a_2 - A_2 + \dots + a_n - A_n| \leq \\ &\leq |a_1 - A_1| + |a_2 - A_2| + \dots + |a_n - A_n| = \Delta a_1 + \Delta a_2 + \dots + \Delta a_n. \end{aligned} \quad (1.2)$$

Из (1.2) получаем оценку с использованием предельных абсолютных погрешностей:

$$\Delta S \leq \Delta a_1 + \Delta a_2 + \dots + \Delta a_n.$$

Для относительной погрешности суммы нескольких чисел справедлива оценка

$$\begin{aligned} \delta S &= \frac{\Delta S}{|S|} \leq \frac{|a_1| \left(\frac{\Delta a_1}{|a_1|} \right) + |a_2| \left(\frac{\Delta a_2}{|a_2|} \right) + \dots}{|S|} = \\ &= \frac{|a_1| \delta a_1 + |a_2| \delta a_2 + \dots}{|S|}, \end{aligned} \quad (1.3)$$

где δa_i , $i=1, 2, \dots, n$ — относительные погрешности представления чисел.

Из (1.3) следует, что относительная погрешность суммы нескольких чисел *одного знака* не превышает наибольшей из относительных погрешностей слагаемых:

$$\delta S \leq \max_k \delta a_k, \quad k=1, 2, \dots, n.$$

Пусть A и B — ненулевые числа одного знака. Тогда для соответствующих приближенных чисел справедливы неравенства

$$\delta a + b \leq \delta_{\max}, \quad \delta a - b \leq \gamma \delta_{\max}, \quad (1.4)$$

где $\delta_{\max} = \max \delta a, \delta b$, $\gamma = |A+B|/|A-B|$.

Для доказательства, учитывая (1.2), рассмотрим выражение

$$\begin{aligned} |A \pm B| \delta a \pm b &= \Delta a \pm b \leq \Delta a + \Delta b = \\ &= |A| \delta a + |B| \delta b \leq |A| + |B| \delta_{\max} = |A+B| \delta_{\max}. \end{aligned} \quad (1.5)$$

Из выражения (1.5) следуют оценки (1.4). Из второго неравенства (1.4) следует, что при сложении чисел разного знака или вычитании чисел одного знака относительная погрешность может быть очень большой (если числа близки между собой). Это происходит из-за того, что величина γ в этой ситуации может быть очень большой. Поэтому вычислительные алгоритмы необходимо строить таким образом, чтобы избегать вычитания близких чисел.

Для предельных относительных погрешностей справедливо равенство

$$\delta_S = \frac{|a_1| \delta_{a_1} + |a_2| \delta_{a_2} + \dots}{|S|}.$$

Отметим также, что погрешности вычислений зависят от порядка вычислений. Рассмотрим пример сложения трех приближенных чисел

$$\begin{aligned} S &= x_1 + x_2 + x_3, \\ \tilde{S}_1 &= x_1 + x_2 \quad 1 \pm \delta, \\ \tilde{S}_2 &= \tilde{S}_1 + x_3 \quad 1 \pm \delta = x_1 + x_2 \quad 1 \pm \delta \quad 1 \pm \delta + x_3 \quad 1 \pm \delta, \end{aligned} \quad (1.6)$$

где δ — предельная относительная погрешность округления.

При другой последовательности действий результат будет другим:

$$\begin{aligned} \tilde{S}_3 &= x_3 + x_2 \quad 1 \pm \delta, \\ \tilde{S}_4 &= \tilde{S}_3 + x_1 \quad 1 \pm \delta = x_3 + x_2 \quad 1 \pm \delta \quad 1 \pm \delta + x_1 \quad 1 \pm \delta. \end{aligned}$$

Из (1.6) видно, что результат выполнения некоторого алгоритма, искаженный погрешностями округления, совпадает с результатом выполнения того же алгоритма, но с неточными исходными данными. Поэтому можно применять *обратный анализ*, то есть свести влияние погрешностей округления к возмущению исходных данных. Другими словами, вместо (1.6) можно записать

$$\tilde{S} = \tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3,$$

где $\tilde{x}_1 = x_1 \quad 1 \pm \delta \quad 1 \pm \delta$, $\tilde{x}_2 = x_2 \quad 1 \pm \delta \quad 1 \pm \delta$, $\tilde{x}_3 = x_3 \quad 1 \pm \delta$.

В *прямом анализе ошибок* пытаются оценить влияние погрешностей исходных данных, метода и округлений на результат. В *обратном анализе ошибок* приближенное решение рассматривается как точное решение задачи, но при возмущенных

исходных данных. В обратном анализе рассматривается влияние ошибок на результат, а не оценивается погрешность результата. Однако оценить эквивалентное возмущение исходных данных весьма сложно, поэтому на практике обратный анализ ошибок пока не получил широкого распространения.

Для относительной трансформированной погрешности произведения справедлива оценка

$$\delta ab \leq \delta a + \delta b + \delta a \delta b . \quad (1.7)$$

Для доказательства рассмотрим выражение

$$\begin{aligned} |AB| \delta ab &= \Delta ab = |ab - AB| = \\ &= \left| A - a \quad B + B - b \quad A - A - a \quad B - b \right| \leq \\ &\leq |B| \Delta a + |A| \Delta b + \Delta a \Delta b = \\ &= |B| \Delta a \frac{|A|}{|A|} + |A| \Delta b \frac{|B|}{|B|} + \Delta a \Delta b \frac{|A|}{|A|} \frac{|B|}{|B|} = \\ &= |AB| \delta a + \delta b + \delta a \delta b , \end{aligned}$$

из которого следует (1.7).

Для относительной погрешности частного справедлива оценка

$$\delta \left(\frac{a}{b} \right) \leq \frac{\delta a + \delta b}{1 - \delta b} .$$

Действительно,

$$\begin{aligned} \delta \left(\frac{a}{b} \right) &= \frac{\left| \frac{a}{b} - \frac{A}{B} \right|}{\frac{|A|}{|B|}} = \frac{|aB - Ab|}{|Ab|} = \frac{|B a - A - A b - B|}{|Ab|} \leq \\ &\leq \frac{|B| \Delta a + |A| \Delta b}{|AB| 1 - \delta b} = \frac{\delta a + \delta b}{1 - \delta b} . \end{aligned} \quad (1.8)$$

В выражении (1.8) учтено неравенство

$$|b| = |B + b - B| \geq |B| - \Delta b = |B| 1 - \delta b .$$

Так как для предельных погрешностей справедливы оценки $\delta_a \ll 1$ и $\delta_b \ll 1$, то для оценки предельных трансформированных погрешностей умножения и деления на практике используются оценки

$$\delta_{ab} \approx \delta_a + \delta_b, \quad \delta_{a/b} \approx \delta_a + \delta_b .$$

Рассмотренные аналитические оценки громоздки и малоприменимы для оценки погрешностей многократно повторенных операций. Кроме того, эти оценки рассчитаны на наихудшие случаи и не учитывают частичную компенсацию погрешностей при выполнении большого числа операций. Ведь часть операций будет иска-

жать результат в большую сторону, а часть — в меньшую. При большом числе n арифметических операций можно пользоваться *приближенной статистической оценкой погрешности арифметических операций*, учитывающей частичную компенсацию погрешностей разных знаков [15]: $\delta_{\Sigma} \approx \delta_{fl} \sqrt{n}$, где δ_{Σ} — суммарная относительная погрешность; $\delta_{fl} \leq \varepsilon$ — относительная погрешность выполнения операций с плавающей точкой; ε — погрешность представления чисел с плавающей точкой. Статистическую оценку погрешности проиллюстрируем на примере суммы большого числа слагаемых $S = \sum_{i=1}^n a_i$. Вследствие погрешностей слагаемые a_i можно рассматривать как случайные величины. Если слагаемые a_i независимы и их среднеквадратические отклонения равны σ_a , то из теории вероятностей известно, что $\sigma_S = \sigma_a \sqrt{n}$. Если принять, что погрешности распределены по нормальному закону с нулевым математическим ожиданием и среднеквадратическим отклонением σ_a , то часто считают, что погрешность не превышает трех σ_a : $\Delta_a \leq 3\sigma_a$ ("правило трех сигма"). Умножая предыдущее равенство на 3, получаем для предельных погрешностей $\Delta_S = \Delta_a \sqrt{n}$.

1.2.5. Трансформированные погрешности вычисления функций

Рассмотрим *трансформированную погрешность вычисления значений функций*. Пусть $y = f(x_1, x_2, \dots, x_m)$ — дифференцируемая функция многих переменных и известны приближенные значения аргументов $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m$. Тогда трансформированная погрешность значения функции равна

$$\Delta y = f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m) - f(x_1, x_2, \dots, x_m) .$$

(1.9)

Выражение (1.9) представляет собой полное приращение функции. Главной линейной частью приращения (1.9) является полный дифференциал, тогда

$$\Delta y \approx dy = \sum_{i=1}^m \frac{\partial y}{\partial x_i} dx_i . \quad (1.10)$$

Из соотношения (1.10) получаем оценку *абсолютной трансформированной погрешности функции многих переменных*

$$\Delta y \approx |dy| = \left| \sum_{i=1}^m \frac{\partial y}{\partial x_i} dx_i \right| \leq \sum_{i=1}^m \left| \frac{\partial y}{\partial x_i} \right| |\tilde{x}_i - x_i| = \sum_{i=1}^m \left| \frac{\partial y}{\partial x_i} \right| \Delta x_i \leq \sum_{i=1}^m \left| \frac{\partial y}{\partial x_i} \right| \Delta_{x_i} , \quad (1.11)$$

где Δ_{x_i} — предельная абсолютная погрешность переменной x_i .

Относительная трансформированная погрешность функции многих переменных равна

$$\begin{aligned} \delta y &= \frac{\Delta y}{|y|} \leq \sum_{i=1}^m \left| \frac{\partial y}{\partial x_i} \frac{\Delta x_i}{|y|} \right| = \sum_{i=1}^m \left| \frac{\partial y}{y \partial x_i} \right| \Delta x_i = \\ &= \sum_{i=1}^m \left| \frac{\partial \ln y}{\partial x_i} \right| \Delta x_i \leq \sum_{i=1}^m \left| \frac{\partial \ln y}{\partial x_i} \right| \Delta x_i. \end{aligned} \quad (1.12)$$

Трансформированные погрешности функции одной переменной получаются из (1.11) и (1.12) при $m = 1$.

С практической точки зрения важно определить *допустимую погрешность аргументов по допустимой погрешности функции* (обратная задача). Эта задача имеет однозначное решение только для дифференцируемых функций одной переменной $y = f(x)$ в случае, когда $f'(x) \neq 0$:

$$\Delta x = \frac{1}{|f'(x)|} \Delta y.$$

Для функций многих переменных задача не имеет однозначного решения, и необходимо ввести дополнительные ограничения. Например, если функция $y = f(x_1, x_2, \dots, x_m)$ наиболее критична к погрешности Δx_i , то, пренебрегая погрешностью других аргументов, получаем

$$\Delta x_i = \Delta y \left/ \left| \frac{\partial f}{\partial x_i} \right| \right|.$$

Если вклад погрешностей всех аргументов примерно одинаков, то применяют *принцип равных влияний*: $\Delta x_i = \Delta y \left/ m \left| \frac{\partial f}{\partial x_i} \right| \right|$, $i = \overline{1, m}$.

1.3. Свойства вычислительных задач и алгоритмов

1.3.1. Корректность вычислительной задачи

Пусть решается операторное уравнение общего вида

$$Ay = f, \quad (1.13)$$

где $A: Y \rightarrow F$ — некоторый оператор, действующий из метрического пространства Y в метрическое пространство F . Тогда задача нахождения элемента $y \in Y$ по заданному элементу $f \in F$ из уравнения (1.13) называется *корректно поставленной по Адамару*, или просто *корректной*, если [17]:

- решение $y \in Y$ существует при каждом $f \in F$;
- это решение единственно;

□ решение непрерывно зависит от правой части уравнения (1.13), то есть из сходимости $f_n \rightarrow f$ по метрике пространства F следует сходимость $y_n \rightarrow y$ по метрике пространства Y , где y_n — решение уравнения (1.13), в котором правая часть заменена на f_n . Другими словами, решение устойчиво по отношению к малым возмущениям входных данных.

Если хотя бы одно из этих условий не выполнено, то задача называется *некорректной*.

Длительное время считалось, что некорректные задачи, решения которых неустойчивы, не имеют физического смысла. Однако выяснилось, что многие важные практические задачи являются некорректными. Например, некорректными являются *обратные задачи* [16], ряд задач распознавания образов и многие другие. Поэтому разработаны методы решения многих классов некорректных задач — так называемые *методы регуляризации* [17]. основополагающие работы в этой области принадлежат российским математикам, в частности А. Н. Тихонову (см., например, [17]). Использование методов регуляризации выходит за рамки нашего курса, поэтому дадим самые общие представления о методах решения некорректных задач. Идея методов регуляризации состоит в учете некоторых дополнительных условий, часто носящих не очень четкую и неоднозначную формулировку. Например, сужается пространство решений Y или учитывается априорная информация о решении.

При решении различных физических задач известна априорная информация о решении. Учет априорной информации позволяет сузить множество решений Y до некоторого множества M , в котором решение задачи устойчиво. Это приводит к понятию задачи, корректной по Тихонову, или условно корректной.

Задача нахождения решения уравнения (1.13) называется *условно корректной* или *корректной по Тихонову*, если [17]:

- решение y существует и принадлежит множеству $M \subseteq Y$ (*множеству корректности*, или *классу корректности*);
- решение y единственно в M ;
- малым вариациям f , не выводящим за пределы M , соответствуют малые вариации y .

Понятие условной корректности позволяет сузить множество, на котором ищется решение, но, как правило, требует привязки задачи (1.13) к конкретным приложениям.

Для решения некорректных задач используется понятие *регуляризирующего оператора* [17] $R_{f,\alpha} : F \rightarrow Y$ для уравнения (1.13). Регуляризирующий оператор (выбор которого в общем случае неоднозначен) должен позволять получить *регуляризованное решение* уравнения (1.13) с приближенной правой частью f , если значение *параметра регуляризации* α согласовано с погрешностью исходных данных. При этом если погрешность исходных данных стремится к нулю, то регуляризованное решение стремится к точному решению, а параметр регуляризации стремится к нулю.

Рассмотрим понятие метода регуляризации на примере линейного оператора A [18—20]. Пусть вместо (1.13) известно приближенное уравнение

$$\tilde{A}y = \tilde{f}, \quad (1.14)$$

где $y \in Y$, $\tilde{f} \in F$.

Задача (1.14) очень чувствительна к погрешностям исходных данных и является неустойчивой, то есть решение сильно изменяется при изменении исходных данных. Близость уравнений (1.13) и (1.14) характеризуется неравенствами

$$\|\tilde{A} - A\| \leq \xi, \quad \|\tilde{f} - f\| \leq \delta, \quad (1.15)$$

где $\|\dots\|$ — некоторая норма.

Для решения уравнения (1.14) используется сглаживающий функционал (*функционал Тихонова*)

$$\Phi_\alpha [y, \tilde{A}, \tilde{f}] = \|\tilde{A}y - \tilde{f}\|^2 + \alpha \Omega y,$$

где $\alpha > 0$ — *параметр регуляризации*; Ωy — *стабилизирующий функционал (стабилизатор)*. Часто берут $\Omega y = \|y\|^2$.

Доказывается [17, 20], что при определенных условиях функционал Тихонова, в частности

$$\Phi_\alpha [y, \tilde{A}, \tilde{f}] = \|\tilde{A}y - \tilde{f}\|^2 + \alpha \|y\|^2, \quad (1.16)$$

имеет единственный минимум y_α , который принимается за регуляризованное решение уравнения (1.14). Регуляризованное решение можно найти методами численной минимизации функционала (1.16) или с помощью решения *уравнения Тихонова*

$$\alpha y_\alpha + \tilde{A}^* \tilde{A} y_\alpha = \tilde{A}^* \tilde{f}, \quad (1.17)$$

где \tilde{A}^* — оператор, сопряженный с \tilde{A} .

Уравнение (1.17) представляет собой необходимое условие экстремума функционала (1.16).

Главной проблемой является выбор параметра регуляризации α . При малом значении α регуляризованное решение y_α обеспечивает малую невязку $\|\tilde{A}y - \tilde{f}\|^2$, но является неустойчивым. При большом значении α решение y_α является устойчивым, что обеспечивается стабилизатором $\|y\|^2$, но сильно отличается от решения системы (1.14). Выбор параметра регуляризации α является сложной задачей (см., например, [20, 22]). Параметр регуляризации должен быть согласован с уровнем погрешности ξ и δ (см. (1.15)).

Например, решение переопределенной системы линейных алгебраических уравнений $\mathbf{Ax} = \mathbf{b}$ методом регуляризации Тихонова сводится к минимизации функционала

$$\Phi_\alpha = \|\mathbf{Ax} - \mathbf{b}\|^2 + \alpha \|\mathbf{x}\|^2, \quad (1.18)$$

где α — параметр регуляризации. Задача минимизации функционала (1.18) эквивалентна решению системы

$$\mathbf{A}^T \mathbf{A} + \alpha \mathbf{E} \mathbf{x} = \mathbf{A}^T \mathbf{b},$$

где \mathbf{E} — единичная матрица. Опуская математические подробности, отметим, что параметр регуляризации согласуется с уровнем погрешности задания элементов матрицы и вектора правой части.

Часто, например при решении обратных задач, используют так называемую *итерационную регуляризацию* [19, 21]. Для простоты опять рассмотрим линейную задачу. Вместо решения некорректной задачи (1.13) можно решить задачу отыскания минимума функционала

$$J y = \|Ay - f\|^2. \quad (1.19)$$

Нахождение минимума (1.19) означает нахождение такого решения y , которое обеспечивает минимум невязки $r = Ay - f$. Для нахождения минимума часто используют итерационные методы численной оптимизации [23]. Для таких итерационных процессов характерно, что первые приближения мало отличаются от соответствующих приближений при точных исходных данных. С ростом числа итераций приближения могут все сильнее отличаться от точного решения, а итерационный процесс может начать расходиться. Метод итерационной регуляризации состоит в том, чтобы по оценке погрешности исходных данных выбрать номер итерации, на котором следует остановить итерационный процесс. Итерационный процесс решения останавливается по критерию, согласованному с точностью исходных данных. Параметром регуляризации здесь является номер итерации.

1.3.2. Обусловленность вычислительной задачи

Рассмотрим корректную вычислительную задачу. Корректная задача теоретически позволяет найти ее решение со сколь угодно малой погрешностью при условии малых погрешностей исходных данных. Погрешности исходных данных не могут быть устранены никакими методами решения, и на практике они всегда конечны и не могут быть сколь угодно малы, хотя в каком-то смысле они все же малы. Поэтому важно знать, как малые погрешности исходных данных трансформируются в погрешность результата. Это существенно еще и потому, что, как показали последние исследования [24—26], существуют задачи (названные "особыми"), у которых малые изменения параметров могут привести к радикальному изменению свойств решения. Например, устойчивое решение системы обыкновенных дифференциальных уравнений, описывающей систему автоматического управления,

может стать неустойчивым при малом изменении параметров. Причем малое изменение параметров реальной системы и, соответственно, модели может произойти в процессе эксплуатации системы, например из-за износа. Неустойчивость системы управления может привести к аварии.

Чувствительность решения вычислительной задачи к малым погрешностям исходных данных называется *обусловленностью задачи*. Задача называется *хорошо обусловленной*, если малым погрешностям исходных данных соответствуют малые погрешности решения. В противном случае задача называется *плохо обусловленной*.

Степень обусловленности задачи определяется *числом обусловленности*. Выделяют *абсолютное число обусловленности* ν_{Δ} , определяемое неравенством

$$\Delta y \leq \nu_{\Delta} \Delta x ,$$

и *относительное число обусловленности* ν_{δ} , определяемое неравенством

$$\delta y \leq \nu_{\delta} \delta x .$$

Чаще всего используется относительное число обусловленности. Если $\nu \gg 1$, то задача плохо обусловлена. Практически можно считать [12], если $\nu_{\delta} \sim 10^N$, то N показывает число верных цифр, которое может быть потеряно по сравнению с числом верных цифр исходных данных.

Например, при вычислении дифференцируемой функции $y = f(x)$ из оценки (1.12) получаем число обусловленности

$$\nu_{\delta} \approx \frac{|x| |f'(x)|}{|f(x)|} .$$

Если вычисляется функция $y = \sin x$, то $\nu_{\delta} \approx |x \cdot \operatorname{ctg} x|$. При $x \rightarrow \pi k$ число обусловленности стремится к бесконечности, то есть задача плохо обусловлена.

В чем различие между некорректными и плохо обусловленными задачами? На практике и в том и в другом случае небольшие изменения исходных данных приводят к большим изменениям результата. В [24, 25] предлагается уточнить классическое определение некорректной задачи. В классической работе [17] рассматриваются некорректные задачи в основном для сложных проблем, описываемых дифференциальными уравнениями в частных производных и интегральными уравнениями. И классическое определение некорректности, безусловно, справедливо для этих задач. Но в целом это определение неудобно. Требования существования и единственности решения часто не являются определяющими.

Например, уравнение $x^2 + 1 = 0$ не имеет решения в области вещественных чисел, уравнение $\sin x = 0$ имеет бесконечное множество решений, но некорректными в бытовом смысле слова эти задачи не являются. В [24] в определении некорректности основное внимание уделяется влиянию исходных данных: "*Некорректными* назовем те задачи, в которых *сколь угодно малым* изменениям коэффициентов, параметров, начальных условий, граничных условий и т. п. соответствуют *конечные* или даже *коренные изменения решения* (под коренными изменениями подразуме-

ваются, например, случаи, когда при номинальных значениях коэффициентов решение отсутствует, а при сколь угодно малых вариациях — появляется). ... *Плохо обусловленными* назовем задачи, в которых малым, но *конечным* изменениям коэффициентов, параметров и т. п. соответствуют большие изменения решения". Определение плохо обусловленной задачи в отличие от некорректной задачи не является математически точным. Из определения видно, что некорректные задачи являются предельным случаем задач плохо обусловленных, когда малые, но конечные изменения параметров переходят в сколь угодно малые ("бесконечно малые"). В [24, 25] показано, что корректность задачи может изменяться при эквивалентных преобразованиях, не изменяющих решения. В указанных работах предлагается ввести *третий класс задач, меняющих свою корректность в ходе эквивалентных преобразований, используемых при их решении*. Другим примером, требующим расширения понятия корректности, являются системы линейных алгебраических уравнений. Общеизвестно, если определитель матрицы системы линейных алгебраических уравнений равен нулю, то такая задача является некорректной. В реальной задаче при неточно заданных элементах матрицы нельзя говорить о точном равенстве нулю определителя. В [26] вводится понятие "очень плохо обусловленных систем уравнений" — систем, у которых при разрешаемых погрешностями изменениях элементов матрицы возникает значение определителя, равное нулю.

В практике решения вычислительных задач большое внимание уделяется *обусловленности систем линейных алгебраических уравнений (СЛАУ)*. Рассмотрим, как влияют погрешности исходных данных на погрешность решения СЛАУ. Корректно поставленные задачи решения СЛАУ (то есть задачи, решение которых существует, единственно и непрерывно зависит от входных данных) в зависимости от чувствительности решения к погрешности исходных данных делятся на хорошо и плохо обусловленные. Рассмотрим пример плохо обусловленной СЛАУ [24]. Система

$$\begin{aligned} 1,1x + y &= 1,1; \\ x + y &= 1 \end{aligned}$$

имеет определитель

$$\begin{vmatrix} 1,1 & 1 \\ 1 & 1 \end{vmatrix} = 0,1$$

и решение $x=1$; $y=0$. С учетом вариации одного из коэффициентов система примет вид

$$\begin{aligned} 1,1x + y &= 1,1; \\ 1 + \varepsilon x + y &= 1 \end{aligned}$$

и имеет решение $x = \frac{1}{1-10\varepsilon}$; $y = \frac{-11\varepsilon}{1-10\varepsilon}$. Решение очень сильно зависит от вариации коэффициента. Например, если $|\varepsilon| \leq 0,001$, то $0,990 \leq x \leq 1,010$; $-0,011 \leq y \leq 0,011$. Если $|\varepsilon| \leq 0,010$, то $0,909 \leq x \leq 1,110$; $-0,122 \leq y \leq 0,100$. Если $|\varepsilon| \leq 0,100$,

то $0,500 \leq x \leq \infty$; $-\infty \leq y \leq 0,550$. Таким образом, уже при $|\varepsilon| = 0,1$ вариация решения может быть сколь угодно большой.

Для СЛАУ из двух уравнений явление плохой обусловленности имеет наглядный геометрический смысл. В этом случае каждое уравнение системы описывает прямую, а координаты точки пересечения прямых являются решением. Рассмотрим СЛАУ [27]

$$1000x_1 + 999x_2 = b_1,$$

$$999x_1 + 998x_2 = b_2.$$

Наклоны двух прямых, описываемых этими уравнениями, соответственно равны: $k_1 = -1000/999 = -1,001001$, $k_2 = -999/998 = -1,001002$. То есть прямые практически параллельны. Найти точку пересечения этих прямых трудно. Небольшое изменение правой части одного из уравнений приводит к небольшому параллельному сдвигу соответствующей прямой. При этом точка пересечения прямых (решение СЛАУ) резко изменится.

Оценим число обусловленности СЛАУ. Рассмотрим систему линейных алгебраических уравнений, записанную в матричном виде

$$\mathbf{Ax} = \mathbf{b}, \quad (1.20)$$

где \mathbf{A} — невырожденная квадратная матрица коэффициентов; \mathbf{b} — ненулевой вектор правой части; \mathbf{x} — вектор искомого решения.

Пусть элементы матрицы \mathbf{A} заданы точно, а правая часть СЛАУ задана неточно, то есть используется возмущенный вектор $\mathbf{b} + \Delta\mathbf{b}$. Тогда изменится и решение системы:

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}. \quad (1.21)$$

Если $\det \mathbf{A} \neq 0$, то из (1.20) и (1.21) получим¹

$$\|\Delta\mathbf{x}\| \leq \|\mathbf{A}^{-1}\| \cdot \|\Delta\mathbf{b}\|, \quad (1.22)$$

где под $\|\Delta\mathbf{x}\|$ и $\|\Delta\mathbf{b}\|$ будем понимать абсолютные погрешности векторов.

Перемножая (1.22) и очевидное неравенство $\|\mathbf{b}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|$, с учетом того, что $\|\mathbf{b}\| \neq 0$ и $\|\mathbf{x}\| \neq 0$, получим

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|},$$

или

$$\delta(\mathbf{x}) \leq \text{cond}(\mathbf{A}) \delta(\mathbf{b}), \quad (1.23)$$

где $\delta \mathbf{x}$ и $\delta \mathbf{b}$ — относительные погрешности решения и правой части, а

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{A}\|$$

¹ Определение и свойства норм векторов и согласованных норм матриц приводятся в разд. 2.1.