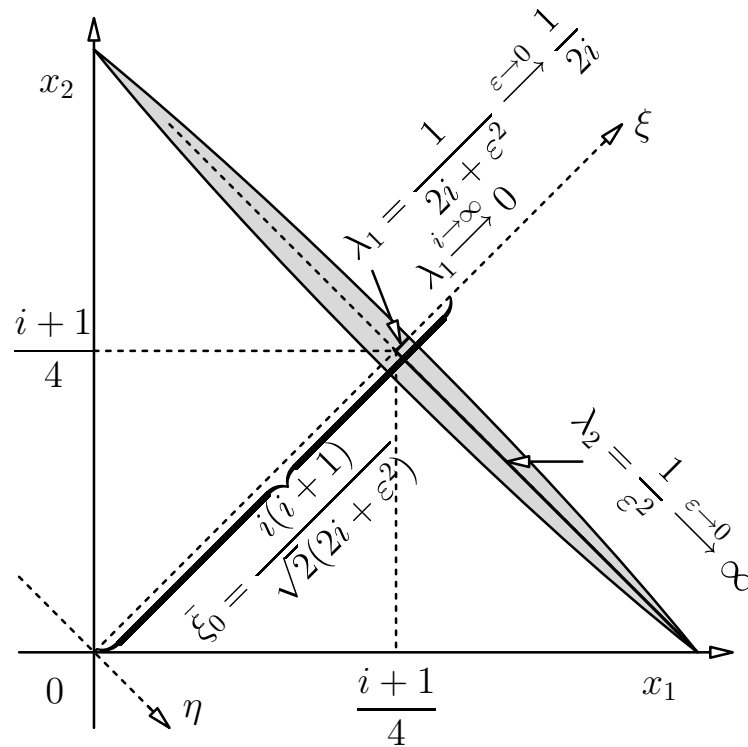


И. В. Семушин

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ И ОЦЕНИВАНИЯ



Ульяновск
2011

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

И. В. Семушин

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ И ОЦЕНИВАНИЯ

Учебное пособие

Ульяновск
УлГТУ
2011

УДК 519.61 + 519.654 + 519.244.4

ББК 22.193я7

С30

Рецензенты: Профессор д-р Б. С. Верховский, Кафедра ВТ,
Технологический Институт Нью Джерси, США
Профессор, д-р техн. наук В. И. Левин,
Заслуженный деятель науки РФ, кафедра математики,
Пензенская государственная технологическая академия

Рецензия кафедры общей информатики, Самарский государственный
аэрокосмический университет им. акад. С. П. Королёва.—Заведующий
кафедрой профессор, д-р техн. наук В. А. Фурсов

Утверждено редакционно-издательским советом университета
в качестве учебного пособия

Семушин, И. В.

С30 Вычислительные методы алгебры и оценивания: учебное пособие /
И. В. Семушин. — Ульяновск: УлГТУ, 2011. — 366 с.
ISBN 978-5-9795-0902-0

Книга сообщает базовые сведения по вычислительной линейной алгебре и линейному оцениванию. Знакомит с эффективными алгоритмами дискретной фильтрации. Включает контрольные задачи и лабораторные проекты с большим числом индивидуальных заданий.

Для студентов, изучающих численные методы по многим учебным программам бакалавриата и специальностей, а также для аспирантов и специалистов в области математического моделирования, численных методов и комплексов программ.

УДК 519.61 + 519.654 + 519.244.4
ББК 22.193я7

ISBN 978-5-9795-0902-0

© Оформление. УлГТУ, 2011
© Семушин, И. В., 2011

RUSSIAN FEDERATION'S MINISTRY FOR EDUCATION AND SCIENCE

Federal State Budget Educational Institution for
Higher Professional Education

«ULYANOVSK STATE TECHNICAL UNIVERSITY»

I. V. Semushin

COMPUTATIONAL METHODS OF ALGEBRA AND ESTIMATION

Textbook

Ulyanovsk
UISTU
2011

UDC 519.61 + 519.654 + 519.244.4

BBK 22.193я7

C30

Reviewers: Professor Dr. Boris S. Verkhovsky, CS Department,
New Jersey Institute of Technology, USA
Professor Dr. Vitaly I. Levin, RF's Honoured Science Worker,
Math Department, Penza State Academy of Technology

Department of Basic Informatics, Academician S. P. Korolyov Samara
State Aerospace University submitted its approval of the book.—Signed by
Professor Dr. Vladimir A. Fursov, Head of Department

Approved by the University Editorial Board
as a textbook

Semushin, I. V.

C30 Computational Methods of Algebra and Estimation: textbook /
I. V. Semushin. — Ulyanovsk: UlSTU, 2011. — 366 c.
ISBN 978-5-9795-0902-0

The book conveys the basic knowledge in Computational Linear Algebra and Linear Estimation. It familiarizes with the efficient algorithms of discrete filtering, and includes test problems and laboratory projects with the great number of individual assignments.

For students who learn Numerical Methods as a discipline of study at the bachelor degree or diploma level as well as for PhD students and specialists in the field of Mathematical Modeling, Numerical Methods and Integrated Software.

UDC 519.61 + 519.654 + 519.244.4
BBK 22.193я7

ISBN 978-5-9795-0902-0

© Design. UlSTU, 2011
© Semushin, I. V., 2011

For the things we have to learn before we can do them,
we learn by doing them.

—Aristotle

Ибо то, что нам надо постичь, чтобы уметь это делать,
мы постигаем, делая это.

—Аристотель

Оглавление

Предисловие	11
1 ВВЕДЕНИЕ	15
1.1 Учебные цели студента	15
1.2 Оценка работы студента	16
1.3 Кодекс студента	21
1.4 Краткое описание курса	23
I Вычислительная линейная алгебра	25
2 СТАНДАРТНЫЕ АЛГОРИТМЫ LU-РАЗЛОЖЕНИЯ	27
2.1 Алгоритмы метода Гаусса	27
2.2 Выбор ведущего элемента	30
2.3 Компактные схемы	34
2.4 Алгоритмы метода Жордана	37
2.5 Вычисление обратной матрицы	41
2.6 Плохо обусловленные матрицы	47
2.7 Задание на лабораторный проект № 1	48
2.8 Варианты задания на лабораторный проект № 1	51
3 ВЕКТОРНО-ОРИЕНТИРОВАННЫЕ АЛГОРИТМЫ LU-РАЗЛОЖЕНИЯ	53
3.1 Гауссово исключение и ijk -алгоритмы	53
3.2 Распараллеливание вычислений	55
3.3 Параллельное умножение матрицы на вектор	58
3.4 Параллельное LU -разложение	59
3.5 LU -разложение и его ijk -формы	62
3.6 Треугольные системы	67
3.7 Задание на лабораторный проект № 2	69
3.8 Варианты задания на лабораторный проект № 2	72

4	АЛГОРИТМЫ ОКАЙМЛЕНИЯ В LU-РАЗЛОЖЕНИИ	73
4.1	Метод окаймления	73
4.2	Окаймление известной части разложения	73
4.3	Окаймление неизвестной части разложения	76
4.4	Задание на лабораторный проект № 3	78
4.5	Варианты задания на лабораторный проект № 3	81
5	РАЗРЕЖЕННЫЕ ФОРМЫ LU-РАЗЛОЖЕНИЯ	82
5.1	Упакованные формы хранения матриц	82
5.2	Выбор ведущего элемента	84
5.3	Задание на лабораторный проект № 4	87
5.4	Варианты задания на лабораторный проект № 4	89
6	РАЗЛОЖЕНИЯ ХОЛЕССКОГО	90
6.1	Положительно определенные матрицы	90
6.2	Квадратные корни из P и алгоритмы Холесского	91
6.3	Программная реализация алгоритмов Холесского	94
6.4	Разложение Холесского: ijk -формы	96
6.5	Разложение Холесского: алгоритмы окаймления	98
6.6	Особенности хранения ПО-матрицы P	101
6.7	Задание на лабораторный проект № 5	102
6.8	Варианты задания на лабораторный проект № 5	106
7	ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ	107
7.1	Ортогональные матрицы и приложения	107
7.2	Линейная задача наименьших квадратов	109
7.3	Ортогональные матрицы и наименьшие квадраты	110
7.4	Преобразование Хаусхолдера	111
7.5	Шаг триангуляризации матрицы методом Хаусхолдера	116
7.6	Решение треугольной системы и обращение матриц	117
7.7	Преобразование Гивенса	120
7.8	Варианты заполнения матрицы R	126
7.9	Правосторонние ортогональные преобразования	127
7.10	Двусторонние ортогональные преобразования	128
7.11	Ортогонализация Грама–Шмидта	131
7.12	Алгоритмы ортогонализации Грама–Шмидта	133
7.13	Решение систем после ортогонализации	137
7.14	Обращение матриц после ортогонализации	137

7.15	Задание на лабораторный проект № 6	137
7.16	Варианты задания на лабораторный проект № 6	140
8	ИТЕРАЦИОННЫЕ МЕТОДЫ	141
8.1	Итерационные методы	141
8.2	Итерационная формула	142
8.3	Метод Якоби	143
8.4	Метод Зейделя	143
8.5	Матричная запись методов Якоби и Зейделя	144
8.6	Каноническая форма одношаговых ИМ	145
8.7	Методы простой итерации, Рундсона и Юнга	145
8.8	Сходимость итерационных методов	146
8.9	Скорость сходимости итерационных методов	147
8.10	Итерационные методы вариационного типа	150
8.11	Другие методы	155
8.12	Задание на лабораторный проект № 7	156
8.13	Варианты задания на лабораторный проект № 7	159
9	Фонд задач	160
9.1	Типовые задачи	160
9.2	Решения и рекомендации к типовым задачам	164
9.3	Варианты контрольных заданий	167
9.4	Задачи для контрольных заданий и экзамена	174
II	Линейное оценивание	197
10	ТЕОРЕТИЧЕСКИЕ ОСНОВЫ	199
10.1	Конечномерные линейные пространства	200
10.2	Обобщение на гильбертовы пространства	204
10.3	Проектирование в конечномерных пространствах	206
10.4	Наименьшие квадраты и псевдоинверсия	209
10.5	Отыскание псевдообратной матрицы	212
10.6	Основные теоремы по МНК и псевдоинверсии	223
10.7	Вычисление матриц проектирования	225
10.8	Рекурсия в задаче МНК	231
10.9	Основные свойства симметрических / эрмитовых матриц	233

11	ОЦЕНИВАНИЕ ПО МЕТОДУ НАИМЕНЬШИХ КВАДРАТОВ . . .	236
11.1	Модели, регрессии и оценки	236
11.2	Линейная задача наименьших квадратов	238
11.3	Статистическая интерпретация	240
11.4	Включение априорных статистических данных	241
11.5	Включение предшествующего МНК-решения	243
11.6	Рекурсия МНК в стандартной информационной форме .	244
11.7	Рекурсия МНК в стандартной ковариационной форме .	246
11.8	Ковариационный алгоритм Поттера для МНК	250
11.9	Полная статистическая интерпретация рекурсии в МНК .	251
11.10	Основные результаты	261
12	ОДНОВРЕМЕННОЕ РЕШЕНИЕ НОРМАЛЬНЫХ УРАВНЕНИЙ . .	264
12.1	Метод нормальных уравнений	264
12.2	Формирование матрицы A	265
12.3	Задание на лабораторный проект № 8	265
12.4	Варианты задания на лабораторный проект № 8	268
13	УСТОЙЧИВЫЕ АЛГОРИТМЫ ФИЛЬТРАЦИИ	270
13.1	Фильтрация Калмана в историческом аспекте	270
13.2	Стандартный фильтр Калмана	272
13.3	Скаляризованная форма фильтра Калмана	274
13.4	Стабилизированный фильтр Калмана–Джозефа	275
13.5	Квадратно-корневой фильтр Поттера	276
13.6	Одноранговое обновление ПО-матриц	278
13.7	Факторизованный фильтр Бирмана	279
13.8	Квадратно-корневой фильтр Карлсона	285
13.9	Редуцированный фильтр Бирмана	287
13.10	Редуцированный фильтр Бар-Ицхака	288
13.11	Редуцированный фильтр Бар-Ицхака–Медана	289
13.12	Задача сопровождения судна на траектории	293
13.13	Пример задачи с мультиколлинеарностью	300
13.14	Задание на лабораторный проект № 9	305
13.15	Варианты задания на лабораторный проект № 9	309
14	ОРТОГОНАЛИЗОВАННЫЕ БЛОЧНЫЕ АЛГОРИТМЫ	316
14.1	Задача оценивания	316
14.2	Блочные алгоритмы в исторической перспективе	317

14.3	Расширенный квадратно-корневой К-фильтр	320
14.4	Расширенный квадратно-корневой И-фильтр	322
14.5	Модифицированный квадратно-корневой И-фильтр . . .	324
14.6	Комбинированный квадратно-корневой фильтр	326
14.7	Скаляризованный квадратно-корневой К-фильтр	327
14.8	Скаляризованный квадратно-корневой И-фильтр	328
14.9	Скаляризованный модифицированный квадратно- корневой И-фильтр	329
14.10	Скаляризованный комбинированный квадратно- корневой фильтр	329
14.11	Задание на лабораторный проект № 10	330
14.12	Варианты задания на лабораторный проект № 10	334
Заключение		335
А	Обоснования алгоритмов для подразд. 14.7–14.10	339
A.1	Построение новых скаляризованных алгоритмов	339
В	К задаче управления	343
B.1	Задача ЛКГ-управления	343
B.2	Решение задачи управления	344
B.3	Двойственность задач фильтрации и управления	347
B.4	Вычислительные алгоритмы задачи управления	347
Список иллюстраций		350
Список таблиц		351
Библиографический список		352
Предметный указатель		361

Предисловие

Кафедра «Информационные системы» УлГТУ решением от 01.12..2011 г. № 12, рекомендует данное учебное пособие для студентов высших учебных заведений, обучающихся по направлениям:

- 230700.62 «Прикладная информатика» — профиль «Прикладная информатика в экономике»,
- 231000.62 «Программная инженерия»,
- 230700.68 «Прикладная информатика»,
- 231000.68 «Программная инженерия», а также
- для аспирантов по специальности 05.13.18 «Математическое моделирование, численные методы и комплексы программ».

Кроме отмеченных, данное пособие может быть рекомендовано студентам и других направлений и специальностей, например,

- 010101 «Математика» при изучении дисциплины «Методы вычислений»,
- 010501 «Прикладная математика и информатика» при изучении дисциплины «Численные методы»,
- 010503 «Математическое обеспечение и администрирование информационных систем» при изучении дисциплины «Вычислительная математика»,
- 210400 «Телекоммуникации» при изучении дисциплины «Вычислительная математика»,
- 230200 «Информационные системы» при изучении дисциплины «Вычислительная математика».

Несмотря на различия в названиях учебных дисциплин, курс «Численные методы» по многим специальностям и направлениям подготовки в университетах преследует следующие общие цели:

- заложить базовые умения и навыки в области разработки вычислительных алгоритмов решения задач, возникающих в процессе математического моделирования законов реального мира;
- обеспечить понимание основных идей численных методов, особенностей и условий их применения;
- подготовить студентов к применению этих знаний в дальнейшей учебе и практической деятельности.

При изучении данного курса значительное время отводят на «Вычислительные методы алгебры», или «Численные методы—I». Согласно требованиям Государственного образовательного стандарта [7], к фундаментальной части численных методов относят следующие темы из «Вычислительной линейной алгебры» (ВЛА):

- тема 1 — методы исключения в решении систем;
- тема 2 — разложения Холецкого положительно определенных матриц;
- тема 3 — методы ортогональных преобразований;
- тема 4 — итерационные методы решения систем;
- тема 5 — методы решения проблемы собственных значений матриц.

Часть I данного учебного пособия составили первые четыре темы из этого списка. Обширная тема 5, хотя и не включена, подготавливается здесь более основательным, чем обычно, изложением темы 3.

К ВЛА иногда относят тему «Линейное программирование» (ЛП), но чаще ее изучают как самостоятельный предмет, базирующийся на ВЛА, или включают ее в первый раздел таких дисциплин как «Методы оптимальных решений» или «Исследование операций». Поэтому тема ЛП в данное пособие не вошла, — по ней существует обширная литература, в частности, специальный компьютерный курс [76].

Часть II данного учебного пособия определена необходимостью изучения важной прикладной темы — метода наименьших квадратов (МНК). В статистической интерпретации МНК означает оценивание неизвестных параметров математической модели (идентификация) или неизвестного состояния системы (фильтрация и предсказание). При том, что пособие предлагает по этой теме необходимый теоретический материал, основное внимание оно уделяет все же практической реализации методов оценивания. Благодаря включению в данное пособие этой части, названной «Линейное оценивание»,

оно поддерживает изучение других прикладных дисциплин, таких как «Прикладная статистика», «Эконометрика» или «Стохастические модели, оценки и управление».

Понимание вопросов реализации методов оценивания, бесспорно, является решающим в приложениях теории. Однако, разработкам и исследованию численно устойчивых методов оценивания в отечественной научной литературе не уделено необходимое внимание. Еще меньшее внимание этому уделено в нашей учебной литературе, хотя численная неустойчивость теоретических методов оценивания существует как серьезный барьер для их практического применения. Акцентом на эффективную вычислительную реализацию методов оценивания в этом учебном пособии мы восполняем отмеченный пробел.

Учебная направленность всего предлагаемого материала выражена в том, что почти по каждой теме он включает объемный учебный лабораторный проект. Для подготовки к его выполнению, а также для контроля знаний студентов в течение семестра и на экзамене, в пособии разработано большое количество контрольных, тестовых заданий.

Значение «Численных методов» во многих областях науки и техники трудно переоценить, — оно растёт очень быстро. В связи с этим важно, чтобы студенты, готовящиеся стать специалистами в области математического моделирования, численных методов и комплексов программ, обладали подлинно глубокими знаниями, т.е. знаниями, имеющими для них практическую ценность в их будущей деятельности. Такое знание достигается не схоластическим изучением теории и не решением элементарных задач в классе, но реальной проектной работой по созданию серьезных программных продуктов высокого профессионального уровня, воплощающих эти численные методы. В связи с этим данное пособие использует так называемый *проектно-ориентированный подход* [134], при котором студенты получают необходимый теоретический материал и закрепляют эти знания в практических лабораторных проектах. После этого итоговая проверка знаний по курсу «Численные методы—I» проводится в форме решения задач на экзамене или же методом тестирования. Последнее предполагает умение быстро отыскивать правильный ответ, решать простые задачи и анализировать алгоритмы. Надеемся, что при таком подходе к преподаванию и изучению студент лучше поймет и оценит этот важный предмет.

Разработка данного пособия велась на протяжении многих лет преподавания этого предмета в Ульяновском государственном университете и в Ульяновском государственном техническом университете. В этой работе в разное время участвовали коллеги: профессор Г. Ю. Куликов (Dr. Gennady Kulikov, CEMAT (Centro de Matemática e Aplicações), Instituto Superior Técnico, LISBOA) — подготовил раздел по решению систем с разреженными матрицами; к. ф.-м. н., доцент Ю. В. Цыганова (Ульяновский государственный университет) — выверяла текст, уточняла наши критерии учебной работы студента (разд. 1) и помогала готовить материал (подразд. 13.7) по устойчивым алгоритмам фильтрации; аспирант К. В. Захаров применил алгоритмы оценивания для решения прикладной (нелинейной) задачи сопровождения судна на траектории с маневрированием (подразд. 13.12); к. ф.-м. н. М. В. Куликова (Dr. Maria Kulikova, Research Fellow at Universidade Técnica de Lisboa, Instituto Superior Técnico, CEMAT (Centro de Matemática e Aplicações), LISBOA) — готовила материал по блочным алгоритмам оценивания (разд. 14) и обосновала ряд новых алгоритмов (разд. А.1). Выражаю моим коллегам благодарность за их вклад в подготовку пособия, за исследовательскую работу и поддержку этого важного направления.

Я признателен уважаемым рецензентам за время, которое они уделили для прочтения рукописи, и за их полезные рекомендации, которые я старался, по возможности, учесть. Конечно, все ошибки, которых не удалось избежать в этой работе, — мои; мне будет приятно, если внимательные студенты их заметят.

Должен сказать, что включение обширного и в значительной мере справочного материала, составившего Часть II, поддержано интересом, который проявил к нему профессор доктор технических наук К. К. Васильев во время рецензирования одной из наших статей на тему вычислительных методов оценивания.

Прохождению материала в печать и его типографскому оформлению оказали содействие и помощь дирекция и сотрудники издательства УлГТУ. Всем, кто прямо или косвенно способствовал выходу в свет этого издания и улучшению его качества, приношу свою благодарность.

Ульяновск,
декабрь 2011



И. В. Семушин

1

ВВЕДЕНИЕ

1.1 Учебные цели студента

Мы живем в высокотехнологичном мире, в котором компьютер с каждым днем становится все более неотъемлемой частью. К тому же, наше общество все больше зависит от математики. Любая проблема решается лучше, если для нее найдена или построена подходящая (удовлетворительная, т. е. адекватная) математическая модель. При том, что для этого может потребоваться различный объем математических знаний, каждому, кто берется решать математически ориентированные проблемы, необходимо иметь навыки аналитического мышления.

Допустим, вы этим обладаете и смогли придать задаче математическую форму, т. е. дали правильную математическую постановку задачи; вопрос заключается в том, существует ли для этой задачи аналитическое решение? Действительность такова, что множество задач, для которых аналитическое решение существует и может быть найдено в конечной форме, невелико. Большинство задач требует численных методов для своего решения. Особенность же этой области знания в том, что «наилучшего» численного метода обычно не существует, так как в одних условиях лучшим будет один метод, в то время как для других условий успешнее работает другой метод. Понять и обосновать, какой же метод выбрать как лучший, можно лишь проводя вычислительные эксперименты с различными методами и для различных задач и условий. Для этого нужно уметь осознанно планировать вычислительные эксперименты, понимать и правильно программировать численные методы и эффективно использовать возможности вычислительной техники.

Таким образом, безусловно каждому из вас потребуется хорошая компьютерная подготовка, чтобы выжить на рынке труда и успешно функционировать среди грамотных компьютерных пользователей. Было бы образователь-

ным преступлением получить диплом выпускника университета и не иметь этих навыков хотя бы на удовлетворительном уровне. В конце концов, для этого вы и посещаете курсы информатики, программирования и численных методов.

Курс численных методов способствует этому, давая богатый набор индивидуальных заданий. В этом курсе мы преследуем три конкретные цели:

1. Студенты научатся выводить и доказывать положения математической теории численных методов, т. е. разовьют *навыки* аналитического мышления. Эти навыки будут проверены посредством *финального экзамена*.
2. Студенты увидят, как математика и компьютеры применяются к проблемам реального мира, т. е. научатся решать задачи. Эти *умения* будут проверены посредством *семестровых контрольных работ*, которые мы рассматриваем как часть распределенного по времени экзамена.
3. Студенты приобретут реальный *опыт* разработки компьютерных программ высокого (почти профессионального) уровня и применения компьютеров посредством написания, отладки и многочисленных прогонов своих программ. Приобретенный опыт будет проверен посредством выполнения *домашних заданий на лабораторные работы*, которые по своей значимости можно трактовать как своего рода учебные программные проекты.

1.2 Оценка работы студента

Выставление финальной оценки. Для оценки того, в какой мере студент приблизился к своим целям — навыки, умения и опыт, — мы применяем следующую *систему оценок*.

- Ваша оценка есть взвешенное среднее посещаемости (A), домашней работы (H) и экзаменов (E), где под «экзаменами» (см. подробнее ниже) понимается учет не только финального экзамена (во время сессии), но и контрольных работ в течение семестра:

5 % — посещаемость.

Этот вес действует только в случае, если вы посещаете занятия. Если вы пропускаете занятия, этот вес прогрессивно возрастает (см. ниже). Вы можете получить «неудовлетворительно» исключительно в результате низкой посещаемости.

30 % — домашняя работа.

65 % — экзамены.

Таким образом, итоговая оценка (final grade, FG) вычисляется по правилу:

$$FG = 0.05A + 0.30H + 0.65E, \quad (1.1)$$

где каждая составляющая:

$A \equiv$ attendance (посещаемость),

$H \equiv$ homework (домашняя работа) и

$E \equiv$ exams (экзамены)

выражается целым числом не выше 100 баллов.

- Эта итоговая оценка затем отображается на стандартную шкалу оценок:

$\{83 \div 100\} \Rightarrow$ «отлично»,

$\{71 \div 82\} \Rightarrow$ «хорошо»,

$\{56 \div 70\} \Rightarrow$ «удовлетворительно»,

$\{0 \div 55\} \Rightarrow$ «неудовлетворительно».

- Пример.

Студент Иван С. имеет следующие баллы:

$A = 90$, $H = 87$, $E = 83$. Тогда $0.05 \times 90 + 0.30 \times 87 + 0.65 \times 83 = 84.6$.

Следовательно, Иван заработал «отлично».

Имейте в виду, что оценки зарабатываются.

- Мы оставляем за собой право дать своего рода «плюс-минус дельта», если студент имеет оценку на границе между оценками (т. е. 82, 70 или 55). Если студент имеет 90 или выше за посещаемость ($A \geq 90$), сдал все домашние задания в установленный срок и проявил хорошее прилежание, тогда мы рассматриваем возможность выставления ему следующей более высокой оценки. Если же студент не продемонстрировал указанных выше качеств, возможность повышения оценки исключается. Мы не рассматриваем возможности повышения оценки, если до граничного значения не хватает хотя бы одного балла.
- Для итоговой оценки мы используем «симметричное» округление, т. е. округляем вверх, если младшая цифра есть 5 или выше, и вниз, если она меньше пяти. При вычислении средней оценки за домашнюю работу и средней за экзамены соответствующие числа H и E округляются до ближайшей десятой и затем они умножаются на свои весовые коэффициенты 0.05 и 0.30; после сложения по формуле (1.1) финальная оценка округляется.

Учет посещаемости (А)

- Каждое учебное занятие, в том числе лекция, начинается с вашей росписи в явочном листе. Поставить свою роспись — ваша личная ответственность. Отсутствие росписи означает ваше отсутствие на занятии. Чтобы ваше отсутствие было расценено как уважительное, вы должны известить об этом преподавателя своевременно (т.е. в течение одной недели до или после занятия). Пожалуйста, оставьте преподавателю телефонное сообщение на рабочий телефон (секретарю кафедры) или записку.
- Ваша оценка за посещаемость будет определена по табл. 1.1.

Таблица 1.1. Влияние неуважительных пропусков на оценку

Число неуважительных пропусков ^а	Балл А	Вклад в FG , вашу итоговую оценку
0	100	+5
1	90	+4.5
2	50	+2.5
3	0	+0
4	−50	−2.5
5	−100	−5
6	−150	−7.5
7	−200	−10
8	−400	−20
9	−600	−30
10	−800	−40

^а Неуважительный пропуск есть пропуск занятия, который не связан с болезнью, с семейной утратой или с факультетским мероприятием.

При числе неуважительных пропусков выше десяти у вас нет шанса получить положительную итоговую оценку за весь курс.

- Вы можете иметь максимум 8 уважительных пропусков. *После этого все пропуски считаются неуважительными.*

- Для спортсмена пропуск занятия считается уважительным, если его тренер известит об этом преподавателя заранее в письменной форме. Если вы больны, позвоните на кафедру, чтобы преподавателя об этом известили. Любой такой пропуск будет неуважительным, если преподавателя не известят в течение одной недели после вашего пропуска занятия. Извещение следует делать в форме телефонного сообщения или записки секретарю кафедры. Ваше извещение должно содержать номер группы, день и время пропускаемого занятия, название предмета и, конечно, ваше имя и фамилию.

Домашняя работа (H)

- Вам будет предложен ряд домашних заданий, которые — по нашему предположению — вы выполните и сдадите. Баллы за отдельные задания складываются и тем самым образуют H , т. е. оценку за этот вид вашей учебной работы. Любая сдача домашнего задания позже установленного срока влечет уменьшение вашей оценки H на 10 баллов. За каждое невыполненное задание в H поступает 0.
- Домашние задания представляют собой задания на лабораторные работы (проекты). Обычно мы предлагаем выполнить 3 таких работы за семестр, т. е. выдаем 3 задания. Максимальное количество баллов H , которое можно заработать за всю домашнюю работу, составляет 100. Эти 100 баллов мы разделяем определенным образом между общим числом выданных домашних заданий.
- Предлагаемые каждому студенту 3 лабораторные работы на первый семестр покрывают три темы из списка на стр. 12, включенные в данное пособие. За выполненное безупречно и в полном объеме задание по теме № 1 (это любой, по выбору студента или преподавателя, лабораторный проект №№ 1, 2, 3 или 4) студент зарабатывает 50 баллов, причем по срокам это задание должно предшествовать всем последующим. Далее, за выполненное безупречно и в полном объеме задание по теме № 2 (это лабораторный проект № 5) студент зарабатывает 20 баллов, а за выполненное безупречно и в полном объеме задание по теме № 3 (это лабораторный проект № 6) — 30 баллов. Заработанное число баллов за каждое задание будет уменьшено, если защита работы не отвечает всем требованиям, изложенным в данном учебном пособии, или не демонстрирует самостоятельность выполнения. Следующие по номерам проекты предлагаются в рамках дополнительного семестра или как курсовые работы.

- Преподаватель, ведущий лабораторные занятия в дисплейном классе, назначит сроки сдачи лабораторных работ и на каждом занятии всегда с готовностью поможет вам, если вы ясно формулируете те конкретные вопросы, которые у вас возникли дома. Преподаватель, ведущий семинарские (практические) занятия, поможет вам и всей аудитории, когда вы будете у доски рассказывать, как вы понимаете и как дома программируете тот или иной алгоритм.

Экзамены (E)

- Ваша оценка за экзамены, т. е. величина E в составе финальной оценки, вычисляемой по формуле (1.1), будет определена как равномерно взвешенное среднее значение результатов P_{KP-1} , P_{KP-2} и P_{KP-3} трех письменных контрольных работ ($KP-1$), ($KP-2$) и ($KP-3$) в течение семестра и результата $P_{УЭ}$ устного экзамена ($УЭ$) во время экзаменационной сессии. Это означает, что

$$E = (P_{KP-1} + P_{KP-2} + P_{KP-3} + P_{УЭ}) / 4, \quad P_{KP-i}, P_{УЭ} \in [0, 100]. \quad (1.2)$$

При том, что контрольные работы письменно проверяют ваше умение решать задачи, устный экзамен есть всеобъемлющая проверка вашего знания основных положений теории, умения доказывать эти положения и делать из них логические выводы. Эти (письменная и устная) части экзамена в совокупности покрывают весь учебный курс. Для этого мы проводим обычно три контрольные работы за семестр.

- Все контрольные работы будут вам объявлены заранее — не позднее, чем за неделю. Если вы собираетесь пропустить контрольную работу (это должен быть уважительный пропуск), мы предпочтем, чтобы вы написали эту работу раньше назначенного срока. Если вы не сможете написать контрольную работу до назначенного срока, то примите все меры к тому, чтобы написать ее в течение недели после контрольного срока. По истечении недели после этого вы получите ноль. Вы также получите ноль за неуважительный пропуск контрольной работы.
- Мы переписываем и заменяем некоторые задания или делаем вариации в постановке экзаменационных вопросов по сравнению с теми, которые опубликованы в наших учебных пособиях, в предыдущем семестре или в типовой рабочей программе на нашем сайте. Об этом будет объявлено за две недели до контрольных работ или финального экзамена.

1.3 Кодекс студента

Академическая честность

- К сожалению, есть люди, не столь честные, как другие, и настолько, что мы должны пояснить, как будем действовать в этом случае.
- За любую контрольную работу, экзамен, программу или любой иной вид работы, который выполнен нечестно, вы получите ноль, и преподаватель будет беседовать с вами. Если такая проблема случится во второй раз, преподаватель направит вас к декану факультета, и вы снова заработаете ноль за этот вид работы. Если вопрос о нечестности возникнет в третий раз, то вы сразу заработаете «неудовлетворительно» за весь предмет и снова будете отправлены к декану.
- Что считается *академической нечестностью*, т.е. обманом? По общепринятому правилу, это означает найти кого-то другого, кто сделает за вас вашу работу, и выдать ее за вашу собственную. Это также включает получение и оказание посторонней помощи на экзамене или во время контрольной работы (от соседа или с помощью шпаргалки).
- *Наши экзамены — это всегда закрытая книга, закрытый конспект, закрытый сосед и открытый ум.* Если в этом правиле появятся какие-либо изменения, об этом будет объявлено заранее.
- Не пользуйтесь шпаргалками. Они приносят больше вреда, чем пользы. Ваше сознание будет раздвоено между попыткой сформулировать ответ и попыткой утаить факт пользования шпаргалкой. Обнаружить такое раздвоенное сознание не составляет никакого труда. Вы будете обескуражены еще больше самыми простыми вопросами экзаменатора.
- При выполнении домашних заданий приемлемо работать с кем-то еще, обсуждая трудные вопросы и помогая тем самым друг другу, но при этом вы должны сами делать свою работу. Например, при написании компьютерных программ вполне нормально — обсуждать синтаксис, детали задания или получать помощь по сообщениям об ошибке. Ненормально, если вы отдаете кому-то копию вашей программы. Неприемлемо, если кто-то другой пишет программу для вас. Недопустимо копировать работу предыдущего семестра.
- В курсовых работах — вообще, в любых письменных работах — *плагиатом* является дословное копирование части чужих трудов, таких как чья-то статья (в том числе и опубликованная в Интернете), книга

или энциклопедия, без использования кавычек и ссылки на источник. Обобщающие заключения и выводы, которые вы пишете, должны быть выражены вашими собственными словами.

- Нечестность, когда она появляется в домашней работе, не столь очевидна. Мы это вполне признаём. Но она так или иначе проявит себя на устном экзамене, так как ваш балл за домашнюю работу будет контрастировать с уровнем вашего ответа. Вы только навредите себе и осложните свое положение очевидной провинностью.

Поведение в аудитории

- Примите все меры к тому, чтобы приходить на занятия вовремя. Если вы все же опаздываете,
 - не спрашивайте разрешения войти и не извиняйтесь за опоздание,
 - не проходите на место перед передним рядом мест,
 - тихо займите ваше место,
 - для получения любого раздаточного материала (если он есть) дождитесь конца занятия,
 - не хлопайте дверью.
- Чтобы выйти из аудитории, просите разрешения.
- Поднимайте руку и ждите, когда на вас обратят внимание, перед тем как задать вопрос.
- Не разговаривайте в аудитории.
- Уберите за собой и поставьте стул в исходное положение.

Путь к успеху

- Приходите на занятие вовремя, принимайте в нем участие и ведите записи.
- Просматривайте задания до занятия.
- Проверяйте ваши записи после занятия.
- Вовремя выполняйте ваши задания.
- Не накапливайте задолженности по самостоятельной работе — чтению учебных материалов, по домашней работе и в целом — по учебе.

- Выполняйте рекомендации по подготовке к контрольным работам и к финальному экзамену. Убедитесь, что вы можете решать типовые задачи и доказывать теоремы, которые во время лекций были отмечены как упражнения для самостоятельной работы.
- Придерживайтесь твердой решимости добиться успеха.
- Если вам нужна помощь, получайте ее безотлагательно.
- Сохраняйте позитивное отношение к делу.

Обратная связь

- По окончании всего курса занятий заполните анонимно лист обратной связи, который выдаст преподаватель. В нем вы можете отметить как положительные, так и отрицательные, на ваш взгляд, стороны преподавания этого предмета в текущем семестре.
- Преподаватель периодически просматривает свой сайт. Вы можете посылать через него ваши кратко сформулированные предложения, которые, по вашему мнению, помогли бы вам повысить эффективность изучения этого предмета.

Добро пожаловать на наши занятия. Вместе мы рассчитываем на большой и продуктивный семестр!

1.4 Краткое описание курса

Этот примерный план (стр. 24) может незначительно варьироваться в соответствии с новыми образовательными стандартами.

Как бы ни называлась дисциплина, соответствующая курсу «Численные методы», — «Вычислительная математика» или «Методы вычислений», — главная отличительная особенность нашего курса и данного пособия выражается в следующем диалоге, который иногда возникает между Студентом и Экзаменатором (автором данного пособия):

СТУДЕНТ: Я знаю этот численный метод и хочу получить более высокую оценку.

ЭКЗАМЕНАТОР: Если вы хорошо знаете этот метод, так научите этому компьютер.

ЧИСЛЕННЫЕ МЕТОДЫ (АЛГЕБРЫ)				
Кредиты: 3 ^a		Семестр: 1 ^b	Обязательный: ДА	Экзамен
Формат:	Лекции	34 ч		
	Семинары	0 ч		
	Лабораторные работы	17 ч		
	Самостоятельная работа	49 ч		
Преподаватель: проф. И. В. Семушин, доц. Ю. В. Цыганова				
Содержание:				
Этот курс излагает численные методы решения задач линейной алгебры: решение систем уравнений, обращение матриц, вычисление определителей, решение нелинейных уравнений, аппроксимация функций и отыскание собственных значений матриц. Программа курса охватывает: методы исключения неизвестных, разложения Холецкого, ортогональные преобразования, итерационные методы, метод наименьших квадратов, устойчивые алгоритмы оценивания и обзор методов решения проблемы собственных значений.				
Ожидаемые результаты изучения: продемонстрировать —				
знание и понимание:	структуры погрешностей решения вычислительных задач; корректности задач; прямых и итерационных методов для систем уравнений; задачи и алгоритмов метода наименьших квадратов; проблемы собственных значений и основ ее решения.			
способность: (теоретические навыки)	понимать и формулировать основные численные процедуры и решать демонстрационные задачи; идентифицировать подходящие методы для конкретных задач линейной алгебры.			
способность: (практические навыки)	понимать реализацию и поведение численных методов и решений на практике; логически формулировать численные методы для решения задач на компьютере с применением языков программирования (C #, Pascal или C/C++).			
способность: (ключевые навыки)	изучать предмет самостоятельно; использовать литературные источники; использовать персональный компьютер для программирования; эффективно конспектировать материал и распоряжаться рабочим временем.			
Оценивание: 5 % за посещаемость (неуважительные пропуски прогрессивно штрафуются); 30 % за семестровые (домашние) задания, 65 % суммарно за три контрольные работы и финальный (устный) экзамен. Практическая работа: Выдаваемые индивидуальные задания включают программирование отдельных методов из числа методов, излагаемых в данном курсе.				
Рекомендуемые учебные материалы: Конспект лекций. В. В. Воеводин. Численные методы алгебры. Теория и алгоритмы. — М.: Наука, 1966. А. А. Самарский, А. В. Гулин. Численные методы. — М.: Наука, 1989. Дополнительное чтение: Н. Н. Калиткин. Численные методы. — М.: Наука, 1978. Н. С. Бахвалов. Численные методы. — М.: Наука, 1975. или: Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. Численные методы. — М.: Наука, 1987.				

^a Число кредитных часов (приравнивается числу аудиторных часов в неделю).^b Продолжительность курса.

I

Вычислительная линейная алгебра

2

СТАНДАРТНЫЕ АЛГОРИТМЫ LU -РАЗЛОЖЕНИЯ

2.1 Алгоритмы метода Гаусса

Обычный метод Гаусса, осуществляющий LU -разложение матрицы A [12, 5, 4, 9, 97], заключается в последовательном исключении переменных из уравнений системы

$$Ax = f. \quad (2.1)$$

На первом шаге для исключения первой переменной x_1 из всех уравнений, лежащих в системе ниже первого уравнения, первое уравнение

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = f_1 \quad (2.2)$$

объявляем *ведущим уравнением*. Это возможно только при $a_{11} \neq 0$. Тогда, разделив обе части (2.2) на a_{11} , это ведущее уравнение получим в виде, в котором коэффициент при x_1 окажется равен 1. Заметим, что эта 1 — строгая (т. е. не приближенная) величина. Это действие — деление уравнения на ведущий элемент (на первом шаге это $a_{11} \neq 0$) — удобно называть *нормировкой*.

Второе действие заключается в серии вычитаний ведущего уравнения из всех нижележащих уравнений, чтобы исключить из них неизвестную x_1 . Для этого умножаем пронормированное уравнение на a_{i1} и вычитаем результат из i -го уравнения системы (2.1), $i = 2, \dots, n$. На этом заканчивается первый шаг алгоритма. После первого шага система (2.1) приведена в виду:

$$\left. \begin{array}{l} 1 \cdot x_1 + a_{12}^{(1)}x_2 + \dots + a_{1n}^{(1)}x_n = f_1^{(1)}, \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = f_2^{(1)}, \\ \dots \\ a_{n2}^{(1)}x_2 + \dots + a_{nn}^{(1)}x_n = f_n^{(1)}. \end{array} \right\} \quad (2.3)$$

Это второе действие удобно называть *обновлением активной* подсистемы (активной подматрицы), т. е. той части системы уравнений (матрицы A), где еще будут продолжаться подобного рода действия.

На втором шаге метода Гаусса описанный алгоритм повторяем для переменной x_2 , т. е. берем систему (2.3), объявляем ведущим второе уравнение (для этого нужно иметь $a_{22}^{(1)} \neq 0$) и нормируем в ней второе уравнение. Получаем его в виде

$$1 \cdot x_2 + a_{23}^{(2)} x_3 + \dots + a_{2n}^{(2)} x_n = f_2^{(2)}$$

(верхний индекс в скобках указывает номер шага, после которого получен текущий результат). После этого исключаем переменную x_2 из оставшихся $n - 2$ уравнений системы (2.3). Таким образом, любой *полный шаг алгоритма метода Гаусса* состоит из двух действий: сначала нормировка ведущей строки матрицы, потом обновление (серия вычитаний) в активной подматрице.

После $n - 1$ полных шагов и n -го неполного шага (поскольку ниже n -го ведущего уравнения больше нет уравнений) получим две системы линейных алгебраических уравнений

$$Ly = f, \quad \bar{U}x = y, \quad (2.4)$$

эквивалентных исходной системе (2.1), где L — нижняя треугольная матрица и \bar{U} верхняя треугольная матрица, на диагонали которой стоят единицы¹. При этом k -й столбец матрицы L (его нетривиальная, т. е. ненулевая часть) запоминает числа для двух действий на k -м шаге алгоритма, а именно: элемент l_{kk} является ведущим элементом, производившим нормировку k -го уравнения, в то время как элементы l_{ki} , $i = k + 1, k + 2, \dots, n$ являются теми коэффициентами, на которые было умножено пронормированное ведущее (k -е) уравнение, чтобы результатом последующего его вычитания из нижележащих уравнений было исключение из них неизвестного x_k . Можно говорить, что роль матрицы L — сохранять «историю» нормировок и вычитаний в процессе исключения неизвестных по методу Гаусса. Роль матрицы \bar{U} — иная. Матрица \bar{U} представляет собою тот эквивалентный вид системы (2.1), который она приобретет по завершении этого процесса.

ОПРЕДЕЛЕНИЕ 2.1. Определители Δ_i подматриц, получаемых оставлением первых i строк и первых i столбцов матрицы, называют *главными минорами* матрицы.

¹ Здесь и далее черта над матрицей означает, что на главной диагонали стоят строгие единицы.

ТЕОРЕМА 2.1. Если все главные миноры матрицы A в системе (2.1) отличны от нуля, то процесс Гаусса исключения неизвестных, протекающий в *прямом направлении*, — начиная от первого неизвестного x_1 и от первого уравнения системы, — эквивалентен разложению $A = L\bar{U}$, которое существует и единственно с нижней треугольной невырожденной матрицей L и верхней треугольной матрицей \bar{U} с единичной диагональю.

ДОКАЗАТЕЛЬСТВО. Докажите самостоятельно индукцией по размеру n матрицы [12]. \square

После разложения вводят правую часть f данной системы (2.1) и находят вектор y решения называется *прямой подстановкой*:

$$y_i = \left(f_i - \sum_{j=1}^{i-1} l_{ij} y_j \right) l_{ii}^{-1}, \quad i = 1, 2, \dots, n.$$

Далее вторая система из (2.4) так же легко решается процедурой *обратной подстановки*:

$$x_i = y_i - \sum_{j=i+1}^n u_{ij} x_j, \quad i = n-1, \dots, 1, \quad x_n = y_n.$$

ЗАМЕЧАНИЕ 2.1. Пересчет элементов вектора f должен быть отложен, т. е. сначала пересчет коэффициентов матрицы A должен привести к разложению матрицы A в произведение матриц L и \bar{U} и только затем должна быть введена и соответственно обработана правая часть — вектор f . При этом вектор y замещает f и затем x замещает y , — *экономия памяти*.

Алгоритм $L\bar{U}$ -разложения матрицы A запишем следующим образом:

Для $k = 1$ до n
 Нормируем первую строку матрицы $A^{(k-1)}$.
 Для $i = k + 1$ до n
 Вычитаем первую строку матрицы $A^{(k-1)}$,
 умноженную на $a_{ik}^{(k-1)}$, из i -й строки.

Здесь $A^{(k-1)}$ означает активную подматрицу матрицы A после $(k-1)$ -го шага метода Гаусса, $k = 1, 2, \dots, n$, причем $A^{(0)} = A$.

Алгоритм $\bar{L}U$ -разложения матрицы A отличается только нормировкой элементов активной подматрицы:

Для $k = 1$ до $n - 1$

Нормируем первый столбец матрицы $A^{(k-1)}$.

Для $i = k + 1$ до n

Вычитаем первую строку матрицы $A^{(k-1)}$,
умноженную на $a_{ik}^{(k-1)}$, из i -й строки.

УПРАЖНЕНИЕ 2.1. Измените направление процесса Гаусса на *обратное*. Докажите, что если процесс Гаусса исключения неизвестных вести от последнего неизвестного x_n и от последнего уравнения системы, двигаясь вверх по нумерации уравнений и влево по нумерации исключаемых неизвестных, то получим разложение $A = U\bar{L}$ (если нормировать строки) и $A = \bar{U}L$ (если нормировать столбцы). Докажите соответствующие теоремы — аналоги теоремы 2.1. Для этого измените определение 2.1 главных миноров.

Описанные выше алгоритмы в том виде, в каком они приведены, на практике используются очень редко, т.е. только в том случае, если можно гарантировать, что в результате гауссова исключения на диагонали не появятся нулевые элементы. Однако это можно гарантировать только для матриц специального вида (например, для положительно определенных матриц, см. разд. 6), поэтому в общем случае используется метод Гаусса с выбором главного (ведущего) элемента.

2.2 Выбор ведущего элемента

ОПРЕДЕЛЕНИЕ 2.2. Матрицей перестановок P называют квадратную матрицу, в которой каждая строка, а также каждый столбец, содержит один ненулевой элемент, равный 1.

ОПРЕДЕЛЕНИЕ 2.3. Элементарной матрицей перестановок P_{ij} называют квадратную матрицу, полученную из единичной матрицы перестановкой двух строк: i -й и j -й, либо перестановкой двух столбцов: i -го и j -го (оба варианта перестановок дают один и тот же результат).

УПРАЖНЕНИЕ 2.2. Докажите справедливость следующих утверждений:

1. Произведение $P_{ij}A$ производит в A перестановку i -й и j -й строк.
2. Произведение AP_{ij} производит в A перестановку i -го и j -го столбцов.

3. $P_{ij}P_{ij} = I$ — свойство *идемпотентности* матриц P_{ij} .
4. Любая матрица перестановок P может быть разложена в произведение элементарных матриц перестановок.
5. $P^{-1} = P^T$ — свойство *ортогональности* матриц перестановок P .

ТЕОРЕМА 2.2. Если $\det A \neq 0$, то существуют матрицы перестановок P и Q , такие что все угловые (т. е. главные) миноры матрицы PA , равно как и матриц AP и APQ , отличны от нуля.

ДОКАЗАТЕЛЬСТВО. Докажите индукцией по размеру матрицы A [12]. \square

СЛЕДСТВИЕ 2.1. Если $\det A \neq 0$, то существуют матрицы перестановок P и Q , такие что следующие варианты разложения матрицы A существуют и единственны: $PA = L\bar{U}$, $PA = \bar{L}U$, $AP = L\bar{U}$, $AP = \bar{L}U$, $PAQ = L\bar{U}$, $PAQ = \bar{L}U$. \square

Отсюда видны *три стратегии выбора главного (ведущего) элемента*: по столбцу, по строке и по активной подматрице.

Стратегия I. Первая стратегия (*по столбцу*) подразумевает, что в качестве главного на k -м шаге метода Гаусса выбирается максимальный по модулю элемент первого столбца активной подматрицы. Затем этот элемент меняется местами с диагональным элементом, что соответствует перестановке строк матрицы $A^{(k-1)}$ и элементов вектора $f^{(k-1)}$. На самом деле строки матрицы $A^{(k-1)}$ и элементы вектора $f^{(k-1)}$ остаются на своих местах, а переставляются только элементы дополнительного вектора, в котором хранятся номера строк исходной матрицы, соответствующие номерам строк матрицы, т. е. элементы так называемого *вектора перестановок*. Все обращения к элементам матриц L , U и вектора f осуществляются через вектор перестановок.

Стратегия II. Следующая стратегия (*по строке*) заключается в выборе в качестве главного элемента максимального по модулю элемента первой строки активной подматрицы. Затем этот элемент обменивается местами с диагональным, что соответствует перестановке столбцов матрицы $A^{(k-1)}$ и элементов вектора x . Как и в предыдущем случае, реально обмениваются только элементы дополнительного вектора, в котором фиксируются перестановки столбцов матрицы A . Доступ к элементам матриц L , U и вектора x осуществляется с использованием этого вектора.

Стратегия III. Последняя стратегия выбора главного элемента (*по активной подматрице*) объединяет две первые стратегии. Здесь в качестве главного выбирается максимальный по модулю элемент активной подматрицы. В общем случае, чтобы поставить этот элемент на место диагонального, требуется обменивать столбцы и строки матрицы $A^{(k-1)}$, что связано с введением двух дополнительных векторов: в одном хранятся перестановки столбцов, а в другом — перестановки строк матрицы A .

Приведенные выше алгоритмы LU -разложения с учетом выбора главного элемента преобразуются к одному из следующих вариантов.

Алгоритм 1. $L\bar{U}$ -разложение по методу Гаусса с выбором главного элемента

Для $k = 1$ до n
 Выбираем главный элемент в $A^{(k-1)}$.
 Нормируем первую строку матрицы $A^{(k-1)}$.
 Для $i = k + 1$ до n
 Вычитаем первую строку матрицы $A^{(k-1)}$,
 умноженную на $a_{ik}^{(k-1)}$, из i -й строки.

Алгоритм 2. $\bar{L}U$ -разложение по методу Гаусса с выбором главного элемента

Для $k = 1$ до $n - 1$
 Выбираем главный элемент в $A^{(k-1)}$.
 Нормируем первый столбец матрицы $A^{(k-1)}$.
 Для $i = k + 1$ до n
 Вычитаем первую строку матрицы $A^{(k-1)}$
 умноженную на $a_{ik}^{(k-1)}$ из i -й строки.

Вышеприведенные алгоритмы называют *исключением по столбцам*, так как они исключают x_k из всей поддиагональной части k -го столбца.

ЗАМЕЧАНИЕ 2.2. Во всех алгоритмах должно быть выполнено *требование к реализации*: все действия должны выполняться в одном и том же массиве чисел. Например, в Алгоритме 1 сначала $A^{(0)} = A$, а в конце на месте этой матрицы получаются нетривиальные элементы матриц L и \bar{U} .

ЗАМЕЧАНИЕ 2.3. Под *выбором главного элемента* здесь и далее понимается любая из описанных выше трех стратегий, которая применима.

ЗАМЕЧАНИЕ 2.4. При UL -разложении матрицы A все действия выполняются аналогично, но в *обратном порядке* нумерации строк и столбцов: снизу-вверх и справа-налево.

Наряду с гауссовым исключением по столбцам, представленным выше, возможно проводить гауссово *исключение по строкам*. Это такая разновидность метода Гаусса, в которой на каждом шаге исключения изменяется одна строка — первая строка активной подматрицы. Рассмотрим *гауссово исключение по строкам с выбором главного элемента по строке*.

Выполняем i -й шаг, т. е. работаем с i -й строкой матрицы. В ней еще не было ни одного исключения неизвестных. Первое действие: из i -й строки вычитаем первую, умноженную на a_{i1} ; затем из i -й строки вычитаем вторую, умноженную на a_{i2} , и так далее; в завершение этой серии вычитаний из i -й строки вычитаем $(i-1)$ -ю, умноженную на $a_{i(i-1)}$. Второе действие: отыскиваем главный элемент в i -й строке и осуществляем (если надо) перестановку столбцов. Третье действие: i -ю строку нормируем (делим на ведущий элемент). Повторяя этот алгоритм n раз, получим $L\bar{U}$ -разложение матрицы AP . При $i = 1$ шаг, очевидно, неполный, т. е. без вычитаний.

Таким образом, отличие гауссова исключения по строкам от гауссова исключения по столбцам сводится в алгоритме к изменению порядка действий: сначала серия вычитаний, а затем — нормировка. Такая возможность (для варианта $A = L\bar{U}$) представлена в следующем алгоритме.

Алгоритм 3. $L\bar{U}$ -разложение по методу Гаусса (по строкам)

Для $k = 1$ до n
 Для $i = 1$ до $k - 1$
 Вычитаем i -ю строку матрицы A ,
 умноженную на a_{ki} , из k -й строки.
 Выбираем главный элемент в k -й строке.
 Нормируем k -ю строку матрицы A .

ЗАМЕЧАНИЕ 2.5. В описаниях алгоритмов упоминаются элементы матрицы A . Естественно, речь идет о текущих, а не об исходных значе-

ниях элементов, занимающих указанные позиции матрицы A . Это связано с выполнением требования к реализации (см. Замечание 2.2).

2.3 Компактные схемы

Следующей разновидностью метода Гаусса являются *компактные схемы*. Первая называется *компактной схемой Краута*, а вторую мы будем называть *компактной схемой Краута «строка за строкой»*. В схеме Краута на каждом шаге исключения изменяются только первый столбец и первая строка активной подматрицы. В схеме «строка за строкой» на k -м шаге изменяется только k -я строка матрицы A .

Выведем формулы схемы Краута для k -го шага. Предположим, что уже сделаны первые $k-1$ шагов, т. е. определены $k-1$ столбец матрицы L и $k-1$ строка матрицы \bar{U} . Из соотношения $A = L\bar{U}$ для (i, j) -го элемента имеем

$$a_{ij} = \sum_{p=1}^n l_{ip} u_{pj}. \quad (2.5)$$

В силу треугольности матриц L и \bar{U} при $p > i$ имеем $l_{ip} = 0$ и при $p > j$ имеем $u_{pj} = 0$. Тогда с учетом того, что $u_{kk} = 1$, для k -го столбца матрицы A находим

$$a_{ik} = l_{ik} + \sum_{p=1}^{k-1} l_{ip} u_{pk}, \quad i \geq k. \quad (2.6)$$

Из (2.6) следует

$$l_{ik} = a_{ik} - \sum_{p=1}^{k-1} l_{ip} u_{pk}, \quad i \geq k. \quad (2.7)$$

Следовательно, k -й столбец матрицы L становится известен. Теперь из (2.5) для k -й строки матрицы A имеем

$$a_{kj} = l_{kk} u_{kj} + \sum_{p=1}^{k-1} l_{kp} u_{pj}, \quad j > k. \quad (2.8)$$

Из (2.8) находим

$$u_{kj} = \left(a_{kj} - \sum_{p=1}^{k-1} l_{kp} u_{pj} \right) / l_{kk}, \quad j > k. \quad (2.9)$$

Таким образом, (2.9) дает способ нахождения k -й строки матрицы \bar{U} . В результате, зная первые $k - 1$ столбцов матрицы L и $k - 1$ строк матрицы \bar{U} , мы можем по формулам (2.7) и (2.9) определить k -й столбец матрицы L и затем k -ю строку матрицы \bar{U} . Первый столбец матрицы L определяется равенствами

$$l_{i1} = a_{i1}, \quad i = 1, 2, \dots, n.$$

Это следует из (2.7) и того, что первым столбцом матрицы \bar{U} является первый координатный вектор e_1 . Здесь, как обычно, предполагаем, что если нижний предел суммирования меньше верхнего, то значение суммы равно нулю. После этого в первом столбце выбираем главный элемент. Затем по формулам

$$u_{1j} = a_{1j}/l_{1j}, \quad j = 2, 3, \dots, n$$

вычисляем первую строку матрицы \bar{U} . Повторяя указанную последовательность действий n раз, с помощью формул (2.7) и (2.9) получаем $L\bar{U}$ -разложение матрицы A .

Алгоритм 4. $L\bar{U}$ -разложение по компактной схеме Краута

Для $k = 1$ до n

По формуле (2.7) вычисляем k -й столбец матрицы L .

Выбираем среди элементов k -го столбца главный элемент.

По формуле (2.9) вычисляем k -ю строку матрицы \bar{U} .

Чтобы получить метод Краута, дающий $\bar{L}U$ -разложение с выбором главного элемента по строке, достаточно поменять местами формулы (2.7) и (2.9), а также последовательность вычисления столбцов матрицы \bar{L} и строк матрицы U . Таким образом, на k -м шаге сначала по формуле

$$u_{kj} = a_{kj} - \sum_{p=1}^{k-1} l_{kp}u_{pj}, \quad j \geq k, \quad (2.10)$$

вычисляется строка матрицы U . Затем в этой строке выбирается главный элемент и находится столбец матрицы \bar{L} по следующей формуле:

$$l_{ik} = \left(a_{ik} - \sum_{p=1}^{k-1} l_{ip}u_{pk} \right) / u_{kk}, \quad i \geq k. \quad (2.11)$$

УПРАЖНЕНИЕ 2.3. Выведите расчетные формулы компактных схем для любого из альтернативных вариантов разложения: $A = U\bar{L}$ или $A = \bar{U}L$. Что изменяется? Дайте ответы в форме условных схем алгоритмов, представленных непосредственно до и после этого упражнения.

Алгоритм 5. $\bar{L}U$ -разложение по компактной схеме Краута

Для $k = 1$ до n

По формуле (2.10) вычисляем k -ю строку матрицы U .

Выбираем среди элементов k -й строки главный элемент.

По формуле (2.11) вычисляем k -й столбец матрицы \bar{L} .

Компактная схема «строка за строкой», дающая $L\bar{U}$ -разложение матрицы A , использует те же самые формулы (2.7) и (2.9). Меняется только последовательность вычисления элементов матрицы L . Рассмотрим подробнее.

Пусть уже обработаны по этому методу первые $k - 1$ строк матрицы A . Следовательно, мы имеем $k - 1$ строку матрицы L и $k - 1$ строку матрицы \bar{U} . Далее по формулам (2.7) вычисляем ненулевые элементы k -й строки матрицы L . По формулам (2.9) без деления на диагональный элемент l_{kk} вычисляем ненулевые элементы k -й строки матрицы \bar{U} . Затем среди вновь вычисленных элементов, от диагонального до n -го, определяем главный элемент, меняем его местами с диагональным и делим элементы k -й строки матрицы \bar{U} на этот элемент. В результате получаем требуемое разложение.

Алгоритм 6. $L\bar{U}$ -разложение по компактной схеме «строка за строкой»

Для $k = 1$ до n

По формуле (2.7) вычисляем элементы k -й строки матрицы L .

По формуле (2.9) без деления на диагональный элемент l_{kk} , вычисляем k -ю строку матрицы \bar{U} .

Среди элементов k -й строки (от диагонального до n -го) определяем главный элемент.

Делим на главный элемент k -ю строку матрицы \bar{U} .

2.4 Алгоритмы метода Жордана

К последней группе методов исключения относятся *алгоритмы метода Жордана*. Эти алгоритмы используют те же самые формулы, что и обычный метод Гаусса, но в отличие от него на k -м шаге метода Жордана пересчитывают все строки матрицы A , а не только строки, находящиеся ниже ведущей строки. Это означает *полное исключение* i -й переменной из всех уравнений, кроме i -го. Таким образом, метод Жордана формально дает решение системы линейных алгебраических уравнений за один проход.

ТЕОРЕМА 2.3. Выполнение действий полного исключения в том же массиве, где первоначально располагалась матрица A , дает то же самое разложение $A = L\bar{U}$, что и метод Гаусса, но существенно в другом виде, а именно: матрица L получается, как и в методе Гаусса, но матрица \bar{U} оказывается полученной не в «чистом виде», а в виде обратной матрицы \bar{U}^{-1} и с той особенностью, что все знаки элементов матрицы \bar{U}^{-1} выше диагонали имеют неправильные (противоположные) знаки.

ДОКАЗАТЕЛЬСТВО. Нужно воспользоваться определениями элементарных матриц специального вида [10] и их свойствами. Приведем эти определения и свойства и затем продолжим доказательство.

ОПРЕДЕЛЕНИЕ 2.4. *Элементарная матрица* E есть любая матрица вида $E = I + B$, где $\text{rank } B = 1$.

УПРАЖНЕНИЕ 2.4. Докажите, что $E = I + xy^T$, где x и y — некоторые векторы.

ОПРЕДЕЛЕНИЕ 2.5. Введем следующие *специальные элементарные матрицы*:

D_k — диагональная k -матрица. Имеет единичную диагональ, кроме k -го элемента, который не тривиален, т. е. не равен нулю или единице.

L_k^C — столбцово-элементарная нижняя треугольная k -матрица. Имеет единичную диагональ и нетривиальные элементы только в k -м столбце.

L_k^R — строчно-элементарная нижняя треугольная k -матрица. Имеет единичную диагональ и нетривиальные элементы только в k -й строке.

U_k^C — столбцово-элементарная верхняя треугольная k -матрица. Имеет единичную диагональ и нетривиальные элементы только в k -м столбце.

Таблица 2.1. Свойства специальных элементарных матриц

Коммутативность в операции умножения	
$L_i^C D_j = D_j L_i^C, \quad i > j$	$U_i^C D_j = D_j U_i^C, \quad i < j$
$L_i^R D_j = D_j L_i^R, \quad i < j$	$U_i^R D_j = D_j U_i^R, \quad i > j$
$L_i^C U_j^C = U_j^C L_i^C, \quad i \geq j$	$L_i^R U_j^R = U_j^R L_i^R, \quad i \leq j$
$L_i^C U_i^C = U_i^C L_i^C = T_i^C$	$L_i^R U_i^R = U_i^R L_i^R = T_i^R$
Операция обращения матриц	
D_i^{-1} получается из D_i заменой нетривиального элемента d_{ii} на d_{ii}^{-1} с сохранением знака этого элемента	E^{-1} , где $E \in \{L_k^C, L_k^R, U_k^C, U_k^R, T_k^C, T_k^R\}$ получается из E заменой знаков нетривиальных элементов на противоположные
Применимость правила суперпозиции вместо перемножения матриц	
$L_i^C L_j^C L_k^C, \quad i < j < k$	$L_i^R L_j^R L_k^R, \quad i < j < k$
$U_i^C U_j^C U_k^C, \quad i > j > k$	$U_i^R U_j^R U_k^R, \quad i > j > k$
$L = D_1 L_1^C D_2 L_2^C \cdots D_{n-1} L_{n-1}^C D_n L_n^C$	$L = L_1^R D_1 L_2^R D_2 \cdots L_{n-1}^R D_{n-1} L_n^R D_n$
$U = D_n U_n^C D_{n-1} U_{n-1}^C \cdots D_2 U_2^C D_1 U_1^C$	$U = U_n^R D_n U_{n-1}^R D_{n-1} \cdots U_2^R D_2 U_1^R D_1$

U_k^R — строчно-элементарная верхняя треугольная k -матрица. Имеет единичную диагональ и нетривиальные элементы только в k -й строке.

T_k^C — полно-столбцово-элементарная верхняя треугольная k -матрица. Содержит единичную диагональ и нетривиальные элементы только в k -м столбце.

T_k^R — полно-строчно-элементарная верхняя треугольная k -матрица. Имеет единичную диагональ и нетривиальные элементы только в k -й строке.

УПРАЖНЕНИЕ 2.5. Докажите свойства этих элементарных матриц, приведенные в табл. 2.1.

Продолжим доказательство теоремы 2.3, прерванное на стр. 37.

Приведение данной матрицы A к единичной матрицы, составляющее суть исключения по методу Гаусса-Жордана, запишем в терминах операций с введенными специальными элементарными матрицами:

$$A^{(n+1)} = (T_n^C)^{-1} D_n^{-1} \cdots (T_2^C)^{-1} D_2^{-1} (T_1^C)^{-1} D_1^{-1} A = I. \quad (2.12)$$

К результату (2.12) приводит следующий *алгоритм Гаусса-Жордана*:

Начальное значение: $A^{(1)} = A$.

Для $k = 1$ до n выполнять

$$A^{(k+1)} = (T_k^C)^{-1} D_k^{-1} A^{(k)},$$

где элементы

$$D_k^{-1}(k, k) = 1/A^{(k)}(k, k),$$

$$T_k^C(i, k) = A^{(k)}(k, k)(i, k), \quad i = 1, 2, \dots, n, \quad i \neq k$$

суть множители для нормировки и вычитаний, соответственно.

Множители, возникающие в этом алгоритме, образуют так называемую *таблицу множителей*. По существу, это матрица, которая займет место исходной матрицы A по окончании всего алгоритма:

$$\begin{bmatrix} D_1^{-1}(1, 1) & T_1^C(1, 2) & T_1^C(1, 3) & \cdots \\ T_1^C(2, 1) & D_1^{-1}(2, 2) & T_1^C(2, 3) & \cdots \\ T_1^C(3, 1) & T_1^C(3, 2) & D_1^{-1}(3, 3) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (2.13)$$

Воспользуемся свойством в четвертой строке табл. 2.1, $T_i^C = L_i^C U_i^C$ в его инверсной форме $(T_i^C)^{-1} = (U_i^C)^{-1} (L_i^C)^{-1}$, и подставим его в (2.12), а также свойствами коммутативности из этой таблицы. Это дает возможность перегруппировать сомножители в (2.12) следующим образом:

$$[(U_n^C)^{-1} \cdots (U_2^C)^{-1} (U_1^C)^{-1}] \cdot [(L_n^C)^{-1} D_n^{-1} \cdots (L_2^C)^{-1} D_2^{-1} (L_1^C)^{-1} D_1^{-1}] A = I.$$

Второй сомножитель в квадратных скобках совпадает с матрицей L^{-1} для $L\bar{U}$ -разложения матрицы A . Первый сомножитель в квадратных скобках есть матрица U^{-1} для этого разложения. Правило суперпозиции, согласно табл. 2.1, действует для первого сомножителя (в нем индексы матриц убывают слева-направо) и не действует для второго сомножителя. Однако для $L = D_1 L_1^C D_2 L_2^C \cdots D_{n-1} L_{n-1}^C D_n L_n^C$ правило суперпозиции действует. Суперпозиция, т. е. постановка элементов матриц на принадлежащие им позиции, реализуется автоматически по ходу алгоритма. Поэтому в нижней треугольной части матрицы (2.13) (кроме диагонали) образуется матрица L , диагональные элементы матрицы L представлены на диагонали, но в инверсном виде (там — обратные значения этих элементов), а выше диагонали расположены элементы той матрицы, для которой действует правило суперпозиции, т. е. элементы матрицы U^{-1} . В работе алгоритма перемены знаков у этих

элементов не совершались (что надо делать при обращении элементарных матриц E , согласно табл. 2.1), поэтому эти знаки — неверные. \square

Чтобы выполнить разложение по методу Жордана, надо воспользоваться следующим алгоритмом. На первом шаге в активной подматрице $A^0 = A$ выбирается главный элемент. Затем первая строка нормируется, домножается на a_{i1} и вычитается из i -й строки, $i = 2, 3, \dots, n$. На втором шаге главный элемент определяется среди элементов активной подматрицы $A^{(1)}$. Потом вторая строка нормируется и после домножения на a_{i2} вычитается из i -й, где $i = 1, 3, \dots, n$. В общем случае на k -м шаге в подматрице $A^{(k-1)}$ выбирается главный элемент. Затем k -я строка нормируется, домножается на a_{ik} и вычитается из i -й, где $i = 1, \dots, k-1, k+1, \dots, n$. В результате, чтобы получить требуемое разложение, остается поменять знак на противоположный у всех элементов, лежащих выше главной диагонали.

Алгоритм 7. « $L\bar{U}^{-1}$ -разложение» $A = L\bar{U}$ по методу Жордана

Для $k = 1$ до n
 Выбираем главный элемент в $A^{(k-1)}$.
 Нормируем первую строку матрицы $A^{(k-1)}$.
 Для $i = 1$ до $k-1$
 Вычитаем первую строку матрицы $A^{(k-1)}$,
 умноженную на $a_{ik}^{(k-1)}$, из i -й строки.
 Для $i = k+1$ до n
 Вычитаем первую строку матрицы $A^{(k-1)}$,
 умноженную на $a_{ik}^{(k-1)}$, из i -й строки.
 Для $i = 1$ до n
 Для $j = i+1$ до n
 $a_{ij} = -a_{ij}$

ЗАМЕЧАНИЕ 2.6. Термин « $L\bar{U}^{-1}$ -разложение», который мы используем здесь для краткости в применении к методу Жордана, не должен вводить в заблуждение. Он самом деле отыскивает $L\bar{U}$ -разложение матрицы A , но при его выполнении в одном и том же массиве дает вместо матрицы \bar{U} обратную матрицу \bar{U}^{-1} , причем в следующем виде: единицы главной диагонали матрицы \bar{U}^{-1} не хранятся, а все другие элементы этой матрицы получают с противоположными знаками.

УПРАЖНЕНИЕ 2.6. Объясните, как следует понимать словосочетание « $\bar{L}^{-1}U$ -разложение» Жордана. Докажите, что в этом случае выполняется разложение $A = \bar{L}U$, но матрица \bar{L} получается в виде обратной матрицы с неправильными знаками ее внедиагональных элементов.

ЗАМЕЧАНИЕ 2.7. Чтобы сэкономить процессорное время, целесообразно везде пользоваться обратными величинами ведущих элементов, как это сделано на диагонали в таблице множителей (2.13).

2.5 Вычисление обратной матрицы

Есть два способа вычисления *обратной матрицы* A^{-1} : через решение системы $Ax = f$ с различными правыми частями f и непосредственно через разложение матрицы A в произведение треугольных матриц. В первом способе правая часть f последовательно пробегает значения столбцов e_i единичной матрицы I , при этом для каждой из них найденное решение x системы $Ax = f$ образует i -й столбец искомой матрицы A^{-1} . Это, очевидно, соответствует решению матричного уравнения $AX = I$, так как $X = A^{-1}$.

Второй способ основан на том, что если $A = L\bar{U}$, то $A^{-1} = \bar{U}^{-1}L^{-1}$. Это называют *элиминативной формой обратной матрицы* [10], так как здесь A^{-1} находят непосредственно по разложению $A = L\bar{U}$, которое само по себе эквивалентно процедуре гауссова исключения (*elimination*) неизвестных. Рассмотрим этот способ и характеризуем особенности его программной реализации более подробно. Для численной иллюстрации рассмотрим следующий пример.

ПРИМЕР 2.1. Пусть для данной матрицы A найдено $A = L\bar{U}$:

$$A = \begin{bmatrix} 2 & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}, \quad L = \begin{bmatrix} 2 & & & \\ 1 & 2 & & \\ 3 & 2 & 3 & \\ 2 & 1 & 2 & 4 \end{bmatrix}, \quad \bar{U} = \begin{bmatrix} 1 & 2 & -2 & 3 \\ & 1 & 2 & -1 \\ & & 1 & -2 \\ & & & 1 \end{bmatrix}.$$

Известная особенность реализации такого разложения заключается в том, что результат разложения замещает исходную матрицу, т. е. имеем

исходный массив \Rightarrow результирующий массив

$$\begin{bmatrix} 2 & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 2 & -2 & 3 \\ 1 & 2 & 2 & -1 \\ 3 & 2 & 3 & -2 \\ 2 & 1 & 2 & 4 \end{bmatrix}. \quad (2.14)$$

Следовательно, до начала вычисления обратной матрицы A^{-1} в наличии имеем две матрицы: матрицу L — в нижней треугольной части массива вместе с диагональю, матрицу \bar{U} — в верхней треугольной части массива без единичной (известной по умолчанию) диагонали. Запишем $A^{-1} = \bar{U}^{-1} \times L^{-1}$, где символ \times обозначает процедуру перемножения треугольных матриц \bar{U}^{-1} и L^{-1} в указанном порядке. Тем самым отмечаем, что это должна быть специальная, а не общая процедура, экономящая время вычислений за счет исключения операций умножения на заведомо нулевые элементы сомножителей \bar{U}^{-1} и L^{-1} .

Сомножители \bar{U}^{-1} и L^{-1} нужно вычислять также по специальным процедурам, для которых исходные данные \bar{U} и L берутся из массива, названного в выражении (2.14) *результатирующим массивом* после факторизации $A = L\bar{U}$. Результаты \bar{U}^{-1} и L^{-1} работы этих процедур записываются в этот же результирующий массив.

Вывод алгоритмов процедур для L^{-1} и для \bar{U}^{-1} основан на свойствах элементарных треугольных матриц, в частности, на свойстве «суперпозиции вместо перемножения». Для L это свойство означает, что произведение $L = L_1 L_2 L_3 L_4$ может быть получено не перемножением элементарных матриц L_1, L_2, L_3, L_4 , а суперпозицией (постановкой на свои позиции) нетривиальных столбцов элементарных матриц-сомножителей:

$$\begin{array}{c} L \\ \left[\begin{array}{cccc} 2 & & & \\ 1 & 2 & & \\ 3 & 2 & 3 & \\ 2 & 1 & 2 & 4 \end{array} \right] \end{array} = \begin{array}{c} L_1 \\ \left[\begin{array}{cccc} 2 & & & \\ 1 & 1 & & \\ 3 & & 1 & \\ 2 & & & 1 \end{array} \right] \end{array} \begin{array}{c} L_2 \\ \left[\begin{array}{cccc} 1 & & & \\ & 2 & & \\ 2 & 1 & & \\ 1 & & 1 & \end{array} \right] \end{array} \begin{array}{c} L_3 \\ \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 3 & \\ & & 2 & 1 \end{array} \right] \end{array} \begin{array}{c} L_4 \\ \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 4 \end{array} \right] \end{array}.$$

Согласно правилу обращения произведения матриц, L^{-1} найдем как результат перемножения следующих обратных матриц:

$$\begin{array}{c} L_4^{-1} \\ \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 4 \end{array} \right]^{-1} \end{array} \times \begin{array}{c} L_3^{-1} \\ \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 3 & \\ & & 2 & 1 \end{array} \right]^{-1} \end{array} \times \begin{array}{c} L_2^{-1} \\ \left[\begin{array}{cccc} 1 & & & \\ & 2 & & \\ & 2 & 1 & \\ & 1 & & 1 \end{array} \right]^{-1} \end{array} \times \begin{array}{c} L_1^{-1} \\ \left[\begin{array}{cccc} 2 & & & \\ 1 & 1 & & \\ 3 & & 1 & \\ 2 & & & 1 \end{array} \right]^{-1} \end{array}.$$

Так как индексы у элементарных нижнетреугольных матриц здесь слева направо уже не возрастают, а убывают, операция перемножения \times матриц не может быть заменена суперпозицией. В программной реализации этот

символ \times должен соответствовать некоторой специальной вычислительной процедуре. Все исходные данные для этой процедуры уже предоставлены полученным разложением (2.14). Действительно, инверсию элементарных матриц, показанных в последнем выражении, получают в самой процедуре применением простых операций: сначала диагональный элемент из нетривиального столбца каждой элементарной матрицы заменяют на обратный по величине; затем полученное число берут с противоположным знаком и умножают на каждый поддиагональный элемент. Эти действия соответствуют указанному выражению, представленному в следующем виде:

$$\begin{bmatrix} & L_4^{-1} & & \\ 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/4 \end{bmatrix} \times \begin{bmatrix} & L_3^{-1} & & \\ & 1 & & \\ & & 1/3 & \\ & & -2/3 & 1 \end{bmatrix} \times \begin{bmatrix} & L_2^{-1} & & \\ & & 1/2 & \\ & -2/2 & 1 & \\ & -1/2 & & 1 \end{bmatrix} \times \begin{bmatrix} & L_1^{-1} & & \\ & & 1/2 & \\ & -1/2 & 1 & \\ & -3/2 & & 1 \\ & -2/2 & & & 1 \end{bmatrix}.$$

В действительности это означает, что сначала — на этапе **1** — результирующий массив из (2.14) пересчитывают по указанным правилам, чтобы найти L^{-1} , приводя этот массив к следующему стартовому виду:

$$\left[\begin{array}{c|ccc} 2 & 2 & -2 & 3 \\ 1 & 2 & 2 & -1 \\ 3 & 2 & 3 & -2 \\ 2 & 1 & 2 & 4 \end{array} \right] \xRightarrow{\textbf{1}} \left[\begin{array}{c|ccc} 1/2 & 2 & -2 & 3 \\ -1/2 & 1/2 & 2 & -1 \\ -3/2 & -2/2 & 1/3 & -2 \\ -2/2 & -1/2 & -2/3 & 1/4 \end{array} \right]. \quad (2.15)$$

Чтобы понять, как в этом массиве должна работать процедура вычисления матрицы L^{-1} , рассмотрим произведение матриц перед выражением (2.15) и формально будем перемножать матрицы $L_4^{-1}, L_3^{-1}, L_2^{-1}, L_1^{-1}$ справа налево, т. е. вычислим $L_4^{-1}(L_3^{-1}(L_2^{-1}L_1^{-1}))$. Процесс такого поэтапного перемножения отразим в табл. 2.2.

Из табл. 2.2 видно, что после получения (2.15), т. е. на этапе **2**, пересчитывают только элементы a_{21}, a_{31} и a_{41} . В данном случае имеем

$$\left[\begin{array}{c|ccc} 1/2 & 2 & -2 & 3 \\ -1/2 & 1/2 & 2 & -1 \\ -3/2 & -2/2 & 1/3 & -2 \\ -2/2 & -1/2 & -2/3 & 1/4 \end{array} \right] \xRightarrow{\textbf{2}} \left[\begin{array}{c|ccc} 1/2 & 2 & -2 & 3 \\ -1/4 & 1/2 & 2 & -1 \\ -1 & -1 & 1/3 & -2 \\ -3/4 & -1/2 & -2/3 & 1/4 \end{array} \right].$$

Таблица 2.2. Поэтапное перемножение $L_4^{-1}(L_3^{-1}(L_2^{-1}L_1^{-1}))$

$$\begin{aligned}
 & \begin{matrix} L_4^{-1} \\ \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/4 \end{bmatrix} \end{matrix} \times \begin{matrix} L_3^{-1} \\ \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1/3 & \\ & & -2/3 & 1 \end{bmatrix} \end{matrix} \times \begin{matrix} L_2^{-1} \\ \begin{bmatrix} 1 & & & \\ & 1/2 & & \\ & -2/2 & 1 & \\ & -1/2 & & 1 \end{bmatrix} \end{matrix} \times \begin{matrix} L_1^{-1} \\ \begin{bmatrix} 1/2 & & & \\ -1/2 & 1 & & \\ -3/2 & & 1 & \\ -2/2 & & & 1 \end{bmatrix} \end{matrix} \\
 & \begin{matrix} L_3^{-1} \\ \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1/3 & \\ & & -2/3 & 1 \end{bmatrix} \end{matrix} \times \begin{matrix} (L_2^{-1}L_1^{-1}) \\ \begin{bmatrix} 1/2 & & & \\ -1/4 & 1/2 & & \\ -1 & -1 & 1 & \\ -3/4 & -1/2 & & 1 \end{bmatrix} \end{matrix} \Leftarrow \textcircled{2} \\
 & \begin{matrix} L_4^{-1} \\ \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/4 \end{bmatrix} \end{matrix} \times \begin{matrix} (L_3^{-1}(L_2^{-1}L_1^{-1})) \\ \begin{bmatrix} 1/2 & & & \\ -1/4 & 1/2 & & \\ -1/3 & -1/3 & 1/3 & \\ -1/12 & 1/6 & -2/3 & 1 \end{bmatrix} \end{matrix} \Leftarrow \textcircled{3} \\
 & \begin{matrix} L_4^{-1}(L_3^{-1}(L_2^{-1}L_1^{-1})) \\ \begin{bmatrix} 1/2 & & & \\ -1/4 & 1/2 & & \\ -1/3 & -1/3 & 1/3 & \\ -1/48 & 1/24 & -1/6 & 1/4 \end{bmatrix} \end{matrix} \Leftarrow \textcircled{4}
 \end{aligned}$$

Далее видно, что на этапе $\textcircled{3}$ пересчитывают только a_{31} , a_{41} , a_{32} и a_{42} , т. е.

$$\left[\begin{array}{cccc} 1/2 & \boxed{2} & -2 & 3 \\ -1/4 & \boxed{1/2} & 2 & -1 \\ -1 & -1 & \boxed{1/3} & -2 \\ -3/4 & -1/2 & -2/3 & \boxed{1/4} \end{array} \right] \xRightarrow{\textcircled{3}} \left[\begin{array}{cccc} 1/2 & \boxed{2} & -2 & 3 \\ -1/4 & \boxed{1/2} & 2 & -1 \\ -1/3 & -1/3 & \boxed{1/3} & -2 \\ -1/12 & 1/6 & -2/3 & \boxed{1/4} \end{array} \right],$$

и на этапе $\textcircled{4}$ пересчитываются только элементы a_{41} , a_{42} и a_{43} , т. е.

$$\left[\begin{array}{cccc} 1/2 & \boxed{2} & -2 & 3 \\ -1/4 & \boxed{1/2} & 2 & -1 \\ -1/3 & -1/3 & \boxed{1/3} & -2 \\ -1/12 & 1/6 & -2/3 & \boxed{1/4} \end{array} \right] \xRightarrow{\textcircled{4}} \left[\begin{array}{cccc} 1/2 & \boxed{2} & -2 & 3 \\ -1/4 & \boxed{1/2} & 2 & -1 \\ -1/3 & -1/3 & \boxed{1/3} & -2 \\ -1/48 & 1/24 & -1/6 & \boxed{1/4} \end{array} \right].$$

Построение алгоритма вычисления матрицы \bar{U}^{-1} проводится аналогично. Для \bar{U} свойство суперпозиции означает, что произведение $\bar{U} = \bar{U}_4\bar{U}_3\bar{U}_2\bar{U}_1$ может быть получено не перемножением элементарных матриц $\bar{U}_4, \bar{U}_3, \bar{U}_2, \bar{U}_1$,

а суперпозицией (постановкой на свои позиции) нетривиальных столбцов элементарных матриц-сомножителей, т. е.

$$\begin{array}{c} \bar{U} \\ \left[\begin{array}{cccc} 1 & 2 & -2 & 3 \\ & 1 & 2 & -1 \\ & & 1 & -2 \\ & & & 1 \end{array} \right] \end{array} = \begin{array}{c} \bar{U}_4 \\ \left[\begin{array}{ccc} 1 & & 3 \\ & 1 & -1 \\ & & 1 & -2 \\ & & & 1 \end{array} \right] \end{array} \begin{array}{c} \bar{U}_3 \\ \left[\begin{array}{ccc} 1 & -2 & \\ & 1 & 2 \\ & & 1 \end{array} \right] \end{array} \begin{array}{c} \bar{U}_2 \\ \left[\begin{array}{ccc} 1 & 2 & \\ & 1 & \\ & & 1 & 1 \end{array} \right] \end{array},$$

где учтено, что крайний справа сомножитель \bar{U}_1 равен единичной матрице, так как по построению \bar{U} — верхнетреугольная матрица с единичной диагональю.

Руководствуясь правилом обращения произведения матриц, \bar{U}^{-1} найдем как результат перемножения следующих обратных матриц:

$$\begin{array}{c} \bar{U}_2^{-1} \\ \left[\begin{array}{ccc} 1 & -2 & \\ & 1 & \\ & & 1 \end{array} \right] \end{array} \times \begin{array}{c} \bar{U}_3^{-1} \\ \left[\begin{array}{ccc} 1 & 2 & \\ & 1 & -2 \\ & & 1 \end{array} \right] \end{array} \times \begin{array}{c} \bar{U}_4^{-1} \\ \left[\begin{array}{ccc} 1 & -3 & \\ & 1 & 1 \\ & & 1 & 2 \\ & & & 1 \end{array} \right] \end{array}. \quad (2.16)$$

Здесь уже отражен тот факт, что обращение элементарных матриц $\bar{U}_4, \bar{U}_3, \bar{U}_2$ сводится в простой замене знака на противоположный у каждого нетривиального (внедиагонального) элемента.

В действительности же это означает, что сначала — на этапе ① для верхнетреугольной части и на уже рассмотренном этапе ❶ для нижнетреугольной части (2.15) — результирующий массив из (2.14) пересчитывают по указанным правилам для вычисления \bar{L}^{-1} и \bar{U}^{-1} , приводя его к следующему стартовому виду:

$$\left[\begin{array}{c|ccc} 2 & 2 & -2 & 3 \\ 1 & 2 & 2 & -1 \\ 3 & 2 & 3 & -2 \\ 2 & 1 & 2 & 4 \end{array} \right] \xrightarrow[\text{❶}]{\text{①}} \left[\begin{array}{c|ccc} 1/2 & -2 & 2 & -3 \\ -1/2 & 1/2 & -2 & 1 \\ -3/2 & -2/2 & 1/3 & 2 \\ -2/2 & -1/2 & -2/3 & 1/4 \end{array} \right]. \quad (2.17)$$

Одинаковые номера этапов ① и ❶ здесь подсказывают, что эти подготовительные действия над верхней и нижней частями массива могут быть совмещены по времени в одном цикле.

Процесс поэтапного перемножения в выражении (2.16) отразим в табл. 2.3.

Таблица 2.3. Поэтапное перемножение $\bar{U}_2^{-1}(\bar{U}_3^{-1}\bar{U}_4^{-1})$

$$\begin{array}{c}
\bar{U}_2^{-1} \qquad \qquad \bar{U}_3^{-1} \qquad \qquad \bar{U}_4^{-1} \\
\begin{bmatrix} 1 & -2 & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \times \begin{bmatrix} 1 & & 2 & \\ & 1 & -2 & \\ & & 1 & \\ & & & 1 \end{bmatrix} \times \begin{bmatrix} 1 & & & -3 \\ & 1 & & 1 \\ & & 1 & 2 \\ & & & 1 \end{bmatrix} \\
\\
\bar{U}_2^{-1} \qquad \qquad (\bar{U}_3^{-1}\bar{U}_4^{-1}) \\
\begin{bmatrix} 1 & -2 & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \times \begin{bmatrix} 1 & & 2 & 1 \\ & 1 & -2 & -3 \\ & & 1 & 2 \\ & & & 1 \end{bmatrix} \quad \Longleftarrow \textcircled{2} \\
\\
\bar{U}_2^{-1}(\bar{U}_3^{-1}\bar{U}_4^{-1}) \\
\begin{bmatrix} 1 & -2 & 6 & 7 \\ & 1 & -2 & -3 \\ & & 1 & 2 \\ & & & 1 \end{bmatrix} \quad \Longleftarrow \textcircled{3}
\end{array}$$

Из табл. 2.3 видно, что после получения верхней треугольной части в (2.17) пересчитывают только следующие элементы: на этапе $\textcircled{2}$ — a_{14} , a_{24} и на этапе $\textcircled{3}$ — a_{13} , a_{14} . Совмещая операции $\textcircled{2}$ и $\textcircled{2}$ после (2.17), получаем

$$\begin{bmatrix} 1/2 & \begin{array}{|c} -2 \end{array} & 2 & -3 \\ -1/2 & 1/2 & \begin{array}{|c} -2 \end{array} & 1 \\ -3/2 & -2/2 & 1/3 & \begin{array}{|c} 2 \end{array} \\ -2/2 & -1/2 & -2/3 & 1/4 \end{bmatrix} \xrightarrow[\textcircled{2}]{\textcircled{2}} \begin{bmatrix} 1/2 & \begin{array}{|c} -2 \end{array} & 2 & 1 \\ -1/4 & 1/2 & \begin{array}{|c} -2 \end{array} & -3 \\ -1 & -1 & 1/3 & \begin{array}{|c} 2 \end{array} \\ -3/4 & -1/2 & -2/3 & 1/4 \end{bmatrix}.$$

Совмещение операций $\textcircled{3}$ и $\textcircled{3}$

$$\begin{bmatrix} 1/2 & \begin{array}{|c} -2 \end{array} & 2 & 1 \\ -1/4 & 1/2 & \begin{array}{|c} -2 \end{array} & -3 \\ -1 & -1 & 1/3 & \begin{array}{|c} 2 \end{array} \\ -3/4 & -1/2 & -2/3 & 1/4 \end{bmatrix} \xrightarrow[\textcircled{3}]{\textcircled{3}} \begin{bmatrix} 1/2 & \begin{array}{|c} -2 \end{array} & 6 & 7 \\ -1/4 & 1/2 & \begin{array}{|c} -2 \end{array} & -3 \\ -1/3 & -1/3 & 1/3 & \begin{array}{|c} 2 \end{array} \\ -1/12 & 1/6 & -2/3 & 1/4 \end{bmatrix}$$

завершает операции над верхней треугольной частью, а для нижней треугольной части завершение $\textcircled{4}$ показано отдельно на стр. 44.

2.6 Плохо обусловленные матрицы

Следующие матрицы [11] часто используют для испытания разработанных программ решения систем и обращения матриц в особо сложных условиях, т. е. в таких случаях, когда матрица системы близка к вырожденной матрице.

Обозначения: a_{ij} — элемент матрицы A , n — ее порядок.

1. Матрица Гильберта.

$$a_{ij} = 1/(i + j - 1).$$

2. Матрица A с элементами:

$$a_{ii} = 1 \text{ для } i = 1, 2, \dots, 20;$$

$$a_{ii+1} = 1 \text{ для } i = 1, 2, \dots, 19;$$

$$a_{ij} = 0 \text{ для остальных значений } i \text{ и } j.$$

$$3. A = \begin{bmatrix} 5 & 4 & 7 & 5 & 6 & 7 & 5 \\ 4 & 12 & 8 & 7 & 8 & 8 & 6 \\ 7 & 8 & 10 & 9 & 8 & 7 & 7 \\ 5 & 7 & 9 & 11 & 9 & 7 & 5 \\ 6 & 8 & 8 & 9 & 10 & 8 & 9 \\ 7 & 8 & 7 & 7 & 8 & 10 & 10 \\ 5 & 6 & 7 & 5 & 9 & 10 & 10 \end{bmatrix}.$$

4. Матрица A с элементами:

$$a_{ii} = 0.01/(n - i + 1)/(i + 1);$$

$$a_{ij} = 0 \text{ для } i < j;$$

$$a_{ij} = i(n - j) \text{ для } i > j.$$

5. Матрица из пункта 4, но

$$a_{ij} = j(n - i) \text{ для } i < j.$$

$$6. A = \begin{bmatrix} R & S & T & T \\ S & R & S & T \\ T & S & R & S \\ T & T & S & R \end{bmatrix}, \quad R = \begin{bmatrix} \operatorname{ctg} \theta & \operatorname{cosec} \theta \\ -\operatorname{cosec} \theta & \operatorname{ctg} \theta \end{bmatrix},$$

$$S = \begin{bmatrix} 1 - \operatorname{ctg} \theta & \operatorname{cosec} \theta \\ -\operatorname{cosec} \theta & 1 + \operatorname{ctg} \theta \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Вычисления проводить при θ близком к нулю или π .

7. Матрица с параметром α :

$$a_{ii} = \alpha^{|n-2i|/2};$$

$$a_{1j} = a_{j1} = a_{11}/\alpha^j;$$

$$a_{nj} = a_{jn} = a_{nn}/\alpha^j;$$

$$a_{ij} = 0 \text{ для остальных значений } i \text{ и } j.$$

8. $a_{ij} = e^{i \cdot j \cdot h}$.

Вычисления проводить при h близких к нулю или 1000.

9. $a_{ij} = c + \log_2(i \cdot j)$.

Вычисления проводить при больших c .

$$10. A = \begin{bmatrix} 0.9143 \cdot 10^{-4} & 0 & 0 & 0 \\ 0.8762 & 0.7156 \cdot 10^{-4} & 0 & 0 \\ 0.7943 & 0.8143 & 0.9504 \cdot 10^{-4} & 0 \\ 0.8017 & 0.6123 & 0.7165 & 0.7123 \cdot 10^{-4} \end{bmatrix}.$$

2.7 Задание на лабораторный проект № 1

Написать и отладить программу, реализующую ваш вариант метода исключения с выбором главного элемента, для численного решения систем линейных алгебраических уравнений $Ax = f$, вычисления $\det A$ и A^{-1} . Предусмотреть сообщения, предупреждающие о невозможности решения указанных задач с заданной матрицей A .

Отделить следующие основные части программы:

- а) подпрограмму факторизации матрицы A , отвечающую вашему варианту метода исключения;
- б) подпрограмму решения систем линейных алгебраических уравнений;
- в) подпрограмму вычисления определителя матриц;
- г) две подпрограммы обращения матриц;
- д) сервисные подпрограммы.

Уделить особое внимание эффективности программы (в смысле экономии оперативной памяти). Предусмотреть пошаговое выполнение алгоритма исключения с выводом результата на экран.

Выполнить следующие пункты задания.

1. Провести подсчет фактического числа выполняемых операций умножения и деления при решении системы линейных алгебраических уравнений, сравнить его с оценочным числом $(n^3/3)$.

2. Определить скорость решения задач (решение систем линейных алгебраических уравнений, обращение матриц) с учетом времени, затрачиваемого на разложение матрицы. Для этого спроектировать и провести эксперимент, который охватывает матрицы порядка от 5 до 100 (через 5 порядков). Представить результаты в виде таблицы и графика зависимости времени выполнения (в минутах и секундах) от порядка матриц. Таблицу и график вывести на экран.

3. Оценить точность решения систем линейных алгебраических уравнений, имеющих тот же самый порядок, что и задачи из п. 2. Для этого сгенерировать случайные матрицы A , задать точное решение x^* и образовать правые части $f = Ax^*$. Провести анализ точности вычисленного решения x от порядка матрицы. Результаты представить в виде таблицы и графика.

Для заполнения матрицы A использовать случайные числа из диапазона от -100 до 100 . В качестве точного решения взять вектор $x^* = (1, 2, \dots, n)^T$, где n — порядок матрицы. Для оценки точности использовать норму вектора

$$\|x\|_{\infty} = \max_i (|x_i|). \quad (2.18)$$

4. Повторить пункт 3 задания для плохо обусловленных матриц (см. подразд. 2.6), имеющих порядок от 4 до 40 с шагом 4.

5. Вычислить матрицу A^{-1} следующими двумя способами.

Способ 1 — через решение системы $AX = I$, где I — единичная матрица.

Способ 2 — через разложение матрицы A в произведение элементарных матриц, обращение которых осуществляется отдельными процедурами, а их произведение дает матрицу A^{-1} .

Сравнить затраты машинного времени (по числу операций) и точность обращения матриц при использовании указанных способов 1 и 2. Эксперименты провести для случайных матриц порядков от 5 до 100 через 5. Для оценки точности в обоих способах использовать оценочную формулу

$$\|A_{\text{т}}^{-1} - A_{\text{пр}}^{-1}\| \leq \|I - AA_{\text{пр}}^{-1}\| \cdot \|A\|^{-1}. \quad (2.19)$$

Использовать *норму матрицы типа «бесконечность»*, т. е. вычислять ее по следующему выражению:

$$\|A\|_{\infty} = \max_i \left(\sum_{j=1}^n |a_{ij}| \right), \quad (2.20)$$

где A_T^{-1} — точное значение обратной матрицы, а $A_{\text{пр}}^{-1}$ — приближенное значение, полученное в результате обращения каждым из способов 1 и 2.

6. Провести подсчет фактического числа выполняемых операций умножения и деления при обращении матриц первым и вторым способами, сравнить его с оценочным числом (n^3) .

ЗАМЕЧАНИЕ 2.8. По ходу проведения численных экспериментов на экран дисплея должны выводиться следующие таблицы.

Решение систем линейных алгебраических уравнений:

Порядок	Время	Точность	Теоретическое число операций	Реальное число операций
---------	-------	----------	---------------------------------	----------------------------

Аналогичная таблица должна быть построена для плохо обусловленных матриц.

Обращение матриц:

Порядок	Время		Точность		Число операций		
	спос. 1	спос. 2	спос. 1	спос. 2	спос. 1	спос. 2	теорет.

ЗАМЕЧАНИЕ 2.9. Результаты экспериментов необходимо вывести на экран в форме следующих графиков. Для случая обращения матриц при построении графиков использовать данные из второй таблицы.

Графики решения систем линейных алгебраических уравнений:

- зависимость реального и оценочного числа операций от порядка матрицы (для разных графиков использовать разные цвета);
- зависимость времени решения от порядка матриц;
- зависимость точности решения от порядка матриц. При построении графиков использовать данные из первой таблицы. Для этого их необходимо записать в текстовый файл.

Графики для обращения матриц:

- зависимость реального и оценочного числа операций от порядка матрицы (для разных графиков использовать разные цвета);
- зависимость времени обращения первым и вторым способом от порядка матриц;
- зависимость точности обращения первым и вторым способом от порядка матриц.

2.8 Варианты задания на лабораторный проект № 1

1. $L\bar{U}$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по столбцу.
2. $L\bar{U}$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по строке.
3. $L\bar{U}$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по активной подматрице.
4. $\bar{L}U$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по столбцу.
5. $\bar{L}U$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по строке.
6. $\bar{L}U$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по активной подматрице.
7. $L\bar{U}$ -разложение на основе гауссова исключения по строкам с выбором главного элемента по строке.
8. $L\bar{U}$ -разложение по компактной схеме Краута с выбором главного элемента по столбцу.
9. $\bar{L}U$ -разложение по компактной схеме Краута с выбором главного элемента по строке.
10. $L\bar{U}$ -разложение по компактной схеме «строка за строкой» с выбором главного элемента по строке.
11. $L\bar{U}^{-1}$ -разложение $A = L\bar{U}$ на основе жорданова исключения с выбором главного элемента по столбцу.
12. $L\bar{U}^{-1}$ -разложение $A = L\bar{U}$ на основе жорданова исключения с выбором главного элемента по строке.
13. $L\bar{U}^{-1}$ -разложение $A = L\bar{U}$ на основе жорданова исключения с выбором главного элемента по активной подматрице.

14. $\bar{U}L$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по столбцу.

15. $\bar{U}L$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по строке.

16. $\bar{U}L$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по активной подматрице.

17. $U\bar{L}$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по столбцу.

18. $U\bar{L}$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по строке.

19. $U\bar{L}$ -разложение на основе гауссова исключения по столбцам с выбором главного элемента по активной подматрице.

20. $U\bar{L}$ -разложение на основе гауссова исключения по строкам с выбором главного элемента по строке.

21. $U\bar{L}$ -разложение по компактной схеме Краута с выбором главного элемента по столбцу.

22. $\bar{U}L$ -разложение по компактной схеме Краута с выбором главного элемента по строке.

23. $U\bar{L}$ -разложение по компактной схеме «строка за строкой» с выбором главного элемента по строке.

24. $\bar{L}^{-1}U$ -разложение $A = \bar{L}U$ на основе жорданова исключения с выбором главного элемента по столбцу.

25. $\bar{L}^{-1}U$ -разложение $A = \bar{L}U$ на основе жорданова исключения с выбором главного элемента по строке.

26. $\bar{L}^{-1}U$ -разложение $A = \bar{L}U$ на основе жорданова исключения с выбором главного элемента по активной подматрице.

Если нет других указаний преподавателя, выбирайте ваш вариант по вашему номеру в журнале студенческой группы.

3

ВЕКТОРНО-ОРИЕНТИРОВАННЫЕ АЛГОРИТМЫ LU -РАЗЛОЖЕНИЯ

3.1 Гауссово исключение и ijk -алгоритмы

Рассмотрим систему линейных уравнений

$$Ax = b \quad (3.1)$$

с невырожденной матрицей A размера $(n \times n)$. Мы считаем A заполненной матрицей. Разреженные матрицы рассматриваются в разд. 5.

Как изложено в подразд. 2.1, наиболее известной формой гауссова исключения является та, в которой система (3.1) приводится к верхнетреугольному виду путем вычитания одних уравнений, умноженных на подходящие числа, из других уравнений; полученная треугольная система решается с помощью обратной подстановки. Математически все это эквивалентно тому, что вначале строится разложение матрицы A , например вида $A = \bar{L}U$, где \bar{L} является нижнетреугольной матрицей с единицами на главной диагонали, а U — верхнетреугольная матрица с ненулевыми элементами на диагонали. Затем решаются треугольные системы

$$\bar{L}y = b, \quad Ux = y. \quad (3.2)$$

Процесс их решения называется, соответственно, *прямой* и *обратной подстановками*.

Мы сосредоточимся вначале на $\bar{L}U$ -разложении, поглощающем большую часть времени всего процесса, а затем вернемся к решению треугольных систем. Псевдокод разложения приведен на рис. 3.1.

В цикле j на рис. 3.1 кратные k -й строки текущей матрицы A вычитаются из расположенных ниже строк. Эти операции представляют собой *триады*, в которых векторами являются строки матрицы A [8].

Для $k = 1$ до $n - 1$
 Для $i = k + 1$ до n
 $l_{ik} = a_{ik}/a_{kk}$
 Для $j = k + 1$ до n
 $a_{ij} = a_{ij} - l_{ik}a_{kj}$

Рис. 3.1. Строчно ориентированная схема LU -разложения

Для $k = 1$ до $n - 1$
 Для $s = k + 1$ до n
 $l_{sk} = a_{sk}/a_{kk}$
 Для $j = k + 1$ до n
 Для $i = k + 1$ до n
 $a_{ij} = a_{ij} - l_{ik}a_{kj}$

Рис. 3.2. Столбцово ориентированная схема LU -разложения

ОПРЕДЕЛЕНИЕ 3.1. *Триадой* называют операцию вида $\mathbf{a} + \alpha \mathbf{b}$, где \mathbf{a} и \mathbf{b} суть векторы, а α — скаляр¹.

Определение триады появилось в связи с использованием векторных компьютеров², требующих, чтобы векторы располагались в последовательно адресуемых ячейках памяти. Для алгоритма на рис. 3.1 удобно предположить, что A хранится по строкам. Соответственно этому, схема на рис. 3.1 названа строчно ориентированной. В ней посредством триад осуществляется *модификация* (т. е. *обновление*) строк матрицы A ; на это приходится основная часть работы в LU -разложении.

Реальные задачи, как правило, требуют выбора главного элемента, и это еще сильнее уменьшает скорость. При использовании одной из стратегий частичного выбора мы должны на первом шаге просмотреть первый столбец в поисках максимального по модулю элемента. Эта стратегия, соответственно, называется *выбором главного элемента по столбцу*, и она приводит к перестановке строк³. Как только положение максимального элемента определено, соответствующую строку можно переставить с первой (точнее, текущей ведущей) строкой или изменить порядок индексации строк. Второй вариант называют *неявной перестановкой* строк. Как именно реализуется стратегия выбора главного элемента, зависит от вашего варианта задания. Однако во всех вариантах лабораторных работ физическая, т. е. явная перестановка строк (или столбцов) запрещена и должна быть заменена изменением порядка нумерации строк (или столбцов), т. е. неявной перестановкой. Это требование соответствует реальным пакетам вычислительной линейной алгебры, т. е. так в реальности всегда и делают. В схеме на рис. 3.1 возможны все три стратегии выбора главного элемента (см. подразд. 2.2).

¹ В зарубежной литературе триаду называют также *saxpy*, что обозначает операцию $y := ax + y$ и заменяет фразу: Single precision (с обычной точностью) ax Plus y .

² По поводу триады и векторных компьютеров см. подразд. 3.2 и 3.3.

³ Подробнее о стратегиях выбора главного элемента см. подразд. 2.2.

Правая часть b системы (3.1) также может обрабатываться в ходе приведения системы к треугольному виду, благодаря чему осуществляется этап прямой подстановки в равенствах (3.2). Такая обработка правой части b , выполняемая одновременно с приведением матрицы A к треугольному виду, также запрещена во всех вариантах лабораторных работ (проектов). Это требование тоже отвечает реальности, так как позволяет экономить значительное время в условиях, когда требуется решать одну и ту же систему (3.1) с различными правыми частями. Ситуация такого рода — обращение матрицы с помощью решения матричного уравнения $AX = I$, где I — единичная матрица, т.е. нахождение $X = A^{-1}$. В этом случае правые части b вводятся в уравнения (3.2) последовательно. При вычислении i -го столбца матрицы $X = A^{-1}$ каждая правая часть равна очередному (i -му) столбцу единичной матрицы. Для каждого b в (3.2) сначала решают первую систему, т.е. вычисляют y , а затем — вторую систему, т.е. вычисляют x . Существенно, что для хранения ни y , ни x затрат памяти не требуется: их можно хранить там же, куда был введен вектор b .

Если A хранится по столбцам, то мы изменим алгоритм LU -разложения, как это показано на рис. 3.2. На k -м шаге измененного алгоритма сначала формируется k -й столбец матрицы L ; это достигается векторной операцией деления. В самом внутреннем цикле (цикле по i) k -й столбец L , умноженный на число, вычитается из j -го столбца текущей матрицы A ; длина столбцов равна $n - k$. Таким образом, основной векторной операцией снова является триада, но теперь в качестве векторов выступают столбцы матрицы L и текущей матрицы A . В данный алгоритм также возможно внедрить любую из трех стратегий выбора главного элемента (см. подразд. 2.2).

3.2 Распараллеливание вычислений

В этом разделе мы, следуя [8], излагаем некоторые понятия параллельных вычислений, которые в практике решения линейных систем имеют большое значение.

Параллельные вычисления реализуются не на обычных (скалярных) компьютерах, какими являются все персональные компьютеры, а на *векторных* или *параллельных компьютерах*. Их отличает то, что операндами команд компьютера являются не отдельные числовые величины (скаляры), а целые группы таких величин, объединяемых в векторы или матрицы. Поэтому векторно-матричные операции для своего выполнения требуют вызова всего

лишь одной команды. Выполнение таких команд начинается, как обычно, с загрузки операндов из памяти в векторно-матричный процессор и завершается обратным действием — записью результата операции в память. В промежутке между этими действиями операции реализуются на аппаратном уровне в процессоре.

Векторные компьютеры. В основе таких компьютеров лежит концепция *конвейеризации*. Это означает явное сегментирование процессора на отдельные части (сегменты), каждая из которых выполняет свою вычислительную подзадачу независимо для соответствующих частей операндов. Например, сумматор процессора для чисел с плавающей точкой разделен на шесть сегментов; каждый сегмент занят реализацией своей части операции сложения чисел. Всякий сегмент может работать только с одной парой операндов, а в целом на конвейере в текущий момент времени могут находиться шесть пар операндов. Преимущество подобной сегментации в том, что результаты выдаются в 6 раз быстрее (а в общем случае в K , где K — число сегментов сумматора), чем в скалярном процессоре, который, получив пару операндов, не принимает новой пары, пока не вычислит результат для первой пары. Для реализации этой возможности ускорения нужно подавать данные из памяти в процессор достаточно быстро, чтобы конвейер был все время загружен данными и работой.

Параллельные компьютеры. В основе такого компьютера лежит идея использовать несколько процессоров, работающих сообща для решения одной задачи. Параллельный компьютер может иметь в своем составе либо очень простые процессоры, пригодные только для малых или ограниченных задач, либо набор полноценных процессоров, либо весьма мощные векторные процессоры. Все процессоры параллельного компьютера в каждый момент времени выполняют одну и ту же команду (или все простаивают) под управление главного процессора, называемого *контроллером*.

Распараллеливание. Независимо от того, на какой аппаратуре реализуется тот или иной вычислительный алгоритм, он обладает собственной, ему присущей характеристикой, показывающей возможности его распараллеливания.

ОПРЕДЕЛЕНИЕ 3.2. *Степенью параллелизма* численной задачи называется число ее операций, которые можно выполнять параллельно.

ПРИМЕР 3.1. Пусть требуется сложить два n -мерных вектора \mathbf{a} и \mathbf{b} . Сложения их элементов

$$a_i + b_i, \quad i = 1, \dots, n \quad (3.3)$$

независимы и потому могут выполняться параллельно. Степень параллелизма этого алгоритма равна n .

ПРИМЕР 3.2. Пусть требуется сложить n чисел a_1, \dots, a_n . Обычный последовательный алгоритм

$$s := a_1, \quad s := s + a_i, \quad i = 1, \dots, n$$

не пригоден для параллельных вычислений. Однако в самой задаче заключен немалый параллелизм. Можно разбить операнды на «двойки», т. е. складывать их по двое на каждом этапе операции. Полностью эффект этой идеи проявляется, когда число операндов равно степени двойки, т. е. $n = 2^q$. Если, например, $q = 3$, то все сложение займет $q = 3$ этапа, на каждом этапе действия выполняются параллельно, как показано на рис. 3.3.

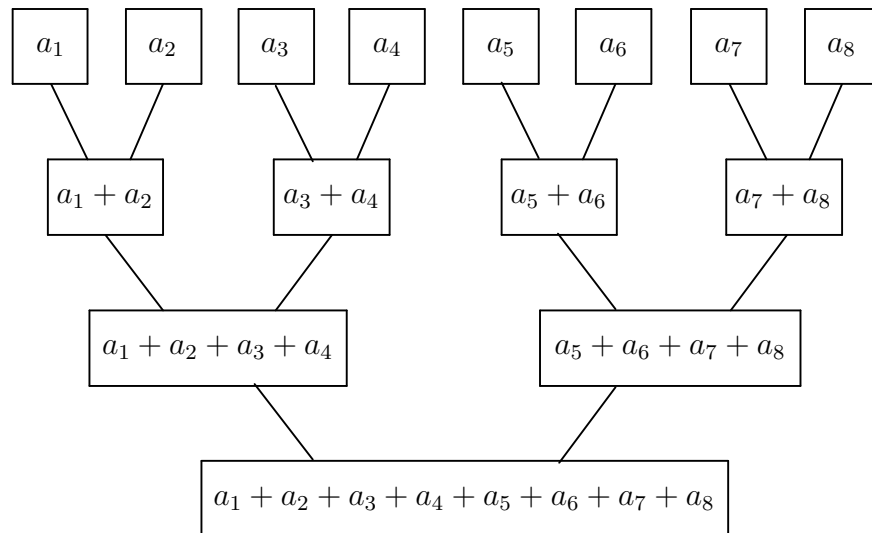


Рис. 3.3. Сложение n чисел методом сдваивания для $n = 8$ [8]

Очевидно, на первом этапе степень параллелизма равна $n/2$, на втором $n/4$ и т. д. В связи с этим приходим к обновленному определению.

ОПРЕДЕЛЕНИЕ 3.3. *Средней степенью параллелизма* численной задачи называется отношение общего числа операций ее вычислительного алгоритма к числу последовательных этапов алгоритма.

Для приведенного примера 3.2 алгоритма сдваивания в задаче сложения n чисел средняя степень параллелизма равна

$$\frac{1}{q} \left(\frac{n}{2} + \frac{n}{4} + \dots + 1 \right) = \frac{2^q - 1}{q} = \frac{n - 1}{\log n},$$

тогда как в предыдущем примере 3.1 средняя степень параллелизма максимальна. Этот алгоритм (3.3) обладает «идеальным» параллелизмом, в то время как для алгоритма на рис. 3.3 средняя степень параллелизма в $\log n$ раз меньше идеальной.

3.3 Параллельное умножение матрицы на вектор

Пусть A — матрица размера $(m \times n)$, а \mathbf{x} — вектор длины n . Тогда

$$A\mathbf{x} = \begin{bmatrix} (\mathbf{a}_1, \mathbf{x}) \\ \vdots \\ (\mathbf{a}_m, \mathbf{x}) \end{bmatrix}, \quad (3.4)$$

где \mathbf{a}_i — i -я строка матрицы A , а $(\mathbf{a}_i, \mathbf{x})$ — обычное скалярное произведение векторов \mathbf{a}_i и \mathbf{x} . Каждое из m имеющихся здесь скалярных произведений, как известно, требует суммирования n поэлементных произведений $a_{ij}x_j$. Как показано в предыдущем подразделе, такое суммирование можно распараллеливать сдваиванием, но такой параллелизм вычисления каждого отдельного скалярного произведения так или иначе неидеален. Однако m скалярных произведений в (3.4) можно вычислять параллельно. Другой способ умножения матрицы на вектор дается формулой

$$A\mathbf{x} = \sum_{j=1}^n x_j \mathbf{a}_j, \quad (3.5)$$

где \mathbf{a}_j теперь обозначает j -й столбец матрицы A .

Различие представлений (3.4) и (3.5) можно рассматривать как различие двух способов доступа к данным в памяти, что показывают две программы на рис. 3.4. Программа слева на рис. 3.4 реализует метод (3.4), тогда как программа справа реализует метод (3.5), и различие здесь только в порядке индексов для циклов. Алгоритм, основанный на представлении (3.5), записывается так:

$$\mathbf{y} = \mathbf{0}, \quad \text{для } j \text{ от } 1 \text{ до } n \text{ выполнить } \mathbf{y} = \mathbf{y} + x_j \mathbf{a}_j.$$

$y_i = 0$ Для $i = 1$ до m Для $j = 1$ до n $y_i = y_i + a_{ij}x_j$	$y_i = 0$ Для $j = 1$ до n Для $i = 1$ до m $y_i = y_i + a_{ij}x_j$
--	--

Рис. 3.4. ij (слева) и ji (справа) формы матрично-векторного умножения [8]

Как выше (в подразд. 3.1) говорилось, такая операция типа «вектор плюс произведение вектора на число», называется триадой (или операцией *saxpy*); некоторые векторные компьютеры выполняют ее особенно эффективно.

Сравнение приведенных способов умножения матрицы на вектор показывает, что на одних векторных компьютерах предпочтителен один способ, на других — другой; многое определяется также и способом хранения данных в памяти. Предположим, что матрица A хранится по столбцам; такое соглашение в отношении двумерных массивов принято в Фортране. (В других языках, например в Паскале, двумерные массивы хранятся по строкам.) Тогда векторы, требуемые для алгоритма (3.5), располагаются в последовательно адресуемых ячейках памяти, в то время как для алгоритма (3.5) строки будут представлять векторы с шагом m . Однако в векторных компьютерах часто существует ограничение на доступ к векторам: в качестве операндов для векторных команд допускаются только векторы с шагом 1 (т. е. такие, которые располагаются в последовательно адресуемых ячейках памяти). Поэтому, если матрица A хранится по столбцам, то эти соображения, связанные с памятью, усиливают аргументацию в пользу алгоритма (3.5). Если же матрица A хранится по строкам, то предпочтительней может оказаться алгоритм (3.4). Только детальный анализ может показать, какой выбор следует сделать для конкретной машины.

3.4 Параллельное LU -разложение

Для распараллеленных вычислений нужны соответствующие параллельные или векторные компьютеры. С середины 70-х годов 20-го столетия фирма CRAY Research, Inc. производит векторные компьютеры, которые могут служить примером процессоров типа «регистр–регистр». Под этим подразумевается, что существуют векторные команды, для которых операндами являются векторы. Эти команды получают свои операнды из очень быстрой памяти, именуемой *векторными регистрами*, и запоминают результаты опять-

таки в векторных регистрах. Для операции сложения векторов это показано на рис. 3.5.

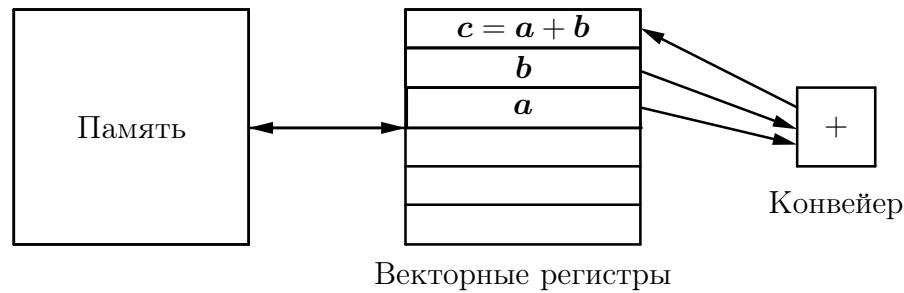


Рис. 3.5. Операция сложения в компьютере типа «регистр–регистр» [8]

Предполагается, что каждый векторный регистр состоит из некоторого числа слов. Например, в машинах CRAY имеется восемь векторных регистров, каждый емкостью в 64 числа с плавающей точкой. До начала сложения операнды загружаются из оперативной памяти в регистры. После завершения сложения векторный результат переписывается из регистровой памяти в оперативную память. Для таких машин желателен иной подход к организации LU -разложения.

Исследуем вначале характер обменов во внутреннем цикле столбцового алгоритма LU -разложения на рис. 3.2. Для простоты предположим, что столбец матрицы A полностью вкладывается в векторный регистр, и начнем с рассмотрения случая, когда на фоне вычислений может выполняться только загрузка или только запись в память. Несколько первых операций указаны в следующем списке:

Сформировать первый столбец матрицы L .

Загрузить второй столбец матрицы A .

$$\left\{ \begin{array}{l} \text{Модифицировать второй столбец матрицы } A; \\ \text{загрузить третий столбец матрицы } A. \end{array} \right. \quad (3.6)$$

$$\text{Записать в память модифицированный второй столбец.} \quad (3.7)$$

$$\left\{ \begin{array}{l} \text{Модифицировать третий столбец матрицы } A; \\ \text{загрузить четвертый столбец матрицы } A. \end{array} \right. \quad (3.8)$$

.....

Согласно (3.6), загрузка следующего столбца матрицы A совмещается с модификацией текущего столбца. Но затем возникает задержка при записи модифицированного второго столбца из регистра в память. Можно так

модифицировать алгоритм, чтобы устранить задержку, вызванную записью в память (3.7). Идея состоит в том, чтобы выполнять необходимую обработку для j -го столбца полностью, прежде чем переходить к $(j + 1)$ -му столбцу. При этом обработка каждого из остальных столбцов матрицы A откладывается до тех пор, пока не наступит время придать этому столбцу окончательный вид. Псевдокод данного алгоритма приведен на рис. 3.6.

Для $j = 2$ до n
 Для $s = j$ до n
 $l_{s,j-1} = a_{s,j-1}/a_{j-1,j-1}$
 Для $k = 1$ до $j - 1$
 Для $i = k + 1$ до n
 $a_{ij} = a_{ij} - l_{ik}a_{kj}$

Рис. 3.6. Столбцово ориентированная схема $\bar{L}U$ -разложения с отложенными модификациями (jki -алгоритм, см. с. 64) [8]

Опишем несколько первых операций j -го шага вычислений, показывая таким образом характер обменов с памятью:

$$\left. \begin{array}{l}
 \text{Загрузить первый столбец матрицы } L. \\
 \text{Загрузить } j\text{-й столбец матрицы } A. \\
 \left\{ \begin{array}{l} \text{Модифицировать } j\text{-й столбец матрицы } A; \\ \text{загрузить второй столбец матрицы } L. \end{array} \right. \\
 \left\{ \begin{array}{l} \text{Модифицировать } j\text{-й столбец матрицы } A; \\ \text{загрузить третий столбец матрицы } L. \end{array} \right. \\
 \dots\dots\dots
 \end{array} \right\} \quad (3.9)$$

Заметим, что в алгоритме (3.9) не производится записей в память, пока вся работа с j -м столбцом матрицы A не завершена. Столбцы матрицы L все время должны загружаться в регистры, но эти загрузки идут на фоне вычислений. Только в начале и в конце каждого шага происходят задержки для загрузок и(или) записей. Вполне вероятно, что транслятор не сумеет распознать возможности оставить текущий j -й столбец в регистре; в этом случае результат, требуемый от алгоритма на рис. 3.6, либо достигается с переходом к программированию на языке ассемблера, либо аппроксимируется путем разворачивания циклов. Еще одна потенциальная проблема при реализации данного алгоритма заключается в том, что длины векторов при

модификациях непостоянны: на j -м шаге мы модифицируем j -й столбец, используя $n - 1$ последних элементов столбца 1, далее $n - 2$ последних элементов столбца 2 и т. д.

Алгоритм с отложенными модификациями не столь нужен для тех машин типа «регистр–регистр», в которых допускается совмещение с арифметикой как загрузок, так и записей в память. В этом случае операцию (3.7) можно было бы удалить, а операцию (3.8) заменить операцией

$$\left\{ \begin{array}{l} \text{Модифицировать третий столбец матрицы } A; \\ \text{загрузить четвертый столбец матрицы } A; \\ \text{записать в память второй столбец матрицы } A. \end{array} \right.$$

Таким образом, запись в память второго столбца матрицы A происходит одновременно с загрузкой четвертого столбца.

ЗАМЕЧАНИЕ 3.1. Материал подразд. 3.2, 3.3 и 3.4 из [8] приведен, чтобы объяснить те приложения, для которых создаются алгоритмы с отложенными модификациями: алгоритм на рис. 3.6 и другие, показанные ниже в подразд. 3.5. Таким образом, включение таких алгоритмов в лабораторную работу (проект) может рассматриваться как задача имитирования алгоритмов векторных или параллельных компьютеров на обычных (скалярных) компьютерах с целью освоения этих современных версий LU -разложения.

3.5 LU -разложение и его ijk -формы

Ниже в описании ijk -алгоритмов факторизации (разложения), основанных на методе Гаусса исключения переменных, используем из [8] следующие обозначения для индексов:

k — номер исключаемой переменной,

i — номер строки, т. е. модифицируемого уравнения,

j — номер столбца, т. е. коэффициента в модифицируемом уравнении.

Тогда общую основу всех алгоритмов удобно определить тройкой вложенных циклов вида

$$\begin{array}{l} \text{Для } \underline{\hspace{1cm}} \\ \quad \text{Для } \underline{\hspace{1cm}} \\ \qquad \text{Для } \underline{\hspace{1cm}} \\ \qquad \qquad a_{ij} = a_{ij} - l_{ik}a_{kj} \end{array}$$

Здесь последняя формула обозначает модификацию j -го элемента i -й строки матрицы A при исключении k -й переменной вектора неизвестных x из уравнений системы $Ax = f$. Перестановки трех индексов для циклов определяют $3! = 6$ возможных вариантов алгоритмов, образуя так называемые ijk -формы, для каждого вида разложения. Для квадратной матрицы A размера $(n \times n)$ возможны четыре вида разложения, а именно:

$$A = L\bar{U}, \quad A = \bar{L}U, \quad A = U\bar{L}, \quad A = \bar{U}L, \quad (3.10)$$

где черта сверху указывает на тот из сомножителей, который имеет единичную главную диагональ. Поэтому всего возможно построить 24 варианта ijk -алгоритмов разложения матрицы для решения различных задач: решения систем, обращения матрицы и вычисления ее определителя.

Рассмотрим все шесть ijk -форм для одного из четырех разложений (3.10), а именно, для $\bar{L}U$ -разложения матрицы A . Для численной иллюстрации возьмем следующий пример.

ПРИМЕР 3.3.

$$A = \begin{bmatrix} 2 & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}, \quad \bar{L} = \begin{bmatrix} 1 & & & \\ 1/2 & 1 & & \\ 3/2 & 1 & 1 & \\ 1 & 1/2 & 2/3 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & 4 & -4 & 6 \\ & 2 & 4 & -2 \\ & & 3 & -6 \\ & & & 4 \end{bmatrix}.$$

Два алгоритма для $\bar{L}U$ -разложения матрицы A с немедленными модификациями

1) kij -алгоритм, рис. 3.1.

Доступ к элементам матрицы A по строкам. Исключение по столбцам. Модификации немедленные. ГЭ — любая из трех стратегий.

Для $k = 1$ до $n - 1$

Для $i = k + 1$ до n

$$l_{ik} = a_{ik}/a_{kk}$$

Для $j = k + 1$ до n

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

2) kji -алгоритм, рис. 3.2.

Доступ к элементам матрицы A по столбцам. Исключение по столбцам. Модификации немедленные. ГЭ — любая из трех стратегий.

Для $k = 1$ до $n - 1$

Для $s = k + 1$ до n

$$l_{sk} = a_{sk}/a_{kk}$$

Для $j = k + 1$ до n

Для $i = k + 1$ до n

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

**Два алгоритма для $\bar{L}U$ -разложения матрицы A
(столбцово ориентированные с отложенными модификациями)**

3) jki -алгоритм, рис. 3.6.

Доступ к элементам матрицы A по столбцам. Исключение по столбцам. Модификации отложенные. ГЭ — по $(j - 1)$ -му столбцу.

Для $j = 2$ до n

Для $s = j$ до n

$$l_{s,j-1} = a_{s,j-1}/a_{j-1,j-1}$$

Для $k = 1$ до $j - 1$

Для $i = k + 1$ до n

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

4) jik -алгоритм.

Доступ к элементам матрицы A по столбцам. Исключение по столбцам. Модификации отложенные. В цикле по s идет нормировка $(j - 1)$ -го столбца. Первый цикл по i вычисляет столбец для U , второй — столбец для \bar{L} . ГЭ — по $(j - 1)$ -му столбцу.

Для $j = 2$ до n

Для $s = j$ до n

$$l_{s,j-1} = a_{s,j-1}/a_{j-1,j-1}$$

Для $i = 2$ до j

Для $k = 1$ до $i - 1$

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

Для $i = j + 1$ до n

Для $k = 1$ до $j - 1$

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

**Два алгоритма ijk -форм для $\bar{L}U$ -разложения матрицы A
(строчно ориентированные с отложенными модификациями)**

5) ikj -алгоритм.

Доступ к элементам матрицы A по строкам. Исключение по строкам. Модификации отложенные. ГЭ — по $(i - 1)$ -й строке.

Для $i = 2$ до n

Для $k = 1$ до $i - 1$

$$l_{i,k} = a_{i,k}/a_{k,k}$$

Для $j = k + 1$ до n

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

6) ijk -алгоритм.

Доступ к элементам матрицы A по строкам. Исключение по строкам. Модификации отложенные. Первый цикл по j находит элементы i -й строки \bar{L} . Второй цикл по j — элементы i -й строки U . ГЭ — по $(i - 1)$ -й строке.

Для $i = 2$ до n

Для $j = 2$ до i

$$l_{i,j-1} = a_{i,j-1}/a_{j-1,j-1}$$

Для $k = 1$ до $j - 1$

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

Для $j = i + 1$ до n

Для $k = 1$ до $i - 1$

$$a_{ij} = a_{ij} - l_{ik}a_{kj}$$

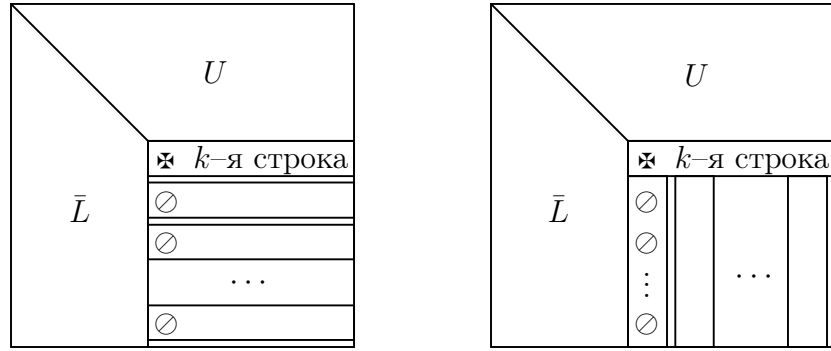


Рис. 3.7. Способ доступа к данным для kji -формы (слева) и для kjk -формы (справа) $\bar{L}U$ -разложения. Обозначения: \bar{L} , U — вычисление закончено, обращений больше нет; \star — главный элемент (ГЭ); \circ — деление на ГЭ (нормировка) [8]

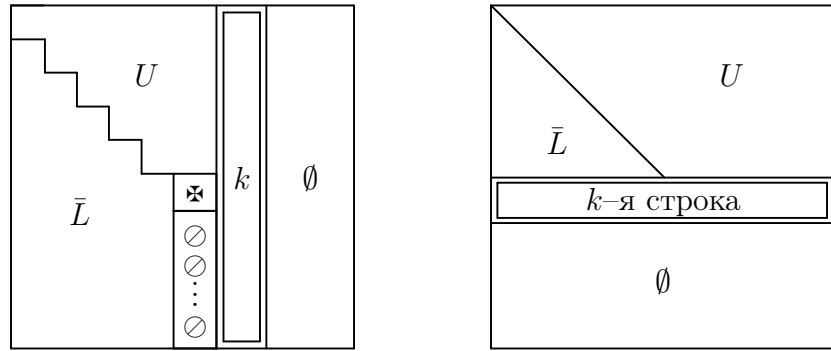


Рис. 3.8. Способ доступа к данным для jki -формы и для jik -формы (слева) и для ikj -формы и для ijk -формы (справа) $\bar{L}U$ -разложения. Обозначения: \bar{L} , U — вычисление закончено, обращения больше не производятся; \star — главный элемент (ГЭ); \circ — деление на ГЭ (нормировка); \emptyset — обращений не было [8]

ЗАМЕЧАНИЕ 3.2. В приведенных алгоритмах не содержится процедура выбора главного элемента. Она дословно переносится из описания лабораторной работы № 1. Аналогичные алгоритмы могут быть написаны для остальных трех видов разложения матрицы A из списка (3.10). При написании программ, соответствующих приведенным алгоритмам, следует выполнить требование, согласно которому все вычисления выполняются в одном и том же двумерном массиве, где сначала хранится матрица A . В процессе вычислений матрица A замещается элементами треугольных матриц, составляющих искомое разложение из списка (3.10). Способ доступа к данным для ijk -форм $\bar{L}U$ -разложения показан на рис. 3.7 и рис. 3.8. Расчеты по алгоритмам kji -формы и kjk -формы $\bar{L}U$ -разложения достаточно очевидны. Для других четырех форм $\bar{L}U$ -разложения эти вычисления поясняются для примера 3.3 в табл. 3.1–3.4.

Таблица 3.1. Вычисления по алгоритму jki -формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом

A	$j = 2$	$j = 3$	$j = 4$
$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 2 & 1 \\ 3/2 & 2 & 1 & 1 \\ 2/2 & 1 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & 1 \\ 3/2 & 2/2 & 7 & 1 \\ 2/2 & 1/2 & 4 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -8 \\ 2/2 & 1/2 & 2/3 & -1 \end{bmatrix}$
$\begin{array}{c} \Uparrow\Uparrow \\ \text{исходная} \\ \text{матрица} \end{array}$	$\begin{array}{c} \Uparrow\Uparrow \\ \text{нормировка} \\ (j-1)\text{-го} \\ \text{столбца;} \\ (j-1)\text{-} \\ \text{-кратное} \\ \text{исключение} \\ \text{в } j\text{-м столбце} \end{array}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & 1 \\ 3/2 & 2/2 & \mathbf{3} & 1 \\ 2/2 & 1/2 & 2 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -6 \\ 2/2 & 1/2 & 2/3 & 0 \end{bmatrix}$
		$\begin{array}{c} \Uparrow\Uparrow \\ \text{нормировка} \\ (j-1)\text{-го} \\ \text{столбца;} \\ (j-1)\text{-кратная} \\ \text{модификация} \\ j\text{-го столбца} \end{array}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -6 \\ 2/2 & 1/2 & 2/3 & 4 \end{bmatrix}$

Таблица 3.2. Вычисления по алгоритму jik -формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом

A	$j = 2$	$j = 3$	$j = 4$
$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 2 & 1 \\ 3/2 & 2 & 1 & 1 \\ 2/2 & 1 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & 1 \\ 3/2 & 2/2 & 7 & 1 \\ 2/2 & 1/2 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -6 \\ 2/2 & 1/2 & 2/3 & 4 \end{bmatrix}$
$\begin{array}{c} \Uparrow\Uparrow \\ \text{исходная} \\ \text{матрица} \end{array}$	$\begin{array}{c} \Uparrow\Uparrow \\ \text{нормировка} \\ (j-1)\text{-го} \\ \text{столбца;} \\ (j-1)\text{-} \\ \text{-кратная} \\ \text{модификация} \\ j\text{-го столбца} \end{array}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & 1 \\ 3/2 & 2/2 & \mathbf{3} & 1 \\ 2/2 & 1/2 & 2 & 5 \end{bmatrix}$	$\begin{array}{c} \Uparrow\Uparrow \\ \text{нормировка} \\ (j-1)\text{-го} \\ \text{столбца;} \\ (j-1)\text{-кратная} \\ \text{модификация} \\ j\text{-го столбца} \end{array}$

Таблица 3.3. Вычисления по алгоритму *ikj*-формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом

A	$i = 2$	$i = 3$	$i = 4$																																																																
<table><tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr><tr><td>1</td><td>4</td><td>2</td><td>1</td></tr><tr><td>3</td><td>8</td><td>1</td><td>1</td></tr><tr><td>2</td><td>5</td><td>0</td><td>5</td></tr></table>	2	4	-4	6	1	4	2	1	3	8	1	1	2	5	0	5	<table><tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr><tr><td>$1/2$</td><td>2</td><td>4</td><td>-2</td></tr><tr><td>3</td><td>8</td><td>1</td><td>1</td></tr><tr><td>2</td><td>5</td><td>0</td><td>5</td></tr></table>	2	4	-4	6	$1/2$	2	4	-2	3	8	1	1	2	5	0	5	<table><tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr><tr><td>$1/2$</td><td>2</td><td>4</td><td>-2</td></tr><tr><td>$3/2$</td><td>2</td><td>7</td><td>-8</td></tr><tr><td>2</td><td>5</td><td>0</td><td>5</td></tr></table>	2	4	-4	6	$1/2$	2	4	-2	$3/2$	2	7	-8	2	5	0	5	<table><tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr><tr><td>$1/2$</td><td>2</td><td>4</td><td>-2</td></tr><tr><td>$3/2$</td><td>$2/2$</td><td>3</td><td>-6</td></tr><tr><td>$2/2$</td><td>1</td><td>4</td><td>-1</td></tr></table>	2	4	-4	6	$1/2$	2	4	-2	$3/2$	$2/2$	3	-6	$2/2$	1	4	-1
2	4	-4	6																																																																
1	4	2	1																																																																
3	8	1	1																																																																
2	5	0	5																																																																
2	4	-4	6																																																																
$1/2$	2	4	-2																																																																
3	8	1	1																																																																
2	5	0	5																																																																
2	4	-4	6																																																																
$1/2$	2	4	-2																																																																
$3/2$	2	7	-8																																																																
2	5	0	5																																																																
2	4	-4	6																																																																
$1/2$	2	4	-2																																																																
$3/2$	$2/2$	3	-6																																																																
$2/2$	1	4	-1																																																																
<table><tr><td>$\Uparrow\Uparrow$</td></tr><tr><td>исходная матрица</td></tr></table>	$\Uparrow\Uparrow$	исходная матрица	<table><tr><td>$\Uparrow\Uparrow$</td></tr><tr><td>$(i - 1)$</td></tr><tr><td>нормировок и вычитаний в i-й строке</td></tr></table>	$\Uparrow\Uparrow$	$(i - 1)$	нормировок и вычитаний в i -й строке	<table><tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr><tr><td>$1/2$</td><td>2</td><td>4</td><td>-2</td></tr><tr><td>$3/2$</td><td>$2/2$</td><td>3</td><td>-6</td></tr><tr><td>2</td><td>5</td><td>0</td><td>5</td></tr></table>	2	4	-4	6	$1/2$	2	4	-2	$3/2$	$2/2$	3	-6	2	5	0	5	<table><tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr><tr><td>$1/2$</td><td>2</td><td>4</td><td>-2</td></tr><tr><td>$3/2$</td><td>$2/2$</td><td>3</td><td>-6</td></tr><tr><td>$2/2$</td><td>$1/2$</td><td>2</td><td>0</td></tr></table>	2	4	-4	6	$1/2$	2	4	-2	$3/2$	$2/2$	3	-6	$2/2$	$1/2$	2	0																											
$\Uparrow\Uparrow$																																																																			
исходная матрица																																																																			
$\Uparrow\Uparrow$																																																																			
$(i - 1)$																																																																			
нормировок и вычитаний в i -й строке																																																																			
2	4	-4	6																																																																
$1/2$	2	4	-2																																																																
$3/2$	$2/2$	3	-6																																																																
2	5	0	5																																																																
2	4	-4	6																																																																
$1/2$	2	4	-2																																																																
$3/2$	$2/2$	3	-6																																																																
$2/2$	$1/2$	2	0																																																																
	<table><tr><td>$\Uparrow\Uparrow$</td></tr><tr><td>$(i - 1)$</td></tr><tr><td>нормировок и вычитаний в i-й строке</td></tr></table>	$\Uparrow\Uparrow$	$(i - 1)$	нормировок и вычитаний в i -й строке		<table><tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr><tr><td>$1/2$</td><td>2</td><td>4</td><td>-2</td></tr><tr><td>$3/2$</td><td>$2/2$</td><td>3</td><td>-6</td></tr><tr><td>$2/2$</td><td>$1/2$</td><td>$2/3$</td><td>4</td></tr></table>	2	4	-4	6	$1/2$	2	4	-2	$3/2$	$2/2$	3	-6	$2/2$	$1/2$	$2/3$	4																																													
$\Uparrow\Uparrow$																																																																			
$(i - 1)$																																																																			
нормировок и вычитаний в i -й строке																																																																			
2	4	-4	6																																																																
$1/2$	2	4	-2																																																																
$3/2$	$2/2$	3	-6																																																																
$2/2$	$1/2$	$2/3$	4																																																																

3.6 Треугольные системы

По окончании этапа приведения в гауссовом исключении нам необходимо решить треугольную систему уравнений

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

Обычный алгоритм обратной подстановки описывается формулами

$$x_i = (c_i - u_{i,i+1}x_{i+1} - \dots - u_{in}x_n)/u_{ii}, \quad i = n, \dots, 1. \quad (3.11)$$

Рассмотрим, как он может быть реализован в векторных операциях. Если U хранится по строкам (так будет, если на этапе приведения A хранилась по строкам), то формулы (3.11) задают *скалярные произведения* с длинами векторов, меняющимися от 1 до $n-1$, и n скалярных делений (рис. 3.9 слева).

Альтернативный алгоритм, полезный, если U хранится по столбцам, представлен в виде псевдокода на рис. 3.9 справа. Он называется *столбцовым алгоритмом* (или *алгоритмом векторных сумм*). Как только найдено

Таблица 3.4. Вычисления по алгоритму ijk -формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом

A	$i = 2, j = 2, k = 1$	$i = 3, j = 2, k = 1$	$i = 4, j = 2, k = 1$
$\begin{bmatrix} 2 & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & 2 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & 2 & 4 & -2 \\ 3/2 & 2 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & 2 & 4 & -2 \\ 3/2 & 2/2 & 3 & -6 \\ 2/2 & 1 & 0 & 5 \end{bmatrix}$
$\begin{matrix} \uparrow\uparrow \\ \text{исходная} \\ \text{матрица} \end{matrix}$	$i = 2, j = 3, k = 1$	$i = 3, j = 3, k = 1$	$i = 4, j = 3, k = 1$
	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & 2 & 4 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 7 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & -6 \\ 2/2 & 1/2 & 4 & 5 \end{bmatrix}$
	$i = 2, j = 4, k = 1$	$i = 3, j = 3, k = 2$	$i = 4, j = 3, k = 2$
	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & 2 & 4 & -2 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & -6 \\ 2/2 & 1/2 & 2 & 5 \end{bmatrix}$
$\begin{matrix} \uparrow\uparrow \\ (i-1) \\ \text{нормировок} \\ \text{в } i\text{-й строке} \end{matrix}$		$i = 3, j = 4, k = 1$	$i = 4, j = 4, k = 1$
		$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & -8 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -6 \\ 2/2 & 1/2 & 2/3 & -1 \end{bmatrix}$
$\begin{matrix} \uparrow\uparrow \\ (n-i)(i-1) + \\ + \frac{i(i-1)}{2} \\ \text{вычитаний} \\ \text{в } i\text{-й строке} \end{matrix}$		$i = 3, j = 4, k = 2$	$i = 4, j = 4, k = 2$
		$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & -6 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & -6 \\ 2/2 & 1/2 & 2/3 & 0 \end{bmatrix}$
			$i = 4, j = 4, k = 3$
		$\begin{matrix} \uparrow\uparrow \\ (i-1) \\ \text{нормировок и} \\ (n-i)(i-1) + \\ + \frac{i(i-1)}{2} \\ \text{вычитаний} \\ \text{в } i\text{-й строке} \end{matrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -6 \\ 2/2 & 1/2 & 2/3 & 4 \end{bmatrix}$

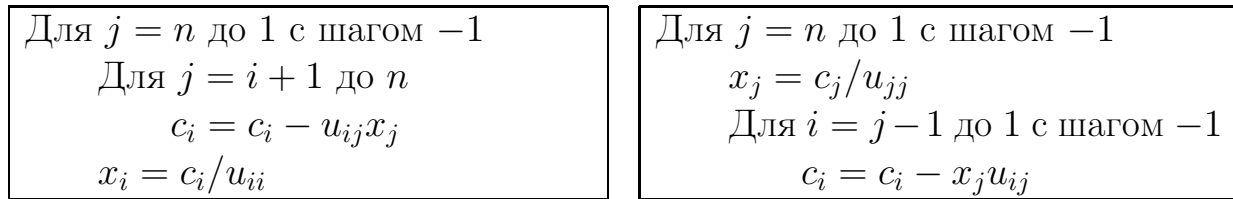


Рис. 3.9. Алгоритмы скалярных произведений (слева) и столбцовый для обратной подстановки (справа)

значение x_n , вычисляются и вычитаются из соответствующих элементов c_i величины произведений $x_n u_{in}$ ($i=1, \dots, n-1$); таким образом, вклад, вносимый x_n в прочие компоненты решения, реализуется до перехода к следующему шагу. Шаг с номером i состоит из скалярного деления, сопровождаемого триадой длины $i-1$ (подразумевается, что шаги нумеруются в обратном порядке: $n, n-1, \dots, 2, 1$). Какой из двух алгоритмов выбрать, диктуется способом хранения матрицы U , если он был определен LU -разложением.

Как алгоритм скалярных произведений, так и столбцовый алгоритм легко переформулировать на случай нижнетреугольных систем, процесс решения которых называется алгоритмом *прямой подстановки*.

3.7 Задание на лабораторный проект № 2

Написать и отладить программу, реализующую ваш вариант метода исключения с выбором главного элемента, для численного решения систем линейных алгебраических уравнений $Ax = f$, вычисления $\det A$ и A^{-1} . Предусмотреть сообщения, предупреждающие о невозможности решения указанных задач с заданной матрицей A .

Отделить следующие основные части программы:

- а) подпрограмму факторизации матрицы A , отвечающую вашему варианту метода исключения;
- б) подпрограмму решения систем линейных алгебраических уравнений;
- в) подпрограмму вычисления определителя матриц;
- г) подпрограммы обращения матриц;
- д) сервисные подпрограммы.

Уделить особое внимание эффективности программы (в смысле экономии оперативной памяти). Предусмотреть пошаговое выполнение алгоритма исключения с выводом результата на экран.

Выполнить следующие пункты задания.

1. Провести подсчет фактического числа выполняемых операций умножения и деления при решении системы линейных алгебраических уравнений, сравнить его с оценочным числом $(n^3/3)$.

2. Определить скорость решения задач (решение систем линейных алгебраических уравнений, обращение матриц) с учетом времени, затрачиваемого на разложение матрицы. Для этого спроектировать и провести эксперимент, который охватывает матрицы порядка от 5 до 100 (через 5 порядков). Представить результаты в виде таблицы и графика зависимости времени выполнения (в минутах и секундах) от порядка матриц. Таблицу и график вывести на экран.

3. Оценить точность решения систем линейных алгебраических уравнений, имеющих тот же самый порядок, что и задачи из пункта 2. Для этого сгенерировать случайные матрицы A , задать точное решение x^* и образовать правые части $f = Ax^*$. Провести анализ точности вычисленного решения x от порядка матрицы. Результаты представить в виде таблицы и графика.

Для заполнения матрицы A использовать случайные числа из диапазона от -100 до 100 . В качестве точного решения взять вектор $x^* = (1, 2, \dots, n)^T$, где n — порядок матрицы. Для оценки точности использовать норму вектора

$$\|x\|_{\infty} = \max_i (|x_i|). \quad (3.12)$$

4. Повторить пункт 3 задания для плохо обусловленных матриц (см. подразд. 2.6), имеющих порядок от 4 до 40 с шагом 4.

5. Вычислить матрицу A^{-1} следующими двумя способами.

Способ 1 — через решение системы $AX = I$, где I — единичная матрица.

Способ 2 — через разложение матрицы A в произведение элементарных матриц, обращение которых осуществляется аналитически, а их произведение дает матрицу A^{-1} .

Сравнить затраты машинного времени и точность обращения матриц при использовании указанных способов 1 и 2. Эксперименты провести для случайных матриц порядков от 5 до 100 через 5. Для оценки точности в обоих способах использовать оценочную формулу

$$\|A_{\text{т}}^{-1} - A_{\text{пр}}^{-1}\| \leq \|I - AA_{\text{пр}}^{-1}\| \cdot \|A\|^{-1}. \quad (3.13)$$

В этом выражении норму матрицы вычислять в соответствии со следующим определением:

$$\|A\|_{\infty} = \max_i \left(\sum_{j=1}^n |a_{ij}| \right), \quad (3.14)$$

где A_T^{-1} — точное значение обратной матрицы, а $A_{пр}^{-1}$ — приближенное значение, полученное в результате обращения каждым из способов 1 и 2.

6. Провести подсчет фактического числа выполняемых операций умножения и деления при обращении матриц первым и вторым способами, сравнить его с оценочным числом (n^3) .

ЗАМЕЧАНИЕ 3.3. По ходу проведения численных экспериментов на экран дисплея должны выводиться следующие таблицы.

Решение систем линейных алгебраических уравнений:

Порядок	Время	Точность	Теоретическое число операций	Реальное число операций
---------	-------	----------	---------------------------------	----------------------------

Аналогичная таблица должна быть построена для плохо обусловленных матриц.

Обращение матриц:

Порядок	Время		Точность		Число операций		
	спос. 1	спос. 2	спос. 1	спос. 2	спос. 1	спос. 2	теорет.

ЗАМЕЧАНИЕ 3.4. Результаты экспериментов необходимо вывести на экран в форме следующих графиков.

Графики решения систем линейных алгебраических уравнений:

- зависимость реального и расчетного числа операций от порядка матрицы (для разных графиков использовать разные цвета);
- зависимости времени и точности решения от порядка матриц.

Графики для обращения матриц:

- зависимость реального и оценочного числа операций от порядка матрицы (для разных графиков использовать разные цвета);
- зависимости времени и точности обращения первым и вторым способом от порядка матриц.

Таблица 3.5. Варианты задания на лабораторный проект № 2

Вид разложения	kij			kji			jki	jik	ikj	ijk
	a	b	c	a	b	c	a	a	b	b
$A = \bar{L}U$	1	2	3	4	5	6	7	8	9	10
$A = L\bar{U}$	11	12	13	14	15	16	17	18	19	20
$A = \bar{U}L$	21	22	23	24	25	26	27	28	29	30
$A = U\bar{L}$	31	32	33	34	35	36	37	38	39	40

^a — выбор ГЭ по столбцу активной подматрицы

^b — выбор ГЭ по строке активной подматрицы

^c — выбор ГЭ по активной подматрице

3.8 Варианты задания на лабораторный проект № 2

В табл. 3.5 приведены 40 номеров вариантов задания на лабораторную работу (проект) № 2 с тремя стратегиями выбора главного элемента.

Если нет других указаний преподавателя, выбирайте ваш вариант по вашему номеру в журнале студенческой группы.

4

АЛГОРИТМЫ ОКАЙМЛЕНИЯ В LU -РАЗЛОЖЕНИИ

4.1 Метод окаймления

Хотя ijk -формы дают шесть различных способов организации LU -разложения, имеются и другие способы, потенциально полезные для векторных компьютеров. Даже тогда, когда та или иная ijk -форма теоретически пригодна для конкретной векторной машины, при ее реализации могут возникнуть проблемы, особенно если применяется язык высокого уровня. Разбираемые ниже способы организации вычислений основаны на операциях с подматрицами; потенциально они проще реализуются и облегчают написание переносимых программ.

В основе этих способов организации лежит идея *окаймления* [14]. Математически ее можно представить следующим образом. Разобьем матрицу A на блоки и, соответственно, разобьем на блоки сомножители \bar{L} и U искомого разложения $\bar{L}U = A$:

$$\begin{bmatrix} \bar{L}_{11} & 0 & 0 \\ \mathbf{l}_{j1}^T & 1 & 0 \\ L_{31} & \mathbf{l}_{3j} & \bar{L}_{33} \end{bmatrix} \begin{bmatrix} U_{11} & \mathbf{u}_{1j} & U_{13} \\ 0 & u_{jj} & \mathbf{u}_{j3}^T \\ 0 & 0 & U_{33} \end{bmatrix} = \begin{bmatrix} A_{11} & \mathbf{a}_{1j} & A_{13} \\ \mathbf{a}_{j1}^T & a_{jj} & \mathbf{a}_{j3}^T \\ A_{31} & \mathbf{a}_{3j} & A_{33} \end{bmatrix}. \quad (4.1)$$

Здесь \mathbf{l}_{j1}^T , \mathbf{u}_{j3}^T , \mathbf{a}_{j1}^T и \mathbf{a}_{j3}^T — векторы-строки, а \mathbf{l}_{3j} , \mathbf{u}_{1j} , \mathbf{a}_{1j} и \mathbf{a}_{3j} — векторы-столбцы; каждый из этих элементов находится на j -й позиции.

4.2 Окаймление известной части разложения

Пусть известно разложение $\bar{L}_{11}U_{11} = A_{11}$, которое можно рассматривать как равенство (4.1) для блока A_{11} . Запишем аналогичные равенства для тех

трех блоков, которые окаймляют эту известную часть разложения, следуя правилу перемножения блок-матриц, а именно, для \mathbf{a}_{j1}^T , \mathbf{a}_{1j} и a_{jj} :

$$\mathbf{l}_{j1}^T U_{11} = \mathbf{a}_{j1}^T, \quad \bar{L}_{11} \mathbf{u}_{1j} = \mathbf{a}_{1j}, \quad \mathbf{l}_{j1}^T \mathbf{u}_{1j} + u_{jj} = a_{jj}. \quad (4.2)$$

Из первого уравнения (4.2), переписанного в виде $U_{11}^T \mathbf{l}_{j1} = \mathbf{a}_{j1}$, находим \mathbf{l}_{j1} , из второго находим \mathbf{u}_{1j} и затем из третьего находим $u_{jj} = a_{jj} - \mathbf{l}_{j1}^T \mathbf{u}_{1j}$. При этом первое и второе уравнения описываются следующими нижнетреугольными системами

$$U_{11}^T \mathbf{l}_{j1} = \mathbf{a}_{j1}, \quad \bar{L}_{11} \mathbf{u}_{1j} = \mathbf{a}_{1j}. \quad (4.3)$$

Существуют два естественных варианта реализации *окаймления известной части LU -разложения*.

В первом варианте треугольные системы (4.3) решаются с помощью столбцового алгоритма, во втором — с помощью алгоритма скалярных произведений. Псевдокоды этих двух вариантов приведены на рис. 4.1.

<p>Для $j = 2$ до n</p> <p style="padding-left: 20px;">Для $k = 1$ до $j - 2$</p> <p style="padding-left: 40px;">Для $i = k + 1$ до $j - 1$</p> <p style="padding-left: 60px;">$a_{ij} = a_{ij} - l_{ik} a_{kj}$</p> <p style="padding-left: 20px;">Для $k = 1$ до $j - 1$</p> <p style="padding-left: 40px;">$l_{jk} = a_{jk} / a_{kk}$</p> <p style="padding-left: 20px;">Для $i = k + 1$ до j</p> <p style="padding-left: 40px;">$a_{ji} = a_{ji} - l_{jk} a_{ki}$</p>	<p>Для $j = 2$ до n</p> <p style="padding-left: 20px;">Для $i = 2$ до $j - 1$</p> <p style="padding-left: 40px;">Для $k = 1$ до $i - 1$</p> <p style="padding-left: 60px;">$a_{ij} = a_{ij} - l_{ik} a_{kj}$</p> <p style="padding-left: 20px;">Для $i = 2$ до j</p> <p style="padding-left: 40px;">$l_{j,i-1} = a_{j,i-1} / a_{i-1,i-1}$</p> <p style="padding-left: 20px;">Для $k = 1$ до $i - 1$</p> <p style="padding-left: 40px;">$a_{ji} = a_{ji} - l_{jk} a_{ki}$</p>
--	---

Рис. 4.1. Алгоритмы окаймления известной части $\bar{L}U$ -разложения: столбцовый (слева) и алгоритм скалярных произведений (справа)

В первом цикле по i на рис. 4.1 (слева) выполняется модификация j -го столбца матрицы A и тем самым вычисляется j -й столбец матрицы U . Во втором цикле по i модифицируется j -я строка матрицы A и вычисляется j -я строка матрицы \bar{L} . Заметим, что при $i = j$ во втором цикле по i пересчитывается элемент (j, j) матрицы A ; в результате, согласно (4.2), получается элемент u_{jj} .

Во второй форме алгоритма окаймления на рис. 4.1 (справа) первый цикл по i, k вычисляет j -й столбец матрицы U , для чего из элементов a_{ij} вычитаются скалярные произведения строк с 2-й по $(j - 1)$ -ю матрицы \bar{L} с j -м столбцом U . Это эквивалентно решению системы $\bar{L}_{11} \mathbf{u}_{1j} = \mathbf{a}_{1j}$. Во втором

цикле по i, k модифицируется j -я строка A путем делений (нормировок) элементов этой строки, сопровождаемых опять-таки вычитаниями скалярных произведений j -й строки \bar{L} и столбцов U . Это эквивалентно решению треугольной системы $U_{11}^T \mathbf{l}_{j1} = \mathbf{a}_{j1}$ относительно j -й строки матрицы \bar{L} . Отметим, что здесь при $j = i$ модифицируется элемент (j, j) матрицы A , и это относится уже к вычислению j -го столбца матрицы U ; в результате получается элемент u_{jj} . Вычисления по этим формам показаны в табл. 4.1.

Таблица 4.1. Вычисления по алгоритмам на рис. 4.1 для примера 3.3. Позиции-элемента-делителя столбца \bar{L} показаны выделенным шрифтом

A	$j = 2$	$j = 3$	$j = 4$																																																																
<table border="1"> <tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr> <tr><td>1</td><td>4</td><td>2</td><td>1</td></tr> <tr><td>3</td><td>8</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>5</td><td>0</td><td>5</td></tr> </table>	2	4	-4	6	1	4	2	1	3	8	1	1	2	5	0	5	<table border="1"> <tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr> <tr><td>$1/2$</td><td>2</td><td>2</td><td>1</td></tr> <tr><td>3</td><td>8</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>5</td><td>0</td><td>5</td></tr> </table>	2	4	-4	6	$1/2$	2	2	1	3	8	1	1	2	5	0	5	<table border="1"> <tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr> <tr><td>$1/2$</td><td>2</td><td>4</td><td>1</td></tr> <tr><td>$3/2$</td><td>$2/2$</td><td>3</td><td>1</td></tr> <tr><td>2</td><td>5</td><td>0</td><td>5</td></tr> </table>	2	4	-4	6	$1/2$	2	4	1	$3/2$	$2/2$	3	1	2	5	0	5	<table border="1"> <tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr> <tr><td>$1/2$</td><td>2</td><td>4</td><td>-2</td></tr> <tr><td>$3/2$</td><td>$2/2$</td><td>3</td><td>-6</td></tr> <tr><td>$2/2$</td><td>$1/2$</td><td>$2/3$</td><td>4</td></tr> </table>	2	4	-4	6	$1/2$	2	4	-2	$3/2$	$2/2$	3	-6	$2/2$	$1/2$	$2/3$	4
2	4	-4	6																																																																
1	4	2	1																																																																
3	8	1	1																																																																
2	5	0	5																																																																
2	4	-4	6																																																																
$1/2$	2	2	1																																																																
3	8	1	1																																																																
2	5	0	5																																																																
2	4	-4	6																																																																
$1/2$	2	4	1																																																																
$3/2$	$2/2$	3	1																																																																
2	5	0	5																																																																
2	4	-4	6																																																																
$1/2$	2	4	-2																																																																
$3/2$	$2/2$	3	-6																																																																
$2/2$	$1/2$	$2/3$	4																																																																

В обеих формах окаймления обращения к данным производятся одинаково, что показано на рис. 4.2.

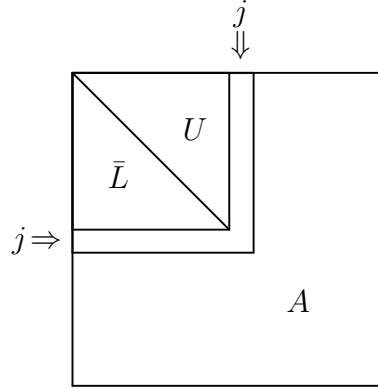


Рис. 4.2. Доступ к данным в алгоритмах окаймления известной части разложения. \bar{L} , U — вычисление закончено, но обращения происходят. A — обращений не было. Вычисляются: j -й столбец матрицы U и j -я строка матрицы \bar{L}

Обратим внимание, что в обоих случаях требуется доступ и к строкам, и к столбцам матрицы A . Поэтому алгоритмы будут неэффективны для векторных компьютеров, требующих, чтобы элементы вектора находились в смежных позициях памяти. Еще одной сложностью является внедрение процедуры выбора главного элемента в эти алгоритмы.

4.3 Окаймление неизвестной части разложения

Основная работа в алгоритмах окаймления приходится на решение треугольных систем (4.3). Это матрично-векторные операции, которые можно реализовать в виде подпрограмм по типу рис. 4.1, добиваясь в них максимальной для данной машины эффективности. Еще один способ организации вычислений, который называют *алгоритмом Донгарры–Айзенштата*, имеет то преимущество, что его основной операцией является матрично-векторное умножение. Математически алгоритм можно описать следующим образом. Выпишем из равенства (4.1) три других соотношения, на этот раз для a_{jj} , \mathbf{a}_{3j} и \mathbf{a}_{j3}^T . Отсюда получим

$$u_{jj} = a_{jj} - \mathbf{l}_{j1}^T \mathbf{u}_{1j}, \quad \mathbf{u}_{j3}^T = \mathbf{a}_{j3}^T - \mathbf{l}_{j1}^T U_{13}, \quad \mathbf{l}_{3j} = (\mathbf{a}_{3j} - L_{31} \mathbf{u}_{1j}) / u_{jj}. \quad (4.4)$$

Характер доступа к данным при таком вычислении показан на рис. 4.3.

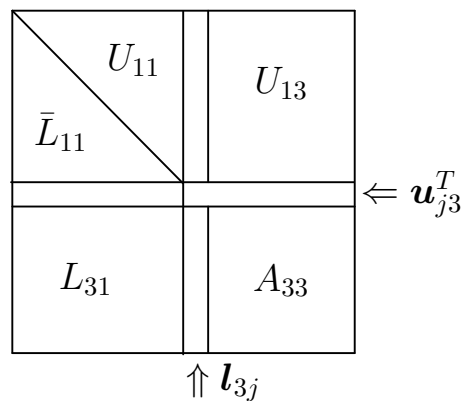


Рис. 4.3. Доступ к данным в алгоритмах окаймления неизвестной части разложения. \bar{L}_{11} , U_{11} — вычисление закончено, обращений больше нет. L_{31} , U_{13} — вычисление закончено, но обращения происходят. A_{33} — обращений не было. Вычисляются: j -й столбец \mathbf{l}_{3j} матрицы \bar{L} и j -я строка \mathbf{u}_{j3}^T матрицы \mathbf{U}

Видно, что блоки U_{13} и L_{31} , необходимые для вычислений (4.4), на каждый такой момент времени уже известны, так же как и все другие величины в правых частях равенств (4.4), поэтому здесь нет решения уравнений.

Основной операцией в (4.4) является умножение вектора на прямоугольную матрицу. Можно реализовать такие умножения посредством скалярных произведений или линейных комбинаций, что приводит к двум различным формам алгоритма, показанным на рис. 4.4.

Для $j = 1$ до n Для $k = 1$ до $j - 1$ Для $i = j$ до n $a_{ji} = a_{ji} - l_{jk}a_{ki}$ Для $k = 1$ до $j - 1$ Для $i = j + 1$ до n $a_{ij} = a_{ij} - l_{ik}a_{kj}$ Для $s = j + 1$ до n $l_{sj} = a_{sj}/a_{jj}$	Для $j = 1$ до n Для $i = j + 1$ до n Для $k = 1$ до $j - 1$ $a_{ij} = a_{ij} - l_{ik}a_{kj}$ Для $i = j$ до n Для $k = 1$ до $j - 1$ $a_{ji} = a_{ji} - l_{jk}a_{ki}$ Для $s = j + 1$ до n $l_{sj} = a_{sj}/a_{jj}$
--	--

Рис. 4.4. Алгоритмы Донгарры–Айзенштата окаймления неизвестной части $\bar{L}U$ -разложения: алгоритм линейных комбинаций (слева) и алгоритм скалярных произведений (справа)

Первый цикл по k, i на рис. 4.4 (слева) производит последовательные модификации j -й строки матрицы A , которая по окончании цикла k превращается в j -ю строку матрицы U . Эти модификации можно рассматривать как вычитание векторно-матричного произведения $\mathbf{l}_{j1}^T U_{13}$ из \mathbf{a}_{j3}^T во второй формуле $\mathbf{u}_{j3}^T = \mathbf{a}_{j3}^T - \mathbf{l}_{j1}^T U_{13}$ в (4.4) с помощью линейных комбинаций строк U . В случае $j = i$ результатом модификации будет первая величина u_{jj} в (4.4). Во второй паре циклов по k и i выполняются модификации j -го столбца матрицы A по формуле $\mathbf{l}_{3j} = (\mathbf{a}_{3j} - L_{31}\mathbf{u}_{1j})$, т.е. по второй формуле (4.4) с точностью до деления на u_{jj} с вычислением матрично-векторного произведения $L_{31}\mathbf{u}_{1j}$ в (4.4) посредством линейных операций столбцов матрицы L . Обратите внимание, — в отличие от алгоритма отложенных модификаций на рис. 3.6, теперь длины векторов, участвующих в линейных комбинациях, одинаковы. Второй оператор цикла с индексом k можно удалить, причем программа снова будет верна; мы вставили этот оператор, чтобы подчеркнуть наличие линейных комбинаций. Отметим еще, что в первом цикле по k, i происходит обращение к строкам матрицы A , а во втором цикле по k, i — к ее столбцам. Следовательно, эта форма неэффективна для векторных компьютеров, требующих размещения элементов вектора в смежных ячейках памяти.

Алгоритм на рис. 4.4 (справа) использует скалярные произведения. На j -м шаге первый цикл по i, k вычисляет, с точностью до финального деления, j -й столбец матрицы L ; с этой целью j -й столбец в A модифицируется посредством скалярных произведений строк L и j -го столбца U . Во втором

цикле по i , k вычисляется j -я строка матрицы U , для чего j -я строка в A модифицируется посредством скалярных произведений j -й строки L и столбцов U . Снова требуется доступ к строкам и столбцам матрицы A . Потенциальное преимущество алгоритма Донгарры–Айзенштата заключается в том, что в некоторых векторных компьютерах матрично-векторные умножения выполняются весьма эффективно. Вычисления по этим формам показаны в табл. 4.2.

Таблица 4.2. Вычисления по алгоритмам на рис. 4.4 для примера 3.3. Позиции-элемента-делителя столбца \bar{L} показаны выделенным шрифтом

$j = 1$				$j = 2$				$j = 3$				$j = 4$			
2	4	−4	6	2	4	−4	6	2	4	−4	6	2	4	−4	6
1/2	4	2	1	1/2	2	4	−2	1/2	2	4	−2	1/2	2	4	−2
3/2	8	1	1	3/2	2/2	1	1	3/2	2/2	3	−6	3/2	2/2	3	−6
2/2	5	0	5	2/2	1/2	0	5	2/2	1/2	2/3	5	2/2	1/2	2/3	4

Как и в алгоритмах окаймления известной части разложения (подразд. 4.2), в данных алгоритмах дополнительную сложность представляет внедрение процедуры выбора главного элемента.

4.4 Задание на лабораторный проект № 3

Написать и отладить программу, реализующую ваш вариант метода исключения с выбором главного элемента, для численного решения систем линейных алгебраических уравнений $Ax = f$, вычисления $\det A$ и A^{-1} . Предусмотреть сообщения, предупреждающие о невозможности решения указанных задач с заданной матрицей A .

Отделить следующие основные части программы:

- подпрограмму факторизации матрицы A , отвечающую вашему варианту метода исключения;
- подпрограмму решения систем линейных алгебраических уравнений;
- подпрограмму вычисления определителя матриц;
- подпрограммы обращения матриц;
- сервисные подпрограммы.

Уделить особое внимание эффективности программы (в смысле экономии оперативной памяти). Предусмотреть пошаговое выполнение алгоритма исключения с выводом результата на экран.

Выполнить следующие пункты задания.

1. Провести подсчет фактического числа выполняемых операций умножения и деления при решении системы линейных алгебраических уравнений, сравнить его с оценочным числом $(n^3/3)$.

2. Определить скорость решения задач (решение систем линейных алгебраических уравнений, обращение матриц) с учетом времени, затрачиваемого на разложение матрицы. Для этого спроектировать и провести эксперимент, который охватывает матрицы порядка от 5 до 100 (через 5 порядков). Представить результаты в виде таблицы и графика зависимости времени выполнения (в минутах и секундах) от порядка матриц. Таблицу и график вывести на экран.

3. Оценить точность решения систем линейных алгебраических уравнений, имеющих тот же самый порядок, что и задачи из пункта 2. Для этого сгенерировать случайные матрицы A , задать точное решение x^* и образовать правые части $f = Ax^*$. Провести анализ точности вычисленного решения x от порядка матрицы. Результаты представить в виде таблицы и графика.

Для заполнения матрицы A использовать случайные числа из диапазона от -100 до 100 . В качестве точного решения взять вектор $x^* = (1, 2, \dots, n)^T$, где n — порядок матрицы. Для оценки точности использовать норму вектора

$$\|x\|_{\infty} = \max_i (|x_i|). \quad (4.5)$$

4. Повторить пункт 3 задания для плохо обусловленных матриц (см. подразд. 2.6), имеющих порядок от 4 до 40 с шагом 4.

5. Вычислить матрицу A^{-1} следующими двумя способами.

Способ 1 — через решение системы $AX = I$, где I — единичная матрица.

Способ 2 — через разложение матрицы A в произведение элементарных матриц, обращение которых осуществляется аналитически, а их произведение дает матрицу A^{-1} .

Сравнить затраты машинного времени и точность обращения матриц при использовании указанных способов 1 и 2. Эксперименты провести для случайных матриц порядков от 5 до 100 через 5. Для оценки точности в обоих способах использовать оценочную формулу

$$\|A_{\text{т}}^{-1} - A_{\text{пр}}^{-1}\| \leq \|I - AA_{\text{пр}}^{-1}\| \cdot \|A\|^{-1}. \quad (4.6)$$

Норму матрицы следует вычислять в соответствии со следующим определением:

$$\|A\|_{\infty} = \max_i \left(\sum_{j=1}^n |a_{ij}| \right), \quad (4.7)$$

где A_T^{-1} — точное значение обратной матрицы, а $A_{пр}^{-1}$ — приближенное значение, полученное в результате обращения каждым из способов 1 и 2.

6. Провести подсчет фактического числа выполняемых операций умножения и деления при обращении матриц первым и вторым способами, сравнить его с оценочным числом (n^3) .

ЗАМЕЧАНИЕ 4.1. По ходу проведения численных экспериментов на экран дисплея должны выводиться следующие таблицы.

Решение систем линейных алгебраических уравнений:

Порядок	Время	Точность	Теоретическое число операций	Реальное число операций
---------	-------	----------	---------------------------------	----------------------------

Аналогичная таблица должна быть построена для плохо обусловленных матриц.

Обращение матриц:

Порядок	Время		Точность		Число операций		
	спос. 1	спос. 2	спос. 1	спос. 2	спос. 1	спос. 2	теорет.

ЗАМЕЧАНИЕ 4.2. Результаты экспериментов необходимо вывести на экран в форме следующих графиков. Для случая обращения матриц при построении графиков использовать данные из второй таблицы.

Графики решения систем линейных алгебраических уравнений:

- зависимость реального и оценочного числа операций от порядка матрицы (для разных графиков использовать разные цвета);
- зависимость времени решения от порядка матриц;
- зависимость точности решения от порядка матриц. При построении графиков использовать данные из первой таблицы. Для этого их необходимо записать в текстовый файл.

Графики для обращения матриц:

- зависимость реального и оценочного числа операций от порядка матрицы (для разных графиков использовать разные цвета);
- зависимость времени обращения первым и вторым способом от порядка матриц;
- зависимость точности обращения первым и вторым способом от порядка матриц.

4.5 Варианты задания на лабораторный проект № 3

В табл. 4.3 приведены 16 номеров вариантов задания на лабораторную работу (проект) № 3.

Если нет других указаний преподавателя, выбирайте ваш вариант по вашему номеру в журнале студенческой группы.

Таблица 4.3. Варианты задания на лабораторный проект № 3

Вид разложения	Окаймление α		Окаймление β	
	Алгоритм a	Алгоритм b	Алгоритм c	Алгоритм b
$A = \bar{L}U$	1	2	3	4
$A = L\bar{U}$	5	6	7	8
$A = \bar{U}L$	9	10	11	12
$A = U\bar{L}$	13	14	15	16

α — известной части разложения;

β — неизвестной части разложения;

a — столбцовый;

b — скалярных произведений;

c — линейных комбинаций.

РАЗРЕЖЕННЫЕ ФОРМЫ LU -РАЗЛОЖЕНИЯ

5.1 Упакованные формы хранения матриц

Для решения систем линейных алгебраических уравнений с разреженными матрицами используются те же самые методы гауссова исключения, что и для линейных систем с заполненными матрицами. Отличие состоит только в выборе главного элемента и в способе хранения матрицы коэффициентов системы уравнений [10].

Так как разреженные матрицы имеют небольшое число ненулевых элементов, в целях экономии оперативной памяти ЭВМ такие матрицы хранят в *упакованном виде*. Рассмотрим четыре наиболее употребимых способа упаковки, используемых для хранения произвольных разреженных матриц.

ПРИМЕР 5.1. В качестве примера возьмем квадратную матрицу A порядка 6 с тринадцатью ненулевыми элементами: $a_{11} = 1, a_{13} = 3, a_{14} = -2, a_{21} = 1, a_{25} = 5, a_{33} = 7, a_{34} = 2, a_{42} = -3, a_{46} = -1, a_{51} = 1, a_{54} = 3, a_{65} = 2, a_{66} = 2$.

В излагаемых ниже схемах хранения разреженной матрицы A упаковка осуществляется по строкам.

Схема 1. Каждому ненулевому элементу матрицы A ставится в соответствие запись, состоящая из двух полей. Первое поле записи содержит номер столбца, а второе — значение элемента. Нуль во втором поле означает начало новой строки. В этом случае первое поле содержит номер новой строки. Нули в обоих полях записи указывают на конец массива, хранящего данную разреженную матрицу A . В соответствии с этой схемой матрица A примера 5.1 будет храниться в виде следующего массива:

1	0	1	1.0	3	3.0	4	-2.0	2	0	1	1.0	5	5.0	⇒
3	0	3	7.0	4	2.0	4	0	2	-3.0	6	-1.0	5	0	⇒
1	1.0	4	3.0	6	0	5	2.0	6	2.0	0	0			

Схема 2. Информация о матрице хранится в трех массивах. В массиве **a** хранятся ненулевые элементы матрицы A . В массиве **b** хранятся индексы столбцов, а в массиве **c** — указатели индексов строк, т. е. на k -м месте в массиве **c** хранится местоположение первого ненулевого элемента k -й строки в массиве **a**. В соответствии с этой схемой матрица A примера 5.1 будет храниться в виде трех массивов

$$\begin{aligned} \mathbf{a} &= (1.0, 3.0, -2.0, 1.0, 5.0, 7.0, 2.0, -3.0, -1.0, 1.0, 3.0, 2.0, 2.0), \\ \mathbf{b} &= (1, 3, 4, 1, 5, 3, 4, 2, 6, 1, 4, 5, 6), \\ \mathbf{c} &= (1, 4, 6, 8, 10, 12). \end{aligned}$$

Схема 3. Каждому ненулевому элементу данной матрицы однозначно ставится в соответствие целое число вида

$$\lambda(i, j) = (i - 1)n + j, \quad a_{ij} \neq 0. \quad (5.1)$$

Хранение ненулевых элементов разреженной матрицы обеспечивается двумя массивами. В массиве **a** хранятся ненулевые элементы матрицы, в массиве **b** хранятся соответствующие им числа $\lambda(i, j)$. В соответствии с этой схемой матрица A примера 5.1 будет храниться в виде двух массивов:

$$\begin{aligned} \mathbf{a} &= (1.0, 3.0, -2.0, 1.0, 5.0, 7.0, 2.0, -3.0, -1.0, 1.0, 3.0, 2.0, 2.0), \\ \mathbf{b} &= (1, 3, 4, 7, 11, 15, 16, 20, 24, 25, 28, 35, 36). \end{aligned}$$

Исходная матрица по этой схеме хранения может быть восстановлена следующим образом. Сначала определяем i как такое наименьшее целое число, что

$$i \geq \lambda(i, j)/n.$$

Затем, зная i , с учетом (5.1) находим j

$$j = \lambda(i, j) - (i - 1)n. \quad (5.2)$$

Схема 4. Для хранения каждого ненулевого элемента матрицы используется запись, состоящая из трех полей. В первом поле хранится номер столбца, в котором стоит этот ненулевой элемент. Во втором поле хранится значение элемента, а в третьем — указатель на следующий ненулевой элемент строки или *nil*, если это последний ненулевой элемент в строке.

Таким образом, разреженная матрица хранится в виде массива указателей на списки, а каждый список содержит все ненулевые элементы одной строки. Упакованную форму матрицы A примера 5.1 в этом случае можно схематично изобразить следующим образом.

$$\begin{array}{lclclcl}
 \boxed{1} & \rightarrow & \boxed{1} \mid \boxed{1.0} \mid \boxed{} & \rightarrow & \boxed{3} \mid \boxed{3.0} \mid \boxed{} & \rightarrow & \boxed{4} \mid \boxed{-2.0} \mid \boxed{} & \rightarrow & nil \\
 \boxed{2} & \rightarrow & \boxed{1} \mid \boxed{1.0} \mid \boxed{} & \rightarrow & \boxed{5} \mid \boxed{5.0} \mid \boxed{} & \rightarrow & nil \\
 \boxed{3} & \rightarrow & \boxed{3} \mid \boxed{7.0} \mid \boxed{} & \rightarrow & \boxed{4} \mid \boxed{2.0} \mid \boxed{} & \rightarrow & nil \\
 \boxed{4} & \rightarrow & \boxed{2} \mid \boxed{-3.0} \mid \boxed{} & \rightarrow & \boxed{6} \mid \boxed{-1.0} \mid \boxed{} & \rightarrow & nil \\
 \boxed{5} & \rightarrow & \boxed{1} \mid \boxed{1.0} \mid \boxed{} & \rightarrow & \boxed{4} \mid \boxed{3.0} \mid \boxed{} & \rightarrow & nil \\
 \boxed{6} & \rightarrow & \boxed{5} \mid \boxed{2.0} \mid \boxed{} & \rightarrow & \boxed{6} \mid \boxed{2.0} \mid \boxed{} & \rightarrow & nil
 \end{array}$$

5.2 Выбор ведущего элемента

Способы 1–4 упаковки матриц позволяют компактно хранить матрицу A коэффициентов системы $Ax = f$. Однако при использовании метода Гаусса (или ему подобных) в результате модификации элементов матрицы A может значительно возрасти число ненулевых элементов. С одной стороны, это требует дополнительной памяти для хранения новых ненулевых элементов, а с другой — приводит к возрастанию числа арифметических операций, что влечет накопление ошибок округления. В связи с этим обстоятельством были предложены стратегии выбора главного элемента, позволяющие минимизировать число новых ненулевых элементов на каждом шаге метода Гаусса.

Назовем *локальным заполнением* на $(k + 1)$ -м шаге метода Гаусса число элементов матрицы A , которые были нулевыми после k -го шага и стали ненулевыми после $(k + 1)$ -го шага метода Гаусса. Таким образом, задача заключается в выборе в качестве главного элемента такого элемента матрицы $A^{(k)}$, который минимизирует локальное заполнение матрицы A на $(k + 1)$ -м шаге (как и прежде, $A^{(k)}$ — активная подматрица матрицы A). При этом, чем больше множество элементов, среди которых выбирается главный, тем меньше локальное заполнение. Но, с другой стороны, в качестве главного можно брать только ненулевой элемент. Поэтому вводится понятие допустимого элемента.

Допустимым элементом на $(k + 1)$ -м шаге метода Гаусса называется такой элемент активной подматрицы $A^{(k)}$, который удовлетворяет неравенству

$$|a_{ij}^{(k)}| > \varepsilon,$$

где ε — некоторое наперед заданное положительное число. В лабораторном проекте № 4, описание которого дано ниже, надо взять $\varepsilon = 10^{-3}$, если используется тип *real*, или $\varepsilon = 10^{-5}$, если используется тип *extended*.

Итак, среди элементов активной подматрицы $A^{(k)}$ в соответствии с критерием (5.2) выбирается множество допустимых элементов, а затем среди них отыскивается элемент, минимизирующий локальное заполнение на текущем шаге. Для этого используются следующие две *стратегии выбора оптимального ведущего элемента*.

Стратегия I. Локальное заполнение на $(k + 1)$ -м шаге метода Гаусса будет минимальным, если в качестве главного (ведущего) выбрать элемент $a_{st}^{(k)}$ на позиции $s = \alpha + k$, $t = \beta + k$, где α и β определяются из формулы

$$g_{\alpha\beta}^{(k+1)} = \min_{i,j} \{e_i^T G_{k+1} e_j\} \quad \text{для всех} \quad \left| a_{i+k,j+k}^{(k)} \right| > \varepsilon.$$

Здесь $G_{k+1} = B_k \bar{B}_k^T B_k$, где B_k — матрица, полученная из $A^{(k)}$ путем замены ненулевых элементов единицами, а $\bar{B}_k = M - B_k$ (M — матрица, состоящая из единиц). Согласно стратегии I, в качестве главного берется тот из числа допустимых элементов активной подматрицы $A^{(k)}$, которому в матрице G_{k+1} соответствует наименьший элемент.

Стратегия II. Локальное заполнение на $k + 1$ -м шаге метода Гаусса будет небольшим, если в качестве главного (ведущего) выбрать элемент $a_{st}^{(k)}$, на позиции $s = \alpha + k$, $t = \beta + k$, где α и β определяются из формулы

$$g_{\alpha\beta}^{(k+1)} = \min_{i,j} \{e_i^T \hat{G}_{k+1} e_j\} \quad \text{для всех} \quad \left| a_{i+k,j+k}^{(k)} \right| > \varepsilon.$$

Здесь $\hat{G}_{k+1} = (B_k - I_{n-k})M(B_k - I_{n-k})$, где матрицы M и B_k имеют тот же самый смысл, что и в стратегии I, а I_{n-k} обозначает единичную матрицу размера $n - k$. Хотя стратегия II не обеспечивает минимальное заполнение на текущем шаге, она очень просто реализуется на практике, и ее применение приводит к сравнительно небольшому числу новых ненулевых элементов.

Использование упакованной формы хранения матрицы A и более сложной процедуры выбора главного элемента требует значительного увеличения затрат машинного времени, поэтому перенос процедур и функций из лабораторного проекта № 1, осуществляющих решение системы линейных алгебраических уравнений, не даст оптимальный по времени результат. Это связано с тем, что поиск (i, j) -го элемента матрицы A в упакованной форме требует существенных затрат машинного времени. Следовательно, при написании оптимальной по времени счета программы таких действий надо избегать.

Это можно сделать, если применить метод Гаусса непосредственно к упакованной форме хранения матрицы A , т. е. только к ее ненулевым элементам.

Поскольку стандартный метод Гаусса (см. подразд. 3.1) оперирует со строками матрицы A , разреженные матрицы удобно хранить по строкам. Это позволит эффективно организовать такие операции, как нормировка строки, умножение строки на число, вычитание строки, потому что в этом случае все ненулевые элементы каждой строки матрицы A в упакованной форме хранения образуют непрерывную последовательность. Следовательно, операции нормировки, умножения и вычитания строки можно проводить только для ненулевых элементов.

Такой подход подразумевает, что ненулевые элементы матрицы A модифицируются в том порядке, в котором они хранятся в упакованной форме, что исключает затраты на поиск нужного элемента. Модификация происходит следующим образом. Берется очередной ненулевой элемент матрицы A . Определяется его местоположение в матрице, т. е. индексы i и j . Затем с помощью дополнительных массивов обратных перестановок определяется местоположение этого элемента в матрице с переставленными столбцами и строками. Если в результате этот элемент лежит в активной подматрице, то он изменяется по известным формулам метода Гаусса. Здесь надо предусмотреть процедуру вставки элемента на случай появления нового ненулевого элемента и процедуру удаления элемента из упакованной формы, если ненулевой элемент стал нулевым. После этого обрабатывается следующий ненулевой элемент матрицы A .

Оптимизация по времени процедуры решения системы линейных алгебраических уравнений позволяет значительно повысить эффективность программы. Однако этого недостаточно. Оптимальной по быстродействию будет лишь та программа, где дополнительно оптимизирована процедура выбора главного элемента. Во-первых, надо подумать над эффективным вычислением элементов матрицы G_k (или \hat{G}_k). Во-вторых, надо организовать вычисление этой матрицы так, чтобы исключить поиск элементов в упакованной форме. Только учет всех этих особенностей решения систем линейных алгебраических уравнений с разреженной матрицей коэффициентов позволит написать эффективную по затратам машинного времени программу.

5.3 Задание на лабораторный проект № 4

Написать и отладить программу, реализующую заданный вариант метода исключения с заданной схемой хранения разреженных матриц и заданной стратегией выбора главного элемента, для численного решения систем вида $Ax = f$. Отделить основные части программы:

- а) подпрограмму упаковки матрицы A ;
- б) подпрограмму метода исключения;
- в) подпрограмму выбора главного элемента;
- г) сервисные подпрограммы.

Уделить особое внимание эффективности программы (в смысле скорости счета). Программа должна решать систему линейных алгебраических уравнений порядка $n = 200$ не более чем за 3 минуты (для персонального компьютера 486 АТ с сопроцессором). Предусмотреть пошаговое выполнение алгоритма исключения с выводом результата на экран.

Выполнить следующие пункты задания:

1. Для заданной матрицы A выдать на экран упакованную форму в соответствии со своим вариантом, построить таблицу зависимости оценочного и реального локального заполнения от номера шага исключения (для этого предусмотреть ввод матрицы с экрана).

2. Оценить точность решения систем линейных алгебраических уравнений, имеющих порядок n от 100 до 200 (через 5). Для этого сгенерировать случайные матрицы A (не более 10 ненулевых элементов в строке), выбрать x^* — точное решение и образовать правые части $f = Ax^*$. Провести анализ точности решения как функцию от n . Результаты вывести в таблицу и на график.

Для случайного заполнения матрицы A использовать алгоритм:

(а) Ненулевыми целыми числами, выбранными случайным образом из интервала $[-100; 100]$, заполнить обратную диагональ матрицы A .

(б) В каждой строке случайным образом выбрать количество ненулевых элементов (от 1 до 10 с учетом элементов по пункту (а)), их местоположение (номер столбца от 1 до n) и значение (ненулевые целые числа, лежащие в интервале от -100 до 100).

В качестве точного решения взять вектор $x^* = (1, 2, \dots, n)$. Если при решении системы $Ax = f$ выяснится, что матрица A вырождена (плохо обусловлена), сгенерировать новую матрицу того же порядка и решить систему линейных алгебраических уравнений с новой матрицей A и новой правой

частью. Для оценки точности решения использовать норму вектора по формуле (4.5).

3. Определить скорость решения систем из пункта 2. Результаты вывести в таблицу и на график.

4. Системы из пункта 2 решить методом исключения переменных двумя способами: способ 1 — из лабораторного проекта № 1, способ 2 — из лабораторного проекта № 2 (в соответствии со своим вариантом по лабораторному проекту № 1 или № 2). В этом случае разреженная матрица должна размещаться в памяти ЭВМ полностью (в распакованном виде). Сравнить точность решения и затраты машинного времени, получаемые, с одной стороны, в лабораторном проекте № 1 (или № 2) и, с другой стороны, в лабораторном проекте № 4.

ЗАМЕЧАНИЕ 5.1. По ходу проведения численных экспериментов на экран дисплея должны выводиться таблицы следующего вида.

Решение систем линейных алгебраических уравнений

Порядок матрицы	Время		Точность	
	Заполненная матрица	Разреженная матрица	Заполненная матрица	Разреженная матрица

ЗАМЕЧАНИЕ 5.2. Некоторые результаты экспериментов необходимо сохранять в текстовый файл, чтобы затем вывести на экран в виде графиков.

Графики решения систем линейных алгебраических уравнений:

- зависимость точности решения от порядка матриц для способа 1 решения (см. п. 4);
- зависимость точности решения от порядка матриц для способа 2 решения (см. п. 4);
- зависимость времени решения от порядка матриц для способа 1 решения (см. п. 4);
- зависимость времени решения от порядка матриц для способа 2 решения (см. п. 4).

Таблица 5.1. Варианты задания на лабораторный проект № 4

Вид разложения	Стратегия I				Стратегия II			
	a	b	c	d	a	b	c	d
$A = \bar{L}U$	1	2	3	4	5	6	7	8
$A = L\bar{U}$	9	10	11	12	13	14	15	16
$A = \bar{U}L$	17	18	19	20	21	22	23	24
$A = U\bar{L}$	25	26	27	28	29	30	31	32
$A = L\bar{U}$ в виде L, \bar{U}^{-1}	33	34	35	36	37	38	39	40
$A = U\bar{L}$ в виде \bar{L}^{-1}, U	41	42	43	44	45	46	47	48

^a — схема 1 хранения разреженной матрицы A ;

^b — схема 2 хранения разреженной матрицы A ;

^c — схема 3 хранения разреженной матрицы A ;

^d — схема 4 хранения разреженной матрицы A .

5.4 Варианты задания на лабораторный проект № 4

Студент определяет номер своего варианта по табл. 5.1 (см. выше) соответственно своему порядковому номеру в списке учебной группы. В таблице приведены 48 номеров вариантов задания на лабораторный проект № 4. Все варианты различаются по следующим признакам:

- стратегии I или II выбора главного элемента;
- четыре схемы упаковки и хранения разреженной матрицы A ;
- шесть вариантов метода исключения, определяемых видом разложения матрицы.

6

РАЗЛОЖЕНИЯ ХОЛЕССКОГО

6.1 Положительно определенные матрицы

ОПРЕДЕЛЕНИЕ 6.1. Симметрическая матрица P , $P(n, n)^1$, называется *положительно определенной* (ПО), если и только если $x^T P x > 0$ для всех n -векторов x с $\|x\| > 0$.

В определении 6.1 выражение $x^T P x$ есть *квадратичная форма* в терминах переменных элементов некоторого (произвольного) вектора $x \in \mathbb{R}^n$. Например, для размерности $n = 3$ этого вектора имеем

$$\begin{aligned} x^T P x &= \sum_{i,j=1}^n p(i, j) x_i x_j = \\ &= p(1, 1)x_1^2 + 2p(1, 2)x_1 x_2 + 2p(1, 3)x_1 x_3 + \\ &\quad + p(2, 2)x_2^2 + 2p(2, 3)x_2 x_3 + \\ &\quad + p(3, 3)x_3^2. \end{aligned}$$

Этот очевидный способ раскрытия формулы для $x^T P x$ справелив и в общем случае (для любого целого $n \geq 2$), — он обобщает формулу квадрата суммы двух или более чисел.

Свойства ПО матриц

Когда P есть симметрическая матрица, обозначение $P > 0$ означает, что P является положительно определенной.

Свойство А. $P > 0$ тогда и только тогда, когда все собственные числа матрицы P положительны.

¹ Т. е. P имеет размер $(n \times n)$.

Свойство В. Если $P > 0$, то все диагональные элементы матрицы P положительны.

Свойство С. Если матрица M невырожденная и $P > 0$, то $M^T P M > 0$.

Свойство D. Если $P > 0$, то P^{-1} существует и $P^{-1} > 0$.

Свойство Е. Если $P > 0$, то матрица, полученная вычеркиванием i -й строки и i -го столбца, также является положительно определенной.

Свойство F. Если $P > 0$ и $\rho(i, j) = p(i, j)/[p(i, i)p(j, j)]^{1/2}$ с индексами элементов $i, j = 1, \dots, n$ при $i \neq j$, то $|\rho(i, j)| < 1$, при этом $\rho(i, j)$ называются *коэффициентами корреляции*.

Признаки положительной определенности матриц

Критерий Сильвестра. Чтобы матрица P была положительно определенной, необходимо и достаточно, чтобы все ее главные миноры были положительны.

Достаточное условие. *Диагональное преобладание*, т. е. свойство

$$\forall i = 1, \dots, n : \quad p(i, i) > \sum_{j=1, j \neq i}^n |p(i, j)|,$$

влечет положительную определенность матрицы $P = [p(i, j)]$.

6.2 Квадратные корни из P и алгоритмы Холесского

ОПРЕДЕЛЕНИЕ 6.2. Если матрица P может быть представлена как

$$P = SS^T \tag{6.1}$$

с квадратной матрицей S , то S называют *квадратным корнем* из P [97]. Квадратные корни матриц, когда они существуют, определяются равенством (6.1) неединственным образом, так как, если S удовлетворяет равенству (6.1) и T есть любая ортогональная ($TT^T = I$ и $T^T T = I$) матрица, то ST также удовлетворяет (6.1).

ЗАМЕЧАНИЕ 6.1. Хотя определение 6.2, т.е. формула (6.1), часто используется, оно не является универсальным. Обобщения заключаются в отказе от квадратной формы матрицы S или (если она остается квадратной) в допущении комплексных значений ее элементов. В последнем случае пишут $P = SS^H$, где S^H обозначает комплексно сопряженную при транспонировании матрицу. Ограничения в определении квадратного корня из матрицы могут заключаться в различных требованиях к ее форме. Например, можно требовать симметричности $S = S^T$ или эрмитовости $S = S^H$. В случае симметричности, т.е. при $S = S^T$, квадратный корень из матрицы P обозначают $S^{1/2}$, т.е. пишут $P = SS^{1/2}$. Если же требовать, чтобы S в (6.1) имела треугольную форму, то в (6.1) приходим к четырем вариантам разложения Холесского [13, 15].

ОПРЕДЕЛЕНИЕ 6.3. *Разложением Холесского* принято называть любое из следующих представлений положительно определенной матрицы P :

$$P = LL^T, \quad P = \bar{L}D\bar{L}^T, \quad P = UU^T, \quad P = \bar{U}D\bar{U}^T, \quad (6.2)$$

где L — нижняя треугольная матрица с положительными элементами на диагонали, U — верхняя треугольная матрица с положительными элементами на диагонали, \bar{L} — нижняя треугольная матрица с единичными элементами на диагонали, D — диагональная матрица с положительными элементами на диагонали, \bar{U} — верхняя треугольная матрица с единичными элементами на диагонали.

ТЕОРЕМА 6.1 (*Нижнее треугольное разложение Холесского*). Симметрическая матрица $P > 0$ имеет разложение $P = LL^T$, где L — нижняя треугольная матрица. Это разложение на сомножители с положительными диагональными элементами в L дается следующим алгоритмом.

Для $j = 1, \dots, n - 1$ рекуррентно выполнять цикл, образованный следующим упорядоченным набором выражений:

$$\left. \begin{aligned} L(j, j) &= P(j, j)^{1/2}, \\ L(k, j) &= P(k, j)/L(j, j), \quad k = j + 1, \dots, n, \\ P(i, k) &:= P(i, k) - L(i, j)L(k, j) \quad \left\{ \begin{array}{l} k = j + 1, \dots, n \\ i = k, \dots, n \end{array} \right. \end{aligned} \right\} L(n, n) = P(n, n)^{1/2}.$$

ЗАМЕЧАНИЕ 6.2. Приведенная формулировка алгоритма использует обозначения элементов матриц P и L для наглядности. Это не должно создавать впечатления, что данные матрицы хранятся в памяти по отдельности; наоборот, в лабораторном проекте все приведенные вычисления должны

проводиться в одном и том же массиве. Это значит, что элементы матрицы P должны замещаться элементами матрицы L по мере вычисления последних.

ЗАМЕЧАНИЕ 6.3. Легко видеть, что если L_1 и L_1 суть два различных варианта факторизации матрицы $P > 0$, то

$$L_1 = L_2 \operatorname{diag} [\pm 1, \dots, \pm 1],$$

т. е. в приведенном алгоритме можно брать $L(j, j) = \pm P(j, j)^{1/2}$. Обычно рекомендуют выбирать единственное решение, отвечающее положительным диагональным элементам матрицы L .

Внимательное прочтение теоремы 6.1 обнаруживает, что данный алгоритм требует вычисления n квадратных корней, что замедляет вычисления. Этот недостаток устраняется, если перейти ко второму варианту разложения Холесского из общего списка (6.2).

СЛЕДСТВИЕ 6.1 (*Нижнее треугольное разложение Холесского без операции квадратного корня*). Симметрическая матрица $P > 0$ имеет разложение $P = \bar{L} D \bar{L}^T$, где \bar{L} — нижняя треугольная матрица с единичной диагональю и $D = \operatorname{diag} (d_1, \dots, d_n)$. Элементы матриц \bar{L} и D даются следующим алгоритмом.

Для $j = 1, \dots, n - 1$ рекуррентно выполнять цикл, образованный следующим упорядоченным набором выражений:

$$\left. \begin{aligned} d_j &= P(j, j), & \bar{L}(j, j) &= 1, \\ P(i, k) &:= P(i, k) - \bar{L}(i, j) P(k, j) & \left\{ \begin{array}{l} k = j + 1, \dots, n \\ i = k, \dots, n \end{array} \right. & \left\{ \begin{array}{l} d(n) = P(n, n), \\ \bar{L}(n, n) = 1. \end{array} \right. \\ \bar{L}(k, j) &= P(k, j) / d(j), & k &= j + 1, \dots, n \end{aligned} \right\}$$

Данный результат легко получается из теоремы 6.1 с использованием обозначений $d_j = L(j, j)^2$ и $\bar{L}(i, j) = L(i, j) / L(j, j)$.

Следующий аналог теоремы 6.1 формулирует алгоритм третьей версии разложения Холесского из списка (6.2).

ТЕОРЕМА 6.2 (*Верхнее треугольное разложение Холесского*). Симметрическая матрица $P > 0$ имеет разложение $P = U U^T$, где U — верхняя треугольная матрица. Это разложение на сомножители с положительными диагональными элементами в U дается следующим алгоритмом.

Для $j = n, n - 1, \dots, 2$ рекуррентно выполнять цикл, образованный следующим упорядоченным набором выражений:

$$\left. \begin{aligned} U(j, j) &= P(j, j)^{1/2}, \\ U(k, j) &= P(k, j)/U(j, j), \quad k = 1, \dots, j - 1, \\ P(i, k) &:= P(i, k) - U(i, j)U(k, j) \quad \left\{ \begin{array}{l} k = 1, \dots, j - 1 \\ i = 1, \dots, k \end{array} \right\} \end{aligned} \right\} U(1, 1) = P(1, 1)^{1/2}.$$

Аналогично следствию 6.1, зафиксируем последний из четырех вариантов разложения Холецкого (6.2).

СЛЕДСТВИЕ 6.2 (*Верхнее треугольное разложение Холецкого без операции квадратного корня*). Симметрическая матрица $P > 0$ имеет разложение $P = \bar{U}D\bar{U}^T$, где \bar{U} — верхняя треугольная матрица с единичной диагональю и $D = \text{diag}(d_1, \dots, d_n)$. Элементы матриц \bar{U} и D даются следующим алгоритмом.

Для $j = n, n - 1, \dots, 2$ рекуррентно выполнять цикл, образованный следующим упорядоченным набором выражений:

$$\left. \begin{aligned} d_j &= P(j, j), \quad \bar{U}(j, j) = 1, \\ \bar{U}(k, j) &= P(k, j)/d_j, \quad k = 1, \dots, j - 1, \\ P(i, k) &:= P(i, k) - \bar{U}(i, j)\bar{U}(k, j)d_j \quad \left\{ \begin{array}{l} k = 1, \dots, j - 1 \\ i = 1, \dots, k \end{array} \right\} \end{aligned} \right\} \begin{aligned} d(1) &= P(1, 1), \\ \bar{U}(1, 1) &= 1. \end{aligned}$$

6.3 Программная реализация алгоритмов Холецкого

В справочных целях включаем реализацию трех из четырех приведенных выше алгоритмов на языке FORTRAN [15]. Эти примеры реализации могут помочь студентам написать свои собственные программы на других языках высокого уровня при выполнении лабораторного проекта № 5, описание которого дано ниже в подразд. 6.7 [97, 15].

Верхнетреугольное разложение Холецкого:

 $P(N, N), \quad P = UU^T, \quad P > 0, \quad U — \text{верхнетреугольная матрица.}$

DO 5 $J = N, 2, -1$ © $j = n, n - 1, \dots, 2$
 $U(J, J) = \text{SQRT}(P(J, J))$
 $\alpha = 1./U(J, J)$
DO 5 $K = 1, J - 1$
 $U(K, J) = \alpha * P(K, J)$
 $\beta = U(K, J)$
DO 5 $I = 1, K$
5 $P(I, K) = P(I, K) - \beta * U(I, J)$
 $U(1, 1) = \text{SQRT}(P(1, 1))$

ЗАМЕЧАНИЕ 6.4. Матрица U должна замещать P в компьютерной памяти. Нижние части матриц U и P не должны использоваться вовсе (память для них не выделяется). В любом случае верхнетреугольная часть матрицы P теряется, так как на ее месте появляется верхнетреугольная часть матрицы U , т. е. все вычисления ведутся в одном и том же массиве P .

Верхнетреугольное без $\sqrt{\cdot}$ разложение Холецкого:

 $P(N, N), \quad P = \bar{U}D\bar{U}^T, \quad P > 0, \quad \bar{U} — \text{верхнетреугольная матрица}$
с единичной диагональю, $D — \text{диагональная матрица.}$

DO 5 $J = N, 2, -1$ © $j = n, n - 1, \dots, 2$
 $D(J) = P(J, J)$
 $\alpha = 1./D(J)$
DO 5 $K = 1, J - 1$
 $\beta = P(K, J)$
 $\bar{U}(K, J) = \alpha * \beta$
DO 5 $I = 1, K$
5 $P(I, K) = P(I, K) - \beta * \bar{U}(I, J)$
 $D(1) = P(1, 1)$

ЗАМЕЧАНИЕ 6.5. В любом случае верхнетреугольная часть матрицы P теряется, так как на ее месте появляется верхнетреугольная часть матрицы \bar{U} , при этом единицы, соответствующие диагонали матрицы \bar{U} , только подразумеваются, а на их месте пишутся диагональные элементы матрицы D . Для поддиагональной части массива P память не выделяется.

Предыдущие замечания 6.4 и 6.5 свидетельствуют, что фактически массив, выделяемый для исходной матрицы P и одновременно для получаемых на ее месте результатов разложений Холесского, должен быть оформлен как одномерный массив. Размер этого массива, очевидно, равен $N(N+1)/2$ элементов. Напишем для предыдущего алгоритма его эквивалентную «одномерную» версию.

Верхнетреугольное без $\sqrt{\cdot}$ «одномерное» разложение $P = \bar{U}D\bar{U}^T$:

Одномерный массив $P(N(N+1)/2)$ соответствует $P = \bar{U}D\bar{U}^T$.

$JJ = N(N+1)/2$	
$JJN = JJ$	
DO 5 $J = N, 2, -1$	© $j = n, n-1, \dots, 2$
$\alpha = 1./P(JJ)$	
$KK = 0$	
$JJN = JJ - J$	© $JJN =$ следующий диагональный элемент
DO 4 $K = 1, J - 1$	
$\beta = P(JJN + K)$	© $JJN + K = (K, J)$
$P(JJN + K) = \alpha * \beta$	
DO 3 $I = 1, K$	
3 $P(KK + I) = P(KK + I) - \beta * P(JJN + I)$	© $KK + I = (I, K)$
4 $KK = KK + K$	© $KK = K(K-1)/2$
5 $JJ = JJN$	© $JJ = J(J-1)/2$

6.4 Разложение Холесского: ijk -формы

Разложение Холесского симметричной положительно определенной матрицы P может быть получено в результате незначительных изменений базовых LU -разложений квадратной матрицы A , изложенных в подразд. 3.5. При этом симметрия матрицы P используется для сокращения числа действий примерно вдвое. Способ хранения матрицы P должен быть компактным, т. е. в одномерном массиве хранится по строкам (или по столбцам) только нижняя (или верхняя) треугольная часть матрицы P вместе с диагональю.

В той же последовательности, как выше изложены (см. подразд. 3.5) ijk -формы $\bar{L}U$ -разложения матрицы A , приведем ijk -формы $\bar{L}D\bar{L}^T$ и LL^T -разложений Холесского матрицы $P > 0$. Из них видно, сколь незначительны

требуемые изменения. В каждом алгоритме объединены оба разложения, при этом те изменения, что относятся к LL^T -разложению, заключены в скобки. В приводимых ниже алгоритмах явно не указано, когда элементы матриц D , \bar{L} и L должны замещать соответствующие элементы исходной матрицы P . Такие замещения могут происходить сразу, а могут откладываться до того момента, когда элементы матрицы P станут ненужными для дальнейших вычислений. В этом отношении не все ijk -формы одинаково экономичны в реализации, и для каждой из них вопрос о скорейшем замещении исходных элементов матрицы P нужно решать отдельно.

**Два алгоритма Холецкого: разложения LL^T и $\bar{L}D\bar{L}^T$
с немедленными модификациями**

1) kij -алгоритм

$$(l_{11} = p_{11}^{1/2})$$

Для $k = 1$ до $n - 1$

Для $i = k + 1$ до n

$l_{ik} = p_{ik}/p_{kk}$

($l_{ik} = p_{ik}/l_{kk}$)

Для $j = k + 1$ до i

$p_{ij} = p_{ij} - l_{ik}p_{jk}$

($p_{ij} = p_{ij} - l_{ik}l_{jk}$)

($l_{k+1,k+1} = p_{k+1,k+1}^{1/2}$)

2) kji -алгоритм

$$(l_{11} = p_{11}^{1/2})$$

Для $k = 1$ до $n - 1$

Для $s = k + 1$ до n

$l_{sk} = p_{sk}/p_{kk}$ (p_{sk}/l_{kk})

Для $j = k + 1$ до n

Для $i = j$ до n

$p_{ij} = p_{ij} - l_{ik}p_{jk}$

($p_{ij} = p_{ij} - l_{ik}l_{jk}$)

($l_{k+1,k+1} = p_{k+1,k+1}^{1/2}$)

**Четыре алгоритма Холецкого: разложения LL^T и $\bar{L}D\bar{L}^T$
с отложенными модификациями**

3) jki -алгоритм

$$(l_{11} = p_{11}^{1/2})$$

Для $j = 2$ до n

Для $s = j$ до n

$l_{s,j-1} = p_{s,j-1}/p_{j-1,j-1}$

($l_{s,j-1} = p_{s,j-1}/l_{j-1,j-1}$)

Для $k = 1$ до $j - 1$

Для $i = j$ до n

$p_{ij} = p_{ij} - l_{ik}p_{jk}$

($p_{ij} = p_{ij} - l_{ik}l_{jk}$)

($l_{j,j} = p_{j,j}^{1/2}$)

4) jik -алгоритм

$$(l_{11} = p_{11}^{1/2})$$

Для $j = 2$ до n

Для $s = j$ до n

$l_{s,j-1} = p_{s,j-1}/p_{j-1,j-1}$

($l_{s,j-1} = p_{s,j-1}/l_{j-1,j-1}$)

Для $i = j$ до n

Для $k = 1$ до $j - 1$

$p_{ij} = p_{ij} - l_{ik}p_{jk}$

($p_{ij} = p_{ij} - l_{ik}l_{jk}$)

($l_{j,j} = p_{j,j}^{1/2}$)

5) *ikj*-алгоритм
$$(l_{11} = p_{11}^{1/2})$$

Для $i = 2$ до n

 Для $k = 1$ до $i - 1$

$l_{i,k} = p_{i,k}/p_{k,k}$

$(l_{i,k} = p_{i,k}/l_{k,k})$

 Для $j = k + 1$ до i

$p_{ij} = p_{ij} - l_{ik}p_{jk}$

$(p_{ij} = p_{ij} - l_{ik}l_{jk})$

$(l_{i,i} = p_{i,i}^{1/2})$

6) *ijk*-алгоритм
$$(l_{11} = p_{11}^{1/2})$$

Для $i = 2$ до n

 Для $j = 2$ до i

$l_{i,j-1} = p_{i,j-1}/p_{j-1,j-1}$

$(l_{i,j-1} = p_{i,j-1}/l_{j-1,j-1})$

 Для $k = 1$ до $j - 1$

$p_{ij} = p_{ij} - l_{ik}p_{jk}$

$(p_{ij} = p_{ij} - l_{ik}l_{jk})$

$(l_{i,i} = p_{i,i}^{1/2})$

ЗАМЕЧАНИЕ 6.6. Приведенные алгоритмы $\bar{L}D\bar{L}^T$ и LL^T -разложений Холецкого матрицы получены из соответствующих *ijk*-алгоритмов $\bar{L}U$ -разложения матрицы A (см. подразд. 3.5). Для получения $\bar{U}D\bar{U}^T$ и UU^T -разложений Холецкого матрицы удобно исходить из $\bar{U}L$ -разложения матрицы A , если для него предварительно построить *ijk*-алгоритмы. Это построение нетрудно выполнить, если учесть, что $\bar{U}L$ -разложение соответствует измененному (инверсному) порядку исключения переменных. В этом случае модификация системы уравнений начинается с последней переменной последнего уравнения.

Суммируя вышеизложенное по *ijk*-формам алгоритмов Холецкого, полученных из *ijk*-форм алгоритмов Гаусса, имеем 24 разновидности разложений симметричной положительно определенной матрицы P :

- 6 *ijk*-форм для $P = \bar{L}D\bar{L}^T$,
- 6 *ijk*-форм для $P = LL^T$,
- 6 *ijk*-форм для $P = \bar{U}D\bar{U}^T$,
- 6 *ijk*-форм для $P = UU^T$.

6.5 Разложение Холецкого: алгоритмы окаймления

Как и для LU -разложения, для разложения Холецкого в любых его вариантах (6.2) существует еще один класс алгоритмов, — так называемые матрично-векторные алгоритмы, объединяемые идеей окаймления. Получение этих алгоритмов базируется на блочном перемножении матриц, участвующих в разложении. Здесь полностью применимы принципы, изложенные в подразд. 6.2.

Покажем, как выводятся такие матрично-векторные алгоритмы на примере одного из четырех вариантов разложения Холецкого (6.2), а именно, варианта $P = LL^T$. Пользуясь этим справочным материалом, любой студент сможет самостоятельно построить родственные алгоритмы для других трех вариантов разложения. Для этого поделим все матрицы в данном варианте на блоки, выделяя в каждой матрице j -ю строку и j -й столбец. Тем самым разложение $P = LL^T$ будет представлено в блочной форме

$$j \Rightarrow \begin{pmatrix} P_{11} & \overset{j}{\downarrow} \mathbf{a} & P_{13} \\ \mathbf{a}^T & p_{jj} & \mathbf{b}^T \\ P_{31} & \mathbf{b} & P_{33} \end{pmatrix} = \begin{pmatrix} L_{11} & \overset{j}{\downarrow} & \\ \mathbf{c}^T & l_{jj} & \\ L_{31} & \mathbf{d} & L_{33} \end{pmatrix} \begin{pmatrix} L_{11}^T & \overset{j}{\downarrow} \mathbf{c} & L_{31}^T \\ & l_{jj} & \mathbf{d}^T \\ & & L_{33}^T \end{pmatrix}, \quad (6.3)$$

где фрагменты j -й строки и j -го столбца обозначены как векторы-столбцы выделенными символами \mathbf{a} , \mathbf{b} , \mathbf{c} и \mathbf{d} , а заглавные буквы обозначают матрицы. Нулевые элементы треугольных матриц не показаны.

Перемножение матриц (6.3), выполняемое поблочно, дает девять соотношений относительно блок-элементов матриц P и L . Пользуясь этим, рассмотрим два основных способа разложения матрицы P методом окаймления.

Окаймление известной части разложения. Из указанных девяти соотношений возьмем только те, что окаймляют блок $P_{11} = L_{11}L_{11}^T$, считая, что в этой части разложение уже сделано, т. е. что блок L_{11} уже вычислен. В силу симметрии P из трех окаймляющих произведений имеем только два:

$$\mathbf{a} = L_{11}\mathbf{c} \quad \text{и} \quad p_{jj} = \mathbf{c}^T\mathbf{c} + l_{jj}^2. \quad (6.4)$$

Отсюда сначала находим \mathbf{c} как решение нижнетреугольной системы уравнений $L_{11}\mathbf{c} = \mathbf{a}$; затем находим $l_{jj} = (p_{jj} - \mathbf{c}^T\mathbf{c})^{1/2}$.

Окаймление неизвестной части разложения. Из указанных соотношений возьмем те, что окаймляют блок P_{33} , считая, что до этого блока разложение уже сделано, т. е. что блоки L_{11} , L_{31} и \mathbf{c} уже найдены. В силу симметрии P из трех окаймляющих произведений имеем только два:

$$p_{jj} = \mathbf{c}^T\mathbf{c} + l_{jj}^2 \quad \text{и} \quad \mathbf{b} = L_{31}\mathbf{c} + \mathbf{d}l_{jj}. \quad (6.5)$$

Отсюда сначала находим $l_{jj} = (p_{jj} - \mathbf{c}^T\mathbf{c})^{1/2}$; затем $\mathbf{d} = (\mathbf{b} - L_{31}\mathbf{c})/l_{jj}$.

Существует два естественных способа реализации окаймления известной части в LL^T -разложении.

В первом варианте треугольная система в (6.4) решается с помощью строчного алгоритма (аналог алгоритма на рис. 4.1 слева), во втором — с помощью алгоритма скалярных произведений (аналог алгоритма на рис. 4.1 справа). Псевдокоды этих двух вариантов приведены на рис. 6.1.

$l_{11} = \sqrt{p_{11}}$ Для $j = 2$ до n Для $k = 1$ до $j - 1$ $l_{jk} = p_{jk}/l_{kk}$ Для $i = k + 1$ до j $p_{ji} = p_{ji} - l_{jk}l_{ik}$ $l_{jj} = \sqrt{p_{jj}}$	$l_{11} = \sqrt{p_{11}}$ Для $j = 2$ до n Для $i = 2$ до j $l_{j,i-1} = p_{j,i-1}/l_{i-1,i-1}$ Для $k = 1$ до $i - 1$ $p_{ji} = p_{ji} - l_{jk}l_{ik}$ $l_{jj} = \sqrt{p_{jj}}$
---	--

Рис. 6.1. Алгоритмы окаймления известной части LL^T -разложения: строчный (слева); алгоритм скалярных произведений (справа)

Для окаймления неизвестной части в LL^T -разложении также существуют два естественных способа реализации выражений (6.5). Здесь основной операцией является умножение вектора на прямоугольную матрицу.

Можно реализовать такие умножения посредством скалярных произведений или линейных комбинаций, что приводит к двум различным формам алгоритма, показанным на рис. 6.2, которые аналогичны алгоритмам Донгарры–Айзенштата на рис. 4.4.

Для $j = 1$ до n Для $k = 1$ до $j - 1$ $p_{jj} = p_{jj} - l_{jk}l_{jk}$ $l_{jj} = \sqrt{p_{jj}}$ Для $k = 1$ до $j - 1$ Для $i = j + 1$ до n $p_{ij} = p_{ij} - l_{ik}l_{jk}$ Для $s = j + 1$ до n $l_{sj} = p_{sj}/l_{jj}$	Для $j = 1$ до n Для $i = j + 1$ до n Для $k = 1$ до $j - 1$ $p_{ij} = p_{ij} - l_{ik}l_{jk}$ Для $k = 1$ до $j - 1$ $p_{jj} = p_{jj} - l_{jk}l_{jk}$ $l_{jj} = \sqrt{p_{jj}}$ Для $s = j + 1$ до n $l_{sj} = p_{sj}/l_{jj}$
--	--

Рис. 6.2. Алгоритмы окаймления неизвестной части LL^T -разложения: алгоритм линейных комбинаций (слева); алгоритм скалярных произведений (справа)

Таким образом, выше показано, что алгоритмы окаймления в LU -разложении (см. разд. 4) легко модифицируются для случая симметриче-

ской положительно определенной матрицы P . Тогда мы имеем 4 варианта разложения Холецкого (6.2), 2 способа окаймления и 2 схемы вычислений для каждого алгоритма окаймления. Всего получается 16 вариантов алгоритмов окаймления для разложения Холецкого симметрической положительно определенной матрицы. Добавляя к ним 24 разновидности ijk -форм, получаем 40 различных вычислительных схем разложений Холецкого, которые и составляют весь набор вариантов (см. подразд. 6.8) задания на лабораторный проект № 5 (см. подразд. 6.7).

6.6 Особенности хранения ПО-матрицы P

Как уже отмечалось (см. стр. 96), особенностью данного проекта является использование *линейных (одномерных) массивов для хранения матрицы P* . Так как матрица P симметрическая, то достаточно хранить только нижнюю (или верхнюю) треугольную часть этой матрицы вместе с диагональю. Причем для хранения заполненной матрицы используется один одномерный массив, а для хранения разреженной — два.

Хранение матрицы P может быть организовано по столбцам или по строкам в зависимости от используемого алгоритма разложения.

Рассмотрим строчный вариант хранения нижней треугольной части заполненной матрицы P . В этом случае все элементы нижней треугольной матрицы записываются построчно в одномерный массив. Так как для хранения первой строки матрицы требуется один элемент массива, для хранения второй строки — два элемента и т. д., то для хранения симметрической матрицы размера n требуется одномерный массив размера $n(n+1)/2$. Положение (i, j) -го элемента матрицы P в массиве определяется по формуле

$$k = (i-1)i/2 + j.$$

Аналогичным образом организуется хранение матрицы P по столбцам.

Как уже отмечалось, для хранения разреженной матрицы P используются два одномерных массива. Так, хранение нижней треугольной части матрицы P по строкам можно организовать следующим образом. В массиве ***a*** хранятся построчно элементы матрицы от первого ненулевого до диагонального включительно. В массиве ***b*** на i -м месте стоит положение i -го диагонального элемента матрицы P в массиве ***a***. Для определения положения (i, j) -го элемента матрицы P в массиве ***a*** надо воспользоваться следующим алгоритмом. Сначала вычисляем $k = b(i) - (i - j)$. Затем, если $k > b(i - 1)$,

то этот элемент стоит на k -м месте. В противном случае (i, j) -й элемент стоит левее первого ненулевого элемента i -й строки, поэтому он равен нулю и в массиве \mathbf{a} не хранится. Способ хранения по столбцам строится аналогичным образом, но в этом случае хранятся все элементы от диагонального до последнего ненулевого элемента столбца включительно.

Таким образом, существуют 4 варианта хранения разреженной ленточной матрицы P , и выбор конкретного варианта должен соответствовать заданному варианту разложения Холесского и разновидности ijk -форм.

ЗАМЕЧАНИЕ 6.7. С учетом положительной определенности матриц этой лабораторной работы, процедура выбора главного элемента, а также процедуры перестановки строк и столбцов матрицы P отсутствуют как для заполненных, так и для разреженных матриц.

6.7 Задание на лабораторный проект № 5

Написать и отладить программу, реализующую заданный вариант алгоритма разложения Холесского, для решения системы $Px = f$, где P — симметричная положительно определенная матрица (ПО-матрица P , или кратко, матрица $P > 0$). Отделить основные части программы:

- а) подпрограмму генерации ПО-матрицы P ;
- б) подпрограмму разложения Холесского;
- в) подпрограмму решения систем линейных алгебраических уравнений;
- г) подпрограмму решения систем линейных алгебраических уравнений с ленточной матрицей P ;
- д) сервисные подпрограммы, включая демонстрацию разложения на экране, подпрограмму контроля правильности разложения и др.

Уделить особое внимание эффективности программы (в смысле экономии оперативной памяти). Для этого в одномерном массиве достаточно хранить только нижнюю (или верхнюю) треугольную часть и диагональ матрицы P . Результат разложения замещает исходную матрицу. Предусмотреть пошаговое выполнение алгоритма исключения с выводом результата на экран.

Выполнить следующие пункты задания:

1. Дать формулировку и доказательство теоремы о заданном варианте разложения Холесского. Теорема может быть сформулирована как утверждение о том, что предлагаемый студентом алгоритм действительно дает

единственное разложение Холецкого. Для иллюстрации дать численный пример работы алгоритма по шагам для матрицы P размера (4×4) .

2. Провести подсчет количества операций:

- а) извлечения квадратного корня;
- б) умножения и деления.

Подсчет выполнить тремя способами:

- а) фактически — при конкретном расчете разложения;
- б) теоретически точно в зависимости от размерности матрицы n ;
- в) теоретически приближенно в зависимости от n при $n \rightarrow \infty$.

3. Определить скорость решения задач (решение систем линейных алгебраических уравнений), для чего спроектировать и провести эксперимент, который охватывает сгенерированные случайным образом ПО-матрицы P порядка от 5 до 100 (через 5 порядков). Результаты представить в виде таблицы и графика зависимости времени выполнения от порядка матриц. Сравнить со своим вариантом из лабораторного проекта № 1.

4. Оценить точность решения систем линейных алгебраических уравнений с матрицами из п. 3. Для этого выбрать точное решение x^* и образовать правые части $f = Px^*$. В качестве точного решения взять вектор $x^* = (1, 2, \dots, n)$. Для оценки точности использовать норму вектора (2.18) из лабораторного проекта № 1. Провести анализ точности вычисленного решения x в зависимости от порядка матрицы. Результаты представить в виде таблицы и графика. Сравнить со своим вариантом из проекта № 1.

5. Для заполнения матрицы P использовать случайные числа из диапазона от -100 до 100 . Сначала заполнить треугольную часть матрицы P , т. е. элементы p_{ij} , где $i > j$. Затем заполнить диагональ. В качестве диагонального элемента p_{ii} , $i = 1, 2, \dots, n$, выбрать случайное число из интервала

$$\left[\sum_{j \neq i} |p_{ij}| + 1, \sum_{j \neq i} |p_{ij}| + 101 \right], \quad (6.6)$$

чтобы обеспечить выполнение условия

$$p_{ii} \geq \sum_{j \neq i} |p_{ij}| + 1,$$

гарантирующего положительную определенность матрицы P .

6. Определить скорость и точность решения систем линейных алгебраических уравнений с разреженными ленточными матрицами. Для этого спроектировать и провести эксперимент для систем порядка от 100 до 200 (через 5). Результаты представить в виде таблиц и графиков зависимости скорости и точности решения от порядка матриц. Для этих же систем найти аналогичные зависимости для обычного метода Холесского. Результаты сравнить.

7. Для случайного заполнения разреженной ленточной матрицы P использовать следующий алгоритм:

а) случайным образом заполнить половину матрицы (верхнюю или нижнюю), включая диагональ;

б) в случае заполнения нижней треугольной части матрицы P в i -й строке, $i = 1, 2, \dots, n$, случайным образом определить количество ненулевых элементов (от 1 до 10), их местоположение (номер столбца от $\max\{1, i - 50\}$ до $i - 1$) и значение (ненулевые целые числа, лежащие в интервале от -100 до 100);

в) при заполнении верхней треугольной части матрицы P применять тот же алгоритм, что и в п. б), с той лишь разницей, что номер столбца лежит в интервале от $i + 1$ до $\min\{i + 50, n\}$;

г) диагональный элемент в i -й строке, $i = 1, 2, \dots, n$, определить случайным образом на интервале (6.6).

В качестве точного решения взять вектор $x^* = (1, 2, \dots, n)$, n — порядок матрицы P . В случае, если при решении системы $Px = f$ выяснится, что матрица P вырождена (плохо обусловлена), то сгенерировать новую матрицу того же порядка и решить систему линейных алгебраических уравнений с новой матрицей P и новой правой частью. Для оценки точности решения использовать норму вектора (2.18) из лабораторного проекта № 1.

ЗАМЕЧАНИЕ 6.8. По ходу проведения всех численных экспериментов на экран дисплея должны выводиться следующие таблицы.

Число вычислительных операций:

Порядок матрицы	Квадратные корни			Умножение и деление		
	а	б	с	а	б	в

где а, б, в означает способ вычисления числа действий (см. п. 2).

**Решение систем линейных алгебраических уравнений
с заполненной матрицей P :**

Порядок матрицы	Время		Точность	
	метод Гаусса	метод Холесского	метод Гаусса	метод Холесского

Таким образом, в данный проект следует включить работу, выполненную ранее в проекте № 1, чтобы иметь возможность сравнения метода Холесского с методом Гаусса как по времени, так и по точности вычислений.

**Решение систем линейных алгебраических уравнений
с разреженной матрицей P :**

Порядок матрицы	Время		Точность	
	Заполненная матрица	Разреженная матрица	Заполненная матрица	Разреженная матрица

Это означает, что для каждого текущего значения n порядка матрицы P необходимо решать две системы: одну — с заполненной матрицей P (см. п. 5 задания), другую — с разреженной матрицей P (см. п. 7 задания).

ЗАМЕЧАНИЕ 6.9. Необходимо вывести на экран следующие графики:

Графики решения систем с заполненной матрицей P :

- зависимость времени решения от порядка матриц для методов Гаусса и Холесского;
- зависимость точности решения от порядка матриц для методов Гаусса и Холесского.

Графики решения систем с разреженной матрицей P :

- зависимость времени решения от порядка матриц для обычного метода Холесского и с учетом разреженности матрицы P ;
- зависимость точности решения от порядка матриц для обычного метода Холесского и с учетом разреженности матрицы P . При построении графиков использовать данные из соответствующей таблицы. Для этого их необходимо записать в текстовый файл.

6.8 Варианты задания на лабораторный проект № 5

Как отмечалось в конце подразд. 6.5, всего по данной теме предлагается 40 различных вычислительных схем разложений Холесского, которые и составляют весь набор вариантов задания на лабораторный проект № 5 (см. подразд. 6.7)

В табл. 6.1 каждому номеру варианта соответствует своя разновидность разложения Холесского и свой способ организации вычислений.

Таблица 6.1. Варианты задания на лабораторный проект № 5

Вид разложения	<i>ijk</i> -формы						Окаймление			
	<i>kij</i>	<i>kji</i>	<i>jki</i>	<i>jik</i>	<i>ikj</i>	<i>ijk</i>	известной части		неизвестной части	
							a	b	c	b
$P = \bar{L}D\bar{L}^T$	1	2	3	4	5	6	7	8	9	10
$P = LL^T$	11	12	13	14	15	16	17	18	19	20
$P = \bar{U}D\bar{U}^T$	21	22	23	24	25	26	27	28	29	30
$P = UU^T$	31	32	33	34	35	36	37	38	39	40

^a — строчный алгоритм;

^b — алгоритм скалярных произведений;

^c — алгоритм линейных комбинаций.

Если нет других указаний преподавателя, выбирайте ваш вариант по вашему номеру в журнале студенческой группы.

ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ

7.1 Ортогональные матрицы и приложения

В этом разделе напомним определение и некоторые свойства ортогональных матриц, полезные для дальнейшего.

ОПРЕДЕЛЕНИЕ 7.1. Матрица T , имеющая размер $(n \times n)$, т. е. $T(n, n)$, есть *ортогональная матрица*, когда $TT^T = I$.

Свойство А. Если T_1 и T_2 суть две ортогональные матрицы, то их произведение T_1T_2 есть тоже ортогональная матрица.

Свойство В. $T^{-1} = T^T$ и $T^TT = I$.

Свойство С. Ортогональное преобразование сохраняет скалярное произведение векторов, т. е. $\forall x, y \in \mathbb{R}^n : y^Tx \triangleq (x, y) = (Tx, Ty)$, в частности, оно сохраняет (евклидову) норму вектора: $\|Ty\| = \|y\|$.

Свойство D. Если v есть вектор случайных переменных с математическим ожиданием $\mathbf{E}\{v\} = 0$ и ковариацией $\mathbf{E}\{vv^T\} = I$, то теми же характеристиками обладает вектор $\bar{v} = Tv$, т. е.

$$\mathbf{E}\{\bar{v}\} = 0, \quad \mathbf{E}\{\bar{v}\bar{v}^T\} = I.$$

Хотя это свойство легко проверяется, немного удивительно, что компоненты преобразованного вектора остаются взаимно некоррелированными.

Свойства С и D играют существенную роль в квадратно-корневых алгоритмах решения прикладных задач оптимального моделирования и оптимального оценивания методом наименьших квадратов (рис. 7.1).

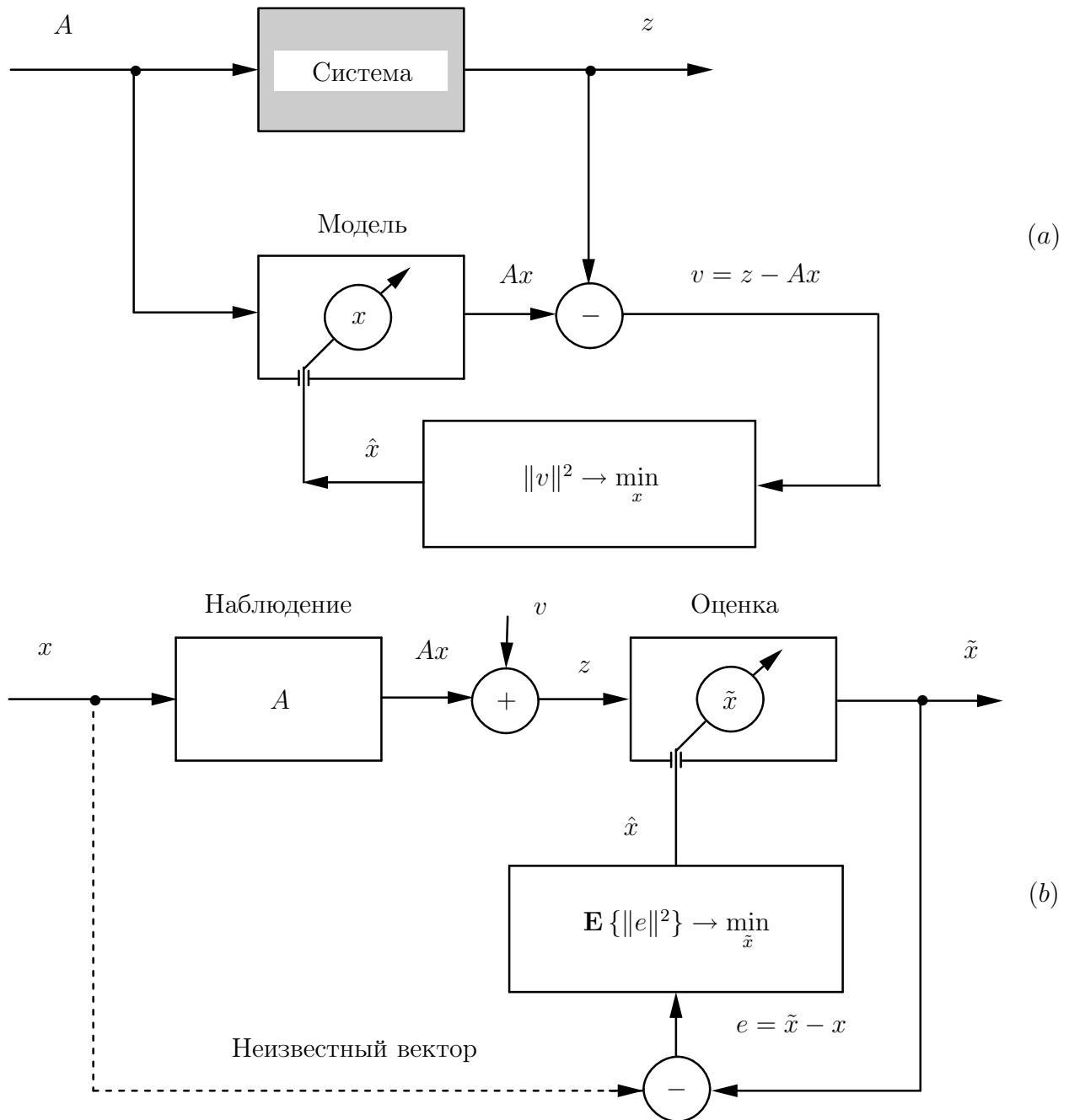


Рис. 7.1. Алгебраически эквивалентные задачи, решаемые методом наименьших квадратов значений невязки v или среднего квадрата погрешности e : (a) — оптимальное моделирование неизвестной системы по экспериментальным условиям A и данным z ; (b) — оптимальное оценивание неизвестного вектора по наблюдениям Ax в присутствии случайных помех v с характеристиками $\mathbf{E}\{v\} = 0$ и $\mathbf{E}\{vv^T\} = I$

7.2 Линейная задача наименьших квадратов

Линейная задача наименьших квадратов (см. рис. 7.1) ставится следующим образом (см. также подразд. 11.2) [13, 15].

Дано линейное уравнение

$$z = Ax + v, \quad (7.1)$$

в котором известны вектор $z \in \mathbb{R}^m$ и $(m \times n)$ -матрица $A \in \mathbb{R}^{m \times n}$, т. е. $A = A(m, n)$. Разностный вектор $v \triangleq z - Ax$, называемый *невязкой*, зависит от переменного вектора $x \in \mathbb{R}^n$. Требуется найти значение \hat{x} вектора x , минимизирующее *квадратический критерий качества*

$$J(x) = (z - Ax)^T(z - Ax) = \|v\|^2 \rightarrow \min. \quad (7.2)$$

Если ни при каком x невязка v не может быть обращена в $\mathbf{0}$ — нулевой вектор, то система $Ax = z$ — несовместная, в противном случае совместная, т. е. критерий (7.2) охватывает оба случая. Однако сам *метод наименьших квадратов* (МНК), выраженный критерием (7.2), создан *Лежандром* в 1805 году как алгебраическая процедура именно для несовместных систем и подтвержден как статистическая процедура *Гауссом* в 1809 году. МНК как алгебраическая процедура проиллюстрирован выше с помощью рис. 7.1(a), а как статистическая процедура — с помощью рис. 7.1(b). Замечательно, что обе процедуры имеют одинаковые решения, т. е. алгебраически эти решения эквивалентны и при $\mathbf{E}\{v\} = 0$ и $\mathbf{E}\{vv^T\} = I$ (см. рис. 7.1(b)) совпадают, поэтому можно говорить о едином МНК-решении \hat{x} .

МНК-решение \hat{x} всегда существует как решение *нормальных уравнений*

$$A^T A \hat{x} = A^T z, \quad (7.3)$$

выражается формулой

$$\hat{x} = A^+ z + (I - A^+ A) y \quad (7.4)$$

через произвольный вектор $y \in \mathbb{R}^n$, где A^+ — *псевдообратная матрица* для матрицы A , и единственно тогда и только тогда, когда $A^+ A = I$, что равносильно условию, что только нулевой вектор составляет ядро (нуль-пространство) матрицы A , т. е. при $\text{rank } A = n$.

Условие $\text{rank } A = n$, называемое *условием полного столбцового ранга* матрицы A , обуславливает случай $m \geq n$, что при $m > n$ означает переопределенную систему полного ранга в (7.1). Этот типичный для практики

случай ниже и рассматривается, при этом из (7.3), (7.4) следует $\hat{x} = A^+z$ и $A^+ = (A^T A)^{-1} A^T$.

ЗАМЕЧАНИЕ 7.1. Слагаемое $\hat{x}_0 \triangleq A^+z$ в (7.4) есть единственное МНК-решение с минимальной нормой, называемое *нормальным псевдорешением*. Оно ортогонально второму слагаемому в (7.4), т. е. $A^+z \perp (I - A^+A)y$, и лежит в пространстве строк матрицы A , т. е. $\hat{x}_0 \in \mathcal{R}(A^T)$ (подробнее см. подразд. 10.4).

Таким образом, типичный для практики случай имеет формальное решение $\hat{x} = \hat{x}_0 = (A^T A)^{-1} A^T z$, и вычислительная задача наименьших квадратов заключается в его эффективном отыскании. Об этом подробнее см. разд. 11.

7.3 Ортогональные матрицы и наименьшие квадраты

В рассматриваемой задаче о наименьших квадратах имеем критерий

$$J(x) = \|z - Ax\|^2, \quad A(m, n), \quad m \geq n, \quad \text{rank } A = n. \quad (7.5)$$

Пусть $T, T(m, m)$, есть матрица некоторого ортогонального преобразования. В силу свойства С (см. подразд. 7.1) запишем

$$J(x) = \|T(z - Ax)\|^2 = \|(Tz) - (TA)x\|^2. \quad (7.6)$$

При таком представлении видно, что минимум критерия $J(x)$, равный $J(\hat{x})$, не зависит от T . Этим фактом можно воспользоваться, т. е. матрицу T можно выбрать так, что (TA) приобретает привлекательную для вычислений форму. Действительно, в подразд. 7.4 и 7.7 мы покажем, как можно выбрать T , чтобы преобразованная матрица имела вид

$$TA = \left[\begin{array}{c} R \\ \mathbf{0} \end{array} \right] \begin{array}{l} \} n \\ \} m - n \end{array} \quad (7.7)$$

с верхнетреугольным блоком R , $\text{rank } R = n$.

Если соответственно этому вектор Tz разбить на блоки, т. е. записать

$$Tz = \left[\begin{array}{c} z_1 \\ z_2 \end{array} \right] \begin{array}{l} \} n \\ \} m - n \end{array}, \quad (7.8)$$

то $J(x)$ от (7.6) приводится к виду

$$J(x) = \|z_1 - Rx\|^2 + \|z_2\|^2. \quad (7.9)$$

Приведение критерия наименьших квадратов к виду (7.9) позволяет заключить, что искомый вектор \hat{x} , минимизирующий этот критерий, должен удовлетворять уравнению

$$R\hat{x} = z_1, \quad (7.10)$$

которое легко решается *обратной подстановкой* (см. подразд. 7.6), и кроме того,

$$\min J(x) = J(\hat{x}) = \|z_2\|^2. \quad (7.11)$$

В вычислительном отношении эти результаты гораздо более элегантны, чем неразумная трата сил на решение нормальных уравнений (7.3). Однако важнее всего то, что решение, использующее ортогональные преобразования (соотношения (7.7), (7.8), (7.10) и (7.11)), менее чувствительно к погрешностям, вызванным ошибками округления в компьютере. Это видно хотя бы из того, что выражение (7.7) влечет равенство

$$R^T R = (TA)^T (TA) = A^T A,$$

которое означает, что R является квадратным корнем из матрицы $(A^T A)$ системы нормальных уравнений (7.3). Следовательно, при решении системы (7.10) вдвое более эффективно используется разрядная сетка компьютера, чем при решении системы (7.3)¹.

7.4 Преобразование Хаусхолдера

Преобразования Хаусхолдера суть матричные представления, которые соответствуют геометрическому понятию отражения [97, 15]. Пусть задан некоторый ненулевой вектор u , который мы называем *направляющим вектором*. Подпространство, ортогональное ему, есть гиперплоскость U_\perp . Если взять произвольный вектор y , то можно отразить его от U_\perp , в точности соблюдая законы обычного оптического отражения от плоского зеркала (рис. 7.2).

Обозначим отраженный вектор y_r . Поскольку положение гиперплоскости U_\perp не зависит от длины направляющего вектора, пронормируем его, т. е. образуем орт $\hat{u} = u/\|u\|$. *Проекция* $(y \mid u)$ вектора y на прямую, задаваемую направлением u , равна $(y^T \hat{u})\hat{u}$. Следовательно,

$$y = (y \mid u) + v, \quad v \perp u, \quad v \in U_\perp. \quad (7.12)$$

¹ Представление в компьютере квадрата a^2 любого действительного числа a требует удвоенной разрядности мантииссы, т. е. счет по уравнению (7.10) равносильен счету с удвоенной разрядностью мантииссы чисел по уравнению (7.3).

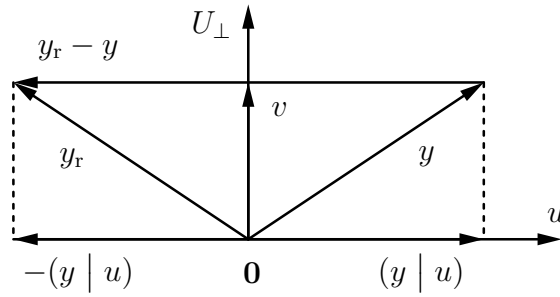


Рис. 7.2. Геометрия преобразования Хаусхолдера. Задача 1 (прямая): даны векторы u и y , найти вектор y_r , отраженный от гиперплоскости U_\perp

Отраженный вектор y_r , как видно из рис. 7.2, имеет разложение

$$y_r = -(y | u) + v, \quad v \perp u, \quad v \in U_\perp \quad (7.13)$$

с той же составляющей v , которая ортогональна вектору u , но с проекцией $-(y | u)$, которая (в силу знака $-$) направлена противоположно проекции $(y | \hat{u})$ вектора y на направление u . Исключая v из (7.12) и (7.13), находим

$$y_r = y - 2(y | u) = (I - \beta u u^T) y = T_u y, \quad (7.14)$$

где $\beta \triangleq 2/\|u\|^2 = 2/u^T u$. Матрица Хаусхолдера $T_u \triangleq (I - \beta u u^T)$, в вычислениях явно не участвующая, имеет фундаментальное значение для приложений в силу своих замечательных свойств.

Свойство 1. $T_u = T_u^T$, т. е. T_u — симметрическая матрица.

Свойство 2. $T_u^2 = I$, т. е. T_u — идемпотентная матрица. Это легко продемонстрировать алгебраически разложением матрицы T_u^2 или же геометрически по рис. 7.2 как двукратное отражение вектора y относительно U_\perp .

Свойство 3. Если $u(j) = 0$, то $(T_u y)(j) = y(j)$, т. е. если j -я компонента вектора u — нулевая, то T_u оставляет j -ю компоненту вектора y неизменной.

Свойство 4. Если $u \perp y$, то $T_u y = y$.

Свойство 5.

$$T_u y = y - \gamma u, \quad \gamma \triangleq 2y^T u / u^T u = \beta y^T u. \quad (7.15)$$

Свойство 5 — важное с практической точки зрения. Формирование матрицы T_u в качестве множителя для y потребовало бы на порядок больше вычислений, чем того требует прямое вычисление $T_u y$ по свойству 5. Это также означает, что не нужно тратить память для хранения T_u , что наиболее существенно проявляется при больших m .

Триангуляризация матрицы методом Хаусхолдера. Обратимся к основному применению ортогональных преобразований. Для этого решим задачу, обратную к той, что рассмотрена выше, а именно: дан вектор y и дано желаемое расположение отраженного вектора y_r , — найти направление u такое, что $T_u y = (s, 0, \dots, 0)^T$ (рис. 7.3). Из свойства С, подразд. 7.1, норма (евклидова длина) вектора y не изменяется при ортогональном преобразовании, следовательно, определим ее как

$$\sigma \triangleq \|T_u y\| = |s| = (y^T y)^{1/2}. \quad (7.16)$$

Направление u может быть получено из свойства 5 (уравнение (7.15)), т. е.

$$u = \text{const} \cdot (y - se_1). \quad (7.17)$$

Этот результат приводит к следующему свойству.

Свойство 6. Пусть $s = -\text{sgn} [y(1)]\sigma$, где $\text{sgn} [\cdot]$ — функция знака,

$$\text{sgn} [x] = \begin{cases} 1, & x \geq 0, \\ -1, & x < 0, \end{cases}$$

и элементы вектора u определены выражением (7.17), т. е. $u(1) = y(1) - s$, $u(i) = y(i)$, $i > 1$. Тогда $T_u y = se_1$ и $\beta \triangleq 2/u^T u = -1/(su(1))$.

ЗАМЕЧАНИЕ 7.2. Геометрический смысл выражения (7.17) ясен из рис. 7.3. Вектор $y - se_1$ ортогонален гиперплоскости U_\perp и коллинеарен вектору u , а именно: $y - se_1 = \gamma u$. Более того, $\gamma = 1$. Выведите это из (7.15).

Непосредственное вычисление $u^T u$ показывает, что $u^T u = -2su(1)$, при этом знак для s выбран противоположно знаку первого элемента $y(1)$, т. е. так, чтобы максимизировать $|u(1)|$ и тем уменьшить относительную погрешность вычисления разности $u(1) = y(1) - s$. Если свойство 6 применить к матрице A , взяв в качестве y ее первый столбец, то это даст первый шаг, который послужит основой приведения матрицы к верхнетреугольному виду. Повторение таких действий шаг за шагом позволит осуществлять верхнюю триангуляризацию любой заданной матрицы A .

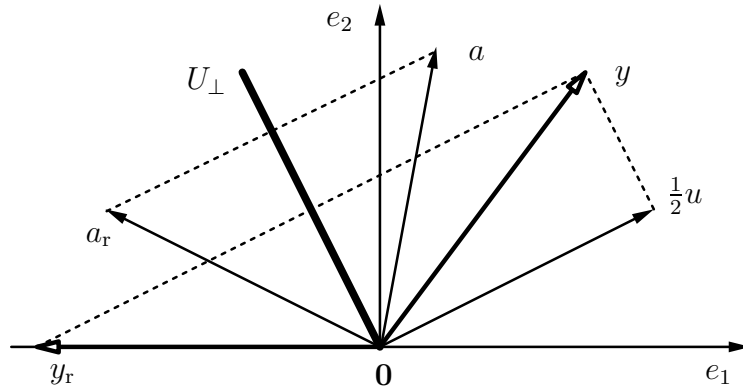


Рис. 7.3. Геометрия преобразования Хаусхолдера. Задача 2 (обратная): даны векторы y и y_r , найти вектор u , задающий отражающую гиперплоскость U_\perp ; здесь $y_r = se_1 = [s \mid 0 \cdots 0]^T$. Докажите, что здесь показан вектор $\frac{1}{2}u$, а не u (см. Замечание 7.2)

ЛЕММА 7.1. Пусть дана матрица $A(m, n)$. Тогда существует ортогональное преобразование Хаусхолдера T_u такое, что

$$T_u A = \begin{matrix} & \overbrace{\quad}^1 & \overbrace{\quad}^{n-1} \\ \begin{matrix} 1\{ \\ m-1\{ \end{matrix} & \left[\begin{array}{c|c} s & \tilde{A} \\ \mathbf{0} & \end{array} \right] \end{matrix}. \quad (7.18)$$

ЗАМЕЧАНИЕ 7.3. Скаляр s и матрица \tilde{A} в (7.18) вычисляются непосредственно по данным в матрице A ; s — по выражению (7.16) и свойству 6, а \tilde{A} — по свойству 5, (7.15). Первый столбец, который уместно назвать *ведущим столбцом в преобразовании Хаусхолдера*, используют как вектор y в задаче 2 (см. рис. 7.3) для определения вектора u . Второй и далее столбцы, обозначенные на рис. 7.3 произвольно как вектор a , отражают от найденной таким образом гиперплоскости U_\perp , решая для этого задачу 1 (см. рис. 7.2) с $y := a$ и тем самым получая блок \tilde{A} .

ТЕОРЕМА 7.1 (*Триангуляризация матрицы по методу Хаусхолдера*). Пусть $A_1 := A(m, n)$ и для каждого j выбрано элементарное преобразование Хаусхолдера T_j так, что

$$T_j A_j = \begin{matrix} & \overbrace{\quad}^1 & \overbrace{\quad}^{n-j} \\ \begin{matrix} 1\{ \\ m-j\{ \end{matrix} & \left[\begin{array}{c|c} s_j & a_j^T \\ \mathbf{0} & A_{j+1} \end{array} \right] \end{matrix}, \quad j = 1, \dots, k; \quad k \leq \min(m-1, n). \quad (7.19)$$

Тогда в процессе после k повторных применений свойства 6 и леммы 7.1 имеем следующий промежуточный результат триангуляризации матрицы A :

$$T^{(k)}A = \left[\begin{array}{ccc|c} s_1 & & & a_1^T \\ & s_2 & & a_2^T \\ & & \ddots & \vdots \\ & & & s_k & a_k^T \\ \mathbf{0} & & & & A_{k+1} \end{array} \right] \quad (7.20)$$

с отвечающей этому моменту процесса итоговой матрицей преобразований

$$T^{(k)} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & T_k \end{bmatrix} \cdots \begin{bmatrix} I_1 & 0 \\ 0 & T_2 \end{bmatrix} T_1. \quad (7.21)$$

ЗАМЕЧАНИЕ 7.4. Важно подчеркнуть, что алгоритм триангуляризации (7.19) не требует вычисления или запоминания ортогональной матрицы $T^{(k)}$, так как правая часть равенства (7.4) вычисляется непосредственно в соответствии с замечанием 7.3. Стоит также заметить, как неявно определяется A_{j+1} рекурсией по j в алгоритме (7.19). Кроме A_{j+1} , на шаге j этой рекурсии определяются скаляр s_j и $(n-j)$ компонент вектор-строки a_j^T . Эти неявные соотношения для s_j , a_j^T и A_{j+1} и весь процесс вычислений (рис. 7.4) представлены в явном виде в подразд. 7.5.

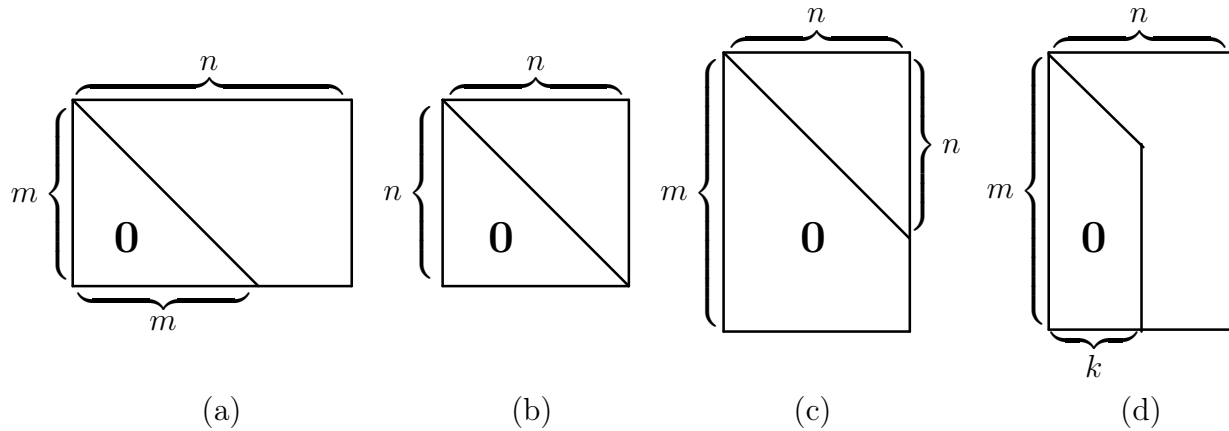


Рис. 7.4. Представление возможных случаев применения теоремы 7.1 к матрице $A(m, n)$; (a) недоопределенная система: $k = m - 1 \leq n$; (b) определенная система: $k = n - 1, m = n$; (c) переопределенная система: $k = n < m$; (d) $k < n < m$

7.5 Шаг триангуляризации матрицы методом Хаусхолдера

Пусть матрица $A = A(m, n)$ задана. Тогда, согласно лемме 7.1, базовая операция процесса триангуляризации заключается в вычислении скаляра s и матрицы $\tilde{A} = \tilde{A}(m, n-1)$ таких, что

$$T_u A = \begin{matrix} 1\{ \\ m-1\{ \end{matrix} \left[\begin{array}{c|c} \overbrace{s}^1 & \overbrace{\tilde{A}}^{n-1} \\ \mathbf{0} & \end{array} \right]. \quad (7.22)$$

Алгоритм. Для вычисления s следует применять свойство 6 (см. стр. 113), т. е. выполнять (7.23) для всех $k = 1$ до $\min(m-1, n)$. Затем для вычисления \tilde{A} следует применять свойство 5 (см. стр. 112), т. е. последовательно в цикле по $j = 2, \dots, n$ для всех $i = k, \dots, m$ выполнять (7.24). Здесь $\lambda \triangleq -\gamma$ (см. (7.15)), $\alpha \triangleq -\beta$ (см. (7.14) и использовано свойство 6).

$$\left. \begin{array}{l} \text{Для } k = 1 \text{ до } \min(m-1, n) \\ s_k = -\operatorname{sgn}[A(k, k)] \left(\sum_{i=k}^m [A(i, k)]^2 \right)^{1/2}, \\ u_k(1) = A(k, k) - s_k, \\ u_k(i) = A(k+i-1, k), \quad i = 2, \dots, m-k+1, \\ \alpha_k = 1/(s_k u_k(1)) \quad (\alpha_k < 0). \end{array} \right\} \quad (7.23)$$

$$\left. \begin{array}{l} \text{Для } j = k+1, \dots, n \\ \lambda := \alpha_k \cdot \sum_{i=k}^m u_k(i-k+1) A(i, j), \\ \text{Для } i = k, k+1, \dots, m \\ A(i, j) := A(i, j) + \lambda u_k(i-k+1). \end{array} \right\} \quad (7.24)$$

Приведенный алгоритм (7.23), (7.24) называют *столбцово ориентированным алгоритмом Хаусхолдера*, так как операции (7.24) вычисляют целиком каждый j -й столбец матрицы, находящийся справа от ведущего, т. е. k -го столбца. Альтернативная схема вычислений называется *строчно ориентированным алгоритмом Хаусхолдера*. Ее можно получить из выражения $T_u = I - \beta u u^T$ для матрицы Хаусхолдера следующим образом.

Введем вспомогательные обозначения: $\mu \triangleq \sqrt{\beta}$, $w \triangleq \mu u$, чтобы записать $T_u = I - ww^T$. Тогда $(T_u A) = A - wz^T$, где $z^T \triangleq w^T A = \mu v^T$, $v^T \triangleq \sum_{i=1}^m u(i)A(i, \cdot)$ и $A(i, \cdot)$ есть i -я строка матрицы $A = A(m, n)$. Введем обозначение $\lambda^T = \alpha v^T$, используя ранее введенное (см. (7.23)) $\alpha \triangleq -\beta$. Отсюда получаем формулу для любой i -й строки $(T_u A)(i, \cdot)$ преобразованной матрицы $(T_u A)$ в виде

$$(T_u A)(i, \cdot) = A(i, \cdot) - w(i)z^T = A(i, \cdot) - \mu^2 u(i)v^T = A(i, \cdot) + \lambda^T u(i).$$

Алгоритм (строчно ориентированный), эквивалентный (7.23) и (7.24).

$$\left. \begin{array}{l} \text{Для } k = 1 \text{ до } \min(m-1, n) \\ \quad s_k = -\operatorname{sgn}[A(k, k)] \left(\sum_{i=k}^m [A(i, k)]^2 \right)^{1/2}, \\ \quad u_k(1) = A(k, k) - s_k, \\ \quad u_k(i) = A(k+i-1, k), \quad i = 2, \dots, m-k+1, \\ \quad \alpha_k = 1/(s_k u_k(1)) \quad (\alpha_k < 0). \end{array} \right\} \quad (7.25)$$

$$\left. \begin{array}{l} \text{Для } j = k+1, \dots, n \\ \quad \lambda_k(j-k) := \alpha_k \cdot \sum_{i=k}^m u_k(i-k+1)A(i, j), \\ \text{Для } i = k, k+1, \dots, m \\ \quad \text{Для } j = k+1, \dots, n \\ \quad \quad A(i, j) := A(i, j) + \lambda_k(j-k)u_k(i-k+1). \end{array} \right\} \quad (7.26)$$

7.6 Решение треугольной системы и обращение матриц

Как отмечено в подразд. 7.3, мы часто заинтересованы в решении уравнения

$$Rx = z, \quad (7.27)$$

где $R = R(n, n)$ — верхняя треугольная невырожденная матрица. Если нужно иметь только решение x , то R^{-1} (для $x = R^{-1}z$) вычислять не надо. Следующий алгоритм обратной подстановки позволяет вычислить решение x непосредственно.

Алгоритм. Для $j = n, n-1, \dots, 1$ вычислять

$$x(j) = \left(z(j) - \sum_{k=j+1}^n R(j, k)x(k) \right) / R(j, j). \quad (7.28)$$

По сложности этот алгоритм почти такой же, как матричное умножение. Он допускает записывать $x(j)$ поверх $z(j)$, что очень удобно в приложениях.

Если все же требуется иметь матрицу $U \triangleq R^{-1}$, то ее можно вычислять по алгоритму окаймления, основанному на следующем легко проверяемом тождестве для треугольных матриц:

$$\begin{bmatrix} R_j & y \\ 0 & \sigma_{j+1} \end{bmatrix}^{-1} = \begin{bmatrix} R_j^{-1} & -R_j^{-1}y\sigma_{j+1}^{-1} \\ 0 & \sigma_{j+1}^{-1} \end{bmatrix} = R_{j+1}^{-1}. \quad (7.29)$$

Это соотношение позволяет вычислять матрицу $U \triangleq R^{-1}$ рекуррентно, а именно: если $R_j^{-1} = U_j$, где R_j обозначает верхнюю левую часть матрицы R , то

$$U_{j+1} = \begin{bmatrix} U_j & -U_j [R(1, j+1), \dots, R(j, j+1)]^T \sigma_{j+1} \\ 0 & \sigma_{j+1} \end{bmatrix}, \quad (7.30)$$

где $\sigma_{j+1} = 1/R(j+1, j+1)$. Полагая $U = R_{-1}^{-1}$, этот результат представим в алгоритмической форме.

Алгоритм. Обращение верхней треугольной матрицы. Задать начальное значение

$$U(1, 1) = 1/R(1, 1). \quad (7.31)$$

Для $j = 2, \dots, n$ вычислять по формулам (7.32) и (7.33):

$$U(j, j) = 1/R(j, j), \quad (7.32)$$

$$U(k, j) = - \left(\sum_{i=k}^{j-1} U(k, i)R(i, j) \right) U(j, j), \quad k = 1, \dots, j-1. \quad (7.33)$$

ЗАМЕЧАНИЕ 7.5. R^{-1} вычисляется по столбцам, при этом элементы матрицы R^{-1} могут записываться поверх элементов исходной матрицы R .

В справочных целях приведем примеры реализации данного алгоритма на языке FORTRAN. Эти примеры могут помочь студентам написать свои собственные программы на других языках высокого уровня при выполнении лабораторного проекта № 6, описание которого дано ниже в подразд. 7.16.

Обращение верхней треугольной матрицы: $U := R^{-1}$. Реализуются формулы (7.31), (7.32) и (7.33). Если нужно, U может замещать R [15].

$R(N, N), \quad U(N, N), \quad R$ и U — верхние треугольные матрицы

```

      U(1, 1) = 1./R(1, 1)
      DO 20 J = 2, N
        U(J, J) = 1./R(J, J)
        JM1 = J - 1
        DO 20 K = 1, JM1
          SUM = 0.
          DO 10 I = K, JM1
            SUM = SUM - U(K, I) * R(I, J)
          10
        20 U(K, J) = SUM * U(J, J)
      
```

В случаях, когда важно или нужно экономить память компьютера, матрицы в программе объявляют как одномерные массивы (см. подразд. 6.3). Хотя в компьютере даже многомерно объявленные массивы всегда хранятся как одномерные, компилятор генерирует индексные выражения с операциями умножения и деления. Операции сложения и вычитания в компьютерах выполняются гораздо быстрее, поэтому индексы для доступа к элементам матриц следует программировать в рекуррентной инкрементной форме, экономя таким образом время процессора (табл. 7.1). В этой программе преобразование в треугольную форму выполняется отождествлением $J(J-1)/1+I$ с (I, J) . Рекуррентное инкрементное вычисление KK , JJ и KK экономит вычисления.

Как отмечалось на с. 118, иногда требуется вычислять R^{-1} . Такая ситуация появляется, если требуется найти A^{-1} , для которой уже выполнено преобразование $TA = R$, где $T = T^{(n-1)}$ по формуле (7.21), так как в теореме 7.1 для этого случая $m = n$ и $A^{-1} = R^{-1}T$. Последнее означает, что то же самое ортогональное преобразование T теперь надо применить к строкам матрицы R^{-1} , но уже в обратном порядке следования элементарных преобразований, составляющих полное преобразование $T = T^{(n-1)}$ по формуле (7.21). Таким образом, возникает проблема запоминания элементарных преобразований, составляющих полное преобразование $T = T^{(n-1)}$, чтобы позже можно было его применить в задаче отыскания A^{-1} или же для решения уравнения $Ax = z$ с невырожденной матрицей A после преобразования $TA = R$.

Таблица 7.1. Эффективное обращение верхней треугольной матрицы $U := R^{-1}$.

$R(N, N), \quad U(N, N), \quad R \text{ и } U — \text{верхнетреугольные матрицы.}$	
$U(1) = 1./R(1)$ $JJ = 1$ DO 20 $J = 2, N$ $\overline{JJ} = JJ$	© $R(1) \equiv R(1, 1)$
$JJ = JJ + J$ $U(JJ) = 1./R(JJ)$ $JM1 = J - 1$ $KK = 0$ DO 20 $K = 1, JM1$ $KK = KK + K$ $KI = KK$ $SUM = 0.$ DO 10 $I = K, JM1$ $SUM = SUM - U(KI) * R(\overline{JJ} + I)$	© $\overline{JJ} = J(J - 1)/2 = (J - 1, J - 1)$ © $JJ = J(J + 1)/2 = (J, J)$ © $KK = K(K + 1)/2$
10 $KI = KI + I$ 20 $U(\overline{JJ} + K) = SUM * U(JJ)$	© $KI = (K, I + 1)$ © $\overline{JJ} + K = (K, J)$
Реализуются формулы (7.31), (7.32) и (7.33). Верхние треугольные матрицы R и U хранятся как векторы размерности $N(N + 1)/2$. Если нужно, U может замещать R .	

Как видно из (7.24), для отражения вектора $y = T^{(k)}z$ от гиперплоскости U_{\perp} , заданной вектором u , требуется иметь сохраненными две величины: вектор u и скаляр α . Поскольку нули ниже диагонали, получающиеся в результате триангуляризации, хранить не нужно, это место можно отвести для сохранения вектора u (вместе с диагональю, поэтому диагональные элементы s_k приходится хранить отдельно). Данное предложение схематически показано на рис. 7.5 (вверху). Каким образом можно выполнить вычисление матрицы A^{-1} , показано на рис. 7.5 (внизу) на конкретном примере размерностей, а именно: $m = 4, n = 4$.

7.7 Преобразование Гивенса

Преобразование Гивенса осуществляет вращение вектора в плоскости двух координат. Очевидно, поворот вектора $y = (y_1 \mid y_2)^T$ на угол θ по часовой стрелке эквивалентен повороту системы координат против часовой стрелки на тот же угол.

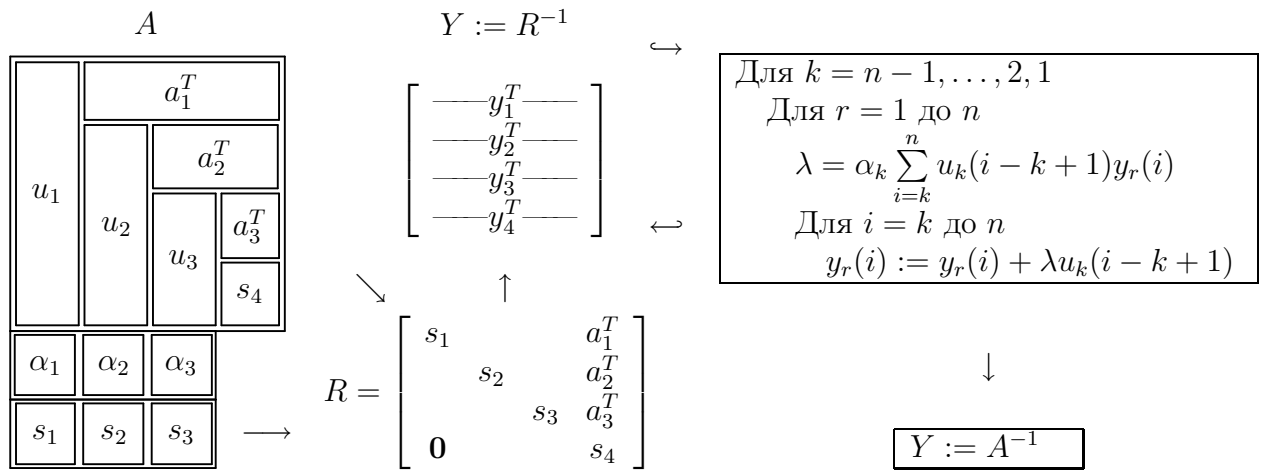
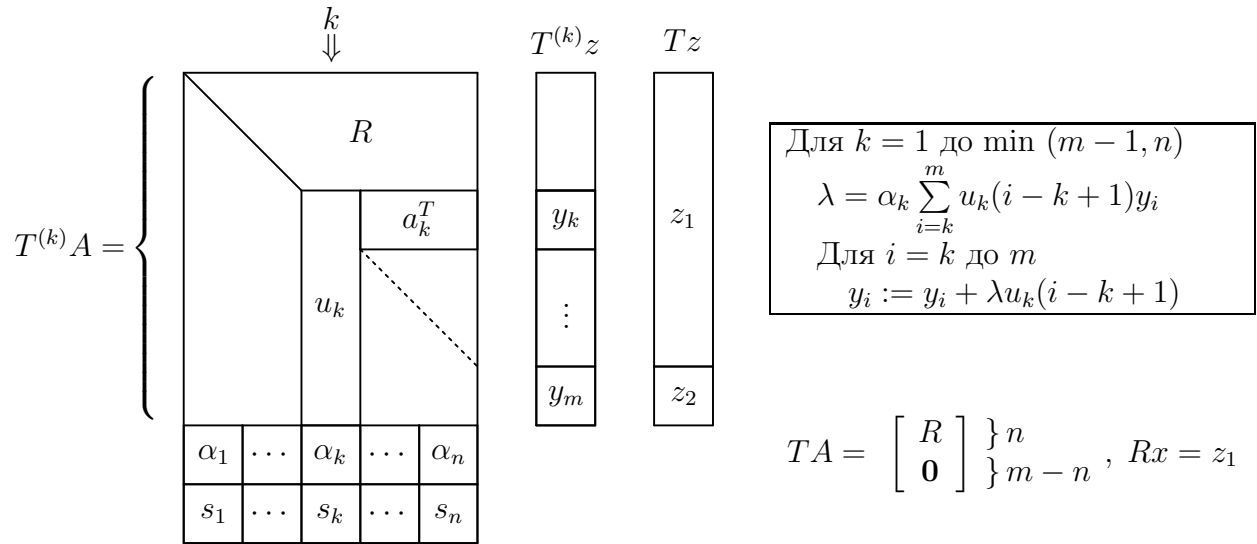


Рис. 7.5. Вверху: Сохранение преобразования T и вычисление вектора $y = Tz, \forall y \in \mathbb{R}^m$. Внизу: Вычисление матрицы A^{-1} после сохранения преобразования T

Следовательно, легко найти (рис. 7.6), что координаты y'_1, y'_2 повернутого вектора $y_r = (y'_1 \mid y'_2)^T$ определяются в виде $y'_1 = y_1 \cos \theta + y_2 \sin \theta$, $y'_2 = -y_1 \sin \theta + y_2 \cos \theta$.

Записывая это в матричной форме и требуя, чтобы поворот $P_{1,2}$ в плоскости (e_1, e_2) происходил до совмещения с первой координатной осью, получим

$$y_r = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} y = P_{1,2} y = \begin{bmatrix} r \\ 0 \end{bmatrix}, \quad \left\{ \begin{array}{l} c \triangleq \cos \theta = y_1/r \\ s \triangleq \sin \theta = y_2/r \end{array} \right\}, \quad r \triangleq \sqrt{y_1^2 + y_2^2},$$

где, очевидно, матрица $P_{1,2}$ плоского вращения ортогональна при любом θ .

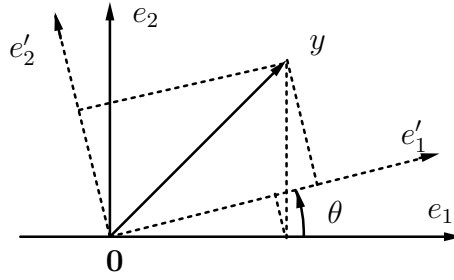


Рис. 7.6. Геометрия вращений

Триангуляризация матрицы преобразованиями Гивенса. Выбор θ такой, что вторая координата вектора y_t становится равной нулю, используем для триангуляризации матрицы $A(m, n)$. На первом шаге нужны преобразования, делающие равными нулю все элементы ниже первого диагонального элемента. Для этого, очевидно, нужно выполнить последовательно элементарные вращения $P_{1,2}, P_{1,3}, \dots, P_{1,m}$. Так определенные преобразования воздействуют на все столбцы матрицы, но только первый столбец, который уместно назвать *ведущим столбцом в преобразовании Гивенса*, приобретает желаемый вид.

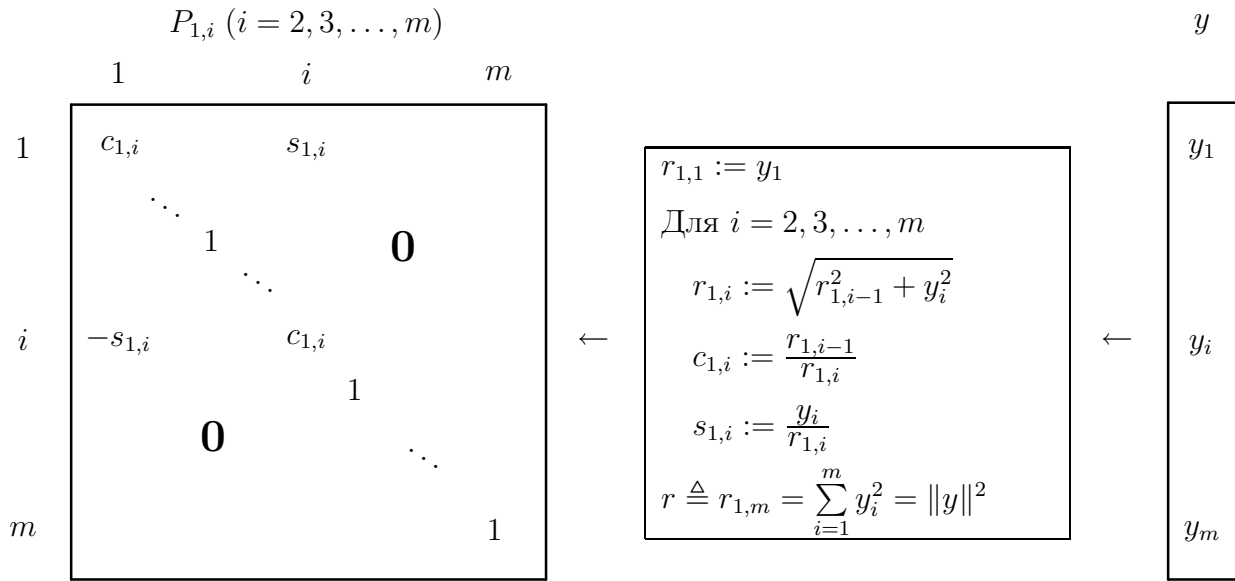
ЛЕММА 7.2. Пусть дана матрица $A(m, n)$ и y — ее ведущий (левый) столбец. Тогда существует ортогональное преобразование Гивенса, задаваемое матрицей $P_1 = P_1(m, m)$, такое, что

$$P_1 A = \begin{matrix} & \overbrace{\quad}^1 & \overbrace{\quad}^{n-1} \\ \begin{matrix} 1\{ \\ m-1\{ \end{matrix} & \left[\begin{array}{c|c} r & \tilde{A} \\ \mathbf{0} & \end{array} \right] \end{matrix}, \quad (7.34)$$

$$P_1 = P_{1,m} \cdots P_{1,3} P_{1,2},$$

и матрицы $P_{1,j}$ определяются по алгоритму на рис. 7.7.

ТЕОРЕМА 7.2 (*Триангуляризация матрицы по методу Гивенса*). Пусть $A_1 := A(m, n)$ и для каждого $j = 1, 2, \dots, k$, $k \leq \min(m-1, n)$ серия элементарных преобразований Гивенса, задаваемая матрицей P_j размера $(m+1-j) \times (m+1-j)$, выбрана, как сказано ниже.


 Рис. 7.7. Вычисление матрицы $P_{1,j}$

Для ведущего (левого) столбца y_j матрицы A_j эта P_j выбрана так, что

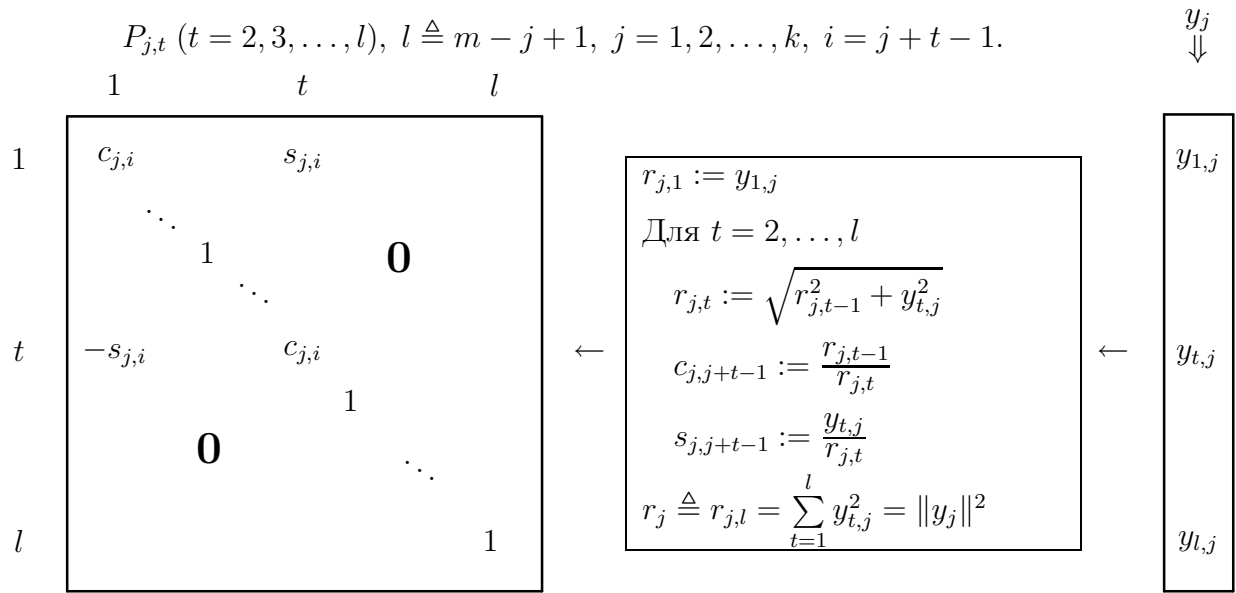
$$P_j A_j = \begin{matrix} 1 \\ m-j \end{matrix} \left\{ \begin{matrix} \overbrace{1}^1 & \overbrace{n-j}^{n-j} \\ \left[\begin{matrix} r_j \\ \mathbf{0} \end{matrix} \right] & \begin{matrix} a_j^T \\ A_{j+1} \end{matrix} \end{matrix} \right\}, \quad j = 1, \dots, k; \quad k \leq \min(m-1, n). \quad (7.35)$$

Тогда после k повторных применений леммы 7.2 имеем следующий промежуточный результат триангуляризации матрицы A :

$$P^{(k)} A = \left[\begin{array}{ccc|ccc} r_1 & & & & a_1^T & \\ & r_2 & & & a_2^T & \\ & & \ddots & & \vdots & \\ & & & r_k & a_k^T & \\ & \mathbf{0} & & & & A_{k+1} \end{array} \right] \quad (7.36)$$

с отвечающей этому моменту процесса итоговой матрицей преобразований

$$P^{(k)} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & P_k \end{bmatrix} \cdots \begin{bmatrix} I_1 & 0 \\ 0 & P_2 \end{bmatrix} P_1, \quad P_j = P_{j,m-j+1} \cdots P_{j,3} P_{j,2}, \quad (7.37)$$



Формула (7.37) имеет рекуррентный вид произведения

$$\left. \begin{aligned} P^{(j)} &= \begin{bmatrix} I_{j-1} & 0 \\ 0 & P_j \end{bmatrix} P^{(j-1)}, \quad P^{(1)} = P_1 \\ P_j &= P_{j,m-j+1} \cdots P_{j,3} P_{j,2}, \quad j = 2, \dots, N \end{aligned} \right\}, \quad N = \min(m-1, n). \quad (7.38)$$

Все участвующие здесь матрицы являются ортогональными, поэтому финальная матрица $P \triangleq P^{(N)}$ также ортогональна. Общее число используемых при этом элементарных матриц вращения равно $(m-1) + (m-2) + \dots + (m-N) = (2m-N-1)N/2$. В результате (в случае $m > n$) получим

$$PA = \begin{bmatrix} R \\ \cdots \\ \mathbf{0} \end{bmatrix}, \quad R = \begin{array}{c|c} \text{треугольник} & \\ \hline \mathbf{0} & \end{array} \triangleq R_{\text{не}}, \quad (7.39)$$

где индекс не подчеркивает, что в треугольной матрице R заполненной частью может быть только «северо-восточная» (*north-by-east*) часть. Полагая $Q = P^T$, при $m = n$ имеем QR -разложение матрицы $A = A(n, n)$, т. е. $A = QR$. Матрицы в (7.37) и (7.38) непосредственно не вычисляются.

Для алгоритма Гивенса — так же, как и для других матричных алгоритмов, — имеются две схемы вычислений: (1) *строчно ориентированная схема Гивенса* и (2) *столбцово ориентированная схема Гивенса* (рис. 7.8). Как и в алгоритме преобразований Хаусхолдера (см. рис. 7.5), здесь обычно требуется сохранять информацию о произведенных по ходу алгоритма (7.38) элементарных преобразованиях.

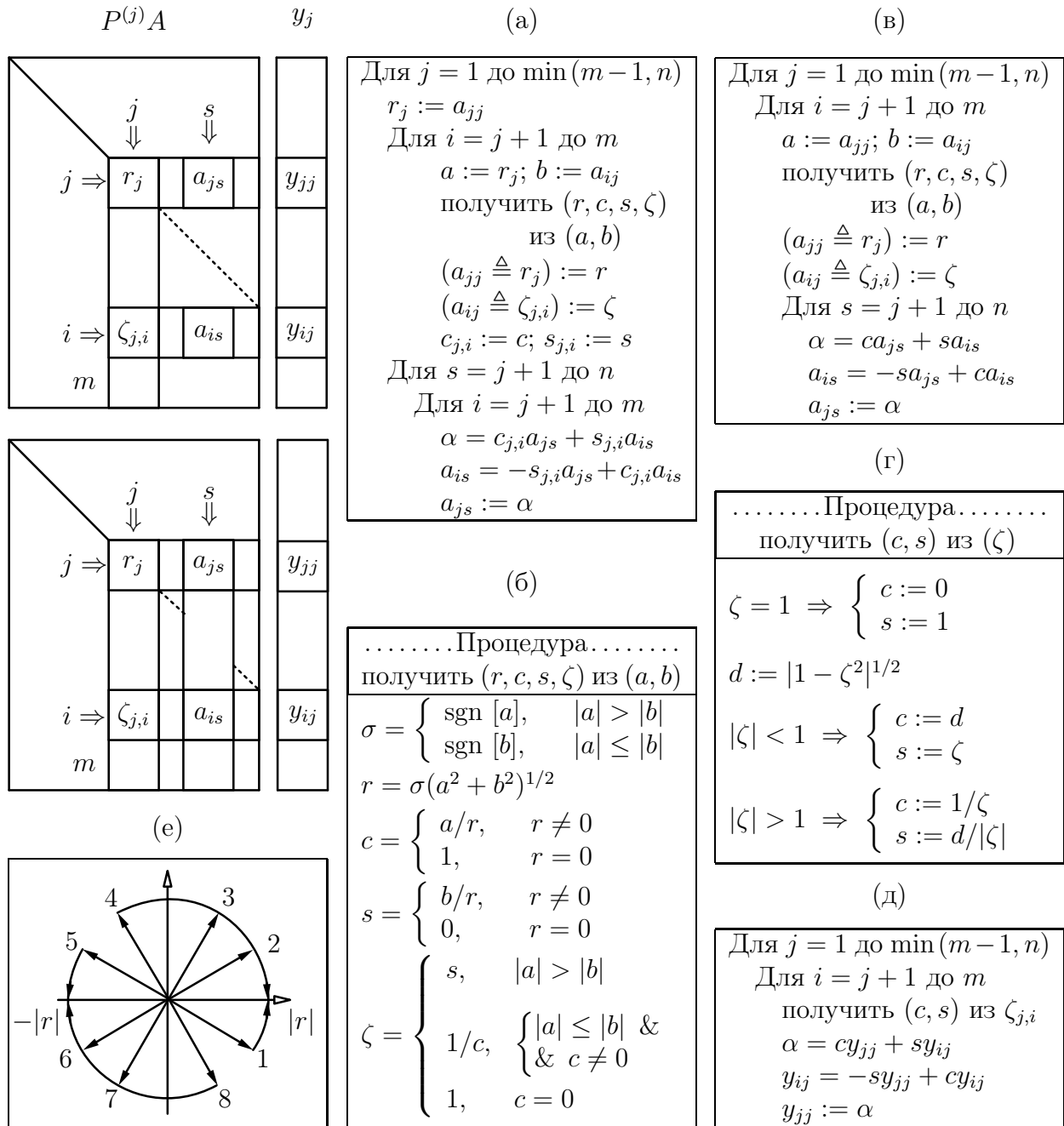


Рис. 7.8. Преобразование Гивенса: (а) столбцово ориентированная схема вычисления матрицы PA , где $P = P^{(j)}$ при $j = \min(m-1, n)$ (нижняя матрица слева); (б) вычисление координаты r вектора $(a, b)^T$, повернутого до совмещения с первой осью, а также косинуса и синуса угла поворота и рабочего признака ζ ; (в) строчно ориентированная схема вычисления матрицы PA (верхняя матрица (г) восстановления косинуса и синуса угла поворота из признака ζ ; (д) получение вектора y теми преобразованиями $P_{j,i}$ произвольного вектора $z \in \mathbb{R}^m$, которые сохранены в рабочих признаках $\zeta_{j,i}$ и восстанавливаются из них; (е) вследствие п. (б) векторы 1, 2, 3 и 4 поворачиваются к положительному направлению первой координатной оси, а векторы 5, 6, 7 и 8 — к отрицательному направлению этой оси.

Сохранение этой информации позволит впоследствии решать системы уравнений $Ax = z$ (совместные или несовместные, в последнем случае — по методу наименьших квадратов, см. подразд. 7.3) или же находить обратную матрицу A^{-1} (когда $m = n$).

Необходимая информация — это значения косинуса и синуса, однако их сохранение было бы неэффективным решением. Gentleman (1973) предложил способ [11], включенный в рис. 7.8(б) и (г) с геометрической иллюстрацией его действия на рис. 7.8(е). Введенный им рабочий признак ζ — это одно число, которое можно хранить в позиции (i, j) как $\zeta_{j,i}$ вместо нулевого элемента, появляющегося в позиции (i, j) матрицы (7.39) в момент преобразования $P_{j,t}$ ($t = i + 1 - j$) в (7.37). Как и с преобразованиями Хаусхолдера, нахождение A^{-1} после преобразований Гивенса требует такой же последовательности процедур: сначала находят R^{-1} (см. рис. 7.5 (внизу)), затем к R^{-1} применяют с правой стороны финальное преобразование $P \triangleq P^{(N)}$ (7.38), так как $A^{-1} = R^{-1}P$. Для этого для рис. 7.5 (внизу) надо взять алгоритм из рис. 7.8(д), который также отыскивает Pz при решении уравнения $Ax = z$.

7.8 Варианты заполнения матрицы R

Традиционно ортогональные преобразования (выше рассмотрены T — преобразование Хаусхолдера и P — преобразование Гивенса) приводят матрицу к виду, показанному на рис. 7.4 или в выражении (7.39). Однако выбор того угла матрицы, который должен остаться треугольно заполненным, естественно, произволен. Предпочтения диктуются целями использования, т. е. предназначением преобразования. Преследуя здесь учебно-тренировочные цели, включим в проект (см. подразд. 7.16) все четыре возможных варианта заполнения матрицы R . Вариант 1 показан в (7.39), другие три имеют следующий вид. Вариант 2:

$$QA = \begin{bmatrix} R \\ \cdots \\ \mathbf{0} \end{bmatrix}, \quad R = \begin{array}{|c|} \hline \text{треугольн.} \\ \hline \mathbf{0} \end{array} \triangleq R_{\text{nw}}, \quad (7.40)$$

где Q обозначает либо T (преобразование Хаусхолдера), либо P (преобразование Гивенса), индекс _{nw} подчеркивает, что в треугольной матрице R заполненной частью может быть только «северо-западная» (*north-by-west*)

часть. Вариант 3:

$$QA = \begin{bmatrix} \mathbf{0} \\ \cdots \\ R \end{bmatrix}, \quad R = \begin{array}{|c|} \hline \mathbf{0} \\ \hline \text{треугольная матрица} \\ \hline \end{array} \triangleq R_{\text{se}}, \quad (7.41)$$

где индекс se подчеркивает, что в треугольной матрице R заполненной частью может быть только «юго-восточная» (*south-by-east*) часть. Вариант 4:

$$PA = \begin{bmatrix} \mathbf{0} \\ \cdots \\ R \end{bmatrix}, \quad R = \begin{array}{|c|} \hline \text{треугольная матрица} \\ \hline \mathbf{0} \\ \hline \end{array} \triangleq R_{\text{sw}}, \quad (7.42)$$

где индекс sw подчеркивает, что в треугольной матрице R заполненной частью может быть только «юго-западная» (*south-by-west*) часть. Вполне очевидно, что эти варианты получаются простым изменением порядка действий в алгоритмах преобразований.

7.9 Правосторонние ортогональные преобразования

С правосторонними ортогональными преобразованиями мы уже сталкивались (см. подразд. 7.6); тогда для квадратной матрицы A после $TA = R$ вычисляли $A^{-1} = R^{-1}T$. Однако, можно начинать с правостороннего преобразования матрицы A ; тогда отыскание A^{-1} потребует, соответственно, левостороннего преобразования.

Пусть $A = A(n, n)$ — квадратная невырожденная матрица. Будем рассматривать ее строки как векторы в \mathbb{R}^n . Преобразования вектора как матрицы-строки в n -мерном линейном пространстве задаются умножением ее на преобразующую матрицу справа. Поэтому правосторонним ортогональным преобразованием Q можно привести матрицу A к виду $AQ = R$, где применена ортогональная матрица Q одного из типов, а R — треугольная матрица, имеющая форму одного из возможных вариантов заполнения (см. подразд. 7.8). При этом преобразованию Q подвергаются не столбцы, а строки матрицы A , и преобразование Q запоминается по принципу, показанному ранее на рис. 7.5 и рис. 7.8, на месте элементов, обращаемых в нуль.

После такого преобразования матрицы A решение системы $Ax = z$ сводится к решению эквивалентной системы с треугольной матрицей $Ry = z$. Затем искомый вектор определяется через сохраненное преобразование Q как $x = Qy$. Обратная матрица A^{-1} , соответственно, находится как решение системы $RY = I$ с последующим преобразованием Q матрицы Y , т. е.

$X = A^{-1} = QY$. Матрица Q не формируется, из чего видна необходимость запоминания преобразований, обеспечивших $AQ = R$.

7.10 Двусторонние ортогональные преобразования

Ортогональные преобразования, будучи применены одновременно слева и справа к данной матрице A , позволяют приводить ее к формам с нулями как ниже, так и выше диагонали. Это, в свою очередь, облегчает решение других сложных задач (матричная проблема собственных значений [143]). С помощью ортогональных преобразований для квадратной матрицы широко распространены: приведение симметрической матрицы к трехдиагональному виду и приведение квадратной матрицы к двухдиагональному виду. При этом в качестве ортогональных преобразований одинаково успешно могут быть использованы преобразования Хаусхолдера или преобразования Гивенса.

Приведение симметрической матрицы к трехдиагональному виду. Применим к симметрической матрице слева и справа преобразование Хаусхолдера (или Гивенса), выбирая его из задачи желаемого преобразования ведущего столбца и ведущей строки, а именно: сохранение первого диагонального элемента, получение ненулевых элементов в двух смежных с ним позициях и получение нулевых элементов в остальных позициях.

ЛЕММА 7.3. Пусть дана матрица $A = A(n, n) = A^T$. Тогда существует ортогональное преобразование Q_2 (Хаусхолдера T_2 или Гивенса P_2) такое, что

$$\left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_2 \end{array} \right] A \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_2^T \end{array} \right] = \begin{array}{c} \overbrace{1} \quad \overbrace{1} \quad \overbrace{n-2} \\ \left\{ \begin{array}{c} 1 \\ 1 \\ n-2 \end{array} \right\} \left[\begin{array}{c|cc} a_1 & s_1 & \mathbf{0} \\ \hline \mathbf{0} & & \tilde{A} \end{array} \right] \end{array}. \quad (7.43)$$

ЗАМЕЧАНИЕ 7.6. В (7.43) транспонирование Q_2^T не требуется, если в качестве Q_2 взято преобразование Хаусхолдера (в силу его симметричности). При этом индекс «2» указывает на позицию того элемента в ведущем столбце (для левостороннего преобразования) или в ведущей строке (для правостороннего преобразования), который остается ненулевым в этом столбце (в результате применения Q_2) или в этой строке (в результате применения Q_2^T).

В данном случае, т.е. в (7.43), эти элементы суть s_1 и s_1 . Элемент a_1 не изменяется, так как I_1 — единичная матрица размера 1×1 .

ТЕОРЕМА 7.3 (*Тридиагонализация симметрической матрицы*). Пусть дана симметрическая матрица $A = A(n, n) = A^T$, $A_1 := A(n, n)$ и для каждого $j = 1, \dots, k$, где $k \leq N = n - 2$, выбрано элементарное преобразование Q_{j+1} (Хаусхолдера T_{j+1} или Гивенса P_{j+1}) так, что

$$\left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_{j+1} \end{array} \right] A_j \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_{j+1}^T \end{array} \right] = \begin{array}{c} \overbrace{1} \{ \left[\begin{array}{c|cc} a_j & s_j & \mathbf{0} \\ \hline s_j & & \\ \mathbf{0} & & A_{j+1} \end{array} \right] \} \\ \overbrace{1} \{ \\ n-j-1 \{ \end{array} \quad (7.44)$$

Тогда после k повторных применений леммы 7.3 имеем отвечающую этому моменту процесса итоговую матрицу преобразований

$$Q^{(k)} = \left[\begin{array}{cc} I_k & 0 \\ 0 & Q_{k+1} \end{array} \right] Q^{(k-1)}, \quad 1 \leq k \leq N = n - 2, \quad Q^{(0)} = I_n \quad (7.45)$$

и промежуточный результат тридиагонализации данной матрицы A в виде

$$Q^{(k)} A (Q^{(k)})^T = \left[\begin{array}{cccccc} a_1 & s_1 & & & & \\ s_1 & a_2 & s_2 & & & \\ & s_2 & \ddots & \ddots & & \\ & & \ddots & a_k & s_k & \\ & & & s_k & & \\ \mathbf{0} & & & & & A_{k+1} \end{array} \right].$$

Приведение квадратной матрицы к двухдиагональному виду.

Применим к произвольной квадратной матрице слева преобразование Q_1 и справа преобразование S_2 (беря любое из них как преобразование Хаусхолдера или как преобразование Гивенса), при этом Q_1 выберем из задачи желаемого преобразования ведущего столбца и S_2 — из задачи желаемого преобразования ведущей строки, а именно: при действии Q_1 — получение ненулевого диагонального элемента и нулевых элементов ниже него в первом (ведущем) столбце; при действии S_2 — сохранение диагонального элемента, получение в смежной с ним позиции ненулевого элемента и нулевых элементов правее него в первой (ведущей) строке.

ЛЕММА 7.4. Пусть дана матрица $A = A(n, n)$. Тогда существуют ортогональное преобразование Q_1 (Хаусхолдера или Гивенса) и ортогональное преобразование S_2 (Хаусхолдера или Гивенса) такие, что

$$Q^{(1)}AS^{(1)} = \begin{matrix} \overbrace{1} & \overbrace{1} & \overbrace{n-2} \\ 1\{ \left[\begin{array}{c|cc} s_1 & a_1 & \mathbf{0} \\ \hline \mathbf{0} & \tilde{A} & \end{array} \right] \end{matrix}_{n-2}, \quad \begin{cases} Q^{(1)} = Q_1, \\ S^{(1)} = \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & S_2 \end{array} \right]. \end{cases} \quad (7.46)$$

ТЕОРЕМА 7.4 (*Бидиагонализация квадратной матрицы*). Пусть дана квадратная матрица $A = A(n, n)$, $A_1 := A$ и для каждого $j = 1, \dots, k$, где $k \leq n - 2$, выбраны элементарное преобразование Q_j (Хаусхолдера типа T_j или Гивенса типа P_j) и элементарное преобразование S_{j+1} (Хаусхолдера типа T_{j+1} или Гивенса типа P_{j+1}) таким образом, что в результате получаем

$$Q_j A_j \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & S_{j+1} \end{array} \right] = \begin{matrix} \overbrace{1} & \overbrace{1} & \overbrace{n-j-1} \\ 1\{ \left[\begin{array}{c|cc} s_j & a_j & \mathbf{0} \\ \hline \mathbf{0} & A_{j+1} & \end{array} \right] \end{matrix}_{n-j}. \quad (7.47)$$

Тогда после k повторных применений леммы 7.4 имеем отвечающие этому моменту процесса итоговые матрицы преобразований

$$\left. \begin{aligned} Q^{(k)} &= \begin{bmatrix} I_{k-1} & 0 \\ 0 & Q_k \end{bmatrix} Q^{(k-1)}, \quad k \leq n-2, \quad Q^{(0)} = I_n, \quad Q^{(1)} = Q_1, \\ S^{(k)} &= S^{(k-1)} \begin{bmatrix} I_k & 0 \\ 0 & S_{k+1} \end{bmatrix}, \quad k \leq n-2, \quad S^{(1)} = \begin{bmatrix} I_1 & 0 \\ 0 & S_2 \end{bmatrix} \end{aligned} \right\} \quad (7.48)$$

и промежуточный результат бидиагонализации данной матрицы A в виде

$$Q^{(k)}AS^{(k)} = \left[\begin{array}{cc|cc} s_1 & a_1 & & \\ & s_2 & a_2 & \\ & & \ddots & \ddots \\ & & & s_k & a_k \\ \hline & & & & A_{k+1} \end{array} \right].$$

Выполнив после $k = n - 2$ еще одно левостороннее преобразование Q_{n-1} (что отвечает применению верхней формулы (7.48) для $k = n - 1$), получаем окончательно

$$Q^{(n-1)}AS^{(n-2)} = \begin{bmatrix} s_1 & a_1 & & & & \\ & s_2 & a_2 & & & \\ & & \ddots & \ddots & & \\ & & & s_{n-1} & a_{n-1} & \\ \mathbf{0} & & & & s_n & \end{bmatrix}.$$

Основное применение указанных двусторонних ортогональных преобразований заключается в вычислении сингулярных значений [13] произвольной матрицы $A = A(n, n)$, а также в решении проблемы собственных значений [143]. Однако эти преобразования можно использовать и для решения системы линейных алгебраических уравнений $Ax = f$. После приведения матрицы к двух- или трехдиагональному виду система уравнений легко решается. Например, в случае с трехдиагональной матрицей система очень эффективно решается методом прогонки [12].

7.11 Ортогонализация Грама–Шмидта

Пусть $A = A(m, n)$ — матрица, имеющая m строк и n столбцов, причем $m \geq n$. Обозначая i -й столбец через a_i , запишем $A = [a_1, a_2, \dots, a_n]$, $a_i \in \mathbb{R}^m$. Рассмотрим случай матрицы полного ранга, т. е. $\text{rank } A = n$. Тогда набор векторов $\{a_i\}$ порождает некоторое подпространство $\mathcal{L} \in \mathbb{R}^m$, т. е. может считаться его базисом. Назовем этот набор исходным базисом и преобразуем его в ортонормированный базис. Такое преобразование называется *процедурой ортогонализации системы векторов* $\{a_1, a_2, \dots, a_n\}$.

Согласно определению, ортонормированным базисом в $\mathcal{L} \in \mathbb{R}^m$ называется система векторов $\{q_1, q_2, \dots, q_n\}$ такая, что

- 1) $\forall i : q_i \in \mathbb{R}^m, m \geq n, q_i^T q_i = \|q_i\|^2 = 1;$
- 2) $\forall i, j, i \neq j : q_i^T q_j = 0$

и любой вектор a_i имеет единственное представление

$$a_i = \sum_{j=1}^n q_j b_{ji}, \quad i = 1, 2, \dots, n,$$

где $b_i^T = (b_{1i}, b_{2i}, \dots, b_{ni})$ — вектор-строка некоторых коэффициентов. Следовательно, матрицу A можно представить в виде произведения двух матриц $A = QB$, где $Q = [q_1, q_2, \dots, q_n]$ — матрица размера $(m \times n)$, составленная из столбцов $q_i \in \mathbb{R}^m$, а $B = [b_1, b_2, \dots, b_n]$ — матрица размера $(n \times n)$, составленная из столбцов $b_i \in \mathbb{R}^n$. Матрица $Q = Q(m, n)$ в этом представлении состоит из ортонормированных векторов-столбцов, в частном случае $m = n$ в качестве Q имеем ортогональную матрицу, т. е. $Q^T Q = I$.

Таким образом, ортогонализация столбцов матрицы A есть представление $A = QB$, где Q — матрица тех же размеров, что и A , но в отличие от A , имеющая ортонормированные столбцы, при этом B — квадратная матрица, обеспечивающая равенство $A = QB$. Очевидно, существует бесконечное множество таких представлений матрицы A , поскольку число ортонормированных базисов не ограничено. Для обеспечения единственности среди множества версий $A = QB$ выберем представление, при котором B — треугольная матрица, которую далее традиционно будем обозначать R , поскольку в ней оказывается заполнен правый (*right*) верхний угол, т. е. $R = R_{\text{не}}$. Хотя традиционно *ортогонализацией Грама–Шмидта* называют отыскание по матрице A такой матрицы Q , что $A = QR$, где $R = R_{\text{не}}$, для R будем допускать все четыре возможных варианта заполнения (см. подразд. 7.8):

- ✧ вариант 1: $R = R_{\text{не}}$, где $R_{\text{не}}$ — верхняя правая треугольная матрица;
- ✧ вариант 2: $R = R_{\text{sw}}$, где R_{sw} — нижняя левая треугольная матрица;
- ✧ вариант 3: $R = R_{\text{se}}$, где R_{se} — нижняя правая треугольная матрица;
- ✧ вариант 4: $R = R_{\text{nw}}$, где R_{nw} — верхняя левая треугольная матрица.

Для ортогонализации системы векторов вычисление матрицы R в явном виде может и не требоваться, хотя такое вычисление всегда присутствует. Ниже, рассматривая ортогонализацию Грама–Шмидта обобщенно, т. е. во всевозможных вариантах треугольного заполнения матрицы R , мы будем требовать явного нахождения факторов (сомножителей) Q и R в разложении $A = QR$. Для любого из вариантов возможны три формы алгоритма, отличающиеся порядком действий.

• Грама–Шмидта Ортогонализация (ГШО)

Этот вариант алгоритма предполагает вычисление ненулевых элементов матрицы R по столбцам, начиная с самого короткого (одноэлементного) столбца.

- **Модифицированная ГШО (МГШО)**

В этом варианте ненулевые элементы матрицы R вычисляются по строкам, начиная с самой длинной (состоящей из n элементов) строки.

- **МГШО с выбором ведущего вектора**

Этот вариант МГШО-алгоритма использует стратегию выбора ведущего вектора. В качестве очередного, подлежащего ортогонализации вектора, выбирается тот из оставшихся столбцов матрицы A , который имеет наибольшую длину (евклидову норму). Хотя эта стратегия требует дополнительных вычислительных затрат, в некоторых плохо обусловленных задачах она так же полезна, как и выбор главного элемента в методе Гаусса.

Таким образом, данной темой — ортогонализация Грама–Шмидта — в предлагаемом проекте (см. подразд. 7.16) охвачено $(4 \times 3) = 12$ различных вариантов задачи разложения $A = QR$.

ЗАМЕЧАНИЕ 7.7. Обратим внимание на различия между двумя типами ортогональных преобразований, а именно: между преобразованиями Хаусхолдера и Гивенса (если говорить о левосторонних их версиях, хотя возможны и правосторонние) — это один тип преобразований, и ортогонализацией Грама–Шмидта — другой тип. Первый тип обеспечивает равенства (7.39), (7.40), (7.41) или (7.42), где преобразованная матрица QA имеет тот же размер, что и исходная матрица A , и в ее составе присутствует блок, появляющийся как треугольная матрица R в одном из четырех углов этой матрицы QA . При этом матрица Q ортогонального преобразования — квадратная. В случае ортогонализации Грама–Шмидта имеем $A = QR$, где Q — матрица того же размера, что и A , но с ортонормированными столбцами, а R — строго квадратная матрица с треугольным заполнением.

7.12 Алгоритмы ортогонализации Грама–Шмидта

Рассмотрим задачу QR -разложения матрицы $A(t, n)$, $t \geq n$, полного ранга на основе ортогонализации Грама–Шмидта.

В данной задаче, рассматривая пример $n = 3$, имеем

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \end{bmatrix},$$

$A = [a_1, a_2, a_3] = [q_1, q_2, q_3]R = [r_{11}q_1, r_{12}q_1 + r_{22}q_2, r_{13}q_1 + r_{23}q_2, r_{33}q_3]$.
В результате получаем линейную систему

$$\begin{aligned} r_{11}q_1 &= a_1, \\ r_{12}q_1 + r_{22}q_2 &= a_2, \\ r_{13}q_1 + r_{23}q_2 + r_{33}q_3 &= a_3. \end{aligned}$$

Ясно, что каждое из этих выражений представляет собой разложение вектора a_1, a_2 или a_3 по системе ортов $\{q_1, q_2, q_3\}$, при этом коэффициент r_{ij} есть алгебраическая проекция вектора a_j на орт q_i .

В силу треугольности матрицы R эта система легко решается. Из первого уравнения находим орт q_1 вдоль вектора a_1 и координату (проекцию как число) первого вектора a_1 вдоль орта q_1 :

$$q_1 = a_1 / \|a_1\|, \quad r_{11} = \|a_1\|.$$

Второе уравнение есть разложение вектора a_2 на сумму проекций вдоль ортов q_1 и q_2 . Так как орт q_1 уже найден, то координата r_{12} легко определяется в виде

$$r_{12} = a_2^T q_1.$$

После этого из второго уравнения имеем

$$r_{22}q_2 = a_2 - r_{12}q_1,$$

следовательно,

$$\begin{aligned} q_2 &= (a_2 - r_{12}q_1) / \|a_2 - r_{12}q_1\|, \\ r_{22} &= \|a_2 - r_{12}q_1\|. \end{aligned}$$

ЗАМЕЧАНИЕ 7.8. По предположению, $\text{rank } A = n$, т.е. ни один из векторов a_i не является нулевым, и все a_i образуют линейно независимую систему. Поэтому $r_{11} \neq 0$, $r_{22} \neq 0$ и $r_{33} \neq 0$, следовательно, существует R^{-1} .

Продолжая решение системы, для третьего уравнения находим

$$r_{13} = a_3^T q_1, \quad r_{23} = a_3^T q_2$$

и затем определяем

$$r_{33}q_3 = a_3 - r_{13}q_1 - r_{23}q_2.$$

Отсюда

$$\begin{aligned} q_3 &= (a_3 - r_{13}q_1 - r_{23}q_2) / \|a_3 - r_{13}q_1 - r_{23}q_2\|, \\ r_{33} &= \|a_3 - r_{13}q_1 - r_{23}q_2\|. \end{aligned}$$

Таким образом, получили классический метод ГШО, отличающийся тем, что матрица R определяется по столбцам с номерами $k = 1, 2, \dots, n$.

В настоящее время существуют две более эффективные версии — ортогонализации Грама–Шмидта:

- Алгоритм МГШО (модифицированная схема).
- Алгоритм МГШО с выбором ведущего вектора.

Первые две версии — классическая и модифицированная — показаны здесь — на стр. 135, а модифицированная с выбором ведущего вектора — на стр. 136.

Алгоритм ГШО (классическая схема)

Для $k = 1$ до n

$$r_{ik} = a_k^T q_i, \quad i = 1, 2, \dots, k-1,$$

$$v = a_k - \sum_{i=1}^{k-1} r_{ik} q_i,$$

$$r_{kk} = (v^T v)^{1/2},$$

$$q_k = v / r_{kk}.$$

Алгоритм МГШО (модифицированная схема)

Для $k = 1$ до n

$$r_{kk} = \|a_k\| = (a_k^T a_k)^{1/2},$$

$$a_k = a_k / r_{kk},$$

Для $j = k+1$ до n

$$r_{kj} = a_j^T a_k,$$

$$a_j = a_j - r_{kj} a_k.$$

Алгоритм МГШО с выбором ведущего вектора

Для $k = 1$ до n

$$q(k) = k,$$

Для $k = 1$ до n

Для $s = k$ до n

$$\text{Найти } \mathbb{N}^\circ l, \text{ для которого } \|a_{q(l)}\| = \max_s \|a_{q(s)}\|,$$

Переставить номера: $q(k) \rightleftharpoons q(l)$,

$$r_{q(k),q(k)} = \|a_{q(k)}\| = (a_{q(k)}^T a_{q(k)})^{1/2},$$

$$a_{q(k)} = a_{q(k)} / r_{q(k),q(k)},$$

Для $j = k + 1$ до n

$$r_{q(k),q(j)} = a_{q(j)}^T a_{q(k)},$$

$$a_{q(j)} = a_{q(j)} - r_{q(k),q(j)} a_{q(k)}.$$

Первая из более современных версий, называемая МГШО (Rice, 1966 [11]), отличается порядком вычислений матрицы R . В этой версии матрица R определяется по строкам с номерами $k = 1, 2, \dots, n$. Этот алгоритм требует меньше оперативной памяти, так как в нем не используется промежуточный вектор v . Кроме того, матрица A заменяется матрицей Q , потому что после операции деления имеем $a_k = q_k$. Одним из его преимуществ является то, что в него легко внедрить процедуру выбора ведущего столбца.

Чтобы получить вторую из более современных версий — так называемую МГШО с выбором ведущего вектора, — нужно изменить алгоритм МГШО таким образом, чтобы очередным ортогонализируемым вектором оказался не k -й, а тот, чья норма наибольшая среди всех оставшихся s -х векторов от $s = k$ до $s = n$. Как и в подразд. 2.2, реально переставляются не столбцы матрицы A , а обмениваются значениями только элементы дополнительного вектора q , в котором фиксируются номера столбцов матрицы A . Доступ к элементам матрицы A осуществляется с использованием этого вектора. Перед началом работы основного алгоритма ортогонализации этот вектор перестановок q заполняется числами от 1 до n в естественном порядке нумерации столбцов матрицы A .

7.13 Решение систем после ортогонализации

1. Пусть дана система линейных алгебраических уравнений с квадратной невырожденной матрицей $Ax = f$. Тогда после ортогонального приведения матрицы с помощью одной из версий ортогонализации Грама–Шмидта имеем представление этой матрицы в виде $A = QR$ и, следовательно, $QRx = f$ и $Rx = Q^T f$.

2. Пусть дана система линейных алгебраических уравнений с прямоугольной матрицей $A(m, n)$, $m > n$, полного ранга. Такая система называется *переопределенной системой*. *Нормальное псевдорешение* \bar{x} , найденное по *методу наименьших квадратов* (МНК), удовлетворяет *нормальным уравнениям*

$$A^T A \bar{x} = A^T f.$$

Поскольку $A = QR$ и $Q^T Q = I$, эти уравнения эквивалентны уравнению

$$R \bar{x} = Q^T f,$$

которое совпадает по виду с уравнением из п. 1.

Чтобы вычислить x (для п. 1) или \bar{x} (для п. 2), находят вектор $f' = Q^T f$, а затем решают систему с треугольной матрицей R (методом подстановки).

7.14 Обращение матриц после ортогонализации

Для матрицы $A = A(n, n)$ имеем $A = QR$, где $Q = Q(n, n)$. Отсюда

$$A^{-1} = R^{-1}Q^{-1} = R^{-1}Q^T.$$

Следовательно, A^{-1} есть решение матричного уравнения $RX = Q^T$. Чтобы найти i -й столбец матрицы A^{-1} , надо в качестве правой части взять i -й столбец матрицы Q^T и решить систему с треугольной матрицей R (как в подразд. 7.13 или подробнее в подразд. 7.6).

7.15 Задание на лабораторный проект № 6

Написать и отладить программу, реализующую ваш вариант ортогонального преобразования для численного решения систем линейных алгебраических уравнений $Ax = f$ с квадратной матрицей A , вычисления

$\pm(\det(A))$ и A^{-1} . Предусмотреть предупреждение о невозможности решения указанных задач из-за присутствия (почти) линейно зависимых векторов среди столбцов матрицы A (в пределах ошибок округления ЭВМ или другого, заранее определенного критерия). Отделить основные части программы:

- а) подпрограмму факторизации матрицы A , отвечающую вашему варианту метода ортогонального приведения;
- б) подпрограмму решения систем линейных алгебраических уравнений;
- в) подпрограмму вычисления определителя матриц;
- г) подпрограмму обращения матриц;
- д) сервисные подпрограммы.

Уделить особое внимание эффективности программы (в смысле экономии оперативной памяти и скорости решения указанных выше задач). Предусмотреть пошаговое выполнение алгоритма ортогонального приведения с выводом результата на экран. Выполнить следующие пункты задания:

1. Провести подсчет фактического количества операций, выполняемых при решении системы линейных алгебраических уравнений (отдельно число операций сложения, число операций умножения, число операций деления и число операций извлечения квадратного корня) и сравнить эти числа с теоретическими (оценочными) числами.

2. Оценить скорость решения задач, т.е. определить время, затраченное на решение системы линейных алгебраических уравнений, и время, затраченное на обращение матриц. Для этого спроектировать и провести эксперимент, который охватывает матрицы порядка от 10 до 100 (через 10 порядков). Представить результаты в виде таблицы и графика зависимости времени выполнения (в минутах и секундах) от порядка матриц. Таблицу и график вывести на экран.

3. Оценить точность решения систем линейных алгебраических уравнений, имеющих тот же самый порядок, что и задачи из п. 2. Для этого сгенерировать случайные матрицы A , выбрать точное решение x^* и образовать правые части $f = Ax^*$. Провести анализ точности вычисленного решения x от порядка матрицы. Результаты представить в виде таблицы и графика.

Для заполнения матрицы A использовать случайные числа из диапазона от -100 до 100 . В качестве точного решения взять вектор $x^* = (1, 2, \dots, n)$, где n — порядок матрицы. Для оценки точности использовать норму вектора

$$\|x\|_{\infty} = \max_i (|x_i|).$$

Таблица 7.2. Варианты задания на лабораторный проект № 6

Вариант заполнения матрицы R	Отражения Хаусхолдера		Вращения Гивенса		Ортогонализация Грама–Шмидта		
	a	b	a	b	c	d	e
 $\triangleq R_{\text{ne}}$	1	2	3	4	5	6	7
 $\triangleq R_{\text{nw}}$	8	9	10	11	12	13	14
 $\triangleq R_{\text{se}}$	15	16	17	18	19	20	21
 $\triangleq R_{\text{sw}}$	22	23	24	25	26	27	28

a — столбцово-ориентированный алгоритм;

b — строчно-ориентированный алгоритм;

c — классическая схема;

d — модифицированная схема;

e — модифицированная схема с выбором ведущего вектора.

4. Повторить п. 3 задания для плохо обусловленных матриц (см. подразд. 2.6 лабораторной работы № 1), имеющих порядок от 4 до 40.

5. Системы из пп. 2 и 3 необходимо решить двумя методами: методом исключения из лабораторной работы № 1 и методом ортогонального приведения из лабораторной работы № 6. Сравнить точность решения и затраты машинного времени. Результаты представить в виде таблицы и графика.

6. Вычислить матрицу A^{-1} двумя способами:

1) через решение системы $AX = I$ на основе метода исключения Гаусса из лабораторной работы № 1 (в соответствии со своим вариантом);

2) через решение системы $AX = I$ на основе метода ортогонального преобразования (в соответствии со своим вариантом).

Сравнить затраты машинного времени и точность обращения способами 1) и 2). Эксперименты провести для матриц порядков от 10 до 100 через 10. Для оценки точности в обоих способах воспользоваться формулой из лабораторной работы (проекта) № 1.

7.16 Варианты задания на лабораторный проект № 6

По теме «Ортогональные преобразования матриц» студентам предлагается выполнение лабораторной работы — индивидуального проекта № 6.

Задание на этот проект содержит 28 вариантов, которые приведены в табл. 7.2 (см. стр. 139). Все варианты различаются по следующим признакам:

- четыре варианта заполнения треугольной матрицы R ;
- три вида ортогональных преобразований:
 - 1) отражения Хаусхолдера,
 - 2) вращения Гивенса,
 - 3) ортогонализация Грама–Шмидта,
- две разновидности алгоритма ортогонализации по методу Хаусхолдера и по методу Гивенса:
 - 1) столбцово-ориентированный алгоритм,
 - 2) строчно-ориентированный алгоритм,
- три разновидности ортогонализации по методу Грама–Шмидта:
 - 1) классическая схема,
 - 2) модифицированная схема,
 - 3) модифицированная схема с выбором ведущего вектора.

Если нет других указаний преподавателя, выбирайте ваш вариант в табл. 7.2 (см. стр. 139) по вашему номеру в журнале студенческой группы.

ИТЕРАЦИОННЫЕ МЕТОДЫ

Приведен справочный материал, необходимый в подразд. 8.12 для выполнения лабораторного проекта по этой теме (подробнее см. [12]).

8.1 Итерационные методы

Любой *итерационный метод* (ИМ) есть метод последовательных приближений. Он изначально предполагает наличие *методической погрешности* (ошибки метода) и этим отличается от *прямых методов* (ПМ), рассмотренных выше, у которых методическая погрешность равна нулю. *Сходящийся ИМ*, — а только такие ИМ и представляют интерес, — обеспечивает стремление этой погрешности к нулю по мере роста числа итераций, $n \rightarrow \infty$.

ИМ позволяют находить решение с некоторой заранее задаваемой погрешностью и при специальных условиях делают это быстрее, чем ПМ. Они запрашивают меньше ресурсов компьютера при решении больших систем. Алгоритмически ИМ более просты, чем ПМ, и в меньшей степени используют структуру матрицы.

Главные вопросы для любого ИМ — сходится ли он? какие условия для этого нужны? Однако для ИМ важно знать не только теоретический факт и условия сходимости. Следующие вопросы — какова скорость сходимости? как можно ее увеличить? Сходимость простых итерационных методов крайне медленная, особенно, в случае плохо обусловленной матрицы, поэтому эти вопросы крайне важны.

Есть два основных «регулятора» скорости ИМ: так называемая *лидирующая матрица* B и *скалярный параметр* τ . Если $B \neq I$, метод называют *неявным ИМ*, поскольку решение на новой итерации приходится искать тем или иным прямым методом применительно в системе с этой матрицей B , обязанной быть более простой, чем исходная матрица A . В силу этого неявные

методы алгоритмически более сложны, чем *явные ИМ* (в которых $B = I$); однако их преимуществом является существенно более быстрая сходимость. Скалярный параметр τ , как и лидирующую матрицу B , можно выбирать на каждой итерации, т. е. поставить их в зависимость от текущего номера итерации n . В этом случае *стационарный ИМ* становится *нестационарным итерационным методом*.

Кроме методических погрешностей на результат итерационного процесса влияют *трансформированные погрешности* алгоритма и *погрешности округления*. Положительным качеством ИМ является то, что при их использовании погрешности округления не накапливаются. Для запуска любого ИМ необходимо задавать начальное приближение к искомому решению.

8.2 Итерационная формула

ИМ могут применяться как для линейных задач, так и для нелинейных. Основой построения любого ИМ является так называемая *итерационная формула* (ИФ), т. е. формула вида $x = \varphi(x)$. Если она есть, то ИМ записывают в виде $x_{n+1} = \varphi(x_n)$, где $n = 0, 1, \dots$.

Построим ИФ для линейной системы

$$Ax = f$$

с обратимой $(m \times m)$ -матрицей $A = [a_{ij}]$, $i, j = 1, 2, \dots, m$, с неизвестным вектором $x = (x_1, x_2, \dots, x_m)^T$ и с заданной правой частью $f = (f_1, f_2, \dots, f_m)^T$. Пусть $\forall i \in \{i, j = 1, 2, \dots, m\} \mid a_{ii} \neq 0$.

Преобразуем данную систему к виду

$$x_i = - \sum_{j=2}^{i-1} \frac{a_{ij}}{a_{ii}} x_j - \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j + \frac{f_i}{a_{ii}}, \quad i = 1, 2, \dots, m. \quad (8.1)$$

Условимся считать значение суммы равным нулю, если верхний предел суммирования меньше нижнего. Так, уравнение (8.1) при $i = 1$ имеет вид:

$$x_1 = - \sum_{j=2}^m \frac{a_{1j}}{a_{11}} x_j + \frac{f_1}{a_{11}}.$$

В дальнейшем верхний индекс будет указывать номер итерации, например,

$$x^n = (x_1^n, x_2^n, \dots, x_m^n)^T,$$

где x_i^n — n -я итерация i -й компоненты вектора x . Число итераций ограничивают *критерием останова*: либо $n < n_{\max}$, либо задают малое ε и проверяют условие

$$\max_{1 \leq i \leq m} |x_i^{n+1} - x_i^n| < \varepsilon.$$

8.3 Метод Якоби

В *итерационном методе Якоби* исходят из записи системы в виде (8.1), причем итерации определяют формулой

$$x_i^{n+1} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^n - \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j^n + \frac{f_i}{a_{ii}}, \quad (8.2)$$

$$n = 0, 1, \dots, n_{\max}, \quad i = 1, 2, \dots, m,$$

где начальные значения $x_i^0, i = 1, 2, \dots, m$ заданы.

8.4 Метод Зейделя

Итерационный метод Зейделя определен формулой

$$x_i^{n+1} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{n+1} - \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j^n + \frac{f_i}{a_{ii}}, \quad (8.3)$$

$$n = 0, 1, \dots, n_{\max}, \quad i = 1, 2, \dots, m,$$

где начальные значения $x_i^0, i = 1, 2, \dots, m$ заданы.

Для наглядности запишем первые два уравнения системы (8.3):

$$x_1^{n+1} = - \sum_{j=2}^m \frac{a_{1j}}{a_{11}} x_j^n + \frac{f_1}{a_{11}}, \quad (8.4)$$

$$x_2^{n+1} = - \frac{a_{21}}{a_{22}} x_1^{n+1} - \sum_{j=2}^m \frac{a_{2j}}{a_{22}} x_j^n + \frac{f_2}{a_{22}}. \quad (8.5)$$

Первую компоненту x_1^{n+1} вектора x^{n+1} находят из (8.4). Для этого используют вектор x^n и значение f_1 . Для вычисления x_2^{n+1} по выражению (8.5) используют только что найденное значение x_1^{n+1} и известные значения $x_j^n, j = 3, \dots, m$ от предыдущей итерации. Таким образом, компоненты x_i^{n+1} вектора x^{n+1} находят из уравнения (8.3) последовательно, начиная с $i = 1$.

8.5 Матричная запись методов Якоби и Зейделя

Сопоставительный анализ итерационных методов упрощается, если записать их не в координатной, а в матричной форме. Представим матрицу A в виде суммы трех матриц

$$A = A_1 + D + A_2, \quad (8.6)$$

где $D = \text{diag} [a_{11}, a_{22}, \dots, a_{mm}]$; — диагональная матрица с той же главной диагональю, что и матрица A , матрица A_1 — нижняя треугольная и матрица A_2 — верхняя треугольная, обе с нулевыми главными диагоналями. Используя «расщепление» (8.6), перепишем систему $Ax = f$ в виде

$$x = -D^{-1}A_1x - D^{-1}A_2x + D^{-1}f.$$

Отсюда видно, что метод Якоби (8.2) представлен формулой

$$x^{n+1} = -D^{-1}A_1x^n - D^{-1}A_2x^n + D^{-1}f,$$

или, что то же самое, формулой

$$Dx^{n+1} + (A_1 + A_2)x^n = f, \quad (8.7)$$

а метод Зейделя (8.3) — формулой

$$x^{n+1} = -D^{-1}A_1x^{n+1} - D^{-1}A_2x^n + D^{-1}f,$$

или, что то же самое, формулой

$$(D + A_1)x^{n+1} + A_2x^n = f. \quad (8.8)$$

Учитывая (8.6), методы (8.7) и (8.8) запишем, соответственно, в виде

$$D(x^{n+1} - x^n) + Ax^n = f, \quad (8.9)$$

$$(D + A_1)(x^{n+1} - x^n) + Ax^n = f. \quad (8.10)$$

Эта запись показывает, что если итерационный метод сходится, то он сходится к решению исходной системы уравнений.

В методы (8.9) и (8.10) можно ввести итерационный параметр τ_{n+1} следующим образом:

$$D \frac{x^{n+1} - x^n}{\tau_{n+1}} + Ax^n = f,$$

$$(D + A_1) \frac{x^{n+1} - x^n}{\tau_{n+1}} + Ax^n = f.$$

Приведенные выше методы Якоби и Зейделя относятся к *одношаговым итерационным методам*. В таких методах для нахождения x^{n+1} используют одну предыдущую итерацию x^n . *Многошаговые итерационные методы* определяют x^{n+1} через значения x^k на двух и более предыдущих итерациях, так что l -шаговый ИМ выглядит как некоторая зависимость $x^{n+1} = \varphi(x^n, \dots, x^{n-l+1})$.

8.6 Каноническая форма одношаговых ИМ

Канонической формой одношагового итерационного метода для решения системы $Ax = f$ называют его представление в виде

$$B_{n+1} \frac{x^{n+1} - x^n}{\tau_{n+1}} + Ax^n = f, \quad n = 0, 1, \dots, n_{\max}. \quad (8.11)$$

Здесь B_{n+1} — *лидирующая матрица*, задающая тот или иной итерационный метод, τ_{n+1} — *итерационный параметр*. Предполагается, что дано начальное приближение x^0 и что существуют B_{n+1} и τ_{n+1} , $n = 0, 1, \dots, n_{\max}$. Тогда из уравнения (8.11) можно последовательно определить все x^{n+1} , $n = 0, 1, \dots, n_{\max}$.

Для нахождения x^{n+1} по известным f и x^n достаточно решить систему

$$B_{n+1}x^{n+1} = F_n$$

с правой частью $F_n = (B_{n+1} - \tau_{n+1}A)x^n + \tau_{n+1}f$.

Стационарный ИМ определяется выполнением двух условий: $B_{n+1} = B = \text{const}$ и $\tau_{n+1} = \tau = \text{const}$; иначе имеем *нестационарный ИМ*.

8.7 Методы простой итерации, Ричардсона и Юнга

Методом простой итерации называют явный метод

$$\frac{x^{n+1} - x^n}{\tau} + Ax^n = f \quad (8.12)$$

с постоянным параметром τ . Явный метод

$$\frac{x^{n+1} - x^n}{\tau_{n+1}} + Ax^n = f \quad (8.13)$$

с переменным τ_{n+1} есть *итерационный метод Ричардсона*. Юнг усовершенствовал метод Зейделя, введя в него параметр $\omega > 0$. Этот метод получил название *метод верхней релаксации*:

$$(D + \omega A_1) \frac{x^{n+1} - x^n}{\omega} + Ax^n = f. \quad (8.14)$$

Расчетная схема для (8.14) с учетом (8.6) получается в виде

$$(I + \omega D^{-1} A_1) x^{n+1} = ((1 - \omega)I - \omega D^{-1} A_2) x^n + \omega D^{-1} f.$$

В покомпонентной записи имеем

$$x_i^{n+1} + \omega \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{n+1} = (1 - \omega) x_i^n - \omega \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j^n + \omega \frac{f_i}{a_{ii}}, \quad i = 1, 2, \dots, m.$$

Последовательно, начиная с $i = 1$, находим все x_i^{n+1} , например, первые два:

$$x_1^{n+1} = (1 - \omega) x_1^n - \omega \sum_{j=2}^m \frac{a_{1j}}{a_{11}} x_j^n + \omega \frac{f_1}{a_{11}},$$

$$x_2^{n+1} = -\omega \frac{a_{21}}{a_{11}} x_1^{n+1} + (1 - \omega) x_2^n - \omega \sum_{j=3}^m \frac{a_{2j}}{a_{22}} x_j^n + \omega \frac{f_2}{a_{22}}.$$

8.8 Сходимость итерационных методов

Запишем *стационарный одношаговый итерационный метод* (8.11) в терминах погрешности $z^n = x^n - x$:

$$B \frac{z^{n+1} - z^n}{\tau} + Az^n = 0, \quad n = 0, 1, \dots, \quad z^0 = x^0 - x. \quad (8.15)$$

ТЕОРЕМА 8.1 ([12]). Пусть $A = A^T > 0$, $\tau > 0$ и выполнено неравенство

$$B - 0.5\tau A > 0. \quad (8.16)$$

Тогда итерационный метод (8.11) сходится.

СЛЕДСТВИЕ 8.1. Пусть $A = A^T > 0$. Тогда метод Якоби сходится [12], если A — матрица с диагональным преобладанием, т. е. при условии

$$a_{ii} > \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, \dots, m. \quad (8.17)$$

СЛЕДСТВИЕ 8.2. Пусть $A = A^T > 0$. Тогда метод верхней релаксации

$$(D + \omega A_1) \frac{x^{n+1} - x^n}{\omega} + Ax^n = f$$

сходится при условии $0 < \omega < 2$. В частности, метод Зейделя сходится [12].

СЛЕДСТВИЕ 8.3. Пусть $A = A^T > 0$. Тогда для метода простой итерации

$$\frac{x^{n+1} - x^n}{\tau} + Ax^n = f$$

необходимым и достаточным условием сходимости является неравенство

$$0 < \tau < 2/\lambda_{\max},$$

где λ_{\max} — наибольшее по абсолютному значению собственное число матрицы A , называемое также *спектральным радиусом* $\rho(A)$ матрицы A [12].

Сходимость итерационного метода (8.11) означает, что $z^n \rightarrow 0$ в некоторой норме при $n \rightarrow \infty$. Перепишав уравнение (8.15), получим:

$$z^{n+1} = Sz^n, \quad n = 0, 1, \dots, \quad (8.18)$$

где

$$S = I - \tau B^{-1}A \quad (8.19)$$

называют *переходной матрицей погрешности* от n -й итерации к $(n+1)$ -й.

ТЕОРЕМА 8.2 ([12]). Итерационный метод

$$B \frac{x^{n+1} - x^n}{\tau} + Ax^n = f, \quad n = 0, 1, \dots, \quad (8.20)$$

сходится при любом начальном приближении тогда и только тогда, когда все собственные значения матрицы (8.19) по модулю меньше единицы.

8.9 Скорость сходимости итерационных методов

Теорема 8.2 о сходимости имеет принципиальное значение и накладывает минимальные ограничения на матрицы A и B . Однако ее непосредственное применение к конкретным итерационным методам не всегда возможно, так как исследование спектра матрицы (8.19) является более трудоемкой задачей, чем решение исходной системы $Ax = f$.

Будем рассматривать решение x системы и последовательные приближения x^n как элементы евклидова пространства, а матрицы A , B и другие — как операторы, действующие в нем.

ЗАМЕЧАНИЕ 8.1. Для двух симметрических матриц A и B неравенство $A \geq B$ означает, что $(Ax, x) \geq (Bx, x)$ для всех $x \in \mathbb{E}$. В случае некоторой симметрической положительно определенной матрицы D будем пользоваться *обобщенной нормой вектора* $\|y\|_D = \sqrt{(Dy, y)}$.

ТЕОРЕМА 8.3 ([12]). Пусть A и B — симметрические положительно определенные матрицы, для которых справедливы неравенства

$$\gamma_1 B \leq A \leq \gamma_2 B, \quad (8.21)$$

где γ_1, γ_2 — положительные константы и $\gamma_1 < \gamma_2$. При $\tau = 2/(\gamma_1 + \gamma_2)$ итерационный метод (8.20) сходится, и для погрешности справедливы оценки:

$$\|x^n - x\|_A \leq \rho^n \|x^0 - x\|_A, \quad n = 0, 1, \dots,$$

$$\|x^n - x\|_B \leq \rho^n \|x^0 - x\|_B, \quad n = 0, 1, \dots,$$

где

$$\rho = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\gamma_1}{\gamma_2}. \quad (8.22)$$

Пусть

$$A\mu = \lambda B\mu. \quad (8.23)$$

Тогда

$$\gamma_1(B\mu, \mu) \leq (A\mu, \mu) = \lambda(B\mu, \mu) \leq \gamma_2(B\mu, \mu)$$

и

$$\gamma_1 \leq \lambda_{\min}(B^{-1}A), \quad \gamma_2 \geq \lambda_{\max}(B^{-1}A), \quad (8.24)$$

где $\lambda_{\min}(B^{-1}A)$ и $\lambda_{\max}(B^{-1}A)$ — минимальное и максимальное по абсолютному значению собственные числа в *обобщенной задаче* (8.23) *на собственные значения*.

Таким образом, наиболее точными константами, с которыми выполняются неравенства (8.21), являются константы

$$\gamma_1 = \lambda_{\min}(B^{-1}A), \quad \gamma_2 = \lambda_{\max}(B^{-1}A).$$

В этом случае параметр

$$\tau = \frac{2}{\lambda_{\min}(B^{-1}A) + \lambda_{\max}(B^{-1}A)} \quad (8.25)$$

называется *оптимальным итерационным параметром*, минимизирующим

$$\rho = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\gamma_1}{\gamma_2}$$

на множестве всех положительных γ_1, γ_2 , удовлетворяющих условиям (8.24).

В случае метода простой итерации ($B = I$) получаем два следствия.

СЛЕДСТВИЕ 8.4. Если $A^T = A > 0$, то для метода простой итерации

$$\frac{x^{n+1} - x^n}{\tau} + Ax^n = f$$

при

$$\tau = \tau_0 = \frac{2}{\lambda_{\min}(A) + \lambda_{\max}(A)}$$

справедлива оценка

$$\|x^n - x\| \leq \rho_0^n \|x^0 - x\|,$$

где [12]

$$\rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}.$$

СЛЕДСТВИЕ 8.5. Для симметрической матрицы A и $\tau_0 = 2/(\lambda_{\min}(A) + \lambda_{\max}(A))$ справедливо равенство

$$\|I - \tau_0 A\| = \rho_0,$$

где [12]

$$\rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}.$$

В приложениях часто встречаются задачи с *плохо обусловленной матрицей* A , когда соотношение $\lambda_{\max}(A)/\lambda_{\min}(A)$ велико. В этом случае число ρ_0 близко к единице, и метод простой итерации сходится медленно. Число итераций $n_0(\varepsilon)$, которое требуется в случае малых ξ для достижения заданной точности ε , т. е. для достижения оценки

$$\|x^n - x\| \leq \varepsilon \|x^0 - x\|,$$

получается из условия $\rho_0^n < \varepsilon$ в виде $n \geq n_0(\varepsilon)$, где

$$n_0(\varepsilon) = \frac{\ln(1/\varepsilon)}{\ln(1/\rho_0)}.$$

Отсюда при малых ξ имеем

$$n_0(\varepsilon) \approx \frac{\ln(1/\varepsilon)}{2\xi} = O\left(\frac{1}{\xi}\right).$$

Это свидетельствует о том, что метод простой итерации в случае малых ξ является медленно сходящимся методом. Ускорить сходимость можно двумя способами: применяя неявный итерационный метод и/или делая $\tau = \tau_{n+1}$ зависящим от номера итерации.

8.10 Итерационные методы вариационного типа

Найти минимальное и максимальное по абсолютному значению собственные числа в обобщенной задаче (8.23) на собственные значения бывает сложно, а без них невозможно задать наилучшее значение итерационного параметра (8.25). В таких случаях можно использовать другой класс итерационных методов — *методы вариационного типа*. Здесь на каждой итерации

$$B \frac{x^{k+1} - x^k}{\tau_{k+1}} + Ax^k = f, \quad (8.26)$$

для параметра τ_{k+1} выбирают то значение, которое минимизирует предопределенный критерий качества, связанный с погрешностью $\|x^{k+1} - x\|_D$, при условии, что предыдущая итерация уже состоялась с погрешностью $\|x^k - x\|_D$. В зависимости от выбора матриц D и B получают различные методы этого типа.

Метод минимальных невязок

Рассмотрим уравнение $Ax = f$ с $A = A^T > 0$. Разность

$$r_k = Ax^k - f, \quad (8.27)$$

которая получается при подстановке приближенного значения x^k в это уравнение, называют *невязкой*. Погрешность $z_k = x^k - x$ и невязка r_k связаны равенством $Az_k = r_k$. Представим *явный итерационный метод*

$$\frac{x^{k+1} - x^k}{\tau_{k+1}} + Ax^k = f \quad (8.28)$$

в виде

$$x^{k+1} = x^k - \tau_{k+1} r_k. \quad (8.29)$$

Метод минимальных невязок есть метод (8.28), в котором параметр τ_{k+1} минимизирует $\|r_{k+1}\|$ при заданной норме $\|r_k\|$ невязки текущего шага. Найдем это значение. Из (8.29) получаем:

$$\begin{aligned} Ax^{k+1} &= Ax^k - \tau_{k+1}Ar_k, \\ r_{k+1} &= r_k - \tau_{k+1}Ar_k. \end{aligned} \quad (8.30)$$

Возводя обе части уравнения (8.30) скалярно в квадрат, получим

$$\|r_{k+1}\|^2 = \|r_k\|^2 - 2\tau_{k+1}(r_k, Ar_k) + \tau_{k+1}^2\|Ar_k\|^2.$$

Отсюда видно, что $\|r_{k+1}\|$ достигает минимума при

$$\tau_{k+1} = \frac{(Ar_k, r_k)}{\|Ar_k\|^2}. \quad (8.31)$$

Таким образом, в методе минимальных невязок переход от k -й итерации к $(k+1)$ -й осуществляется по следующему алгоритму:

по найденному значению x^k вычисляют вектор невязки $r_k = Ax^k - f$,
по формуле (8.31) находят параметр τ_{k+1} ,
по формуле (8.29) определяют вектор x^{k+1} .

ТЕОРЕМА 8.4 ([12]). Пусть A — симметрическая положительно определенная матрица. Для погрешности метода минимальных невязок справедлива оценка

$$\|A(x^n - x)\| \leq \rho_0^n \|A(x^0 - x)\|, \quad n = 0, 1, \dots,$$

где

$$\rho = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}.$$

Иными словами, метод минимальных невязок сходится с той же скоростью, что и метод простой итерации с оптимальным параметром τ .

Метод минимальных поправок

Запишем неявный итерационный метод (8.26) в виде

$$x^{k+1} = x^k - \tau B^{-1}r_k,$$

где $r_k = Ax^k - f$ — невязка. Вектор $\omega_k = B^{-1}r_k$ называют *поправкой итерационного метода* на $(k+1)$ -й итерации. Поправка ω_k удовлетворяет тому же уравнению, что и погрешность $z_k = x^k - x$ неявного метода, т. е. уравнению

$$B \frac{\omega_{k+1} - \omega_k}{\tau_{k+1}} + A\omega_k = 0. \quad (8.32)$$

Пусть B — симметрическая положительно определенная матрица. Тогда *метод минимальных поправок* — это метод (8.26), в котором параметр τ_{k+1} минимизирует норму $\|\omega_{k+1}\|_B = (B\omega_{k+1}, \omega_{k+1})^{1/2}$ при ранее полученном векторе ω^k . В случае $B = I$ метод минимальных поправок совпадает с методом минимальных невязок.

Перепишем (8.32) в виде

$$\omega_{k+1} = \omega_k - \tau_{k+1} B^{-1} A \omega_k$$

и вычислим

$$\|\omega_{k+1}\|_B^2 = \|\omega_k\|_B^2 - 2\tau_{k+1}(A\omega_k, \omega_k) + \tau_{k+1}^2(B^{-1}A\omega_k, A\omega_k) .$$

Минимум $\|\omega_{k+1}\|_B^2$ достигается, если и только если

$$\tau_{k+1} = \frac{(A\omega_k, \omega_k)}{(B^{-1}A\omega_k, A\omega_k)} . \quad (8.33)$$

Для реализации метода минимальных поправок требуется на каждой итерации решать систему уравнений $B\omega_k = r_k$ относительно поправки ω_k и затем решать систему уравнений $Bv_k = A\omega_k$, откуда находят вектор $v_k = B^{-1}A\omega_k$, необходимый для вычисления параметра τ_{k+1} .

ТЕОРЕМА 8.5 ([12]). Пусть A и B — симметрические положительно определенные матрицы и $\lambda_{\min}(B^{-1}A)$, $\lambda_{\max}(B^{-1}A)$ — наименьшее и наибольшее собственные значения в задаче $Ax = \lambda Bx$. Для погрешности метода минимальных поправок справедлива оценка

$$\|A(x^n - x)\|_{B^{-1}} \leq \rho_0^n \|A(x^0 - x)\|_{B^{-1}} , \quad n = 0, 1, \dots ,$$

где

$$\rho_0 = \frac{1 - \xi}{1 + \xi} , \quad \xi = \frac{\lambda_{\min}(B^{-1}A)}{\lambda_{\max}(B^{-1}A)} .$$

Метод скорейшего спуска

Возьмем явный метод (8.13) и выберем итерационный параметр τ_{k+1} из условия минимума $\|z_{k+1}\|_A$ при заданном векторе z_k , где $z_{k+1} = x^{k+1} - x$. Поскольку погрешность z_k удовлетворяет уравнению

$$z_{k+1} = z_k - \tau_{k+1} A z_k ,$$

имеем

$$\|z_{k+1}\|_A^2 = \|z_k\|_A^2 - 2\tau_{k+1}(Az_k, Az_k) + \tau_{k+1}^2(A^2z_k, Az_k) .$$

Минимум нормы $\|z_{k+1}\|_A^2$ достигается при $\tau_{k+1} = \frac{(Az_k, Az_k)}{(A^2 z_k, Az_k)}$.

Величина $z_k = x^k - x$ неизвестна, но $Az_k = r_k = Ax^k - f$. Поэтому вычисление τ_{k+1} проводят по формуле

$$\tau_{k+1} = \frac{(r_k, r_k)}{(Ar_k, r_k)}.$$

ТЕОРЕМА 8.6 ([12]). Для погрешности явного метода скорейшего спуска справедлива оценка

$$\|x^n - x\|_A \leq \rho_0^n \|x^0 - x\|_A, \quad n = 0, 1, \dots,$$

где

$$\rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}.$$

Если вместо (8.13) взять неявный метод (8.26) и параметр τ_{k+1} выбирать из условия минимума $\|z_{k+1}\|_A$, то получим *неявный метод наискорейшего спуска*. Для него

$$\|z_{k+1}\|_A^2 = \|z_k\|_A^2 - 2\tau_{k+1}(Az_k, B^{-1}Az_k) + \tau_{k+1}^2(AB^{-1}Az_k, B^{-1}Az_k),$$

или

$$\|z_{k+1}\|_A^2 = \|z_k\|_A^2 - 2\tau_{k+1}(r_k, \omega_k) + \tau_{k+1}^2(A\omega_k, \omega_k).$$

Следовательно, норма $\|z_{k+1}\|_A^2$ будет минимальной при

$$\tau_{k+1} = \frac{(r_k, \omega_k)}{(A\omega_k, \omega_k)}.$$

ТЕОРЕМА 8.7 ([12]). Для неявного метода скорейшего спуска справедлива оценка

$$\|x^n - x\|_A \leq \rho_0^n \|x^0 - x\|_A, \quad n = 0, 1, \dots,$$

где

$$\rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\lambda_{\min}(B^{-1}A)}{\lambda_{\max}(B^{-1}A)}.$$

Метод сопряженных градиентов

Этот метод исходит из задачи минимизации функции

$$J(x) = \frac{1}{2}(Ax, x) - (b, x), \quad (8.34)$$

решение которой совпадает с решением системы

$$Ax = f, \quad A = A^T > 0. \quad (8.35)$$

Полный вывод *метода сопряженных градиентов* можно найти в [12]. Опуская детали, приведем окончательный результат.

Метод сопряженных градиентов для решения системы $Ax = f$ состоит в вычислениях по следующим формулам:

$$\left. \begin{aligned} r^k &= b - Ax^k, \quad k = 0, 1, \dots, \\ p^{k+1} &= r^k + \beta_{k+1}p^k, \quad k = 1, 2, \dots, \quad p^1 = r^0, \\ x^{k+1} &= x^k + \alpha_{k+1}p^{k+1}, \quad k = 0, 1, \dots, \quad x^0 = 0, \\ \alpha_{k+1} &= (r^k, p^{k+1})/(p^{k+1}, Ap^{k+1}), \quad k = 0, 1, \dots, \\ \beta_{k+1} &= -(Ap^k, r^k)/(Ap^k, p^k), \quad k = 1, 2, \dots \end{aligned} \right\} \quad (8.36)$$

ТЕОРЕМА 8.8 ([12]). Для метода сопряженных градиентов (8.36) справедливо

$$\|x^k - x\|_A \leq 2 \left[(1 - \sqrt{\lambda_{\min}/\lambda_{\max}})/(1 + \sqrt{\lambda_{\min}/\lambda_{\max}}) \right]^k \|x\|_A,$$

где λ_{\min} и λ_{\max} — минимальное и максимальное собственные значения матрицы A .

Следуя [12], преобразуем соотношения (8.36). В этих соотношениях наиболее трудоемкими являются две операции: вычисление векторов Ax^k и Ap^k . Однако операцию вычисления вектора Ax^k можно исключить. Поскольку этот вектор нужен только для вычисления невязки r^k , то можно заменить первую из формул (8.36) на

$$r^k = r^{k-1} - \alpha_k Ap^k, \quad k = 1, 2, \dots, \quad r^0 = b. \quad (8.37)$$

Преобразуем формулы для вычисления параметров α_{k+1} и β_{k+1} . Подставляя второе из соотношений (8.36) в четвертое, найдем

$$\alpha_{k+1} = (r^k, r^k)/(p^{k+1}, \alpha p^{k+1}), \quad k = 0, 1, \dots \quad (8.38)$$

Заменяя здесь $k + 1$ на k и подставляя полученное выражение для (p^k, Ap^k) в последнее из соотношений (8.36), получим

$$\beta_{k+1} = -\alpha_k \frac{(Ap^k, r^k)}{(r^{k-1}, r^{k-1})}.$$

Теперь подставим сюда вместо Ap^k его выражение из (8.37).

ТЕОРЕМА 8.9 ([12]). После k шагов метода сопряженных градиентов невязки r^0, r^1, \dots, r^k взаимно ортогональны.

Принимая это во внимание, найдем

$$\beta_{k+1} = \frac{(r^k, r^k)}{(r^{k-1}, r^{k-1})}, \quad k = 1, 2, \dots. \quad (8.39)$$

С учетом (8.37)–(8.39) формулы метода сопряженных градиентов (8.36) преобразуются к виду

$$\left. \begin{aligned} r^k &= r^{k-1} - \alpha_k Ap^k, \quad k = 1, 2, \dots, \quad r^0 = b, \\ p^{k+1} &= r^k + \beta_{k+1} p^k, \quad k = 1, 2, \dots, \quad p^1 = r^0, \\ x^{k+1} &= x^k + \alpha_{k+1} p^{k+1}, \quad k = 0, 1, \dots, \quad x^0 = 0, \\ \alpha_{k+1} &= \|r^k\|^2 / (p^{k+1}, Ap^{k+1}), \quad k = 0, 1, \dots, \\ \beta_{k+1} &= \|r^k\|^2 / \|r^{k-1}\|^2, \quad k = 1, 2, \dots. \end{aligned} \right\} \quad (8.40)$$

Легко проверить, что эти вычисления проводят в следующем порядке:

$$\begin{aligned} r^0 &= b, \quad p^1 = r^0, \quad Ap^1, \quad \alpha_1, \quad x^1, \\ r^1, \quad \beta_2, \quad p^2, \quad Ap^2, \quad \alpha_2, \quad x^2, \quad \dots \end{aligned}$$

8.11 Другие методы

Область итерационных методов решения систем линейных алгебраических уравнений обширна. Она включает гораздо большее количество методов, чем то, что приведено выше.

В итерационных методах нашли применение полиномы Чебышёва, благодаря которым можно решать задачу оптимального выбора итерационных параметров как для явных ИМ, так и для неявных ИМ [12].

Стационарные методы, широко применявшиеся в 1950–1980 годах, сейчас чаще применяются [126] как средство сглаживания в многосеточных алгоритмах [102, 118, 144] или для предобуславливания в алгоритмах Крылова [139].

Идея сопряженных градиентов [136] оказалась очень плодотворной, и наиболее широкое воплощение она нашла при опоре на метод подпространств Крылова, который является одним из методов решения проблемы собственных значений и собственных векторов для очень больших разреженных матриц [143]. Переход к методу подпространств Крылова в этой проблеме вызван тем, что преобразования подобия, лежащие в основе ее решения для небольших матриц, выполнять для очень больших матриц практически невозможно. В то же время достаточно легко выполнять однотипные операции умножения матрицы на вектор: взять вектор x и затем, умножая слева на A , построить последовательность Крылова x, Ax, A^2x, A^3x, \dots и, соответственно, получить *пространства Крылова*

$$\mathcal{K}_j(A, x) = \text{span} \{x, Ax, A^2x, A^3x, \dots, A^{j-1}x\}.$$

В настоящее время *алгоритмы Крылова с предобуславливанием* применяются в большинстве итерационных методов решения больших разреженных линейных систем [126]. Успешной альтернативой методам Крылова являются *многосеточные методы*, по которым за последние 30–40 лет появилось огромное число публикаций [102, 118, 126, 144].

Вместе с этими мощными ветвями роста, в практике решения линейных систем итерационными методами встречаются решения, которые могут быть классифицированы как *Inventive Math*. Это решения, по которым пока не найдено строгих доказательств, но которые подтверждают свою работоспособность методом широкого вычислительного эксперимента. Примером такого подхода является метод *делинеаризации* для линейных систем [140, 141, 142].

8.12 Задание на лабораторный проект № 7

Написать и отладить программу, реализующую ваш вариант задания в соответствии с табл. 8.1 (см. ниже стр. 159), включающий два итерационных метода для численного решения систем линейных алгебраических уравнений $Ax = f$ с квадратной матрицей A и отыскания обратной матрицы A^{-1} . Предусмотреть сообщение о невозможности решения указанных задач из-за превышения допустимого числа итераций. Отделить основные части программы:

- а) подпрограмму решения систем линейных алгебраических уравнений;
- б) подпрограмму обращения матриц;

в) сервисные подпрограммы.

Уделить особое внимание эффективности программы (в смысле экономии оперативной памяти и скорости решения указанных выше задач). Предусмотреть пошаговое выполнение алгоритма с выводом x^k на каждой итерации (для тестовой задачи небольшой размерности, см. ниже п. 4).

В качестве ε (см. критерий остановки в подразд. 8.2) для обоих итерационных методов использовать погрешность решения данной системы линейных алгебраических уравнений (СЛАУ) методом исключения переменных из лабораторной работы № 1.

Выполнить следующие пункты задания:

1. Провести подсчет фактического количества операций умножения и деления, выполняемых при решении системы линейных алгебраических уравнений с выводом результата на экран. Сравнить с методом исключения переменных из лабораторной работы № 1. Вывести таблицу и график.

2. Оценить скорость решения задач, т. е. определить время, затраченное на решение СЛАУ, и время, затраченное на обращение матриц. Для этого спроектировать и провести эксперимент, который охватывает матрицы порядка от 10 до 200 (через 10 порядков). Представить результаты в виде таблицы и графика зависимости времени выполнения от порядка матриц для трех алгоритмов (двух итерационных методов, соответствующих варианту, и методу исключения переменных из лабораторной работы № 1). Таблицу и графики вывести на экран.

3. Оценить точность решения систем линейных алгебраических уравнений, указанных в п. 2. Для этого сгенерировать случайные матрицы A , задать точное решение x^* и образовать правые части $f = Ax^*$. Провести анализ точности вычисленного решения x от порядка матрицы для трех алгоритмов (аналогично п. 2). В качестве точного решения взять вектор $x^* = (1, 2, \dots, m)^T$, где m — порядок матрицы. Для оценки точности решения использовать норму вектора

$$\|x\| = \max_i |x_i|.$$

Результаты по п. 3 представить в виде таблицы и графиков.

ЗАМЕЧАНИЕ 8.2. Для проведения вычислительного эксперимента по пп. 2 и 3 применять симметрические положительно определенные матрицы A с диагональным преобладанием. Для заполнения матрицы A использовать случайные числа из диапазона от -100 до 100 . Сначала заполнить нижнюю треугольную часть матрицы A , т. е. элементы a_{ij} , где $i > j$. Верхнюю

треугольную часть, где $i < j$, заполнить симметрично нижней части. Затем заполнить диагональ. В качестве диагонального элемента a_{ii} , $i = 1, 2, \dots, m$, выбрать случайное число из интервала

$$\left[\sum_{j \neq i} |a_{ij}| + 1, \sum_{j \neq i} |a_{ij}| + 101 \right],$$

чтобы обеспечить выполнение условия

$$a_{ii} \geq \sum_{j \neq i} |a_{ij}| + 1,$$

гарантирующего положительную определенность матрицы A .

4. До проведения вычислительного эксперимента по пп. 2 и 3 выполнить отладку программы. Для отладки программы, а также для сравнительного тестирования двух заданных итерационных методов использовать следующую тестовую задачу [99]:

$$A = \begin{bmatrix} 4 & 0 & 1 & 1 \\ 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 1 & 1 & 0 & 4 \end{bmatrix}, \quad f = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad x^* = \begin{bmatrix} -41/209 = -0.196172 \\ 53/209 = 0.253589 \\ 167/209 = 0.799043 \\ 206/209 = 0.985646 \end{bmatrix}.$$

5. Для тех вариантов задания, в которых присутствует метод Юнга (верхней релаксации), провести специальный вычислительный эксперимент решения тестовой задачи по п. 4. Цель этого эксперимента — исследование скорости сходимости метода в зависимости от коэффициента релаксации ω в формуле (8.14).

Изменение параметра ω организовать с шагом $\Delta\omega = 0.05$ равномерно в интервале теоретической сходимости метода: $0 < \omega < 2$, т. е. по алгоритму:

$\omega_0 := \omega_{\text{start}}$ Для $i = 0, 1, \dots, 20$ выполнять $\omega_{i+1} := \omega_i + \Delta\omega$

Стартовое значение задавать с клавиатуры, например, $\omega_{\text{start}} = 0.50$.

6. Повторить п. 3 задания для плохо обусловленных матриц (см. подразд. 2.6 лабораторной работы № 1), имеющих порядок от 4 до 40.

Таблица 8.1. Варианты задания на лабораторный проект № 7

Варианты итерационных методов	a	b	c	d	e	f	g
a	-	13	14	1	2	3	4
b	-	-	15	5	6	7	8
c	-	-	-	9	10	11	12

a — метод Якоби;

b — метод Зейделя;

c — метод Юнга;

d — метод минимальных невязок;

e — метод минимальных поправок;

f — метод скорейшего спуска;

g — метод сопряженных градиентов.

7. Вычислить матрицу A^{-1} двумя способами:

1) через решение системы $AX = I$ на основе метода исключения Гаусса из лабораторной работы № 1 (в соответствии со своим вариантом);

2) через решение системы $AX = I$ на основе любого из двух заданных итерационных методов.

Сравнить затраты машинного времени и точность обращения способами 1) и 2). Эксперименты провести для матриц порядков от 10 до 100 через 10, сгенерированных согласно замечанию 8.2. Для оценки точности в обоих способах воспользоваться формулой из лабораторной работы (проекта) № 1.

8.13 Варианты задания на лабораторный проект № 7

По теме «Итерационные методы» студентам предлагается 15 вариантов лабораторной работы № 7, сведенных в табл. 8.1.

Если нет других указаний преподавателя, выбирайте ваш вариант из табл. 8.1 по вашему номеру в журнале студенческой группы.

ФОНД ЗАДАЧ

Целью настоящего раздела является формирование и проверка у студентов, изучающих курс «Численные методы», базовых навыков в области решения задач вычислительной линейной алгебры. Предлагаемые в этом разделе задачи охватывают широкий спектр методов: гауссово исключение переменных, итерационные методы решения систем линейных алгебраических уравнений, включая методы вариационного типа, факторизацию положительно определенных матриц и ортогональные преобразования. Приводимые ниже задачи могут быть использованы как для практических занятий и контрольных работ в аудитории, так и для самостоятельной работы студента, а также для проверки практических навыков студентов во время экзамена. Данный материал позволяет не только проверить знание базовых алгоритмов в области вычислительной линейной алгебры, но и определить уровень владения вычислительной техникой для решения тех или иных прикладных задач. Большое разнообразие и количество задач создает возможность формирования индивидуального задания для каждого студента.

9.1 Типовые задачи

Начнем прежде всего, с разбора типовых задач. В соответствии с вышесказанным настоящее учебное пособие содержит задачи по следующим пяти темам: метод исключения Гаусса, итерационные методы, итерационные методы вариационного типа, разложение Холецкого для симметричных положительно определенных матриц и методы ортогонального приведения.

Задача 1 (см. ниже) является типичным представителем задач на метод Гаусса исключения переменных. Целью задачи является проверка знания базовых алгоритмов для разложения невырожденной матрицы в произведение нижней и верхней треугольных матриц, а также для решения системы

линейных уравнений с разложенной матрицей коэффициентов и обращения матрицы. Разнообразие задач достигается за счет вовлечения различных вариантов разложения (см. лабораторную работу № 1, подразд. 2.8) и использования разных исходных матриц.

Задача 1

Для матрицы

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 6 & 2 & 1 \\ -2 & -2 & -1 \end{pmatrix}$$

выполнить следующее:

- а. Построить $\bar{L}U$ -разложение матрицы A (\bar{L} с единицами на главной диагонали).
- б. С помощью $\bar{L}U$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (0, 3, 1)^T$.

- в. С помощью $\bar{L}U$ -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 2 является типичным представителем задач на итерационные методы решения систем линейных алгебраических уравнений. Целью задачи является проверка знания базовых итерационных алгоритмов, а также необходимых и достаточных условий их сходимости и критериев для оценки точности решения. Разнообразие задач достигается за счет вовлечения различных итерационных методов, изучаемых в курсе «Численные методы», и использования разных исходных матриц.

Задача 2

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 5 & -1 & 0 \\ -1 & 4 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

и вектор $b = (4, 2, -1)^T$, выполнить следующее:

- а.** Сформулировать метод Зейделя в координатном и каноническом виде.
- б.** Определить является ли он сходящимся с нулевым начальным приближением, т. е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.
- в.** Вычислить две итерации по методу Зейделя и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Следующая задача посвящена итерационным методам вариационного типа. Ее целью является проверка знания алгоритмов построения итерационных методов вариационного типа, формул для вычисления оптимального итерационного параметра и критериев для оценки точности решения. Разнообразие задач достигается за счет вовлечения различных способов построения итерационных методов вариационного типа, изучаемых в курсе «Численные методы», и использования разных исходных матриц.

Задача 3

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 5 & -1 & 0 \\ -1 & 4 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

и вектор $b = (4, 2, -1)^T$, выполнить следующее:

- а.** На основе метода Зейделя сформулировать неявный метод скорейшего спуска в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т. е. $x^0 = (0, 0, 0)^T$?
- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 4 является типичным представителем задач на разложение Холецкого для симметричных положительно определенных матриц. Целью

задачи является проверка знания базовых алгоритмов Холецкого для разложения симметричной положительно определенной матрицы, а также способов решения системы линейных уравнений с разложенной матрицей коэффициентов и обращения матрицы. Разнообразие задач достигается за счет вовлечения различных вариантов разложения Холецкого (см. лабораторную работу № 5, подразд. 6.8) и использования разных исходных матриц.

Задача 4

Для матрицы

$$P = \begin{pmatrix} 18 & -10 & 3 & 10 \\ -10 & 105 & -8 & 25 \\ 3 & -8 & 1 & 0 \\ 10 & 25 & 0 & 25 \end{pmatrix}$$

выполнить следующее:

а. Построить UDU^T -разложение матрицы P (U — верхняя треугольная матрица с единицами на главной диагонали, D — диагональная матрица с положительными элементами на диагонали).

б. С помощью UDU^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (21, 112, -4, 60)^T$.

в. С помощью разложения и решения системы найти величину квадратичной формы $J(x) = x^T Px$, где x — решение из п.б.

Последняя задача этого раздела посвящена соответственно последней теме, а именно, — методам ортогонального разложения матриц. Целью настоящей задачи является проверка базовых знаний по алгоритмам ортогонального разложения матрицы, а также по способам решения системы линейных уравнений с разложенной матрицей коэффициентов и обращения матрицы. Разнообразие задач достигается за счет вовлечения различных вариантов ортогонального разложения (см. лабораторную работу № 6, подразд. 7.16) и использования разных исходных матриц.

Задача 5

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & -6 \\ -2 & 6 & -3 \\ -2 & 7 & 3 \end{pmatrix}$$

выполнить следующее:

- а. Построить QR -разложение матрицы A с помощью преобразований отражения (Хаусхолдера).
- б. С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-3, 1, 8)^T$.

- в. С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

9.2 Решения и рекомендации к типовым задачам

Задача 1

Решение.

- а. Используя метод исключения переменных Гаусса, нетрудно получить

$$\bar{L}U = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 2 \end{pmatrix}.$$

- б. Решая две линейные системы с треугольными матрицами, получаем

$$x = (1, -1, -1)^T.$$

- в. Три раза решая линейные системы с правой частью в виде столбцов единичной матрицы, получаем

$$A^{-1} = \begin{pmatrix} 0.00 & 0.25 & 0.25 \\ -1.00 & 0.00 & -1.00 \\ 2.00 & -0.50 & 0.50 \end{pmatrix}, \quad M_A = 9 \cdot 3 = 27.$$

Задача 2

Решение.

а. Метод Зейделя в координатном и каноническом виде:

$$\begin{aligned}x_1^{n+1} &= 0.2x_2^n + 0.8, \\x_2^{n+1} &= 0.25x_1^{n+1} - 0.25x_3^n + 0.5, \\x_3^{n+1} &= -0.5x_2^{n+1} - 0.5; \\(D + A_1)(x^{n+1} - x^n) + Ax^n &= b, \quad n = 0, 1, \dots,\end{aligned}$$

где

$$D + A_1 = \begin{pmatrix} 5 & 0 & 0 \\ -1 & 4 & 0 \\ 0 & 1 & 2 \end{pmatrix}.$$

б. Метод сходится. Для доказательства нужно воспользоваться теоремой о необходимом и достаточном условии сходимости одношагового стационарного итерационного метода.

в. Используя метод Зейделя в координатном виде, вычисляем необходимые итерации, а воспользовавшись формулой для апостериорной оценки точности, находим норму ошибки на этих итерациях.

$$\begin{aligned}x^1 &= (0.80, 0.70, -0.85)^T, \quad \|\Delta x^1\|_\infty \leq 0.36, \\x^2 &= (0.94, 0.95, -0.98)^T, \quad \|\Delta x^2\|_\infty \leq 0.1.\end{aligned}$$

Задача 3

Решение.

а. Неявный метод скорейшего спуска в каноническом виде:

$$B \frac{(x^{n+1} - x^n)}{\tau_{n+1}} + Ax^n = b, \quad n = 0, 1, \dots,$$

где $\tau_{n+1} = \frac{(r^n, w^n)}{(Aw^n, w^n)}$, а $w^n = B^{-1}r^n$ и $r^n = Ax^n - b$,

$$B = D + A_1 = \begin{pmatrix} 5 & 0 & 0 \\ -1 & 4 & 0 \\ 0 & 1 & 2 \end{pmatrix}.$$

- б. Используя формулу для вычисления оптимального итерационного параметра метода скорейшего спуска, получаем

$$\tau_1 = 1.27.$$

- в. Записывая метод скорейшего спуска в координатном виде, вычисляем первую итерацию для оптимального параметра, найденного в п. б., а воспользовавшись формулой для апостериорной оценки точности, находим норму ошибки на этой итерации.

$$x^1 = (1.02, 0.89, -1.08)^T, \quad \|\Delta x^1\|_\infty \leq 1.17.$$

Задача 4

Решение.

а.

$$U = \begin{pmatrix} 1 & 1/4 & 3 & 2/5 \\ 0 & 1 & -8 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 25 \end{pmatrix}.$$

- б. $x = (1, 1, 1, 1)^T$.

- в. $J(x) = 189$.

Задача 5

Решение.

а.

$$\bar{Q}R = \frac{1}{3} \begin{pmatrix} -1 & -2.8 & -0.4 \\ 2 & -0.4 & -2.2 \\ 2 & -1 & 2 \end{pmatrix} \begin{pmatrix} 3 & 8 & 2 \\ 0 & -5 & 5 \\ 0 & 0 & 5 \end{pmatrix}.$$

б.

$$x = (1, 1, 1)^T.$$

в.

$$A^{-1} = \frac{1}{75} \begin{pmatrix} 39 & -48 & 30 \\ 12 & -9 & 15 \\ -2 & -11 & 10 \end{pmatrix}, \quad M_A = 12 \cdot 117/75 = 18.72.$$

9.3 Варианты контрольных заданий

В этом подразделе приведены примеры того, как составляются варианты контрольных заданий для всеобъемлющей проверки базовых теоретических знаний и практических навыков по четырем основным темам курса «Численные методы (алгебры)»: решение систем уравнений методом исключения неизвестных, решение систем уравнений итерационными методами, факторизация положительно определенных матриц и ортогональные преобразования. Каждый из четырех представленных вариантов содержит четыре задания по этим темам. Реальное разнообразие вариантов достигается применением различных алгоритмов и исходных данных.

Вариант I

Задание 1. Для матрицы

$$A = \begin{pmatrix} -1 & 4 & 1 \\ 1 & -2 & 2 \\ -2 & 8 & 3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $\bar{L}U$ -разложение матрицы A (\bar{L} с единицами на главной диагонали).
- б.** С помощью $\bar{L}U$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-2, -1, -5)^T$.

- в.** С помощью $\bar{L}U$ -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 2. Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 5 & -1 \\ -6 & 20 & 4 \\ 2 & -3 & -10 \end{pmatrix}$$

и вектор $b = (-4, 22, 5)^T$, выполнить следующее:

- а. Выписать метод Якоби в координатном и каноническом виде.
- б. Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.
- в. Вычислить две итерации по методу Якоби и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 3. Для матрицы

$$P = \begin{pmatrix} 1 & 2 & -2 & 3 \\ 2 & 8 & 0 & 8 \\ -2 & 0 & 17 & -10 \\ 3 & 8 & -10 & 15 \end{pmatrix}$$

выполнить следующее:

- а. Построить LL^T -разложение матрицы P (L — нижняя треугольная матрица с положительными элементами главной диагонали).
- б. С помощью LL^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (4, 18, 5, 16)^T$.

- в. С помощью разложения и решения системы по пп. а, б найти величину квадратной формы $J(x) = x^T Px$, где x — решение из п.б.

Задание 4. Для матрицы

$$A = \begin{pmatrix} 1 & 2 & 6 \\ -2 & 6 & -7 \\ -2 & 7 & 1 \end{pmatrix}$$

выполнить следующее:

- а. Построить QR -разложение матрицы A с помощью ортогональных преобразований (отражения Хаусхолдера).
- б. С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (5, -15, -8)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Вариант II

Задание 1. Для матрицы

$$A = \begin{pmatrix} 8 & -3 & 1 \\ -5 & -1 & -2 \\ 3 & 0 & 1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $\bar{U}L$ -разложение матрицы A (\bar{U} с единицами на главной диагонали).
- б.** С помощью $\bar{U}L$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (8, 0, 1)^T$.

- в.** С помощью $\bar{U}L$ -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 2. Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 10 & -3 \\ 0 & -3 & 5 \end{pmatrix}$$

и вектор $b = (-1, 3, 7)^T$, выполнить следующее:

- а.** Выписать метод Зейделя в координатном и каноническом виде.
- б.** Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.
- в.** Вычислить две итерации по методу Зейделя и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 3. Для матрицы

$$P = \begin{pmatrix} 21 & 3 & -12 & 3 \\ 3 & 9 & 4 & 1 \\ -12 & 4 & 13 & -2 \\ 3 & 1 & -2 & 1 \end{pmatrix}$$

выполнить следующее:

- а. Построить UU^T -разложение матрицы P (U — верхняя треугольная матрица с положительными элементами главной диагонали).
- б. С помощью UU^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (12, 17, 3, 3)^T$.

- в. С помощью разложения и решения системы по пп. а,б найти величину квадратной формы $J(x) = x^T Px$, где x — решение из п.б.

Задание 4. Для матрицы

$$A = \begin{pmatrix} 1 & 2 & -3 \\ -2 & 6 & 1 \\ -2 & 7 & -7 \end{pmatrix}$$

выполнить следующее:

- а. Построить QR -разложение матрицы A с помощью ортогональных преобразований (вращения Гивенса).
- б. С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (6, 3, 12)^T$.

- в. С помощью QR -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Вариант III

Задание 1. Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 9 & 5 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $L\bar{U}$ -разложение матрицы A (\bar{U} с единицами на главной диагонали).
- б.** С помощью $L\bar{U}$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (3, 7, 13)^T$.

- в.** С помощью $L\bar{U}$ -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 2. Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & -2 & 1 \\ 2 & -20 & 4 \\ -3 & 1 & 10 \end{pmatrix}$$

и вектор $b = (16, 50, 15)^T$, выполнить следующее:

- а.** Выписать метод Якоби в координатном и каноническом виде.
- б.** Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.
- в.** Вычислить две итерации по методу Якоби и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 3. Для матрицы

$$P = \begin{pmatrix} 1 & 2 & -2 & 3 \\ 2 & 6 & 0 & 8 \\ -2 & 0 & 15 & -8 \\ 3 & 8 & -8 & 24 \end{pmatrix}$$

выполнить следующее:

а. Построить LDL^T -разложение матрицы P (L — нижняя треугольная матрица с единицами на главной диагонали, D — диагональная матрица с положительными элементами на диагонали).

б. С помощью LDL^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (4, 16, 5, 27)^T$.

в. С помощью разложения и решения системы найти величину квадратной формы $J(x) = x^T Px$, где x — решение из п.б.

Задание 4. Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 6 \\ -2 & 4 & -7 \\ -2 & 5 & 1 \end{pmatrix}$$

выполнить следующее:

а. Построить QR -разложение матрицы A с помощью преобразований (Грама–Шмидта ортогонализация).

б. С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (8, -1, 8)^T$.

в. С помощью QR -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Вариант IV

Задание 1. Для матрицы

$$A = \begin{pmatrix} 1 & 4 & 1 \\ 4 & 14 & 3 \\ 2 & 4 & 1 \end{pmatrix}$$

выполнить следующее:

а. Построить $U\bar{L}$ -разложение матрицы A (\bar{L} с единицами на главной диагонали).

- б. С помощью $U\bar{L}$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-6, -22, -8)^T$.

- в. С помощью $U\bar{L}$ -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 2. Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 4 & -2 & 0 \\ -2 & 5 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

и вектор $b = (0, 7, 0)^T$, выполнить следующее:

- а. Выписать метод Зейделя в координатном и каноническом виде.
- б. Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.
- в. Вычислить две итерации по методу Зейделя и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задание 3. Для матрицы

$$P = \begin{pmatrix} 30 & -5 & -12 & 3 \\ -5 & 15 & 4 & 1 \\ -12 & 4 & 7 & -2 \\ 3 & 1 & -2 & 1 \end{pmatrix}$$

выполнить следующее:

- а. Построить UDU^T -разложение матрицы P (U — верхняя треугольная матрица с единицами на главной диагонали, D — диагональная матрица с положительными элементами на диагонали).
- б. С помощью UDU^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (16, 15, -3, 3)^T$.

- в.** С помощью разложения и решения системы найти величину квадратной формы $J(x) = x^T P x$, где x — решение из п.б.

Задание 4. Для матрицы

$$A = \begin{pmatrix} 1 & 3 & -3 \\ -2 & 4 & 1 \\ -2 & 5 & -7 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью преобразований (модифицированная Грама–Шмидта ортогонализация).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-1, -3, 4)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число обусловленности матрицы A (M_A) в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

9.4 Задачи для контрольных заданий и экзамена

Задача 1

Для матрицы

$$A = \begin{pmatrix} 2 & 0 & 2 \\ 4 & -1 & 3 \\ -2 & -3 & -2 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $\bar{L}U$ -разложение матрицы A (\bar{L} с единицами на главной диагонали).
- б.** С помощью $\bar{L}U$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (0, 0, -3)^T$.

- в.** С помощью $\bar{L}U$ -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 2

Для матрицы

$$A = \begin{pmatrix} 3 & 6 & 2 \\ -5 & -10 & -4 \\ 1 & 3 & 1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $\bar{U}L$ -разложение матрицы A (\bar{U} с единицами на главной диагонали).
- б.** С помощью $\bar{U}L$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (10, -16, 5)^T$.

- в.** С помощью $\bar{U}L$ -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 3

Для матрицы

$$A = \begin{pmatrix} 3 & 0 & 3 \\ -1 & 1 & -2 \\ 1 & 2 & 1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $L\bar{U}$ -разложение матрицы A (\bar{U} с единицами на главной диагонали).
- б.** С помощью $L\bar{U}$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (0, -2, -2)^T$.

- в.** С помощью $L\bar{U}$ -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 4

Для матрицы

$$A = \begin{pmatrix} -3 & 1 & 1 \\ 2 & 1 & 2 \\ 4 & 0 & 2 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $U\bar{L}$ -разложение матрицы A (\bar{L} с единицами на главной диагонали).
- б.** С помощью $U\bar{L}$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (5, 2, 0)^T$.

- в.** С помощью $U\bar{L}$ -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 5

Для матрицы

$$A = \begin{pmatrix} 2 & 2 & -4 \\ 1 & 2 & -2 \\ 2 & 1 & -1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $L\bar{U}^{-1}$ -разложение матрицы A (\bar{U}^{-1} с единицами на главной диагонали).
- б.** С помощью $L\bar{U}^{-1}$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (0, -1, -2)^T$.

- в.** С помощью $L\bar{U}^{-1}$ -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 6

Для матрицы

$$A = \begin{pmatrix} 6 & 1 & -1 \\ 5 & 1 & -2 \\ -8 & 0 & 4 \end{pmatrix}$$

выполнить следующее:

- а.** Построить $\bar{L}^{-1}U$ -разложение матрицы A (\bar{L}^{-1} с единицами на главной диагонали).
- б.** С помощью $\bar{L}^{-1}U$ -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-3, 0, 0)^T$.

- в.** С помощью $\bar{L}^{-1}U$ -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 7

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 1 & -1 \\ -1 & 5 & 0.5 \\ 1 & 1 & -10 \end{pmatrix}$$

и вектор $b = (-18, 1, 18)^T$, выполнить следующее:

- а.** Сформулировать метод Якоби в координатном и каноническом виде.
- б.** Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.

- в.** Вычислить две итерации по методу Якоби и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 8

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} -10 & 3 & -1 \\ 1 & -5 & 1 \\ 1 & 1 & 10 \end{pmatrix}$$

и вектор $b = (5, -7, -19)^T$, выполнить следующее:

- а.** Сформулировать метод Якоби в координатном и каноническом виде.
- б.** Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.
- в.** Вычислить две итерации по методу Якоби и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 9

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 4 & 0 & -1 \\ 0 & 5 & 2 \\ -1 & 2 & 10 \end{pmatrix}$$

и вектор $b = (-3, -2, -9)^T$, выполнить следующее:

- а.** Сформулировать метод Зейделя в координатном и каноническом виде.
- б.** Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.

- в.** Вычислить две итерации по методу Зейделя и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 10

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 2 & 0 \\ 2 & 5 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

и вектор $b = (8, -4, 3)^T$, выполнить следующее:

- а.** Сформулировать метод Зейделя в координатном и каноническом виде.
- б.** Определить является ли он сходящимся с нулевым начальным приближением, т.е. $x^0 = (0, 0, 0)^T$? Ответ обосновать.
- в.** Вычислить две итерации по методу Зейделя и найти апостериорную оценку ошибки на каждой из них в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 11

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 1 & -1 \\ -1 & 5 & 0.5 \\ 1 & 1 & 10 \end{pmatrix}$$

и вектор $b = (-9, 6, 0)^T$, выполнить следующее:

- а.** Сформулировать метод минимальных невязок в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?

- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 12

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 1 & -1 \\ -1 & 5 & 0.5 \\ 1 & 1 & 10 \end{pmatrix}$$

и вектор $b = (-9, 6, 0)^T$, выполнить следующее:

- а.** Сформулировать явный метод скорейшего спуска в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?
- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 13

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 3 & -1 \\ 1 & 5 & 1 \\ 2 & 1 & 10 \end{pmatrix}$$

и вектор $b = (11, 0, -8)^T$, выполнить следующее:

- а.** Сформулировать метод минимальных невязок в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?

- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 14

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 3 & -1 \\ 1 & 5 & 1 \\ 2 & 1 & 10 \end{pmatrix}$$

и вектор $b = (11, 0, -8)^T$, выполнить следующее:

- а.** Сформулировать явный метод скорейшего спуска в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?
- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 15

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 4 & 0 & -1 \\ 0 & 5 & 2 \\ -1 & 2 & 10 \end{pmatrix}$$

и вектор $b = (-3, -2, -9)^T$, выполнить следующее:

- а.** На основе метода Зейделя сформулировать метод минимальных поправок в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?

- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 16

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 4 & 0 & -1 \\ 0 & 5 & 2 \\ -1 & 2 & 10 \end{pmatrix}$$

и вектор $b = (-3, -2, -9)^T$, выполнить следующее:

- а.** На основе метода Зейделя сформулировать неявный метод скорейшего спуска в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?
- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 17

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 2 & 0 \\ 2 & 5 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

и вектор $b = (8, -4, 3)^T$, выполнить следующее:

- а.** На основе метода Зейделя сформулировать метод минимальных поправок в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?

- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 18

Для системы линейных алгебраических уравнений вида

$$Ax = b,$$

где матрица

$$A = \begin{pmatrix} 10 & 2 & 0 \\ 2 & 5 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

и вектор $b = (8, -4, 3)^T$, выполнить следующее:

- а.** На основе метода Зейделя сформулировать неявный метод скорейшего спуска в каноническом виде.
- б.** Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?
- в.** Вычислить одну итерацию и найти апостериорную оценку ошибки в норме $\|\cdot\|_\infty = \max_{i=1,2,3}\{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 19

Для матрицы

$$P = \begin{pmatrix} 4 & -2 & 2 & 4 \\ -2 & 2 & -3 & 3 \\ 2 & -3 & 14 & -8 \\ 4 & 3 & -8 & 33 \end{pmatrix}$$

выполнить следующее:

- а.** Построить LL^T -разложение матрицы P (L — нижняя треугольная матрица с положительными элементами главной диагонали).
- б.** С помощью LL^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (4, -10, 27, -40)^T$.

- в.** С помощью разложения и решения системы по пп. а,б найти величину квадратичной формы $J(x) = x^T P x$, где x — решение из п.б.

Задача 20

Для матрицы

$$P = \begin{pmatrix} 10 & 7 & 3 & 4 \\ 7 & 30 & -6 & 10 \\ 3 & -6 & 9 & 0 \\ 4 & 10 & 0 & 4 \end{pmatrix}$$

выполнить следующее:

- а.** Построить UU^T -разложение матрицы P (U — верхняя треугольная матрица с положительными элементами главной диагонали).
- б.** С помощью UU^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (4, -7, 0, -2)^T$.

- в.** С помощью разложения и решения системы по пп. а,б найти величину квадратичной формы $J(x) = x^T P x$, где x — решение из п.б.

Задача 21

Для матрицы

$$P = \begin{pmatrix} 4 & -2 & 2 & 4 \\ -2 & 2 & -3 & 3 \\ 2 & -3 & 14 & -8 \\ 4 & 3 & -8 & 33 \end{pmatrix}$$

выполнить следующее:

- а.** Построить LDL^T -разложение матрицы P (L — нижняя треугольная матрица с единицами на главной диагонали, D — диагональная матрица с положительными элементами на диагонали).
- б.** С помощью LDL^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (8, 0, 5, 32)^T$.

- в.** С помощью разложения и решения системы найти величину квадратичной формы $J(x) = x^T P x$, где x — решение из п.б.

Задача 22

Для матрицы

$$P = \begin{pmatrix} 14 & -1 & -1 & -3 \\ -1 & 10 & -2 & 0 \\ -1 & -2 & 5 & 1 \\ -3 & 0 & 1 & 1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить UDU^T -разложение матрицы P (U — верхняя треугольная матрица с единицами на главной диагонали, D — диагональная матрица с положительными элементами на диагонали).
- б.** С помощью UDU^T -разложения матрицы P решить систему

$$Px = b,$$

с вектором $b = (19, -9, -5, -5)^T$.

- в.** С помощью разложения и решения системы найти величину квадратичной формы $J(x) = x^T P x$, где x — решение из п.б.

Задача 23

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & 6 \\ -2 & 6 & -7 \\ -2 & 7 & 1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (5, -15, -8)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 24

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -2 & 6 & -1 \\ -2 & 7 & 7 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (6, 3, 12)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 25

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & -5 \\ -2 & 6 & 5 \\ -2 & 7 & -3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (8, -1, 8)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 26

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & 6 \\ -2 & 6 & 3 \\ -2 & 7 & -3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (5, -5, -12)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 27

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & 5 \\ -2 & 6 & -5 \\ -2 & 7 & 3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (6, 3, 12)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 28

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & 1 \\ -2 & 6 & 1 \\ -2 & 7 & 9 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-1, -3, 4)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 29

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & 7 \\ -2 & 6 & -9 \\ -2 & 7 & -1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-8, 1, -8)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 30

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & -3 \\ -2 & 6 & 1 \\ -2 & 7 & -7 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (6, 3, 12)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 31

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & -6 \\ -2 & 6 & 7 \\ -2 & 7 & -1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (7, 1, 10)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 32

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & -2 \\ -2 & 6 & -1 \\ -2 & 7 & -9 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (1, 3, -4)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 33

Для матрицы

$$A = \begin{pmatrix} 1 & 2 & -7 \\ -2 & 6 & 9 \\ -2 & 7 & 1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (10, -5, 4)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 34

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & -6 \\ -2 & 4 & -3 \\ -2 & 5 & 3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (8, 9, 4)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 35

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 3 \\ -2 & 4 & -1 \\ -2 & 5 & 7 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (5, 5, 14)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 36

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & -5 \\ -2 & 4 & 5 \\ -2 & 5 & -3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-7, -1, -10)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 37

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 6 \\ -2 & 4 & 3 \\ -2 & 5 & -3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (2, 1, -6)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 38

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 5 \\ -2 & 4 & -5 \\ -2 & 5 & 3 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-1, 7, 0)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 39

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 2 \\ -2 & 4 & 1 \\ -2 & 5 & 9 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-4, -7, -16)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 40

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 6 \\ -2 & 4 & -7 \\ -2 & 5 & 1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (8, -1, 8)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 41

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & 7 \\ -2 & 4 & -9 \\ -2 & 5 & -1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-8, 11, 3)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 42

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & -3 \\ -2 & 4 & 1 \\ -2 & 5 & -7 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-1, -3, 4)^T$.

- в.** С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 43

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & -6 \\ -2 & 4 & 7 \\ -2 & 5 & -1 \end{pmatrix}$$

выполнить следующее:

- а.** Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б.** С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (9, -3, 6)^T$.

- в. С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 44

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & -2 \\ -2 & 4 & -1 \\ -2 & 5 & -9 \end{pmatrix}$$

выполнить следующее:

- а. Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б. С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (-2, -6, -7)^T$.

- в. С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

Задача 45

Для матрицы

$$A = \begin{pmatrix} 1 & 3 & -7 \\ -2 & 4 & 9 \\ -2 & 5 & 1 \end{pmatrix}$$

выполнить следующее:

- а. Построить QR -разложение матрицы A с помощью ортогональных преобразований (Хаусхолдера / Гивенса / ГШО / МГШО).
- б. С помощью QR -разложения матрицы A решить систему линейных уравнений

$$Ax = b,$$

где вектор $b = (2, 6, 7)^T$.

- в. С помощью QR -разложения найти матрицу A^{-1} и вычислить число M_A обусловленности матрицы A в норме $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

II

Линейное оценивание

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Пусть два вектора $x \in \mathbb{R}^n$ и $z \in \mathbb{R}^m$ связаны (в общем случае) приближенным равенством $z \approx Ax$. Требуется выбрать такое значение \bar{x} вектора x , которое минимизирует квадрат отклонения $v \triangleq z - Ax$, т. е. найти

$$\bar{x} = \arg \min_x (z - Ax)^T (z - Ax).$$

Метод отыскания такого \bar{x} был предложен *Лежандром* в 1805 году в виде алгебраической процедуры и обоснован как статистическая процедура *Гауссом* в 1809 году (хотя утверждается, что рукопись Гаусса была известна на немецком языке уже с 1806 года [97]). Гаусс в 1809 году заявлял, что пользовался этим методом как алгебраической процедурой уже в 1795 году, чем вызвал немалое раздражение Лежандра [137].

Таким образом, с момента возникновения метод наименьших квадратов (МНК) рассматривается как процедура алгебраическая или процедура статистическая, но какого-либо противопоставления этих процедур нет. Есть лишь разные точки зрения: ставить задачу с позиций Линейной алгебры (ЛА) или же с позиций Теории вероятностей (ТВ) и Математической статистики (МС). ЛА придает задаче наименьших квадратов лаконичную форму, позволяет анализировать все решения этой задачи и формулировать метод. ТВ и МС дают возможность подходить к этой задаче не формально алгебраически, а с точки зрения неопределенности, присущей реальным экспериментальным данным z . Замечательно, что МС, формулируя задачу независимо и совершенно в других терминах, приводит к тем же аналитическим решениям, что и ЛА. Можно говорить, что МС дает статистическую интерпретацию для чисто алгебраической задачи и алгебраического метода решения несовместных систем линейных алгебраических уравнений $z \approx Ax$. Вычислительная линейная алгебра (ВЛА) предлагает множество идей и подходов для эффективной численной реализации МНК.

В настоящее время МНК имеет множество приложений и эффективных численных реализаций и часто излагается как раздел Эконометрики [19], Математических методов обработки данных [49, 50], как Прикладной регрессионный анализ [34] или Регрессионное моделирование [26].

Приводимый ниже материал имеет базовый теоретический характер. В последующих разделах методы ВЛА применяются для изложения эффективных численных алгоритмов решения этих задач.

10.1 Конечномерные линейные пространства

Множество \mathcal{L} в \mathbb{R}^n ($\mathcal{L} \subseteq \mathbb{R}^n$) называется *линейным подпространством* пространства \mathbb{R}^n всех вещественных n -мерных¹ векторов, или, короче, подпространством в \mathbb{R}^n , если при любых скалярных величинах α и β принадлежность $x \in \mathcal{L}$ и $y \in \mathcal{L}$ влечет принадлежность $\alpha x + \beta y \in \mathcal{L}$. Это выражается следующей формулой:²

$$\forall \alpha, \beta \in \mathbb{R}^1 ((x \in \mathcal{L} \& y \in \mathcal{L}) \Rightarrow (\alpha x + \beta y \in \mathcal{L})) .$$

Линейная независимость системы $\{a^1, \dots, a^m\} \in \mathbb{R}^n$ означает, что справедлива импликация

$$\left(\sum_{i=1}^m \lambda_i a^i = 0 \right) \Rightarrow \left(\bigvee_{i=1}^m \lambda_i = 0 \right) ,$$

в противном случае система называется *линейно зависимой*.

Подпространство $\mathcal{L} \subseteq \mathbb{R}^n$ называется m -мерным, т. е. имеющим *размерность* $\dim \mathcal{L} = m$, если в нем имеется такая линейно независимая система векторов $\{a^1, \dots, a^m\}$, что добавление к этой системе любого вектора $a \in \mathbb{R}^n$ дает уже линейно зависимую систему, содержащую $m + 1$ векторов.³ Это выражается следующей формулой:

$$\left(\exists \{a^1, \dots, a^m\} \in \mathcal{L} \left(\left(\sum_{i=1}^m \lambda_i a^i = 0 \right) \Rightarrow \left(\bigvee_{i=1}^m \lambda_i = 0 \right) \right) \right) \& \\ \& \left(\forall a \in \mathcal{L} \left(\left(\sum_{i=1}^m \lambda_i a^i + \lambda_{i+1} a = 0 \right) \& (\lambda_{i+1} \neq 0) \right) \right) .$$

¹ n -мерный вектор содержит n -компонент.

² Импликация $\mathcal{A} \Rightarrow \mathcal{B}$ означает: «из \mathcal{A} следует \mathcal{B} », или « \mathcal{A} влечет \mathcal{B} », или «если \mathcal{A} , то \mathcal{B} ».

³ Такая система $\{a^1, \dots, a^m\}$ называется *максимальной* линейно независимой системой.

Так как $\lambda_{m+1} \neq 0$, то

$$a = \sum_{i=1}^m \mu_i a^i, \quad \mu_i = -\lambda_i / \lambda_{m+1}.$$

Следовательно, любой вектор $a \in \mathfrak{L}$, где $\dim \mathfrak{L} = m$, может быть представлен в виде линейной комбинации векторов $\{a^1, \dots, a^m\}$. Система векторов $\{a^1, \dots, a^m\}$, обладающая этим свойством, образует *базис* в \mathfrak{L} . В этом случае записывают $\mathfrak{L} = \mathfrak{L}(a^1, \dots, a^m)$ и говорят, что \mathfrak{L} *натянута* на $\{a^1, \dots, a^m\}$. Если каждый вектор из \mathfrak{L} может быть выражен линейной комбинацией системы векторов a^1, \dots, a^m , то \mathfrak{L} называют *линейной оболочкой* векторов $\{a^1, \dots, a^m\}$.

Множество \mathfrak{L}_0 , состоящее из одного нулевого вектора 0, является подпространством в \mathbb{R}^n размерности 0 и называется *нулевым подпространством*.

n-мерное евклидово пространство \mathbb{E}^n — это пространство \mathbb{R}^n , в котором определено *скалярное произведение* двух векторов x и y по формуле $(x, y) = x_1 y_1 + \dots + x_n y_n = x^T y$.⁴ Норма $\|x\|$ вектора $x \in \mathbb{E}^n$ равна $(x, x)^{1/2}$, то есть $\|x\|^2 = x^T x$. Расстояние между $x, y \in \mathbb{E}^n$ есть $\|x - y\|$. Векторы x и y *ортогональны*, $x \perp y$, если их скалярное произведение равно нулю: $(x, y) = x^T y = 0$. Вектор $v \in \mathbb{R}^n$ ортогонален к \mathfrak{L} , $v \perp \mathfrak{L}$, если он ортогонален к любому вектору $u \in \mathfrak{L}$. *Ортогональное дополнение* к \mathfrak{L} , обозначаемое \mathfrak{L}^\perp , есть множество всех векторов в \mathbb{R}^n , каждый из которых ортогонален к \mathfrak{L} .

ТЕОРЕМА 10.1 ([25]). Пусть $\mathfrak{L}_0 \subset \mathfrak{L} \subset \mathbb{R}^n$. Тогда:

- (1) существуют целое число m , $1 \leq m < n$, и ортонормированный базис $\{a^1, \dots, a^m\}$ в \mathfrak{L} . Если этот базис продолжить любым способом до ортонормированного базиса

$$a^1, \dots, a^m, a^{m+1}, \dots, a^n \tag{10.1}$$

в \mathbb{R}^n , то линейное подпространство \mathfrak{M} с базисом a^{m+1}, \dots, a^n обладает свойствами:

- (2) $\mathfrak{M} = \mathfrak{L}^\perp$,
- (3) $\mathfrak{L} = \mathfrak{M}^\perp$,
- (4) для любого вектора $x \in \mathbb{R}^n$ существует единственное разложение

$$x = \hat{x} + \tilde{x}, \quad \hat{x} \in \mathfrak{L}, \quad \tilde{x} \in \mathfrak{M}. \tag{10.2}$$

⁴ Везде, где не оговорено противное, в этой книге рассматривается пространство вещественных чисел и, кроме того, любой вектор ассоциируется с матрицей-столбцом.

ДОКАЗАТЕЛЬСТВО.

- (1) Так как \mathfrak{L} отлично от нулевого подпространства, $\mathfrak{L} \supset \mathfrak{L}_0$, то в нем существует вектор a , отличный от 0. Образует нормальный (т. е. с единичной нормой) вектор $a^1 = a/\|a\|$. Если в \mathfrak{L} найдется вектор, ортогональный к a^1 , нормируем его аналогично и обозначим a^2 . Если найдется вектор, ортогональный к a^1 и к a^2 , нормируем его и обозначим a^3 . Продолжая этот процесс, завершим его построением *ортонормированной системы векторов* $\{a^1, \dots, a^m\}$, где $m \geq 1$ и $m < n$. Действительно, $a^1 \in \mathfrak{L}$, но m не может быть равно n . В противном случае векторы $\{a^1, \dots, a^n\}$ образовали бы базис в \mathfrak{L} (так как ортонормированные векторы линейно независимы), что означало бы $\mathfrak{L} = \mathbb{R}^n$. Однако по условию \mathfrak{L} не совпадает с \mathbb{R}^n ($\mathfrak{L} \subset \mathbb{R}^n$), поэтому $m < n$.

Построенная система $\{a^1, \dots, a^m\}$ есть базис в \mathfrak{L} . Действительно, вместе с векторами $\{a^1, \dots, a^m\}$ к \mathfrak{L} принадлежат и все их линейные комбинации, то есть векторы вида

$$u = \sum_{i=1}^m \lambda_i a^i, \quad \lambda_i = (u, a^i). \quad (10.3)$$

Однако кроме них, других векторов в \mathfrak{L} нет. Допустив противное, следовало бы считать, что и векторы вида

$$x = \sum_{i=1}^m (x, a^i) a^i + y, \quad (10.4)$$

где $y \neq 0$, входят в \mathfrak{L} , $x \in \mathfrak{L}$. А так как все $a^i \in \mathfrak{L}$, то и вектор

$$y = x - \sum_{i=1}^m (x, a^i) a^i$$

следовало бы включить в \mathfrak{L} . Но для всех $k = 1, \dots, m$ имеем

$$(y, a^k) = (x, a^k) - \sum_{i=1}^m (x, a^i) (a^i, a^k) = 0, \quad (10.5)$$

то есть $\forall k = 1, \dots, m$ ($y \perp a^k$). Пронормированный вектор $y/\|y\|$ мог бы стать тем a^{m+1} , который расширил бы систему $\{a^1, \dots, a^m\}$ до системы $\{a^1, \dots, a^m, a^{m+1}\}$. Однако это невозможно из-за доказанного ограничения для числа m . Тем самым утверждение (1) теоремы доказано.

- (2) Поскольку $m < n$, в \mathbb{R}^n существует вектор x , не зависящий линейно от $\{a^1, \dots, a^m\}$, то есть выражаемый равенством (10.4), в котором $y \neq 0$, $y \in \mathbb{R}^n$. Так как для y справедливо (10.5), построение базиса в \mathbb{R}^n можно продолжить, как только что указано, до (10.1). Но любой вектор $v \in \mathfrak{M}$ определяется, подобно (10.3), в виде

$$v = \sum_{i=m+1}^n \lambda_i a^i, \quad \lambda_i = (v, a^i). \quad (10.6)$$

Очевидно, что $v \perp u$, и если известно, что какой-нибудь вектор $a \in \mathbb{R}^n$ ортогонален ко всем векторам из \mathfrak{L} , $a \perp \mathfrak{L}$, то $a \in \mathfrak{M}$. Тем самым доказано утверждение (2) теоремы.

- (3) Имеем $u \perp v$. Если для какого-нибудь вектора $a \in \mathbb{R}^n$ известно, что $a \perp \mathfrak{M}$, то $a \in \mathfrak{L}$. Доказано утверждение (3) теоремы.
- (4) Наконец, если x – произвольный вектор в \mathbb{R}^n , то единственно его представление

$$x = \sum_{i=1}^n \mu_i a^i,$$

так как единственным образом определяются числа $\mu_i = (x, a^i) = x^T a^i$, называемые *числовыми проекциями вектора x на направление единичного вектора a^i* . Отсюда единственно разложение (10.2), причем

$$\hat{x} = \sum_{k=1}^m \mu_k a^k \in \mathfrak{L}, \quad \tilde{x} = \sum_{k=m+1}^n \mu_k a^k \in \mathfrak{M}.$$

□

Теорема 10.1 содержится в стандартных курсах линейной алгебры [25] и для метода наименьших квадратов может считаться отправной точкой. Ее называют *теоремой об ортогональном разложении пространства \mathbb{R}^n* и формулируют также следующим образом [13].

ТЕОРЕМА 10.2. Если оба \mathfrak{L} и $\mathfrak{M} \in \mathbb{R}^n$ и выполнено хотя бы одно из следующих условий:

- (1) $\mathfrak{L} = \mathfrak{M}^\perp$ (\mathfrak{L} состоит из векторов, ортогональных к \mathfrak{M}),
- (2) $\mathfrak{M} = \mathfrak{L}^\perp$ (\mathfrak{M} состоит из векторов, ортогональных к \mathfrak{L}),
- (3) \mathfrak{L} и \mathfrak{M} взаимно ортогональны и $\dim \mathfrak{L} + \dim \mathfrak{M} = n$,

то \mathfrak{L} и \mathfrak{M} являются ортогональными дополнениями друг к другу. Если выполнено хотя бы одно из этих трех эквивалентных условий, то любой вектор $x \in \mathbb{R}^n$ может быть разложен единственным образом в сумму $x = \hat{x} + \tilde{x}$, где $\hat{x} \in \mathfrak{L}$, $\tilde{x} \in \mathfrak{M}$. Составляющие этого разложения \hat{x} и \tilde{x} являются проекциями вектора x на подпространства \mathfrak{L} и \mathfrak{M} , соответственно, и они взаимно ортогональны, т. е. $\tilde{x}^T \hat{x} = 0$.

ДОКАЗАТЕЛЬСТВО. Все, что есть в этой формулировке, уже доказано в Теореме 10.1, кроме последнего утверждения о проекциях вектора x . Чтобы доказать и это, напомним, что *проекцией вектора x на подпространство \mathfrak{L}* называется вектор из \mathfrak{L} , ближайший к вектору x .

Для любого $y \in \mathfrak{L}$ имеем

$$\|x - y\|^2 = \|\hat{x} + \tilde{x} - y\|^2 = \|(\hat{x} - y) + \tilde{x}\|^2 = \|\hat{x} - y\|^2 + \|\tilde{x}\|^2,$$

так как $\hat{x} - y \in \mathfrak{L}$ и $(\hat{x} - y) \perp \tilde{x}$. Поэтому $\|x - y\|^2 \geq \|\tilde{x}\|^2$, и строгое неравенство выполняется, если и только если $y \neq \hat{x}$. Следовательно, \hat{x} — проекция x на \mathfrak{L} . \square

ЗАМЕЧАНИЕ 10.1. Проекцию \hat{x} вектора x на подпространство \mathfrak{L} будем обозначать следующим образом: $\hat{x} = (x \mid \mathfrak{L})$.

10.2 Обобщение на гильбертовы пространства

Замечательно, что возможности линейной алгебры не ограничиваются конечномерными пространствами, состоящими лишь из детерминистских векторов. Реальные прикладные задачи имеют дело с функциями, причем не обязательно детерминистскими, а случайными. Необходимость оперировать со значениями функции на интервалах независимой переменной приводит к понятию бесконечномерного вектора. Действительно: любую функцию на интервале значений аргумента можно рассматривать как вектор с континуальным количеством компонент, равных значениям этой функции при изменении аргумента в своем интервале. Дальнейшее усложнение картины мы получаем, когда рассматриваем случайные функции, то есть функции, принимающие значения из некоторого выборочного, вероятностного пространства. Эти обобщения содержатся в *теории гильбертовых пространств* случайных переменных, строгое построение которой можно найти, например, в фундаментальной монографии [92].

В гильбертовых пространствах, обозначаемых \mathbb{H} , сохраняются все основные определения.

ОПРЕДЕЛЕНИЕ 10.1. *Подпространством \mathfrak{L} векторного пространства \mathfrak{X} называют любое подмножество \mathfrak{L} данного множества \mathfrak{X} , $\mathfrak{L} \subseteq \mathfrak{X}$, на котором $x, y \in \mathfrak{L}$ влечет $\alpha x + \beta y \in \mathfrak{L}$ для всех $\alpha, \beta \in \mathbb{R}$. При этом \mathfrak{L} называют *собственным подпространством*, если $\mathfrak{L} \neq \mathfrak{X}$, т. е. $\mathfrak{L} \subset \mathfrak{X}$. Для любого множества $\mathfrak{L} \subseteq \mathfrak{X}$ обозначение $\text{Sp} \{\mathfrak{L}\}$ используют для множества всех конечных линейных комбинаций элементов множества \mathfrak{L} . Множество $\text{Sp} \{\mathfrak{L}\}$ называют *линейной оболочкой*⁵.*

В гильбертовых пространствах, как и в конечномерных, понятие внутреннего (скалярного) произведения дает абстрактную формулировку концепции угла, и как результат, обобщенное понятие перпендикулярности.

ОПРЕДЕЛЕНИЕ 10.2. Векторы $x, y \in \mathbb{H}$ называются *ортгоналными*, если $(x, y) = 0$, и это обозначают записью $x \perp y$. Если $\mathbf{S} \subset \mathbb{H}$ есть любое подмножество (и, в особенности, если \mathbf{S} есть подпространство) в \mathbb{H} , тогда пишут $x \perp \mathbf{S}$, если $\forall s \in \mathbf{S}, x \perp s$. Подобно этому, обозначение $\mathbf{S} \perp \mathbf{T}$ для двух подмножеств \mathbf{S} и \mathbf{T} в \mathbb{H} , указывает, что все элементы подмножества \mathbf{S} ортгональны всем элементам подмножества \mathbf{T} .

Приведем главный результат геометрии гильбертовых пространств. Он называется *Теоремой о проекции*. Как и в случае конечномерных пространств (см. подразд. 10.1) он просто устанавливает тот факт, что кратчайший путь от точки до плоскости лежит на перпендикуляре от этой точки до данной плоскости (рис. 10.1). Нижеследующая Теорема о проекции обобщает этот результат на гильбертовы пространства.

ТЕОРЕМА 10.3 (*Теорема об ортгональной проекции* [92]). Пусть \mathfrak{L} есть собственное подпространство некоторого гильбертова пространства \mathbb{H} и пусть x есть точка в \mathbb{H} . Тогда x может быть единственным образом представлено в форме

$$x = y + z$$

с $y \in \mathfrak{L}$ и $z \perp \mathfrak{L}$. Кроме того, для всех $w \in \mathfrak{L}$ имеем

$$\|x - w\| \geq \|x - y\|,$$

где равенство возможно, если и только если $w = y$ (см. рис. 10.1).

ДОКАЗАТЕЛЬСТВО. См. [92]. □

⁵ Span = оболочка.

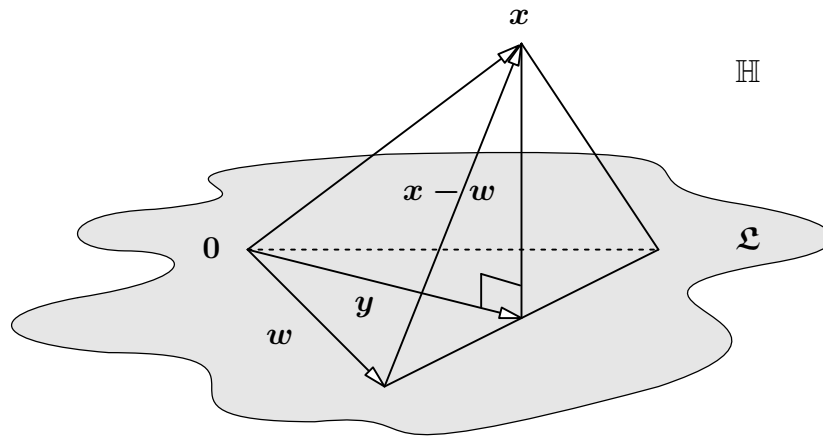


Рис. 10.1. Теорема об ортогональной проекции

Метод наименьших квадратов заключается в практическом применении этой теоремы, т. е. в отыскании вектора $y \in \mathcal{L} \subset \mathbb{H}$, ближайшего к заданному вектору $x \in \mathbb{H}$ и $x \notin \mathcal{L}$. Обобщение на гильбертовы пространства позволяет пользоваться этим методом для обоснования многих теоретических результатов в широком классе линейных стохастических систем, включая: теорию оценивания, теорию оптимальной фильтрации и предсказания, теорию параметрической идентификации стохастических динамических систем и теорию адаптивного управления [92].

10.3 Проектирование в конечномерных пространствах

Вернемся к конечномерным пространствам. На основании Теоремы 10.2 может быть построено доказательство следующего утверждения, в котором используется понятие *ранга матрицы* A , а именно: $\text{rank } A$ равен максимальному числу строк (или столбцов) матрицы A , образующих линейно независимую систему.

ТЕОРЕМА 10.4 (Основная теорема линейной алгебры). С любой $(n \times m)$ -матрицей A , имеющей $\text{rank } A = r$, ассоциированы четыре фундаментальных подпространства:

- (1) $\mathcal{R}(A^T)$ — пространство строк матрицы A , $\dim \mathcal{R}(A^T) = r$,
- (2) $\mathcal{N}(A)$ — нуль-пространство, т. е. ядро матрицы A , $\dim \mathcal{N}(A) = m - r$,
- (3) $\mathcal{R}(A)$ — пространство столбцов (образ) матрицы A , $\dim \mathcal{R}(A) = r$,

(4) $\mathcal{N}(A^T)$ — левое нуль-пространство матрицы A , $\dim \mathcal{N}(A^T) = n - r$,

при этом

$$\begin{aligned}\mathcal{N}(A) &= (\mathcal{R}(A^T))^\perp, & \mathcal{R}(A^T) &= (\mathcal{N}(A))^\perp, \\ \mathcal{N}(A^T) &= (\mathcal{R}(A))^\perp, & \mathcal{R}(A) &= (\mathcal{N}(A^T))^\perp.\end{aligned}$$

Последнее означает, что система $Ax = z$ имеет решение тогда и только тогда, когда вектор $z \perp \mathcal{N}(A^T)$, то есть $z \in \mathcal{R}(A)$ тогда и только тогда, когда z ортогонален к каждому решению системы $A^T y = 0$.

Базовые понятия, содержащиеся в этой теореме, широко используются в дальнейшем.

ОПРЕДЕЛЕНИЕ 10.3. Квадратная матрица P называется *матрицей проектирования* или *проектором* на \mathfrak{L} , если $(x - Px) \perp \mathfrak{L}$, т. е. для всех $x \in \mathbb{R}^n$ проекция $\hat{x} = Px$.

ТЕОРЕМА 10.5.

(1) Проекционная матрица P обладает свойствами:

$$(i) P^2 = P \text{ (идемпотентность)}, \quad (ii) P = P^T \text{ (симметричность)}.$$

(2) Любая матрица P , обладающая этими свойствами, является проекционной матрицей, причем она проектирует любой вектор на свое пространство столбцов, $\mathcal{R}(P)$.

(3) Если $\mathfrak{L} = \mathfrak{L}(a^1, \dots, a^m)$, $a^i \in \mathbb{R}^n$ и $\{a^1, \dots, a^m\}$ — базис в \mathfrak{L} , то $P = P_A = A(A^T A)^{-1} A^T$, где $A = [a^1 \mid \dots \mid a^m]$ — матрица, столбцами которой служат векторы a^i ($i = 1, 2, \dots, m$).

ДОКАЗАТЕЛЬСТВО.

(1) *Свойство (i).* Определение проектора на \mathfrak{L} означает, что

$$\forall x \quad \{Px = \hat{x}, \quad x = \hat{x} + \tilde{x}, \quad \hat{x} \in \mathfrak{L}, \quad \tilde{x} \perp \mathfrak{L}, \quad P\tilde{x} = 0\}.$$

Отсюда

$$\forall x \quad \{P^2 x = P\hat{x} = P(x - \tilde{x}) = Px - P\tilde{x} = Px, \quad (P^2 - P)x = 0\}.$$

Следовательно, $P^2 = P$.

Свойство (ii). Имеем $\forall x, y \quad \{Px \in \mathfrak{L}, \quad (I - P)y \in \mathfrak{L}^\perp\}$. Отсюда $(Px)^T(I - P)y = 0$, $x^T(P^T - P^T P)y = 0$, $P^T = P^T P$, $P = (P^T P)^T = P^T P$. Следовательно, $P = P^T$.

- (2) Пусть P есть $(n \times n)$ -матрица со свойствами (i) и (ii) из предыдущего пункта теоремы. Имеем $Px \in \mathcal{R}(P)$. В то же время $\mathcal{R}(P)$ есть множество векторов вида Pu , где u — любой вектор из \mathbb{R}^n .

Найдем скалярное произведение

$$\begin{aligned}(x - Px)^T Pu &= x^T(I - P^T)Pu = x^T(P - P^T P)u = \\ &= x^T(P - P^2)u = x^T(P - P)u = 0.\end{aligned}$$

Следовательно, для любого $x \in \mathbb{R}^n$ произведение Px есть проекция вектора x на $\mathcal{R}(P)$.

- (3) Пусть $P = A(A^T A)^{-1}A^T$. Имеем $Px \in \mathcal{L} = \mathcal{R}(A)$. В то же время $\mathcal{R}(A)$ есть множество векторов вида Au , где u — любой вектор из \mathbb{R}^n . Найдем скалярное произведение

$$\begin{aligned}(Au)^T(x - Px) &= u^T A^T[I - A(A^T A)^{-1}A^T]x = \\ &= u^T[A^T - A^T A(A^T A)^{-1}A^T]x = 0.\end{aligned}$$

Следовательно, $x - Px \perp \mathcal{L}$ для всех $x \in \mathbb{R}^n$. □

В качестве простого следствия отсюда легко видеть, что *проекция вектора x на $\mathcal{L}(y)$* задается формулой $(x^T y)y/\|y\|^2$, если $y \neq 0$. Также в качестве следствия можно получить так называемую *теорему разложения Фурье* [1].

ТЕОРЕМА 10.6 (*Теорема разложения Фурье* [1]). Пусть дано собственное подпространство $\mathcal{L} = \mathcal{L}(a^1, \dots, a^m) \subset \mathbb{R}^n$ и $\{a^1, \dots, a^m\}$ — ортонормированный базис в \mathcal{L} . Тогда для произвольного $x \in \mathbb{R}^n$ проекция \hat{x} на \mathcal{L} задается формулой

$$\hat{x} = \left[\sum_{i=1}^m a^i (a^i)^T \right] x = AA^T x,$$

где $A = [a^1 \mid \dots \mid a^m]$, т. е. $P = P_A = AA^T$, или же равносильной формулой

$$\hat{x} = \sum_{i=1}^m (x^T a^i) a^i = \sum_{i=1}^m (x, a^i) a^i.$$

Кроме того, легко видеть в качестве следствия Теоремы 10.5, что если P есть проектор на \mathcal{L} , то $(I - P)$ есть проектор на \mathcal{L}^\perp , где $\mathcal{L} = \mathcal{R}(P)$ и $\mathcal{L}^\perp = \mathcal{N}(P^T)$.

ЗАМЕЧАНИЕ 10.2. Теорема 10.5 в пункте (3) определяет матрицу проектирования на пространство $\mathcal{R}(A)$ столбцов данной $(n \times m)$ -матрицы

A , когда $\text{rank } A = m$. Эту матрицу удобно обозначить P_A . Теорема 10.5 устанавливает, что в этом случае $P_A = A(A^T A)^{-1} A^T$, в то время как Теорема 10.6 рассматривает частный, но для вычислений очень удобный подслучай, когда $A^T A = I$, при этом $P_A = A A^T$. Наиболее общий случай проектирования на $\mathcal{R}(A)$, когда $(n \times m)$ -матрица A произвольна и не ограничена условием $\text{rank } A = m$, рассмотрен ниже (подразд. 10.4).

10.4 Наименьшие квадраты и псевдоинверсия

Линейная задача наименьших квадратов возникает из необходимости решать систему линейных алгебраических уравнений $Ax = z$ с произвольно заданными $(n \times m)$ -матрицей A и правой частью $z \in \mathbb{R}^n$. При этом в силу произвольности A и z система может оказаться несовместной (в этом случае правильнее писать $Ax \approx z$) или же иметь одно или бесконечно много решений $x \in \mathbb{R}^m$. Решение *линейной системы* $Ax \approx z$ в смысле *наименьших квадратов*⁶ определяют как вектор \bar{x} , доставляющий *наименьшее* значение *квадратическому* критерию качества

$$J(x) = \|z - Ax\|^2 = (z - Ax)^T(z - Ax). \quad (10.7)$$

Таким образом,

$$\bar{x} \triangleq \arg \min_{x \in \mathbb{R}^m} (z - Ax)^T(z - Ax).$$

Очевидно, данное определение эквивалентно соглашению принять в качестве МНК-решения любой вектор \bar{x} , если и только если $A\bar{x} = \hat{z}$, где \hat{z} — проекция z на $\mathcal{R}(A)$.

ТЕОРЕМА 10.7. МНК-решение \bar{x} системы $Ax = z$, где $A = A(n, m)$, удовлетворяет системе *нормальных уравнений*:

$$A^T A \bar{x} = A^T z. \quad (10.8)$$

Это решение всегда существует, хотя может быть не единственным. Оно единственно тогда и только тогда, когда A имеет полный столбцовый ранг, $\text{rank } A = m$.

ДОКАЗАТЕЛЬСТВО. По определению \bar{x} имеем $A\bar{x} = \hat{z}$, $(z - \hat{z}) \perp \mathcal{A}c$. Здесь c — любой вектор из \mathbb{R}^m . Это означает, что

$$\forall c \in \mathbb{R}^m, \quad (Ac)^T(z - \hat{z}) = c^T(A^T z - A^T A \bar{x}) = 0.$$

⁶ Иначе, *МНК-решение*.

Следовательно, $A^T z - A^T A \bar{x} = 0$, т.е. \bar{x} удовлетворяет (10.8). Эта система всегда совместна, так как оба вектора $A^T z$ и $A^T(A\bar{x})$ принадлежат одному и тому пространству $\mathcal{R}(A^T)$ при $\forall \bar{x} \in \mathbb{R}^m$. Для установления условия единственности решения \bar{x} докажем промежуточный результат.

ЛЕММА 10.1. Справедливы равенства:

$$\begin{aligned} (1) \quad \mathcal{R}(A^T) &= \mathcal{R}(A^T A), & (2) \quad \mathcal{R}(A) &= \mathcal{R}(A A^T), \\ (3) \quad \mathcal{N}(A^T) &= \mathcal{N}(A A^T), & (4) \quad \mathcal{N}(A) &= \mathcal{N}(A^T A). \end{aligned}$$

ДОКАЗАТЕЛЬСТВО. На основании соотношений подпространств из Теоремы (10.4), утверждение (1) Леммы 10.1 эквивалентно утверждению (4); аналогично этому, утверждение (2) Леммы 10.1 эквивалентно утверждению (3). Поэтому достаточно доказать утверждение (3) и утверждение (4). Чтобы доказать совпадение $\mathcal{N}(A)$ с $\mathcal{N}(A^T A)$, заметим, что $A^T A x = 0$, если $A x = 0$. В обратную сторону: если $A^T A x = 0$, то $x^T A^T A x = 0$, т.е. $\|A x\|^2 = 0$, что влечет $A x = 0$. Таким образом, $A x = 0$ эквивалентно $A^T A x = 0$, т.е. $\mathcal{N}(A) = \mathcal{N}(A^T A)$. Аналогично устанавливается утверждение (3). \square

Завершая доказательство Теоремы 10.7, используем из только что доказанной Леммы 10.1 утверждение (1). Согласно этому утверждению, $\text{rank } A = \text{rank } A^T A$. По условию теоремы, $\text{rank } A = m$. Отсюда, $(m \times m)$ -матрица $A^T A$ системы (10.8) имеет полный ранг, т.е. $(A^T A)^{-1}$ существует. В этом случае имеем единственное МНК-решение $\bar{x} = (A^T A)^{-1} A^T z$. Очевидно, это возможно только при $n \geq m$ (переопределенные системы $A x = z$) и $\text{rank } A = m$ (полный столбцовый ранг). В других случаях: (i) при $n \geq m$, но $\text{rank } A = r < m$, или (ii) при $n < m$, — решение \bar{x} не может быть единственным. \square

Каким образом в случае неединственности \bar{x} выбрать среди \bar{x} единственный вектор \bar{x}_0 , в некотором смысле оптимальный?

ОПРЕДЕЛЕНИЕ 10.4. Оптимальным МНК-решением, или иначе — *нормальным псевдорешением* системы $A x = z$, называется вектор \bar{x}_0 , который имеет минимальную (евклидову) норму среди всех \bar{x} , удовлетворяющих системе $A \bar{x} = \hat{z}$, $\hat{z} \in \mathcal{R}(A)$, $(z - \hat{z}) \perp \mathcal{R}(A)$.

ЗАМЕЧАНИЕ 10.3. Пусть $\text{rank } A = m$, когда в согласии с Теоремой (10.7) имеем единственное $\bar{x} = (A^T A)^{-1} A^T z$, оно же \bar{x}_0 . Если теперь

записать $\bar{x}_0 = A^+z$, где A^+ — некоторая матрица, то для этого случая она определяется как $A^+ = (A^T A)^{-1} A^T$. Эта формула (при условии $\text{rank } A = m$) включает в себя наиболее простой случай, когда $n = m$. Тогда A^{-1} существует, $A^+ = A^{-1}$ и $\bar{x}_0 = A^{-1}z$, что совпадает с обычным решением системы $Ax = z$, которая при этих условиях есть стандартная совместная система с квадратной матрицей A . Таким образом, матрица A^+ обобщает понятие обратной матрицы A^{-1} на случай, когда матрица A в системе $Ax = z$ произвольна по своим размерам и рангу. В связи с этим она названа *псевдообратной матрицей* A^+ к $(n \times m)$ -матрице A ; при $\text{rank } A = m$ она, как выше отмечено, равна $(A^T A)^{-1} A^T$. Перейдем к общему случаю для A^+ .

ОПРЕДЕЛЕНИЕ 10.5. *Псевдообратная матрица* (в общем случае произвольной матрицы A) есть такая матрица A^+ , которая из всех решений \bar{x} системы $A\bar{x} = \hat{z}$, где $\hat{z} \in \mathcal{R}(A)$ и $(z - \hat{z}) \perp \mathcal{R}(A)$ при произвольном векторе z , выбирает \bar{x}_0 с минимальной нормой, определяя его как $\bar{x}_0 = A^+z$.

СЛЕДСТВИЕ 10.1. *Проектор на $\mathcal{R}(A) = \mathfrak{L}(a^1, \dots, a^m)$, где все столбцы $a^1, \dots, a^m \in \mathbb{R}^n$ матрицы A не обязательно образуют линейно независимую систему векторов, определяется выражением $P_A = AA^+$. Соответственно, $(I - AA^+)$ — проектор на $\mathcal{N}(A^T) = \mathcal{R}(A)^\perp$.*

Действительно, проекция любого вектора z на $\mathcal{R}(A)$ равна $\hat{z} = A\bar{x}_0 = (AA^+)z = P_A z$, а $\tilde{z} = z - \hat{z} = (I - P_A)z$. \square

То, что \bar{x}_0 (а значит, и A^+) существует, ясно из Теоремы 10.7. Однако, единствен ли вектор \bar{x}_0 ? В каком подпространстве он лежит и как его найти? Ответить на эти вопросы означает, по существу, выяснить все свойства псевдообратной матрицы A^+ , поскольку приведенное для нее Определение 10.3 ее конструктивно не характеризует.

ТЕОРЕМА 10.8 ([1]). Пусть $x \in \mathbb{R}^n$ и A — матрица размера $(n \times m)$. Среди всех векторов \bar{x} , минимизирующих $\|z - Ax\|^2$, то есть удовлетворяющих уравнению

$$A\bar{x} = \hat{z}, \quad \hat{z} \in \mathcal{R}(A), \quad (z - \hat{z}) \perp \mathcal{R}(A), \quad (10.9)$$

или, что эквивалентно, системе нормальных уравнений $A^T A\bar{x} = A^T z$, вектор \bar{x}_0 , имеющий минимальную норму, является единственным вектором из $\mathcal{R}(A^T)$, то есть вектором вида

$$\bar{x}_0 = A^T y, \quad y \in \mathbb{R}^m.$$

ДОКАЗАТЕЛЬСТВО. Каждый вектор \bar{x} , согласно Теореме 10.1 (или Теоремам 10.2, 10.4), может быть разложен в сумму

$$\bar{x} = \bar{x}_r + \bar{x}_n, \quad \bar{x}_r \in \mathcal{R}(A^T), \quad \bar{x}_n \in \mathcal{N}(A), \quad \bar{x}_r \perp \bar{x}_n.$$

Поэтому (теорема Пифагора)

$$\|\bar{x}\|^2 = \|\bar{x}_r\|^2 + \|\bar{x}_n\|^2 \geq \|\bar{x}_r\|^2.$$

Компонента \bar{x}_r сама является решением уравнения $A\bar{x}_r = \hat{z}$, так как $A\bar{x}_n = 0$ по определению нуль-пространства $\mathcal{N}(A)$. Все \bar{x} отличаются от \bar{x}_r добавлением всевозможных ортогональных компонент \bar{x}_n , причем $\|\bar{x}\| > \|\bar{x}_r\|$ при единственном условии: $\|\bar{x}_n\| \neq 0$. Наименьшее значение $\|\bar{x}\| = \|\bar{x}_r\|$ требует равенства $\|\bar{x}_n\| = 0$, т. е. достигается при единственном значении $\bar{x}_n = 0$. Тем самым доказано, что $\bar{x}_0 = \bar{x}_r$. Чтобы установить единственность вектора $\bar{x}_0 = \bar{x}_r$ с минимальной нормой, предположим, что существуют два различных \bar{x}_r^1 и \bar{x}_r^2 , оба из $\mathcal{R}(A^T)$, такие что для них выполняется (10.9):

$$A\bar{x}_r^1 = \hat{z}, \quad A\bar{x}_r^2 = \hat{z}.$$

Тогда, очевидно, имеем $A(\bar{x}_r^1 - \bar{x}_r^2) = 0$, так что $(\bar{x}_r^1 - \bar{x}_r^2) \in \mathcal{N}(A) = (\mathcal{R}(A^T))^\perp$. Оказалось, что вектор $(\bar{x}_r^1 - \bar{x}_r^2)$ ортогонален сам себе, т. е. $(\bar{x}_r^1 - \bar{x}_r^2)^T(\bar{x}_r^1 - \bar{x}_r^2) = \|\bar{x}_r^1 - \bar{x}_r^2\|^2 = 0$, и тем самым $\bar{x}_r^1 = \bar{x}_r^2$. □

ЗАМЕЧАНИЕ 10.4. Теорема 10.8 устанавливает, что при любом $z \in \mathbb{R}^n$ \bar{x}_0 может быть получен из любого вектора $y \in \mathbb{R}^n$, найденного как решение совместной системы $A^T A(A^T y) = A^T z$, по формуле $\bar{x}_0 = A^T y$.

10.5 Отыскание псевдообратной матрицы

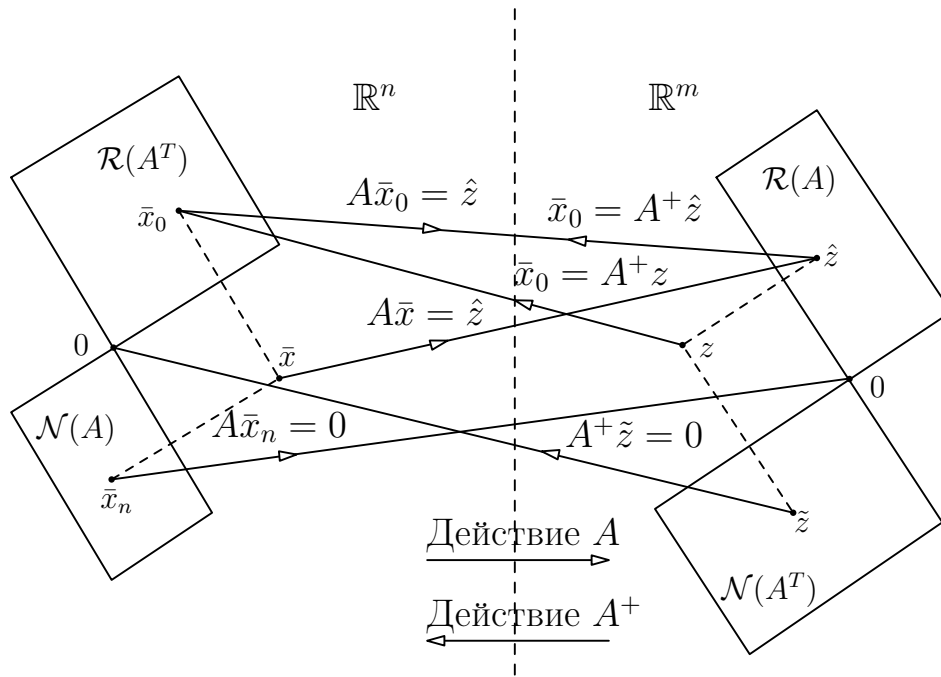
Переформулируем Определение 10.3 псевдообратной матрицы.

ОПРЕДЕЛЕНИЕ 10.6. *Псевдообратная матрица A^+ к матрице A есть такая матрица, что $\forall z (\exists \bar{x}_0, \bar{x}_0 = A^+ z)$, для которого выполнены условия:*

- (1) $A\bar{x}_0 = \hat{z}$, где \hat{z} — проекция вектора z на $\mathcal{R}(A)$: $z - \hat{z} \perp \mathcal{R}(A)$;
- (2) $\bar{x}_0 \in \mathcal{R}(A^T)$.

ПРИМЕР 10.1. [13]

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$


 Рис. 10.2. Матрица A и ее псевдообратная A^+ [13]

$$\mathcal{R}(A) = \mathcal{R}(A^T) = \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = x_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = x_1 e_1 + x_2 e_2.$$

$$\mathcal{R}(A) = \mathcal{L}(e_1, e_2).$$

$$\mathcal{R}(A^T) = \mathcal{L}(e_1, e_2).$$

$$\mathcal{N}(A) = \begin{pmatrix} 0 \\ 0 \\ x_3 \end{pmatrix} = x_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}; \quad \mathcal{N}(A) = \mathcal{L}(e_3); \quad \mathcal{N}(A) \perp \mathcal{R}(A^T).$$

$$1^\circ \text{ Проектируем } z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \text{ на } \mathcal{R}(A). \text{ Находим } \hat{z} = \begin{pmatrix} z_1 \\ z_2 \\ 0 \end{pmatrix}.$$

$$2^\circ \text{ Решаем систему } A\bar{x} = \hat{z}. \text{ Имеем } \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ 0 \end{pmatrix} \Rightarrow \bar{x} =$$

$$= \begin{pmatrix} z_1 \\ z_2 \\ \bar{x}_3 \end{pmatrix} \left. \begin{array}{l} \left\{ \leftarrow \text{фиксированные,} \right. \\ \leftarrow \text{произвольный.} \end{array} \right\}$$

3° Выбираем из \bar{x} тот $\bar{x}_0 = \begin{pmatrix} z_1 \\ z_2 \\ 0 \end{pmatrix}$, который имеет минимальную норму.

Видно, что $\bar{x}_0 \in \mathcal{R}(A^T)$, так как $\bar{x}_0 = z_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + z_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = z_1 e_1 + z_2 e_2$.

4° Находим A^+ такую, что $A^+ z = \bar{x}_0$:

$$\begin{pmatrix} A^+ \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ 0 \end{pmatrix} \implies A^+ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

ПРИМЕР 10.2. [13]

$$A = \begin{bmatrix} \mu_1 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad \mu_1 > 0, \quad \mu_2 > 0.$$

$$\mathcal{R}(A) = \mathcal{L}(e_1, e_2) \subset \mathbb{R}^3 = \mathbb{E}(e_1, e_2, e_3).$$

1° Проектируем $z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$ на $\mathcal{R}(A) \implies z_3 = 0$. Имеем

$$\hat{z} = \begin{pmatrix} z_1 \\ z_2 \\ 0 \end{pmatrix} = z_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + z_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = z_1 e_1 + z_2 e_2 \in \mathcal{R}(A).$$

2° Решаем систему $A\bar{x} = \hat{z}$:
$$\begin{pmatrix} \mu_1 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \bar{x} =$$

$$= \left\{ \begin{pmatrix} z_1/\mu_1 \\ z_2/\mu_2 \\ \bar{x}_3 \\ \bar{x}_4 \end{pmatrix} \right\} \leftarrow \begin{matrix} \text{фиксированные,} \\ \text{произвольные.} \end{matrix}$$

3° Выбираем из \bar{x} тот $\bar{x}_0 = \begin{pmatrix} z_1/\mu_1 \\ z_2/\mu_2 \\ 0 \\ 0 \end{pmatrix}$, у которого $\|\bar{x}_0\| = \min_{\bar{x}} \|\bar{x}\|$.

4° Находим A^+ такую, что $A^+z = \bar{x}_0$:

$$\begin{pmatrix} A^+ \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} z_1/\mu_1 \\ z_2/\mu_2 \\ 0 \\ 0 \end{pmatrix} \Rightarrow A^+ = \begin{pmatrix} \mu_1^{-1} & 0 & 0 \\ 0 & \mu_2^{-1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

ПРИМЕР 10.3. (Обобщающий вышеприведенные примеры 10.1 и 10.2.)

Рассмотрим класс матриц вида $\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} = \Sigma(m, n)$, где $D = \text{diag}(\mu_1, \mu_2, \dots, \mu_r)$. Имеем, на основании примеров 10.1 и 10.2, что всегда $\Sigma^+ = \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \Sigma^+(n, m)$.

ТЕОРЕМА 10.9 (О сингулярном разложении матрицы $A = A(m, n)$).

Для произвольной матрицы $A = A(m, n)$ ранга r существуют две ортогональные матрицы $Q_1 = Q_1(m, m)$ и $Q_2 = Q_2(n, n)$ и положительные действительные числа $\mu_1 \geq \mu_2 \geq \dots \geq \mu_r > 0$, такие что:

1. Справедливы равенства

$$A = Q_1 \Sigma Q_2^T, \quad \Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} = \Sigma(m, n), \quad D = \text{diag}(\mu_1, \mu_2, \dots, \mu_r), \quad (10.10)$$

причем $\mu_i^2 = \lambda_i$, где λ_i — собственные числа матрицы $A^T A$.

2. Для псевдообратной матрицы A^+ справедливо выражение

$$A^+ = Q_2 \Sigma^+ Q_1^T, \quad \Sigma^+ = \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

ОПРЕДЕЛЕНИЕ 10.7. Числа μ_i называются *сингулярными числами матрицы A* , и разложение (10.10) называется *сингулярным разложением матрицы A* .

ДОКАЗАТЕЛЬСТВО.

1. Рассмотрим матрицу $A^T A$. Она — симметрическая или эрмитова (в комплексном случае). Если A — вещественная, то $A^T A$ — симметрическая, то есть она совпадает со своей транспонированной матрицей: $(A^T A)^T = A^T A$. Если A — комплексная, то $A^* A$ — эрмитова, то есть она совпадает со своей сопряженно-транспонированной: $(A^* A)^* = A^* A$.

Фундаментальное свойство таких матриц заключается в следующем. Только эрмитовы матрицы обладают одновременно (подробнее см. ниже стр. 233):

- вещественными неотрицательными собственными значениями,
- ортонормированными собственными векторами.

Имеем: матрица $A^T A$ ($n \times n$) эрмитова.

Обозначим: $\{x_1, x_2, \dots, x_n\}$ — набор собственных векторов в \mathbb{R}^n ,

$\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ — соответствующие собственные значения.

Запишем

$$A^T A x_i = \lambda_i x_i, \quad \begin{cases} x_i^T x_j = 1, & i = j \\ x_i^T x_j = 0, & i \neq j \end{cases}, \quad \text{где } i, j = 1, 2, \dots, n.$$

Умножим скалярно на x_i :

$$x_i^T (A^T A) x_i = \lambda_i x_i^T x_i = \lambda_i \|x_i\|^2 = \lambda_i,$$

$$\|A x_i\|^2 = \lambda_i \implies \lambda_i \geq 0.$$

Пронумеруем λ_i так, чтобы $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$, а остальные $\lambda_i = 0$, $i = r + 1, r + 2, \dots, n$. Это возможно, так как $\text{rank}(A^T A) = \text{rank } A = r$. Вычислим $\mu_i = \sqrt{\lambda_i}$ для $i = 1, 2, \dots, r$. Заметим, что $\mu_i = \sqrt{\lambda_i} = 0$ для $i = r + 1, r + 2, \dots, n$. Для μ_i , $i = 1, 2, \dots, r$ снова запишем:

$$A^T A x_i = \lambda_i x_i \implies x_i^T (A^T A) x_i = \lambda_i = \mu_i^2,$$

$$\frac{x_i^T A^T A x_i}{\mu_i^2} = 1, \quad \frac{x_i^T A^T}{\mu_i} \cdot \frac{A x_i}{\mu_i} = 1, \quad A x_i \neq 0.$$

Обозначим $y_i = A x_i / \mu_i$, $i = 1, 2, \dots, r$. Тогда

$$y_i^T y_j = \frac{x_i^T A^T}{\mu_i} \cdot \frac{A x_j}{\mu_j} = \frac{x_i^T A^T A x_j}{\mu_i \mu_j} = \frac{x_i^T \lambda_j x_j}{\mu_i \mu_j} = \begin{cases} 1, & i = j = 1, 2, \dots, r, \\ 0, & i \neq j. \end{cases}$$

Следовательно, $\{y_1, y_2, \dots, y_r\}$ — ортонормированы в \mathbb{R}^m , так как $y_i \in \mathbb{R}^m$. Все y_i — линейные комбинации столбцов матрицы A : $y_i \in \mathcal{R}(A)$. Они могут быть дополнены в \mathbb{R}^m до полного ортонормированного базиса:

$$\{y_1, y_2, \dots, y_r, y_{r+1}, \dots, y_m\}.$$

Таким образом, имеем:

- в \mathbb{R}^m — ортонормированный базис из векторов $\{y_1, y_2, \dots, y_m\}$,

- в \mathbb{R}^n — ортонормированную систему векторов $\{x_1, x_2, \dots, x_n\}$.

Соберем эти векторы в две матрицы:

$$Q_1 = [y_1, y_2, \dots, y_m], \quad Q_2 = [x_1, x_2, \dots, x_n].$$

а) Имеем для $i = 1, 2, \dots, r$:

$$y_i \mu_i = Ax_i, \quad \mu_i > 0, \quad Ax_i \neq 0. \quad (10.11)$$

б) Имеем для $i = r + 1, r + 2, \dots, m$:

$$\|Ax_i\|^2 = \lambda_i = 0 \implies Ax_i = 0.$$

Поэтому в записи

$$y_i \mu_i = Ax_i \text{ следует считать } \mu_i = 0, \quad i = r + 1, r + 2, \dots, m. \quad (10.12)$$

Это означает, что (10.11) и (10.12) равносильны записи:

$$\underbrace{[y_1, y_2, \dots, y_m]}_{Q_1} \underbrace{\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}}_{\Sigma} = A \underbrace{[x_1, x_2, \dots, x_n]}_{Q_2}$$

где $D = \text{diag}(\mu_1, \mu_2, \dots, \mu_r)$, то есть

$$Q_1 \Sigma = A Q_2, \quad \text{или} \quad Q_1 \Sigma Q_2^T = A, \quad \text{или} \quad \Sigma = Q_1^T A Q_2.$$

Утверждение 1 теоремы доказано.

2. Умножение на ортогональную матрицу не изменяет евклидовой нормы вектора, поэтому имеем цепочку равенств:

$$\|Ax - b\|^2 = \|Q_1 \Sigma Q_2^T x - b\|^2 = \|\Sigma Q_2^T x - Q_1^T b\|^2 = \|\Sigma y - Q_1^T b\|^2 = \|\Sigma y - c\|^2.$$

Введем новый вектор

$$y = Q_2^T x = Q_2^{-1} x.$$

Для него $\|y\| = \|x\|$, так как Q_2^T — ортогональная матрица.

Из приведенной цепочки равенств получаем следующий вывод: отыскание нормального псевдорешения системы $Ax = b$ эквивалентно отысканию нормального псевдорешения системы $\Sigma y = c$, где $c = Q_1^T b$. Имеем эти нормальные псевдорешения: \bar{y}_0 и \bar{x}_0 . Они равны:

$$\begin{aligned} \bar{y}_0 &= \Sigma^+ c, \\ \bar{x}_0 &= A^+ b. \end{aligned}$$

В силу связи x и y имеем

$$\bar{y}_0 = Q_2^T \bar{x}_0 \implies \bar{x}_0 = Q_2 \bar{y}_0.$$

Поэтому

$$\bar{x}_0 = Q_2 \Sigma^+ c = Q_2 \Sigma^+ Q_1^T b = A^+ b.$$

Но b — любой вектор в \mathbb{R}^m . Отсюда

$$A^+ = Q_2 \Sigma^+ Q_1^T.$$

□

ПРИМЕР 10.4. Дана матрица [13] $A = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$.

Тогда $A^T A = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 25 \end{bmatrix}$. Поэтому единственный собственный вектор матрицы $A^T A$ есть: $x_1 = [1]$. Это и есть матрица Q_2 , $Q_2 = [1]$. Найдем λ_1 :

$$(A^T A)x_1 = \lambda_1 x_1 \implies [25] [1] = \lambda_1 [1] \implies \lambda_1 = 25.$$

Найдем μ_1 :

$$\mu_1 = \sqrt{\lambda_1} = \sqrt{25} = 5.$$

Найдем вектор y_1 :

$$y_1 = \frac{Ax_1}{\mu_1} = \frac{1}{5} \begin{bmatrix} 3 \\ 4 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 3/5 \\ 4/5 \end{bmatrix}.$$

y_1 — есть первый столбец матрицы $Q_1 = [y_1, y_2]$. Вторым столбцом y_2 матрицы Q_1 должен быть выбран так, чтобы быть в R^2 ортогональным вектору $y_1 = \begin{bmatrix} 3/5 \\ 4/5 \end{bmatrix}$.

В общем виде $y_2 = \begin{bmatrix} y_{12} \\ y_{22} \end{bmatrix}$. Из условия ортогональности $y_1^T y_2 = 0$ имеем

$$\frac{3}{5}y_{12} + \frac{4}{5}y_{22} = 0.$$

Кроме того, y_2 должен быть ортонормирован, то есть $y_2^T y_2 = 1$. Отсюда следует

$$y_{12}^2 + y_{22}^2 = 1.$$

Из первого условия имеем $y_{12} = -y_{22}4/3$. Подставляя значение y_{12} во второе условие, выводим y_{22} :

$$\begin{aligned} y_{22}^2(4/3)^2 + y_{22}^2 &= 1 \implies y_{22}^2[1 + (4/3)^2] = 1 \implies \\ \implies y_{22}^2 &= \frac{1}{1 + (4/3)^2} = \frac{1}{1 + 16/9} = \frac{9}{25}. \end{aligned}$$

Можно выбрать одно из двух значений:

(а) $y_{22} = 3/5$ либо

(б) $y_{22} = -3/5$.

Отсюда, соответственно,

(а) $y_{12} = (-3/5)(4/3) = -4/5$, либо

(б) $y_{12} = -(-3/5)(4/3) = 4/5$.

Следовательно, возможны два варианта:

Вариант 1

$$Q_1 = \begin{bmatrix} 3/5 & -4/5 \\ 4/5 & 3/5 \end{bmatrix},$$

Вариант 2

$$Q_1 = \begin{bmatrix} 3/5 & 4/5 \\ 4/5 & -3/5 \end{bmatrix}.$$

В варианте 1 имеем

$$\Sigma = Q_1^T A Q_2 = \begin{bmatrix} 3/5 & 4/5 \\ -4/5 & 3/5 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 9/5 + 16/5 \\ -12/5 + 12/5 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}.$$

В варианте 2 имеем

$$\Sigma = Q_1^T A Q_2 = \begin{bmatrix} 3/5 & 4/5 \\ 4/5 & -3/5 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 9/5 + 16/5 \\ 12/5 - 12/5 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}.$$

В любом варианте $Q_1^T A Q_2 = \Sigma$, где $\Sigma = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$, что и должно быть.

Теперь найдем Σ^+ и A^+ . Имеем (из примеров 10.1, 10.2 и 10.3): $\Sigma^+ = \begin{bmatrix} 1/5 & | & 0 \end{bmatrix}$. Поэтому ищем $A^+ = Q_2 \Sigma^+ Q_1^T$.

В варианте 1

$$A^+ = Q_2 \Sigma^+ Q_1^T = \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 1/5 & 0 \end{bmatrix} \begin{bmatrix} 3/5 & 4/5 \\ -4/5 & 3/5 \end{bmatrix} = \begin{bmatrix} 3/25 & 4/25 \end{bmatrix}.$$

В варианте 2

$$A^+ = Q_2 \Sigma^+ Q_1^T = \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 1/5 & 0 \end{bmatrix} \begin{bmatrix} 3/5 & 4/5 \\ 4/5 & -3/5 \end{bmatrix} = \begin{bmatrix} 3/25 & 4/25 \end{bmatrix}.$$

Здесь также имеем

$$A^+ A = \begin{bmatrix} 3/25 & 4/25 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix} = I.$$

Это произошло в силу того, что $\text{rank } A = r = n = 1$. Как мы знаем, в этом случае

$$A^+ = (A^T A)^{-1} A^T = \begin{bmatrix} 25 \end{bmatrix}^{-1} \begin{bmatrix} 3 & 4 \end{bmatrix} = \begin{bmatrix} 3/25 & 4/25 \end{bmatrix}.$$

Вывод из примера 10.4. Пользоваться сингулярным разложением $A = Q_1 \Sigma Q_2^T$, чтобы находить A^+ , непросто. Для этого нужно:

1. Найти собственные векторы x_1, x_2, \dots, x_n матрицы $A^T A$.
Тогда $Q_2 = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$.
2. Найти собственные значения матрицы $A^T A$: $\lambda_1 > \lambda_2 > \dots > \lambda_n \geq 0$.
3. Оценить ранг r матрицы A , т. е. разграничить $\{\lambda_1 > \lambda_2 > \dots > \lambda_r > 0\}$ и $\{\lambda_{r+1} = \lambda_{r+2} = \dots = \lambda_n = 0\}$ и образовать $\mu_i = \sqrt{\lambda_i}$, $D = \text{diag}(\mu_1, \mu_2, \dots, \mu_r)$.
4. Найти $y_i = Ax_i/\mu_i$, где $\mu_i = \sqrt{\lambda_i}$, $i = 1, 2, \dots, r$.
5. Доопределить каким-либо образом систему $\{y_i\}$ до ортонормированного базиса в \mathbb{R}^m .
6. Образовать $Q_1 = \begin{bmatrix} y_1 & y_2 & \dots & y_m \end{bmatrix}$.

В качестве альтернативного решения для нахождения A^+ рассмотрим на следующем примере так называемое «усеченное» $\bar{L}\bar{U}$ -разложение матрицы.

ПРИМЕР 10.5. Дана матрица [13]

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 5 \\ 3 & 7 \end{bmatrix}, \quad \text{rank } A = 2 = n.$$

Выполним полное LU -разложение и затем «усеченное» $\bar{L}\bar{U}$ -разложение:

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 5 \\ 3 & 7 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}_L \underbrace{\begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_U = \underbrace{\begin{bmatrix} 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}}_{\bar{L}} \underbrace{\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}}_{\bar{U}} = \bar{L}\bar{U},$$

$$\text{rank } \bar{L} = \text{rank } \bar{U} = \text{rank } A = 2.$$

В общем случае: пусть $\text{rank } A = r$, $A = LU$, тогда:

1. L — всегда квадратная с единичной диагональю, т.е. $L = L(m, m)$ и $\det L = 1$,
2. U — имеет $(m - r)$ нулевых нижних строк.

Отбросим последние $(m - r)$ строк в U и последние $(m - r)$ столбцов в L (в этом и заключается переход к «усеченному» разложению), тогда получим:

$$A = \bar{L}\bar{U}, \quad \bar{L} = \bar{L}(m, r), \quad \bar{U} = \bar{U}(r, n),$$

$$\text{rank } \bar{L} = r = \text{rank } \bar{U} = \text{rank } A.$$

Свойства \bar{L} и \bar{U} (так как $A = \bar{L}\bar{U}$ и $\text{rank } A = \text{rank } \bar{L} = r$):

- 1° \bar{U} : $\mathcal{R}(\bar{U}^T) = \mathcal{R}(A^T)$ — совпадение пространств строк,
- 2° \bar{L} : $\mathcal{R}(\bar{L}) = \mathcal{R}(A)$ — совпадение пространств столбцов.

ТЕОРЕМА 10.10 ([13]).

$$A^+ = \bar{U}^T(\bar{U}\bar{U}^T)^{-1}(\bar{L}^T\bar{L})^{-1}\bar{L}^T.$$

ДОКАЗАТЕЛЬСТВО. Пусть b — произвольный вектор в \mathbb{R}^m . Рассмотрим вектор

$$y = [\bar{U}^T(\bar{U}\bar{U}^T)^{-1}(\bar{L}^T\bar{L})^{-1}\bar{L}^T] b.$$

Вектор $y \in \mathcal{R}(U^T) = \mathcal{R}(A^T)$. Умножим y слева на $A = \bar{L}\bar{U}$:

$$Ay = \bar{L}\bar{U} [\bar{U}^T(\bar{U}\bar{U}^T)^{-1}(\bar{L}^T\bar{L})^{-1}\bar{L}^T] b = \bar{L}(\bar{L}^T\bar{L})^{-1}\bar{L}^T b.$$

По определению, $\bar{L}(\bar{L}^T\bar{L})^{-1}\bar{L}^T$, есть матрица проектирования на $\mathcal{R}(\bar{L})$, т.е. на $\mathcal{R}(A)$, поэтому

$$A [\bar{U}^T(\bar{U}\bar{U}^T)^{-1}(\bar{L}^T\bar{L})^{-1}\bar{L}^T] b = p,$$

где p — проекция b на $\mathcal{R}(A)$.

Таким образом, вектор y удовлетворяет следующим условиям:

1. $y \in (A^T)$.
2. $Ay = p$, где p — проекция любого вектора b на $\mathcal{R}(A)$.

$$3. y = [\bar{U}^T(\bar{U}\bar{U}^T)^{-1}(\bar{L}^T\bar{L})^{-1}\bar{L}^T] b.$$

$$\text{Отсюда } [\bar{U}^T(\bar{U}\bar{U}^T)^{-1}(\bar{L}^T\bar{L})^{-1}\bar{L}^T] = A^+.$$

□

Приведем заключительные примеры для лучшего понимания структуры и свойств псевдообратной матрицы A^+ .

ПРИМЕР 10.6. Если $A = A(1, 1)$, то

$$A^+ = \begin{cases} 0, & \text{если } A = 0, \\ 1/A, & \text{если } A \neq 0. \end{cases}$$

ПРИМЕР 10.7. Если $A = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, то

$$A^+ = \text{diag}(\lambda_1^+, \lambda_2^+, \dots, \lambda_n^+), \quad \text{где } \lambda_j^+ = \begin{cases} 0, & \text{если } \lambda_j = 0, \\ 1/\lambda_j, & \text{если } \lambda_j \neq 0. \end{cases}$$

ПРИМЕР 10.8. Матрицы

$$A_1 = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{и} \quad A_2 = \begin{pmatrix} 4 & 0 \\ 0 & 10^{-10} \end{pmatrix}$$

почти одинаковы. Однако, их псевдообратные матрицы

$$A_1^+ = \begin{pmatrix} 1/4 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{и} \quad A_2^+ = \begin{pmatrix} 1/4 & 0 \\ 0 & 10^{10} \end{pmatrix}$$

сильно отличаются.

Эти примеры, имеющие общее объяснение в теореме о сингулярном разложении, говорят о том, что *операция псевдообращения — разрывная*. Действительно, функция

$$\lambda^+ = \begin{cases} 1/\lambda, & \text{если } \lambda \neq 0, \\ 0, & \text{если } \lambda = 0 \end{cases}$$

имеет разрыв в точке $\lambda = 0$.

Отсутствие непрерывности операции псевдообращения приводит к серьезным вычислительным трудностям и, возможно, к большим вычислительным ошибкам при решении задачи МНК, особенно, если $r < n$ или $r < m$.

К счастью, на практике задача МНК обычно возникает при $m > n = r$, т. е. с матрицами $A = A(m, n)$ максимального столбцового ранга для переопределенных систем $Ax = z$. В этом случае $A^+ = (A^T A)^{-1} A^T$ и МНК-решение \bar{x} единственно и равно \bar{x}_0 .

10.6 Основные теоремы по МНК и псевдоинверсии

ТЕОРЕМА 10.11 ([1]). Пусть $z \in \mathbb{R}^m$ и $A = A(m, n)$. Тогда:

- (1) найдется $\bar{x}_0 \in \mathbb{R}^n$ единственный, если это вектор с минимальной нормой, минимизирующий $\|z - Ax\|^2$,
- (2) вектор \bar{x}_0 является единственным вектором из $\mathcal{R}(A^T)$, удовлетворяющим уравнению $A\bar{x} = \hat{z}$, где $\hat{z} = p$ — проекция вектора z на $\mathcal{R}(A)$.

ТЕОРЕМА 10.12 ([1]). Среди всех векторов \bar{x} , минимизирующих $\|z - Ax\|^2$, вектор \bar{x}_0 , имеющий минимальную норму, является единственным вектором вида

$$\bar{x}_0 = A^T y,$$

удовлетворяющим уравнению

$$A^T A \bar{x}_0 = A^T z.$$

Иными словами, вектор \bar{x}_0 может быть получен с помощью любого вектора y_0 , удовлетворяющего уравнению

$$A^T A A^T y = A^T z,$$

по формуле $\bar{x}_0 = A^T y$.

ТЕОРЕМА 10.13 ([1]). Для всякой матрицы $A = A(m, n)$ существует псевдообратная матрица $A^+ = A^+(n, m)$, такая что для произвольного вектора $z \in \mathbb{R}^m$

$$\bar{x}_0 = A^+ z$$

является вектором с минимальной нормой среди всех векторов \bar{x} , минимизирующих $\|z - Ax\|^2$.

ТЕОРЕМА 10.14 ([1]). Для любой матрицы $A = A(m, n)$ псевдообратная матрица A^+ обладает свойствами:

- | | |
|---------------------------------|---------------------------------|
| (1) $A^+ = (A^T A)^+ A^T$, | (6) $(A A^T)^+ = (A^T)^+ A^+$, |
| (2) $(A^T)^+ = (A^+)^T$, | (7) $A A^+ A = A$, |
| (3) $A^+ = A^T (A A^T)^+$, | (8) $A^+ A A^+ = A^+$, |
| (4) $(A^+)^+ = A$, | (9) $A A^+ = (A A^+)^T$, |
| (5) $(A^T A)^+ = A^+ (A^T)^+$, | (10) $A^+ A = (A^+ A)^T$. |

Основное правило обращения произведения матриц, $(BA)^{-1} = A^{-1}B^{-1}$, не выполняется для псевдообратных матриц, то есть

$$(BA)^+ \neq A^+B^+.$$

ТЕОРЕМА 10.15 ([1]). Для любой симметрической матрицы $A = A(n, n)$ с действительными элементами предельная матрица

$$P_A = \lim_{\delta \rightarrow 0} (A + \delta I)^{-1} A = \lim_{\delta \rightarrow 0} A(A + \delta I)^{-1}$$

существует. Она является матрицей проектирования на $\mathcal{R}(A) = \mathcal{R}(A^T)$. Это означает, что для любого вектора $z \in \mathbb{R}^n$ вектор

$$\hat{z} = P_A z$$

является проекцией z на $\mathcal{R}(A) = \mathcal{R}(A^T)$ — на пространство строк или столбцов данной симметрической матрицы.

ТЕОРЕМА 10.16 ([1]).

- (1) Для произвольных $z \in \mathbb{R}^m$ и $A = A(m, n)$ вектор \bar{x} минимизирует критерий $\|z - Ax\|^2$ тогда и только тогда, когда \bar{x} имеет вид:

$$\bar{x} = A^+ z + (I - A^+ A)y$$

для некоторого $y \in \mathbb{R}^n$.

- (2) Вектор \bar{x} , минимизирующий $\|z - Ax\|^2$, является единственным тогда и только тогда, когда $A^+ A = I$. Последнее условие выполняется тогда и только тогда, когда только нулевой вектор составляет ядро (нуль-пространство) матрицы A , т. е. при $r = n$.
- (3) Уравнение $Ax = z$ имеет решение тогда и только тогда, когда $A^+ A z = z$. Это выполняется в том и только том случае, когда $z \in \mathcal{R}(A)$. Вектор x является решением уравнения $Ax = z$ тогда и только тогда, когда он задается в виде

$$x = A^+ z + (I - A^+ A)y$$

для произвольного $y \in \mathbb{R}^n$. Это решение единственно (и тогда оно равно $A^+ z$), если и только если

$$AA^+ z = z \quad \text{и} \quad A^+ A = I.$$

10.7 Вычисление матриц проектирования

Дана матрица A . Возьмем $B = A^T$. Имеем

$$\left. \begin{array}{l} \mathcal{R}(A^T) \\ \mathcal{R}(B) \end{array} \right\} \begin{array}{l} \text{— пространство строк матрицы } A, \\ \text{— пространство столбцов матрицы } A^T. \end{array}$$

ОПРЕДЕЛЕНИЕ 10.8. Матрица проектирования P на $\mathcal{R}(B)$ есть такая матрица, которая обладает свойством:

$$(b - Pb) \perp \mathcal{R}(B),$$

где $Pb = \hat{b}$ — проекция вектора b на $\mathcal{R}(B)$, $(b - Pb) = \tilde{b}$ — перпендикуляр к $\mathcal{R}(B)$.

Запишем

$$\tilde{b} = b - Pb = (I - P)b,$$

где $I - P$ — матрица проектирования на $\mathcal{R}^\perp(B) = \mathcal{N}(B^T) = \mathcal{N}(A)$.

Из определения псевдообратной матрицы (см. определение 10.5) следует, что матрица проектирования P вектора b на $\mathcal{R}(A)$ в общем случае такова:

$$Pb = p, \quad Pb = A\bar{x}_0 = AA^+b \implies P = AA^+.$$

Если взять $B = A^T$ и проектировать вектор b на $\mathcal{R}(B)$, то мы должны взять матрицу проектирования в виде $P = BB^+$. Но $B^+ = (A^T)^+ = (A^+)^T$, следовательно $P = A^T(A^+)^T$.

Если $A = A(m, n)$, то P имеет размеры: $(n \times m)(n \times m)^T = (n \times n)$. Если $B = B(n, m)$, то $B^+ = B^+(m, n)$, тогда P имеет размеры: $(n \times m)(m \times n) = (n \times n)$.

Выводы:

1. Если ищут матрицу вида $P = I - A^T(AA^T)^{-1}A$, то это есть матрица проектирования любого вектора на ядро, т. е. на нуль-пространство $\mathcal{N}(A)$ матрицы A . Но в таком виде ее можно определить, только если $(AA^T)^{-1}$ существует, т. е. если $\text{rank } A = m$ (A имеет полный строчный ранг).
2. Если $\text{rank } A = r < m$, что возможно иногда при $m \leq n$, то $(AA^T)^{-1}$ не существует. В этом случае для этой же матрицы P справедливо

наиболее общее выражение, а именно: $P = I - A^T(A^+)^T$, где также можно иметь в виду, что всегда $(A^+)^T = (A^T)^+$. Это означает, что нужно уметь отыскивать A^+ . Для этого есть различные, не очень простые, вычислительные методы (см. книгу [1], а также книгу [13]).

ПРИМЕР 10.9. Определение проектора $P_A = AA^+$. Дано:

$$A = A_1 = \begin{bmatrix} 1 & -1 & 0 & 1 \\ 2 & -2 & -1 & 3 \\ -1 & 1 & -2 & 1 \end{bmatrix} = A(n, m) = A(3, 4),$$

$$n = 3, \quad m = 4, \quad r = \text{rank}(A) = 2 < n = 3.$$

Делаем LU -разложение:

$$A = \underbrace{\begin{bmatrix} 1 & -1 & 0 & 1 \\ 2 & -2 & -1 & 3 \\ -1 & 1 & -2 & 1 \end{bmatrix}}_L \underbrace{\begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_U.$$

Находим усеченное LU -разложение:

$$A = \bar{L}\bar{U} = \underbrace{\begin{bmatrix} 1 & -1 & 0 & 1 \\ 2 & -2 & -1 & 3 \\ -1 & 1 & -2 & 1 \end{bmatrix}}_{\bar{L}} \underbrace{\begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\bar{U}}.$$

Отсюда следует:

$$A^+ = \bar{U}^+ \bar{L}^+ = \underbrace{\bar{U}^T (\bar{U} \bar{U}^T)^{-1}}_{\bar{U}^+} \underbrace{(\bar{L}^T \bar{L})^{-1} \bar{L}^T}_{\bar{L}^+} = \frac{1}{150} \begin{bmatrix} 10 & 14 & -22 \\ -10 & -14 & 22 \\ 5 & -8 & -41 \\ 5 & 22 & 19 \end{bmatrix}.$$

Найдем матрицы проектирования как на пространство строк матрицы A , т. е. на $\mathcal{R}(A^T)$, так и на пространство столбцов матрицы A , т. е. на $\mathcal{R}(A)$.

1. Матрица проектирования на $\mathcal{R}(A^T)$:

$$P_A = A^T(A^+)^T = \begin{bmatrix} 1 & 2 & -1 \\ -1 & -2 & 1 \\ 0 & -1 & -2 \\ 1 & 3 & 1 \end{bmatrix} \frac{1}{150} \begin{bmatrix} 10 & -10 & 5 & 5 \\ 14 & -14 & -8 & 22 \\ -22 & 22 & -41 & 19 \end{bmatrix} =$$

$$= \frac{1}{150} \begin{bmatrix} 60 & -60 & 30 & 30 \\ -60 & 60 & -30 & -30 \\ 30 & -30 & 90 & -60 \\ 30 & -30 & -60 & 90 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -2 & 2 & -1 & -1 \\ 1 & -1 & 3 & -2 \\ 1 & -1 & -2 & 3 \end{bmatrix}.$$

$$I - P_A = I - A^T(A^+)^T = \frac{1}{5} \begin{bmatrix} 3 & -2 & 1 & 1 \\ -2 & 3 & -1 & -1 \\ 1 & -1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix}.$$

Здесь: $\begin{cases} P_A & \text{проектирует на } \mathcal{R}(A^T), \\ I - P_A & \text{проектирует на } \mathcal{N}(A). \end{cases}$

Найдем базис пространства $\mathcal{N}(A)$. Он образован следующими двумя векторами (так как $n - r = 2$):

$$b_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{и} \quad b_2 = \begin{pmatrix} -1 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

Любой вектор в $\mathcal{N}(A)$ задается в виде:

$$y_1 \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + y_2 \begin{pmatrix} -1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \quad \text{где } y_1, y_2 \text{ — числа.}$$

2. В то же время матрица проектирования на $\mathcal{R}(A)$:

$$P_A = AA^+ = \begin{bmatrix} 1 & -1 & 0 & 1 \\ 2 & -2 & -1 & 3 \\ -1 & 1 & -2 & 1 \end{bmatrix} \frac{1}{150} \begin{bmatrix} 10 & 14 & -22 \\ -10 & -14 & 22 \\ 5 & -8 & -41 \\ 5 & 22 & 19 \end{bmatrix} =$$

$$= \frac{1}{150} \begin{bmatrix} 25 & 50 & -25 \\ 50 & 130 & 10 \\ -25 & 10 & 145 \end{bmatrix} = \frac{1}{30} \begin{bmatrix} 5 & 10 & -5 \\ 10 & 26 & 2 \\ -5 & 2 & 29 \end{bmatrix}.$$

$$I - P_A = I - AA^+ = \frac{1}{30} \begin{bmatrix} 25 & -10 & 5 \\ -10 & 4 & -2 \\ 5 & -2 & 1 \end{bmatrix}.$$

Здесь: $\begin{cases} P_A & \text{проектирует на } \mathcal{R}(A), \\ I - P_A & \text{проектирует на } \mathcal{N}(A^T) = \mathcal{R}(A)^\perp. \end{cases}$

Найдем базис пространства $\mathcal{R}(A)$:

$$v_1 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ -1 \\ -2 \end{pmatrix}.$$

Эти два вектора оказались взаимно ортогональны:

$$v_1^T v_2 = \begin{pmatrix} 1 & 2 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \\ -2 \end{pmatrix} = -2 + 2 = 0.$$

Найдем базис пространства $\mathcal{N}(A^T)$. Это пространство определяется как совокупность векторов y , таких что: $y^T A = 0$. Или иначе: $y^T v_1 = 0$ и $y^T v_2 = 0$. Отсюда найдем:

$$y = \begin{pmatrix} 5 \\ -2 \\ 1 \end{pmatrix}, \quad \text{если выбрать } y_3 = 1.$$

Действительно,

$$\begin{pmatrix} 5 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} = 0 \quad \text{и} \quad \begin{pmatrix} 5 & -2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \\ -2 \end{pmatrix} = 0.$$

Базис пространства $\mathcal{N}(A^T)$ состоит из одного вектора. Возьмем в качестве базисного вектора $v_3 = \frac{1}{2}y$, т. е. вектор

$$v_3 = \begin{pmatrix} 2.5 \\ -1 \\ 0.5 \end{pmatrix}.$$

В качестве произвольного вектора для проектирования возьмем (рис. 10.3)

$$z = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix} = 6 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

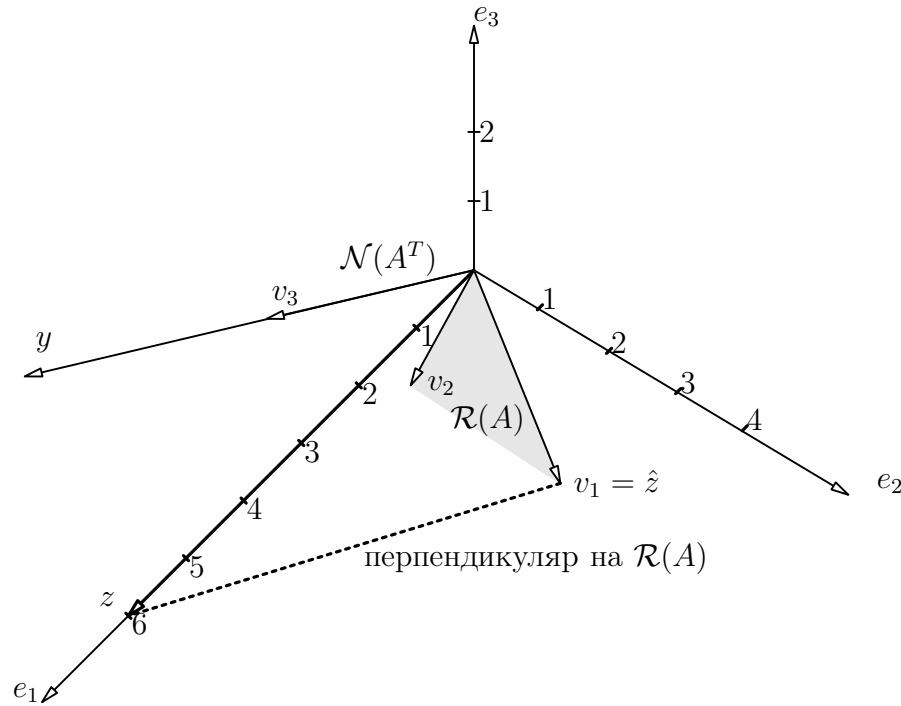


Рис. 10.3. Проектирование вектора на линейное подпространство

Найдем его проекцию на $\mathcal{R}(A)$:

$$\hat{z} = P_A z = \frac{1}{30} \begin{pmatrix} 5 & 10 & -5 \\ 10 & 26 & 2 \\ -5 & 2 & 29 \end{pmatrix} 6 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} = v_1 \quad (\text{совпало с } v_1).$$

Найдем его проекцию на $\mathcal{N}(A^T)$:

$$\tilde{z} = z - \hat{z} = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} = \begin{pmatrix} 5 \\ -2 \\ 1 \end{pmatrix} = y \quad (\text{совпало с } y).$$

ПРИМЕР 10.10.

$$A = A_2 = \begin{bmatrix} 1 & 2 & -1 \\ -1 & -2 & 1 \\ 0 & -1 & -2 \\ 1 & 3 & 1 \end{bmatrix} = A(n, m) = A(4, 3),$$

$$n = 4, \quad m = 3, \quad r = \text{rank}(A) = 2 < n.$$

По сравнению с примером 10.9 данная матрица совпадает с транспонированной матрицей примера 10.9. Используем свойство $(A^+)^T = (A^T)^+$ для

нахождения псевдообратной матрицы для нашего примера:

$$A^+ = \frac{1}{150} \begin{bmatrix} 10 & -10 & 5 & 5 \\ 14 & -14 & -8 & 22 \\ -22 & 22 & -41 & 19 \end{bmatrix}.$$

Отсюда, найдем матрицу проектирования на пространство строк матрицы A , т. е. на $\mathcal{R}(A^T)$:

$$\begin{aligned} P_A = A^T(A^+)^T &= \begin{bmatrix} 1 & -1 & 0 & 1 \\ 2 & -2 & -1 & 3 \\ -1 & 1 & -2 & 1 \end{bmatrix} \frac{1}{150} \begin{bmatrix} 10 & 14 & -22 \\ -10 & -14 & 22 \\ 5 & -8 & -41 \\ 5 & 22 & 19 \end{bmatrix} = \\ &= \frac{1}{150} \begin{bmatrix} 25 & 50 & -25 \\ 50 & 130 & 10 \\ -25 & 10 & 145 \end{bmatrix} = \frac{1}{30} \begin{bmatrix} 5 & 10 & -5 \\ 10 & 26 & 2 \\ -5 & 2 & 29 \end{bmatrix}. \end{aligned}$$

Следовательно,

$$I - P_A = I - A^T(A^+)^T = \frac{1}{30} \begin{bmatrix} 25 & -10 & 5 \\ -10 & 4 & -2 \\ 5 & -2 & 1 \end{bmatrix}.$$

Здесь:

$$\begin{cases} P_A & \text{проектирует на пространство строк матрицы } A, \mathcal{R}(A^T), \\ I - P_A & \text{проектирует на нуль-пространство матрицы } A, \mathcal{N}(A). \end{cases}$$

Геометрическая интерпретация. Для примера 10.9 построить картинку невозможно (так как там $b \in R^4$), а для примера 10.10 — можно (в этом случае $b \in R^3$). Существует связь матриц двух последних примеров, а именно: $A_1 = A_2^T$. Соответственно, имеем

$$\begin{array}{l|l} \mathcal{R}(A_1^T) = \mathcal{R}(A_2) & \mathcal{R}(A_2^T) = \mathcal{R}(A_1) \\ \mathcal{N}(A_1) = \mathcal{N}(A_2^T) & \mathcal{N}(A_2) = \mathcal{N}(A_1^T) \end{array}$$

Из примера 10.9 имеем матрицу P_1 , которая проектирует на $\mathcal{N}(A_1)$, и матрицу $I - P_1$, которая проектирует на $\mathcal{R}(A_1^T)$. Из примера 10.10 имеем матрицу P_2 , которая проектирует на $\mathcal{N}(A_1^T)$, и матрицу $I - P_2$, которая проектирует на $\mathcal{R}(A_1)$ (см. рис. 10.2 на стр. 213).

10.8 Рекурсия в задаче МНК

Постановка вопроса. Даны матрица $A = A(m, n)$ и вектор $z \in \mathbb{R}^m$. Требуется найти единственный вектор \bar{x}_0 с минимальной нормой, минимизирующий $\|z - Ax\|^2$.

Можно ли искать его последовательно?

- 1° Известно, что $A\bar{x}_0 = \hat{z}$, где $\hat{z} \in \mathcal{R}(A)$, $z - \hat{z} \perp \mathcal{R}(A)$, $\bar{x}_0 \in \mathcal{R}(A^T)$. Нормальное псевдорешение \bar{x}_0 несовместной системы $Ax = z$ удовлетворяет равенству

$$\bar{x}_0 = A^+ z,$$

где A^+ — псевдообратная матрица.

- 2° Произвольно расцепим матрицу A на блоки, $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ и, соответственно, вектор z на подвекторы, $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$, так чтобы $A_1 = A_1(k, n)$, $A_2 = A_2(s, n)$, $k + s = m$, $z_1 \in \mathbb{R}^k$, $z_2 \in \mathbb{R}^s$.

- 3° На первом шаге найдем нормальное псевдорешение только первой системы $Ax_1 = z_1$. Оно равно $\tilde{x}_0 = A_1^+ z_1$. Проекция вектора z_1 на $\mathcal{R}(A_1)$:

$$\hat{z}_1 = A_1 \tilde{x}_0 = (A_1 A_1^+) z_1.$$

- 4° Рассмотрим систему

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x = \begin{bmatrix} \hat{z}_1 \\ z_2 \end{bmatrix}, \quad (10.13)$$

которая отличается от исходной системы

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (10.14)$$

тем, что вместо z_1 используется \hat{z}_1 . Для нее нормальное псевдорешение обозначим \hat{x}_0 . Оно равно

$$\hat{x}_0 = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}^+ \begin{bmatrix} \hat{z}_1 \\ z_2 \end{bmatrix}.$$

Вопрос: верно ли равенство $\hat{x}_0 = \bar{x}_0$? Напомним, что

$$\bar{x}_0 = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}^+ \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}.$$

Решение вопроса:

Для системы (10.13) запишем нормальные уравнения:

$$\begin{aligned} \begin{bmatrix} A_1^T & A_2^T \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x &= \begin{bmatrix} A_1^T & A_2^T \end{bmatrix} \begin{bmatrix} \hat{z}_1 \\ z_2 \end{bmatrix}, \\ \begin{bmatrix} A_1^T A_1 + A_2^T A_2 \end{bmatrix} x &= A_1^T \hat{z}_1 + A_2^T z_2. \end{aligned} \quad (10.15)$$

Затем для системы (10.14) запишем нормальные уравнения:

$$\begin{aligned} \begin{bmatrix} A_1^T & A_2^T \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x &= \begin{bmatrix} A_1^T & A_2^T \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \\ \begin{bmatrix} A_1^T A_1 + A_2^T A_2 \end{bmatrix} x &= A_1^T z_1 + A_2^T z_2. \end{aligned} \quad (10.16)$$

Сравним в правой части $A_1^T \hat{z}_1$ и $A_1^T z_1$. Имеем

$$A_1^T \hat{z}_1 = A_1^T (A_1 A_1^+) z_1.$$

Более простое доказательство:

$$A^T (A A^+) = A^T (A A^+)^T = (A A^+ A)^T = A^T.$$

ЛЕММА 10.2. Докажем, что

$$A^T = A^T (A A^+).$$

ДОКАЗАТЕЛЬСТВО. Имеем (см. стр. 220):

$$A = \bar{L} \bar{U}, \quad A^T = \bar{U}^T \bar{L}^T, \quad A^+ = \bar{U}^+ \bar{L}^+, \quad A A^+ = \bar{L} \bar{U} \bar{U}^+ \bar{L}^+,$$

$$A^T (A A^+) = \bar{U}^T \bar{L}^T \cdot \bar{L} \bar{U} \cdot \bar{U}^+ \bar{L}^+ = \bar{U}^T \bar{L}^T = A^T.$$

Лемма доказана. Поэтому $A_1^T \hat{z}_1 = A_1^T z_1$.

Докажем то же самое другим способом:

$$A_1^T (\hat{z}_1) = A_1^T (z - \tilde{z}) = A_1^T z - A_1^T \tilde{z}, \quad \text{где } \tilde{z} \in \mathcal{N}(A_1^T), \quad \text{т. е. } A_1^T \tilde{z} = 0.$$

Поэтому $A_1^T \hat{z}_1 = A_1^T z_1$. Таким образом, правые части уравнений (10.15) и (10.16) совпадают. Поэтому решения уравнений (10.15) и (10.16) одинаковы, то есть

$$\hat{x}_0 = \bar{x}_0.$$

10.9 Основные свойства симметрических / эрмитовых матриц

Свойства симметрических матриц, $A = A^T$, в теории МНК служат основой многих результатов. В алгебре эти свойства обобщены на случай *эрмитовых матриц*, т. е. матриц A , в которых понятие симметричности в отношении комплекснозначных элементов расширено добавлением требования комплексной сопряженности $a_{ij} = \bar{a}_{ji}$, что выражается записью $A = A^*$, где A^* — сопряженно транспонированная матрица к A .

Напомним некоторые определения, а затем приведем четыре основные свойства эрмитовых матриц.

ОПРЕДЕЛЕНИЕ 10.9. Число λ называется *собственным значением* $(n \times n)$ -матрицы A с соответствующим *собственным вектором* x , если $Ax = \lambda x$, т. е. λ есть любой из n корней уравнения

$$\det(A - \lambda I) = 0.$$

Это — *характеристическое уравнение* для матрицы A .

ОПРЕДЕЛЕНИЕ 10.10. *Ортогональной матрицей* называется квадратная вещественная матрица Q , столбцы которой ортонормированы, то есть выполнено условие $Q^T Q = I$.

УПРАЖНЕНИЕ 10.1. Матрица Q^T ортогональна тогда и только тогда, когда ортогональна матрица Q .

ОПРЕДЕЛЕНИЕ 10.11. *Унитарной матрицей* называется квадратная матрица U , столбцы которой ортонормированы с добавлением требования комплексной сопряженности элементов столбцов: $U^* U = I$.

УПРАЖНЕНИЕ 10.2. Матрица U^* унитарна тогда и только тогда, когда унитарна матрица U .

ТЕОРЕМА 10.17. Если $A = A^*$, то для всех комплексных векторов x число $x^* A x$ вещественно.

ДОКАЗАТЕЛЬСТВО. Вычислите $(x^* A x)^*$ как эрмитову (1×1) -матрицу и примите во внимание, что в любой эрмитовой матрице диагональные элементы должны быть вещественными.

ТЕОРЕМА 10.18. Каждое собственное значение эрмитовой матрицы вещественно.

ДОКАЗАТЕЛЬСТВО. Умножьте $Ax = \lambda x$ слева на x^* и используйте предыдущую теорему 10.17.

ТЕОРЕМА 10.19 ([13]). Собственные векторы эрмитовой матрицы, отвечающие различным собственным значениям, взаимно ортогональны.

ДОКАЗАТЕЛЬСТВО. Пусть $\lambda \neq \mu$ и $Ax = \lambda x$, $Ay = \mu y$. Возьмем $(1 \times n)$ -матрицу $x^*A^* = \bar{\lambda}x^*$, сопряженно транспонированную к $Ax = \lambda x$, заменим $A^* = A$ и учтем, что $\lambda = \bar{\lambda}$ (по теореме 10.18), т.е. запишем $x^*A = \lambda x^*$. Умножая это равенство справа на y , а равенство $Ay = \mu y$ слева на x^* , получим

$$x^*Ay = \lambda x^*y, \quad x^*Ay = \mu x^*y.$$

Следовательно, $\lambda x^*y = \mu x^*y$, а так как $\lambda \neq \mu$, то $x^*y = 0$, т.е. вектор x ортогонален к y . \square

Этот результат является основным по своей важности. Разумеется, любые кратные собственным векторам x/α и y/β , равным образом остаются собственными векторами. При выборе $\alpha = \|x\|$, $\beta = \|y\|$ и т.п. мы имеем возможность нормировать все собственные векторы к единичной длине и, таким образом, далее использовать ортонормированные собственные векторы $\{u_1, u_2, \dots, u_n\}$ эрмитовой матрицы A . Запишем их в виде матрицы $U = [u_1 | u_2 | \dots | u_n]$, которая, по определению, является унитарной: $U^*U = I$. Имеем $Au_i = \lambda_i u_i$, по определению собственных значений λ_i , $i = 1, 2, \dots, n$. Эту систему уравнений перепишем в виде одного уравнения, если введем обозначение $\Lambda = \text{diag} \{\lambda_1, \lambda_2, \dots, \lambda_n\}$:

$$AU = U\Lambda,$$

что равносильно соотношениям

$$U^*AU = \Lambda, \quad A = U\Lambda U^*.$$

Тем самым приходим к следующей теореме о *спектральном разложении эрмитовой матрицы*.

ТЕОРЕМА 10.20. Если $A = A^*$, то существует диагонализующая матрица T , которая является также и унитарной, $T^*T = I$, такая что

$$T^*AT = \Lambda, \quad A = T\Lambda T^*, \quad (10.17)$$

где $\Lambda = \text{diag} \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ — диагональная матрица, составленная из собственных значений матрицы A . Если эти значения простые (среди λ_i нет

кратных), то $T = U$, где $U = [u_1 | u_2 | \dots | u_n]$ — матрица, составленная из соответствующих собственных векторов матрицы A . В этом случае спектральное разложение матрицы A имеет вид

$$A = U\Lambda U^* = \lambda_1 u_1 u_1^* + \lambda_2 u_2 u_2^* + \dots + \lambda_n u_n u_n^*.$$

При наличии кратных собственных значений любая эрмитова матрица A по-прежнему имеет полный набор ортонормированных собственных векторов и, следовательно, может быть диагонализирована с помощью некоторой унитарной матрицы T , т. е. (10.17) справедливо в общем случае. Каждая эрмитова матрица с k различными собственными значениями имеет свое «спектральное разложение»

$$A = T\Lambda T^* = \lambda_1 P_1 + \lambda_2 P_2 + \dots + \lambda_k P_k,$$

где P_i есть проекция матрицы A на собственное подпространство, соответствующее значению λ_i .

ДОКАЗАТЕЛЬСТВО. Полное доказательство можно найти в [13] или [23].

Это очень важный результат, имеющий и «обратную» формулировку: только эрмитовы матрицы обладают одновременно и вещественными собственными значениями, и ортонормированными собственными векторами. Если $T^{-1}AT$ равняется некоторой вещественной диагональной матрице Λ и матрица T унитарна, $T^{-1} = T^*$, то матрица A обязательно является эрмитовой: $A^* = (T\Lambda T^*)^* = T\Lambda T^* = A$. Теорема 10.20 имеет множество применений при обработке экспериментальных данных, — не только в регрессионном анализе, но также в области так называемого многомерного анализа. Интересны примеры, когда по экспериментальной корреляционной матрице R нужно провести анализ основных компонент и факторный анализ [13]. Для специалистов-прикладников эти примеры очень поучительны. Они показывают, как два специалиста, вводя различное число p факторов и диагональную компоненту D в разложении $R = FF^T + D$, а также любую ортогональную $(p \times p)$ -матрицу Q в представлении $\tilde{F} = FQ$, могут получать совершенно различные интерпретации F и \tilde{F} (матрицы факторных коэффициентов) одних и тех же экспериментальных данных. Мы эти частные вопросы не рассматриваем и за деталями отсылаем к [13] и специальной литературе по факторному анализу. Когда $A = A^T$, (10.17) принимает следующий вид:

$$T^T AT = \Lambda, \quad A = T\Lambda T^T.$$

ОЦЕНИВАНИЕ ПО МЕТОДУ НАИМЕНЬШИХ КВАДРАТОВ

11.1 Модели, регрессии и оценки

Изучение объектов, явлений или процессов реального мира заключается в построении их моделей, т. е. тех или иных форм описания для выявления существенных закономерностей. Когда тип, т. е. форма модели выбрана, требуется определить ее наилучшее параметрическое наполнение. При этом критерием естественно считать соответствие между откликами объекта и модели, когда они погружены в одни и те же или одинаковые внешние условия. Иными словами, та модель будет наилучшей, чей отклик, по данным проведенного наблюдения, менее всего уклоняется от отклика реального объекта в разнообразных, но одинаковых для модели и объекта условиях.

Таким образом, оптимальная модель строится на основе опытного анализа прошлых поведений объекта, но способна предсказывать и ближайшее будущее поведение. Обращение к прошлому опыту для объяснения причин или закономерностей называют *регрессией*.

Регрессионный анализ в статистике, регрессионное моделирование, идентификация моделей и оценивание состояния объекта в теории систем, адаптивная обработка сигналов — это примеры очень близких или тесно связанных задач, имеющих общую математическую основу, один и тот же математический метод решения [21, 26, 27, 32, 33, 34, 50, 53, 62, 64, 66, 69, 82, 88]. Если модель — линейная по параметрам, а критерий ее качества есть квадрат расхождения (невязки) между откликами модели и объекта, этот метод и есть знаменитый линейный *Метод Наименьших Квадратов* (МНК) [47, 53], возникновение которого связывается с работами Гаусса и Лежандра в начале 19 столетия, см. подробнее стр. 199. При отказе от линейности модели по ее

параметрам получаем нелинейную задачу о наименьших квадратах [22, 35].

В разделах 11 и 12 этой книги мы ограничиваемся обыкновенной линейной статической детерминистской задачей НК [15]. Она хорошо разработана, но имеет обобщения по разным направлениям (для них она является базовой). Основные обобщения:

1. Динамическая задача: оцениваемый вектор не постоянен, а эволюционирует во времени, подчиняясь какому-либо уравнению состояния [61, 115]. Это обобщение присутствует в разд. 13, 14.
2. Стохастическая задача: оцениваемый вектор постоянен, но случаен, и он измеряется на фоне случайных помех [61, 115]. Это обобщение начинается в подразд. 11.9. Если задача, к тому же, динамическая, то допускаются случайные возмущения в уравнении состояния или/и в уравнении наблюдений [61, 115]. Это обобщение присутствует в разд. 13, 14.
3. Полная МНК-задача: имеются возмущения в матрице наблюдений, т. е. в матрице регрессоров. Это обобщение здесь отсутствует. См. Total Least Squares в [18, 38, 100].
4. Неоднородность погрешностей в исходных данных [49]. Это обобщение присутствует в разд. 13, 14.
5. Зависимость ковариации погрешностей от неизвестного параметра (Generalized Least Squares) [106]. Это обобщение здесь отсутствует. Возможен переход к адаптивному оцениванию [73].
6. Зависимость от неизвестного параметра не только ковариации погрешностей, но и других элементов стохастической модели источника данных. Это обобщение здесь отсутствует. Возможен переход к адаптивному оцениванию [73].
7. Улучшение вычислительных схем [15, 89, 93, 95, 103, 104, 107, 108, 119, 123, 125, 135, 129]. Это обобщение присутствует в разд. 13, 14 и составляет их основное содержание.

Вычислительные схемы МНК имеют важнейшее значение для любых обобщений и любых приложений. Неиспользование эффективных, т. е. численно устойчивых и экономичных алгоритмов может увести исследователя в сторону надуманных проблем. И наоборот, — затруднения в численной реализации могут быть с успехом преодолены, если знать и умело использовать

эффективные схемы вычислений. Кроме того, как и любое знание, наука вычислений несет с собой важные, плодотворные идеи, которые полезны сами по себе. Применительно к задаче МНК, эти базовые идеи следующие: рекурсия алгоритма, скаляризация обработки векторных данных, факторизация матриц (ковариационных и информационных), ортогонализация матриц, и взаимная инверсность (двойственность форм) вычислительных алгоритмов.

11.2 Линейная задача наименьших квадратов

Во многих приложениях, связанных с обработкой экспериментальных данных, необходимо отыскивать такой вектор $x \in \mathbb{R}^n$, линейные комбинации компонент которого, Ax , где $A = A(m, n)$ — матрица размера $(m \times n)$, как можно более близки или, еще лучше, равны данным значениям, образующим вектор $z \in \mathbb{R}^m$, т.е. $Ax \approx z$. Если мерой близости двух векторов считать квадрат евклидовой нормы разностного вектора, в данном случае, вектора $v \triangleq z - Ax$, то указанная задача есть линейная задача о наименьших квадратах (см. также подразд. 10.4).

Возможность сделать равным нулю вектор невязок $v = z - Ax$ существует тогда и только тогда, когда $z \in \mathcal{R}(A)$, где $\mathcal{R}(A)$ — пространство столбцов матрицы A . В этом случае имеем совместную систему уравнений $Ax = z$. Однако, z — вектор наблюдений, то есть экспериментальных данных и A — матрица, которую задают до того, как получают z и которую в различных приложениях называют либо матрицей регрессоров, либо матрицей наблюдений, либо матрицей плана эксперимента. Совсем не обязательно, что условие $z \in \mathcal{R}(A)$ будет выполнено, например, из-за случайных погрешностей v во время регистрации экспериментальных данных. Тогда

$$z = Ax + v, \quad (11.1)$$

и решение по методу наименьших квадратов (для краткости, *МНК-решение*) есть вектор \bar{x} , доставляющий минимум функционалу качества (см. рис. 11.1):

$$J(x) = (z - Ax)^T(z - Ax) = \sum_{j=1}^m v(j)^2 = v^T v. \quad (11.2)$$

Требую минимум этого критерия, для искомого \bar{x} получаем так называемые *нормальные уравнения* (см. подразд. 10.4, стр. 209):

$$A^T A \bar{x} = A^T z. \quad (11.3)$$

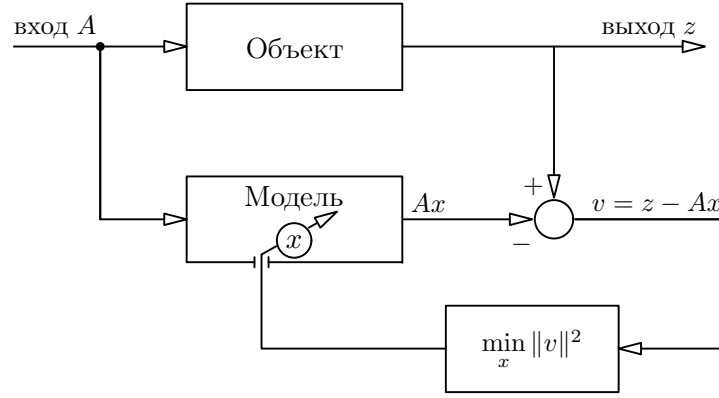


Рис. 11.1. Линейная задача наименьших квадратов

Их решение всегда существует (обе части равенства (11.3) принадлежат одному и тому же пространству $\mathcal{R}(A^T)$ столбцов матрицы A^T), но может быть не единственным (если $\text{rank } A < n$). В последнем случае из всех \bar{x} выбирают то единственное, \bar{x}_0 , которое имеет минимальную норму $\|\bar{x}_0\|$. Этот вектор называют *нормальным псевдорешением*. Известно (см. подразд. 10.4), что

$$\bar{x}_0 = A^+ z, \quad (11.4)$$

где A^+ — *псевдообратная матрица* к A . Как уже отмечалось, в качестве определения A^+ применяют различные формулировки. Здесь для этого используем геометрический подход: A^+ есть такая матрица в выражении (11.4), что для любого $z \in \mathbb{R}^m$ вектор $\bar{x}_0 \in \mathbb{R}^n$ удовлетворяет двум условиям:

$$A\bar{x}_0 = p, \quad p \in \mathcal{R}(A), \quad z - p \perp \mathcal{R}(A). \quad (11.5)$$

$$\bar{x}_0 \in \mathcal{R}(A^T). \quad (11.6)$$

Условие (11.5) требует, чтобы \bar{x}_0 отвечал совместной системе $A\bar{x}_0 = p$, где p — проекция вектора z на $\mathcal{R}(A)$, а условие (11.6) требует, чтобы этот \bar{x}_0 был взят из пространства $\mathcal{R}(A^T)$ строк матрицы A . Условие (11.5), таким образом, выбирает $\bar{x}_0 = \bar{x}$, чтобы минимизировать функционал (11.2), а условие (11.6) среди всех таких \bar{x} выбирает единственный \bar{x}_0 с минимальной нормой.

Часто матрицу A выбирают так, чтобы она имела полный столбцовый ранг, $\text{rank } A = n$. В этом случае $m \geq n$, \bar{x} единственно и равно \bar{x}_0 , $A^+ = (A^T A)^{-1} A^T$ и

$$\bar{x}_0 = (A^T A)^{-1} A^T z. \quad (11.8)$$

Однако иногда такое условие не выполняется, и тогда $\bar{x}_0 = A^+ z$, где A^+ — псевдообратная матрица (см. подразд. 10.4–10.5, стр. 209–212).

11.3 Статистическая интерпретация

Предположим, что вектор ошибок v в уравнении (11.1) образован из случайных величин с нулевым средним и известной матрицей ковариации

$$\mathbf{E}\{v\} = 0, \quad \mathbf{E}\{vv^T\} = P_v, \quad (11.9)$$

где $\mathbf{E}\{\cdot\}$ — оператор математического ожидания (среднего) над \cdot , и P_v — ПО (положительно определенная) матрица. Найдем квадратно-корневое разложение $P_v = SS^T$ (например, разложение Холецкого). Если теперь умножить вектор z (11.1) на S^{-1} , то данные $\bar{z} = S^{-1}z$ получают представление

$$\bar{z} = \bar{A}x + \bar{v} \quad (11.10)$$

с матрицей $\bar{A} = S^{-1}A$ и ошибками $\bar{v} = S^{-1}v$. Этот вектор ошибок всегда имеет единичную ковариацию:

$$\mathbf{E}\{\bar{v}\bar{v}^T\} = \mathbf{E}\{S^{-1}vv^TS^{-T}\} = S^{-1}\mathbf{E}\{vv^T\}S^{-T} = S^{-1}SS^TS^{-T} = I_m,$$

где I_m — единичная матрица размера $(m \times m)$. Вследствие этого данные \bar{z} называют *нормализованными экспериментальными данными*. Значение представления (11.10) заключается в том, что оно демонстрирует, как сконструировать вектор некоррелированных между собой измерений с единичной дисперсией из вектора, элементы которого произвольно взаимно коррелированы (декоррелировать и нормализовать его). Ниже предполагаем, что данные z (11.1) уже декоррелированы и нормализованы, так что

$$\mathbf{E}\{v\} = 0, \quad \mathbf{E}\{vv^T\} = I_m, \quad (11.11)$$

где I_m — единичная матрица размера $(m \times m)$. При этом из (11.3) находим

$$A^T A \bar{x} = A^T z = A^T Ax + A^T v,$$

$$A^T A(\bar{x} - x) = A^T v.$$

Отсюда, если $\det(A^T A) \neq 0$, имеем

$$\mathbf{E}\{\bar{x}\} = x, \quad (11.12)$$

$$(A^T A) \mathbf{E}\{(\bar{x} - x)(\bar{x} - x)^T\}(A^T A) = A^T \mathbf{E}\{vv^T\} A = A^T A. \quad (11.13)$$

Соотношение (11.12) выражает собой *свойство несмещенности решения* (оценки) \bar{x} относительно неизвестного (постоянного) вектора x , измеряемого в виде экспериментальных данных z , (11.1) или (11.10). Соотношение (11.13) дает выражение для ковариации оценки \bar{x} в виде

$$P_{\bar{x}} = \mathbf{E} \{ (\bar{x} - x)(\bar{x} - x)^T \} = (A^T A)^{-1}. \quad (11.14)$$

при определении \bar{x} по нормализованным экспериментальным данным.

Обратная матрица $P_{\bar{x}}^{-1}$ от ковариации $P_{\bar{x}}$ называется *информационной матрицей*. Ее обозначение будет $\Lambda_{\bar{x}}$ или просто Λ . При использовании нормализованных данных она равна $A^T A$, а в более общем случае (11.9) она равна $\Lambda = A^T P_v^{-1} A$.

11.4 Включение априорных статистических данных

Предположим, что в добавление к линейной системе (11.1) мы имеем априорную несмещенную оценку неизвестного вектора x в виде \tilde{x} и соответствующую априорную информационную матрицу $\tilde{\Lambda}$. Это означает, что $\mathbf{E} \{ \tilde{x} \} = x$ и

$$\tilde{\Lambda}^{-1} = \mathbf{E} \{ (\tilde{x} - x)(\tilde{x} - x)^T \} = \tilde{P}, \quad (11.15)$$

где \tilde{P} — *ковариация* оценки \tilde{x} . Найдем какой-нибудь квадратный корень $\tilde{\Lambda}^{1/2}$ из матрицы $\tilde{\Lambda}$, например, по одному из разложений Холецкого (см. разд. 6):

$$\tilde{\Lambda} = \tilde{\Lambda}^{1/2} \tilde{\Lambda}^{T/2} = \tilde{R}^T \tilde{R},$$

где $\tilde{\Lambda}^{1/2} = \tilde{R}^T$. Образует вектор $\tilde{v} = (\tilde{\Lambda}^{1/2})^T (\tilde{x} - x) = \tilde{R}(\tilde{x} - x)$. Этот вектор имеет смысл *нормализованной ошибки* для априорной оценки \tilde{x} вектора x . Действительно, его ковариация равна единичной матрице размера $(n \times n)$:

$$\mathbf{E} \{ \tilde{v} \tilde{v}^T \} = \tilde{R} \mathbf{E} \{ (\tilde{x} - x)(\tilde{x} - x)^T \} \tilde{R}^T = \tilde{\Lambda}^{T/2} \tilde{\Lambda}^{-1} \tilde{\Lambda}^{1/2} = I_n.$$

Так как о векторе x , кроме экспериментальных данных z , (11.1), известна априорная оценка \tilde{x} с ковариацией $\tilde{P} = \tilde{\Lambda}^{-1}$, эту информацию целесообразно включить в контекст задачи о наименьших квадратах, рассматривая модифицированный функционал качества $J_1(x) = \tilde{v}^T \tilde{v} + v^T v$ вместо (11.2). Он соединяет в себе квадрат нормы нормализованной ошибки (невязки) априорной оценки

$$\tilde{v} = \tilde{R}(\tilde{x} - x) = \tilde{\Lambda}^{T/2}(\tilde{x} - x),$$

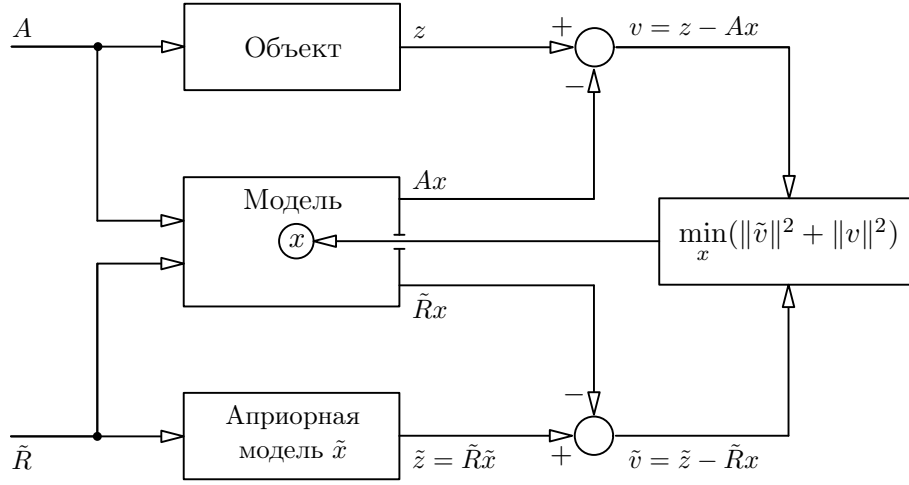


Рис. 11.2. Включение априорных данных в линейную задачу НК

с квадратом нормы нормализованной ошибки (невязки) экспериментальных данных $v = z - Ax$. Так как

$$\begin{aligned} J_1(x) &= (\tilde{x} - x)^T \tilde{\Lambda} (\tilde{x} - x) + (z - Ax)^T (z - Ax) = \\ &= (\tilde{z} - \tilde{R}x)^T (\tilde{z} - \tilde{R}x) + (z - Ax)^T (z - Ax), \end{aligned} \quad (11.16)$$

где $\tilde{z} = \tilde{R}\tilde{x}$, то $J_1(x)$ может быть интерпретирован просто как критерий качества метода наименьших квадратов применительно к *расширенной системе* (рис. 11.2)

$$\begin{bmatrix} \tilde{z} \\ z \end{bmatrix} = \begin{bmatrix} \tilde{R} \\ A \end{bmatrix} x + \begin{bmatrix} \tilde{v} \\ v \end{bmatrix}, \quad (11.17)$$

включающей, помимо текущих экспериментальных данных z , «дополнительные» экспериментальные данные \tilde{z} , соответствующие имеющейся в наличии априорной информации \tilde{x} , $\tilde{\Lambda}$.

Обозначим через \hat{x} МНК-решение расширенной системы (11.17), доставляющее минимум функционалу (11.16). Из этого критерия для \hat{x} получаем, аналогично (11.3), нормальные уравнения

$$(\tilde{\Lambda} + A^T A) \hat{x} = \tilde{\Lambda} \tilde{x} + A^T z. \quad (11.18)$$

Простая модификация данной в п. 11.3 статистической интерпретации приводит к выражению

$$\hat{\Lambda}(\hat{x} - x) = \tilde{\Lambda}(\tilde{x} - x) + A^T v. \quad (11.19)$$

Так как $\mathbf{E}\{\tilde{x} - x\} = 0$ и $\mathbf{E}\{v\} = 0$, то и $\mathbf{E}\{\hat{x} - x\} = 0$, то есть \hat{x} есть также несмещенная оценка: $\mathbf{E}\{\hat{x}\} = x$. Если ошибка априорной оценки и ошибка измерения взаимно некоррелированы, $\mathbf{E}\{(\tilde{x} - x)v^T\} = 0$, то после «возведения в квадрат» обеих частей (11.19) и осреднения получим ковариацию

$$P_{\hat{x}} = \mathbf{E}\{(\hat{x} - x)(\hat{x} - x)^T\} = (\tilde{\Lambda} + A^T A)^{-1} = \hat{\Lambda}^{-1}, \quad (11.20)$$

где через $\hat{\Lambda}$ обозначена информационная матрица *апостериорной* оценки \hat{x} , $\hat{\Lambda} = \tilde{\Lambda} + A^T A$.

ЗАМЕЧАНИЕ 11.1. Матрица $\tilde{\Lambda}$ не обязана быть невырожденной, хотя в (11.15) это формально предполагалось. В действительности, априорное знание некоторых (или всех) компонент вектора x может быть исчезающе мало, так что соответствующие строки и столбцы в информационной матрице $\tilde{\Lambda}$ заполняются исчезающе малыми числами или даже нулями. При этом соотношение (11.15) сохраняет силу в пределе, в том смысле, что в ковариационной матрице \tilde{P} соответствующие диагональные элементы стремятся к $+\infty$, в то время как другие остаются ограниченными. Заметьте, что (11.18) как условие минимума функционала (11.16) получается при произвольной неотрицательно определенной матрице $\tilde{\Lambda}$. Если $\tilde{\Lambda} = 0$, то (11.18) сводится к (11.3) и (11.20) сводится к (11.14). Таким образом, информационная и ковариационная матрицы взаимно обратны не только формально в силу определения (11.15), но и по существу как меры достоверности/недостоверности априорной оценки \tilde{x} вектора x .

11.5 Включение предшествующего МНК-решения

Расширенная система (11.17) показала, что априорные статистические сведения о векторе x , поступающие в виде несмещенной оценки \tilde{x} и ее ковариации \tilde{P} , могут быть интерпретированы как воображаемые добавочные результаты $\tilde{z} = \tilde{R}\tilde{x}$ некоего эксперимента. Это наводит на мысль, что и чисто алгебраическую задачу отыскания МНК-решения системы уравнений можно решать последовательно: предварительно найти \tilde{x} как МНК-решение части системы, а затем включить это \tilde{x} в полную систему, чтобы найти ее МНК-решение \hat{x} . Такое разделение системы на две части — априорную и текущую — можно назвать ее «*расщеплением*» (см. формальное решение рассматриваемого здесь вопроса о расщеплении в подразд. 10.8).

Пусть система уравнений произвольно расщеплена на эти две подсистемы:

$$\begin{bmatrix} f \\ z \end{bmatrix} = \begin{bmatrix} \tilde{R} \\ A \end{bmatrix} x + \begin{bmatrix} w \\ v \end{bmatrix}. \quad (11.21)$$

МНК-решение \hat{x} этой полной системы, доставляющее минимум функционалу $J_1(x) = w^T w + v^T v$, есть решение нормальных уравнений

$$\begin{bmatrix} \tilde{R}^T & | & A^T \end{bmatrix} \begin{bmatrix} \tilde{R} \\ A \end{bmatrix} \hat{x} = \begin{bmatrix} \tilde{R}^T & | & A^T \end{bmatrix} \begin{bmatrix} f \\ z \end{bmatrix}. \quad (11.22)$$

Допустим, найдено МНК-решение \tilde{x} для подсистемы $f = \tilde{R}x + w$ из критерия минимума функционала $J(x) = w^T w$. Как отмечено в (11.5)–(11.6), оно удовлетворяет двум условиям:

$$\begin{aligned} \tilde{R}\tilde{x} &= \tilde{z}, \quad \tilde{z} \in \mathcal{R}(\tilde{R}), \quad f - \tilde{z} \perp \mathcal{R}(\tilde{R}), \\ \tilde{x} &\in \mathcal{R}(\tilde{R}^T). \end{aligned}$$

Разностный вектор $r = f - \tilde{z}$ ортогонален пространству столбцов $\mathcal{R}(\tilde{R})$ матрицы \tilde{R} и, следовательно, лежит в левом нуль-пространстве $\mathcal{N}(\tilde{R}^T)$, определяемом как все векторы y , удовлетворяющие уравнению $\tilde{R}^T y = 0$. Поэтому

$$\tilde{R}^T f = \tilde{R}^T(\tilde{z} + r) = \tilde{R}^T \tilde{z} + \tilde{R}^T r = \tilde{R}^T \tilde{z}.$$

Следовательно, уравнения (11.22) совпадают с уравнениями

$$\begin{bmatrix} \tilde{R}^T & | & A^T \end{bmatrix} \begin{bmatrix} \tilde{R} \\ A \end{bmatrix} \hat{x} = \begin{bmatrix} \tilde{R}^T & | & A^T \end{bmatrix} \begin{bmatrix} \tilde{z} \\ z \end{bmatrix},$$

которые, в свою очередь совпадают с уравнениями (11.18), так как $\tilde{R}^T \tilde{R} = \tilde{\Lambda}$. Тем самым доказано, что МНК-решение \hat{x} данной системы (11.21) совпадает с МНК-решением системы (11.17), отличающейся от (11.21) тем, что в нее вместо f включен вектор \tilde{z} , равный проекции f на $\mathcal{R}(\tilde{R})$, $\tilde{z} = \tilde{R}\tilde{x}$, где \tilde{x} — МНК-решение левой подсистемы в (11.21).

11.6 Рекурсия МНК в стандартной информационной форме

Интерпретация априорных статистических данных как дополнительных наблюдений, или, что равносильно, учет имеющегося МНК-решения подсистемы после добавления в систему новой порции уравнений, является краеугольным камнем рекурсии для МНК. Это дает возможность обрабатывать

экспериментальные данные по мере их поступления, то есть решать задачу о наименьших квадратах по мере поступления новых уравнений. Это равносильно также тому, что исходную большую совокупность экспериментальных данных можно «расщеплять» произвольно на порции и последовательно включать их в обработку. Результат такой последовательной обработки, как доказано выше (подразд. 10.8 и 11.5), будет равносильен результату обработки всей совокупности экспериментальных данных целиком.

Результаты при статистической интерпретации рекурсивны потому, что текущие величины — оценка \hat{x} и ковариация $P_{\hat{x}}$ — становятся априорными и комбинируются с новыми данными, чтобы образовать обновленные оценку и ковариацию. При этом существенно, что результаты (11.18) и (11.20) не зависят (теоретически) от того, какой квадратный корень \tilde{R} в разложении $\tilde{\Lambda} = \tilde{R}^T \tilde{R}$ использован. Эта свобода позволяет выбирать \tilde{R} из соображений большей вычислительной точности. Кроме того, если лишь окончательная оценка (МНК-решение) необходима, то лучше не находить промежуточных оценок, а просто накапливать информационную матрицу $\sum A_j^T A_j$ и сумму $\sum A_j^T z_j$, и лишь в нужный момент (например, в самом конце) вычислить решение.

Информационную форму последовательного МНК запишем, вводя матрицу $(\Lambda \mid d)$.

I. *Инициализация.* Устанавливают начальные значения x_0, Λ_0 :

$$d_0 = \Lambda_0 x_0, \quad (\Lambda \mid d) := (\Lambda_0 \mid d_0).$$

Эти начальные значения берут из априорных данных: $x_0 = \tilde{x}, \Lambda_0 = \tilde{\Lambda}$.

II. *Обработка наблюдений.* Вводят очередную «порцию» наблюдений $z = Ax + v$:

$$(\Lambda \mid d) := (\Lambda \mid d) + A^T (A \mid z). \quad (11.23)$$

В общем случае ненормализованных статистических данных z вместо (11.23) используют алгоритм:

$$(\Lambda \mid d) := (\Lambda \mid d) + A^T R^{-1} (A \mid z). \quad (11.24)$$

III. *Выдача результата.* После последовательной обработки всех порций наблюдений или в нужный момент, когда Λ^{-1} существует, вычисляют

$$\hat{x} = \Lambda^{-1} d, \quad P_{\hat{x}} = \Lambda^{-1}.$$

11.7 Рекурсия МНК в стандартной ковариационной форме

Пусть априорная и апостериорная оценки, \tilde{x} и \hat{x} , характеризуются невырожденными информационными матрицами $\tilde{\Lambda}$ и $\hat{\Lambda}$, соответственно. Тогда существуют обратные к ним ковариационные матрицы, (11.15) и (11.20). Разрешим (11.18) относительно \hat{x} :

$$\hat{x} = (\tilde{\Lambda} + A^T A)^{-1} \tilde{\Lambda} \tilde{x} + (\tilde{\Lambda} + A^T A)^{-1} A^T z.$$

Обозначим

$$L = (\tilde{\Lambda} + A^T A)^{-1} \tilde{\Lambda}, \quad K = (\tilde{\Lambda} + A^T A)^{-1} A^T \quad (11.25)$$

и преобразуем:

$$\hat{x} = L \tilde{x} + K z - K A \tilde{x} + K A \tilde{x} = \tilde{x} + K(z - A \tilde{x}),$$

так как $L + K A = I_n$. Для определения K воспользуемся в (11.25) следующей важной леммой.

ЛЕММА 11.1.

$$(\Lambda_1 - \Lambda_{12} \Lambda_2^{-1} \Lambda_{21})^{-1} = \Lambda_1^{-1} + \Lambda_1^{-1} \Lambda_{12} (\Lambda_2 - \Lambda_{21} \Lambda_1^{-1} \Lambda_{12})^{-1} \Lambda_{21} \Lambda_1^{-1}, \quad (11.26)$$

где предполагается, что все матрицы согласованы по размерам и все обратные матрицы существуют.

УПРАЖНЕНИЕ 11.1. Докажите лемму 11.1, рассматривая блочные матрицы

$$\begin{bmatrix} \Lambda_1 & \Lambda_{12} \\ \Lambda_{21} & \Lambda_2 \end{bmatrix}^{-1} = \begin{bmatrix} P_1 & P_{12} \\ P_{21} & P_2 \end{bmatrix},$$

и выписывая поблочно равенства $\Lambda P = I$ и $P \Lambda = I$.

Применим лемму 11.1 при $\Lambda_1 = \tilde{\Lambda}$, $\Lambda_{12} = A^T$, $\Lambda_{21} = \Lambda_{12}^T = A$, $\Lambda_2^{-1} = -I$. Получим

$$(\tilde{\Lambda} + A^T A)^{-1} = \tilde{\Lambda}^{-1} - \tilde{\Lambda}^{-1} A^T (A \tilde{\Lambda}^{-1} A^T + I)^{-1} A \tilde{\Lambda}^{-1}.$$

Обозначим: $\tilde{P} = \tilde{\Lambda}^{-1}$, $\hat{P} = \hat{\Lambda}^{-1}$. Имеем из подразд. 11.5 выражение (11.23): $\hat{\Lambda} = \tilde{\Lambda} + A^T A$. Следовательно, $\hat{P} = \tilde{P} - \tilde{P} A^T (A \tilde{P} A^T + I)^{-1} A \tilde{P}$. Так как $K = \hat{P} A^T$, то

$$\begin{aligned} K &= \tilde{P} A^T - \tilde{P} A^T (A \tilde{P} A^T + I)^{-1} A \tilde{P} A^T = \\ &= \tilde{P} A^T (A \tilde{P} A^T + I)^{-1} [A \tilde{P} A^T + I - A \tilde{P} A^T] = \tilde{P} A^T (A \tilde{P} A^T + I)^{-1}. \end{aligned}$$

Таким образом, при статистической интерпретации (см. подразд. 11.4) получаем возможность уточнять априорную несмещенную оценку \tilde{x} и уменьшать ее ковариацию \tilde{P} , благодаря включению в процесс обработки вновь поступивших результатов наблюдений $z = Ax + v$ и применяя к ним следующий алгоритм, известный как *стандартный алгоритм Калмана* (рис. 11.3).

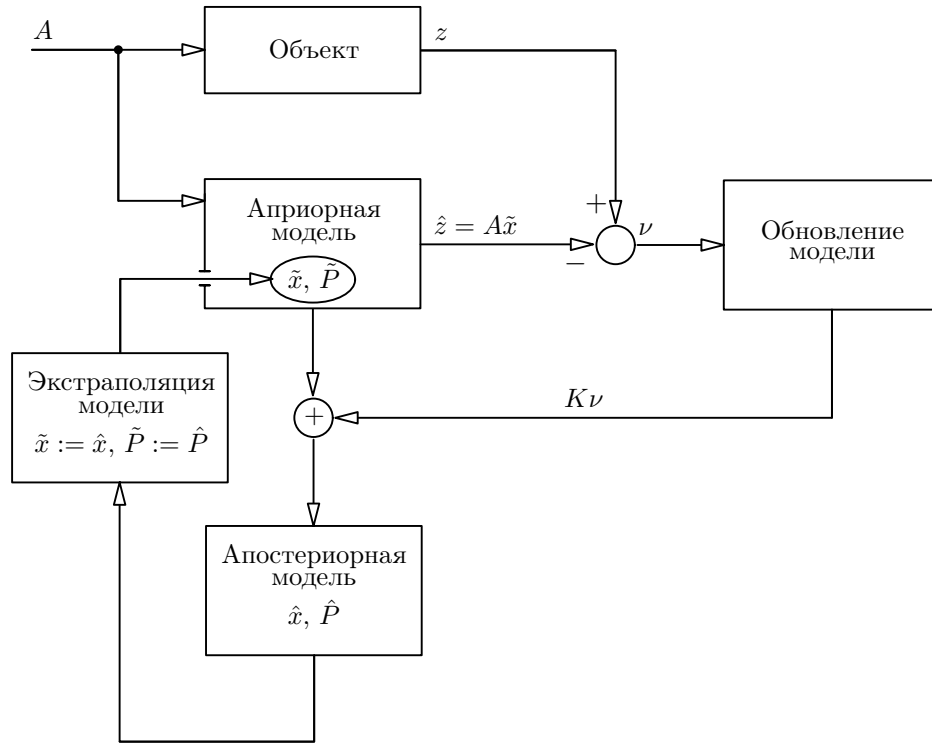


Рис. 11.3. Рекурсия МНК в стандартной ковариационной форме (схема Калмана). Этап 1 — обновление модели по наблюдениям. Этап 2 — экстраполяция модели между наблюдениями. Априорная модель дает предсказание \hat{z} для отклика z объекта. Разность ν имеет смысл обновляющего процесса. Весовая матрица K , умноженная на ν , обеспечивает обновление априорной модели по наблюдению z , т.е. переход к апостериорной модели

I. *Инициализация.* Начальные значения x_0, P_0 :

$$\tilde{x} := x_0, \quad \tilde{P} := P_0. \quad (11.27)$$

II. *Обработка наблюдений.* Очередная порция наблюдений $z = Ax + v$:

$$\begin{aligned} K &= \tilde{P}A^T(A\tilde{P}A^T + I)^{-1}, \\ \hat{P} &= \tilde{P} - KA\tilde{P}, \\ \hat{x} &= \tilde{x} + K(z - A\tilde{x}). \end{aligned} \quad (11.28)$$

III. *Экстраполяция.* Распространение оценки \hat{x} и ее ковариации \hat{P} между наблюдениями, т. е. к моменту повторения этапа II со следующей порцией наблюдений:

$$\tilde{P} := \hat{P}, \quad \tilde{x} := \hat{x}. \quad (11.29)$$

ЗАМЕЧАНИЕ 11.2. Первая и вторая формулы в (11.29) выражают правило экстраполяции только для статической МНК-задачи оценивания, т. е. для случая, когда оцениваемый вектор x не изменяется в процессе наблюдения: $x = \text{const}$. Они заменяются другими в случае динамической МНК-задачи оценивания — первыми двумя формулами в системе уравнений (13.3) (см. ниже стр. 273).

Равным образом данный алгоритм пригоден и без статистической интерпретации (см. подразд. 11.5), когда алгебраическая задача отыскания МНК-решения переопределенной системы решается последовательно. Такое решение может стартовать с условно «пустой» системы уравнений. Практически, это должно отвечать условию $\tilde{\Lambda} = 0$, которое легко реализовать в информационной форме (подразд. 11.5). В ковариационной форме данное условие можно реализовать лишь приближенно, например, полагая $P_0 = \varepsilon^{-2}I$, где $\varepsilon \rightarrow 0$. При таком выборе величина x_0 практически не имеет влияния на дальнейший процесс, так что она может быть взята равной нулевому вектору, $x_0 = 0$. После такой инициализации уравнения исходной переопределенной системы могут вводиться в этап обработки измерений последовательными порциями, и в порциях может содержаться любое, не обязательно одно и то же, число уравнений, выражаемое в числе строк матрицы A .

Как следует из подразд. 11.5, от указанного числа МНК-решение всей алгебраической системы уравнений не зависит. В связи с этим, с вычислительной точки зрения удобным оказывается добавление в очередной порции лишь по одному уравнению. Тогда матрица A содержит всего одну строку, которую теперь обозначим как транспонированный вектор-столбец a , $a^T = (a_1, a_2, \dots, a_n)$. В этом случае

$$z = a^T x + v, \quad (11.30)$$

и обработка наблюдений (11.28) принимает особенно простой вид:

$$\alpha = a^T \tilde{P} a + 1, \quad K = \tilde{P} a / \alpha, \quad \hat{P} = \tilde{P} - K a^T \tilde{P}, \quad \hat{x} = \tilde{x} + K(z - a^T \tilde{x}). \quad (11.31)$$

Это алгоритм скалярной (последовательной) обработки. В нем умножение на обратную матрицу $(A\tilde{P}A^T + I)^{-1}$ заменяется делением на скалярную величину α .

УПРАЖНЕНИЕ 11.2. С применением леммы 11.1 к выражению (11.24) докажите, что в общем случае ненормализованных экспериментальных данных (11.1) при их статистической интерпретации с ковариацией ошибок v , равной R , матрица Калмана K в алгоритме (11.28) определяется выражением

$$K = \tilde{P}A^T(A\tilde{P}A^T + R)^{-1}. \quad (11.32)$$

УПРАЖНЕНИЕ 11.3. Статистическая интерпретация алгоритма (11.31) дана в подразд. 11.3 для случая, когда экспериментальные данные (11.1) нормализованы, то есть характеризуются математическими ожиданиями (11.11). Это выражается добавлением «+1» в выражении для α , (11.31), причем эта «+1» есть не что иное как $\mathbf{E}\{vv^T\} = 1$ для ошибки v в (11.30). Докажите, что в более общем случае, когда матрица R в (11.32) является диагональной,

$$R = \text{diag}(r_1, r_2, \dots, r_m), \quad (11.33)$$

и только в этом случае матричный алгоритм (11.28) с матрицей K из (11.32) эквивалентен m -кратному повторению скалярного алгоритма вида (11.31) с $\alpha = a^T \tilde{P}a + r$, где a^T — i -я строка матрицы A , $r = r_i$ и z — i -й элемент вектора z , (11.1), при $i = 1, 2, \dots, m$.

Таким образом, *скаляризованный алгоритм Калмана* (11.31) имеет следующее общее представление:

$$\alpha = a^T \tilde{P}a + r, \quad K = \tilde{P}a/\alpha, \quad \hat{P} = \tilde{P} - Ka^T \tilde{P}, \quad \hat{x} = \tilde{x} + K(z - a^T \tilde{x}), \quad (11.34)$$

если наблюдения (11.1) в их статистической интерпретации составлены из m отдельных, независимых друг от друга, скалярных данных вида (11.30), каждое с ковариацией $r = r_i$, $i = 1, 2, \dots, m$.

Еще раз отметим, что в применении к решению переопределенной системы алгебраических уравнений в (11.34) следует считать $r = 1$, то есть использовать (11.31).

ЗАМЕЧАНИЕ 11.3. Условие (11.33) не является ни в коей мере ограничительным для использования (11.34). Используя разложение Холеского без квадратных корней (см. разд. 6), любую $R > 0$ можно представить в виде $R = UDU^T$ или $R = LDL^T$ и затем перейти к измерениям $\bar{z} = U^{-1}z$ или $\bar{z} = L^{-1}z$, чтобы диагонализировать матрицу ковариаций ошибок наблюдений.

УПРАЖНЕНИЕ 11.4. Докажите, что в алгоритме (11.28) с определением K по выражению (11.32) вычисление \hat{P} может быть представлено в так называемой симметричной форме Джозефа:

$$\hat{P} = (I - KA)\tilde{P}(I - KA)^T + KRK^T,$$

которую иначе называют *стабилизированным алгоритмом Калмана*, так как она предотвращает возможную потерю положительной определенности матрицы \hat{P} , присущую стандартному алгоритму (11.28) с $\hat{P} = \tilde{P} - KAP$. При скалярной обработке наблюдений в алгоритме (11.34) это выражение для \hat{P} , соответственно, заменяется на *скаляризованный алгоритм Джозефа*

$$\hat{P} = (I - K\alpha^T)\tilde{P}(I - \alpha K^T) + rKK^T.$$

11.8 Ковариационный алгоритм Поттера для МНК

Вместо матриц \tilde{P} и \hat{P} , по своей природе положительно определенных, далее оперируем с квадратными корнями из них, соответственно, \tilde{S} и \hat{S} , отвечающими равенствам $\tilde{S}\tilde{S}^T$ и $\hat{S}\hat{S}^T$. Перепишем выражение для \hat{P} в (11.34) в виде

$$\hat{S}\hat{S}^T = \tilde{S}(I_n - ff^T/\alpha)\tilde{S}^T, \quad f = \tilde{S}^T a, \quad \alpha = r + f^T f,$$

где n — размерность векторов \hat{x} , \tilde{x} , и потребуем так выбрать число β , чтобы обеспечить справедливость следующего разложения:

$$I_n - ff^T/\alpha = (I_n - \beta ff^T)(I_n - \beta ff^T).$$

Отсюда для β получаем квадратное уравнение и из двух его решений выбираем

$$\beta = (1/\alpha)/(1 + \sqrt{r/\alpha}),$$

поскольку выбор знака ”+” обеспечивает меньший уровень относительных ошибок при этих вычислениях. Обозначим

$$\gamma = 1/(1 + \sqrt{r/\alpha}),$$

тогда $\beta = \gamma/\alpha$. В итоге вместо (11.34) получаем следующий ряд формул:

$$\left. \begin{aligned} f &= \tilde{S}^T a, \\ \alpha &= f^T f + r, \\ \gamma &= 1/(1 + \sqrt{r/\alpha}), \\ K &= \tilde{S}f/\alpha, \\ \hat{S} &= \tilde{S} - \gamma K f^T, \\ \hat{x} &= \tilde{x} + K(z - a^T \tilde{x}), \end{aligned} \right\} \quad (11.35)$$

который и составляет *алгоритм Поттера*. Он численно более устойчив, чем стандартный ковариационный алгоритм Калмана (11.34), но ему эквивалентен. В целом, для него характерно следующее.

Вычисление \hat{S} в (11.35) равносильно счету с двойной точностью матрицы \hat{P} в (11.34) при использовании той же разрядности чисел в компьютере, или, иначе, равносильная точность вычислений матрицы \hat{P} может быть достигнута значительно быстрее. Для матрицы \hat{P} теперь отсутствует опасность потери положительной определенности, присущая операции вычитания в (11.34), поскольку здесь вычисляют \hat{S} , а $\hat{P} = \hat{S}\hat{S}^T$ и $\hat{P} > 0$ при $\det(\hat{S}) \neq 0$. Недостатком алгоритма (11.35) является наличие операции извлечения квадратного корня, отдельной для каждого скалярного наблюдения $z = a^T x + v$, и возможная потеря специального (желательно, треугольного) вида матрицы \hat{S} в общем случае. Действительно, для экономии памяти и объема вычислений обычно стремятся иметь матрицы \hat{S} и \tilde{S} в виде треугольных матриц (обе — нижние треугольные или обе — верхние треугольные), что соответствует разложениям Холецкого: $\hat{P} = \hat{S}\hat{S}^T$ и $\tilde{P} = \tilde{S}\tilde{S}^T$. Однако, если стартовать с матрицы \tilde{S} треугольного вида, выполняя инициализацию в соответствии с (11.27), то из-за вычитания в (11.35) матрица \hat{S} в общем случае не остается треугольной. Например, пусть для \hat{S} и \tilde{S} выбрана верхняя треугольная форма. Тогда только при $a = \lambda(1, 0, \dots, 0)^T$, где λ — скаляр, для \hat{S} в (11.35) будет сохранена та же верхняя треугольная форма, благодаря чему выполнение этапа экстраполяции, согласно (11.29), сводится к простому присваиванию: $\tilde{S} := \hat{S}$. Если же выбранная для \tilde{S} треугольная форма будет утрачена, то этап экстраполяции матрицы потребует предварительной триангуляризации матрицы \hat{S} , то есть операции $\tilde{S} := \text{triang}(\hat{S})$. Триангуляризация $\text{triang}(\cdot)$ должна быть выполнена ортогональным преобразованием матрицы (\cdot) , например, преобразованиями Хаусхолдера, Гивенса или же Грама–Шмидта, которые рассматривались выше (разд. 7).

11.9 Полная статистическая интерпретация рекурсии в МНК

Простая статистическая интерпретация МНК-решения, данная в п. 11.3, позволила объяснить в п. 11.4 идею включения априорных данных в процесс решения задачи о наименьших квадратах. Она была «простой» в том смысле, что предполагала случайную природу лишь для ошибки v в экспериментальных данных z , (11.1), при необходимости находить апостериорную

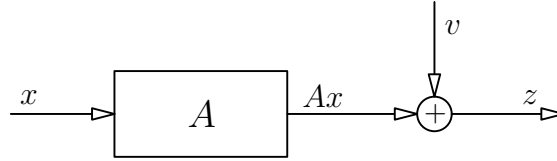


Рис. 11.4. Представление экспериментальных данных z в виде результата измерения неизвестного вектора x с помощью матрицы A в присутствии случайных ошибок v

оценку \hat{x} для некоторого (постоянного) неизвестного вектора x с учетом некоторой имеющейся (априорной) оценки \tilde{x} (см. рис. 11.4). Для той интерпретации было достаточно использовать первые два момента случайной ошибки v : математическое ожидание $\mathbf{E}\{v\} = 0$ и ковариацию $\mathbf{E}\{vv^T\} = P_v$. Ниже, описывая случайные векторы, мы также ограничиваемся первыми двумя моментами распределения вероятностей, то есть фактически принимаем гипотезу о *нормальном* (гауссовом) распределении. Отличие будет в том, что с этого момента мы вводим статистическое описание не только для v , но и для оцениваемого вектора x . При этом надо иметь в виду, что такое описание для x должно всегда рассматриваться как *условное* распределение. Именно это обстоятельство делает приводимую ниже *статистическую интерпретацию МНК полной*, т. е. учитывающей рекурсию процесса обновления оценок в два этапа.

1. Экстраполяция оценок по времени между моментами измерений; означает обновление не по измерению, а по времени; его иногда называют «*темпоральным обновлением*»; эти оценки по отношению к следующему этапу 2 действуют как априорные оценки.
2. Обновление оценок по измерениям; означает переход от текущих априорных оценок (текущего априорного условного распределения) к следующим апостериорным оценкам (новому апостериорному условному распределению вероятностей) для оцениваемого вектора x .

Предположим, что ошибка наблюдения v в уравнении (11.1) есть случайный гауссов вектор. Мы предполагаем, что это уравнение нормализовано, т. е. записано по типу уравнения (11.10), поэтому

$$\mathbf{E}\{v\} = 0, \quad \mathbf{E}\{vv^T\} = I_m, \quad (11.36)$$

где $\mathbf{E}\{\cdot\}$ — оператор математического ожидания, 0 — нулевой m -вектор, I_m — единичная матрица ($m \times m$)-матрица.

ЗАМЕЧАНИЕ 11.4. Предположение о предварительной нормализации не нарушает общности и введено только для упрощения записей. В любой момент оно может быть отозвано путем умножения нормализованных данных на корень квадратный L из матрицы R , где $LL^T = R$ (см. п. 11.3) и R — положительно определенная матрица ($R > 0$). Последнее, т. е. $R > 0$, означает, что не найдется такого невырожденного преобразования вектора наблюдений z , (11.1), после которого результат содержал бы элементы, свободные от случайных ошибок. Иными словами, все экспериментальные данные содержат случайные погрешности. Случай, когда некоторые измерения являются точными, может быть рассмотрен особо [115].

Таким образом, плотность распределения вероятностей вектора v определена выражением

$$f_v(\rho) = [(2\pi)^m |R|]^{-1/2} \exp \left\{ -\frac{1}{2} \rho^T R^{-1} \rho \right\}, \quad (11.37)$$

где согласно (11.36), $R = \mathbf{E} \{vv^T\} = I_m$.

Допустим вместе с этим, что еще до проведения эксперимента по схеме

$$z = Ax + v \quad (11.38)$$

и независимо от него вы располагаете априорной информацией о неизвестном векторе x . Это предварительное знание, явившееся результатом ваших предыдущих экспериментов либо просто кем-то другим полученное и вам сообщенное, выражается величиной \tilde{x} . Сама по себе величина \tilde{x} тоже может быть не вполне надежной, то есть вам следует ее рассматривать как случайную, а то конкретное ее значение, которое вам сообщено, вы обозначаете $\tilde{\xi}$. Это $\tilde{\xi}$ имеет смысл реализованного значения случайного вектора \tilde{x} . Кроме этого $\tilde{\xi}$, которое вы принимаете за центр распределения вектора \tilde{x} , вам нужно иметь сведения о разбросе величины \tilde{x} относительно центра, т. е. нужна ковариация \tilde{P} . Имея $\tilde{\xi}$ и \tilde{P} и продолжая работать лишь с нормальными распределениями, вы выражаете эту *априорную информацию* следующей условной (*априорно принятой*) плотностью распределения вероятностей вектора x :

$$f_{x|\tilde{x}}(\xi|\tilde{\xi}) = [(2\pi)^n |\tilde{P}|]^{-1/2} \exp \left\{ -\frac{1}{2} (\xi - \tilde{\xi})^T \tilde{P}^{-1} (\xi - \tilde{\xi}) \right\}. \quad (11.39)$$

Здесь вы формально предполагаете, что \tilde{P} обратима (т. е. $\tilde{P} > 0$), чтобы иметь возможность записать это выражение.

Теперь вы проводите свое наблюдение по схеме (11.38) и получаете конкретный его результат ζ , — реализованное значение случайного вектора z . Вы хотите использовать результат ζ , чтобы выработать наилучшую оценку \hat{x} для вектора x . Как для этого скомбинировать априорное знание $(\tilde{\xi}, \tilde{P})$ с полученным результатом ζ ? Ответ на этот вопрос кроется в апостериорном распределении, т. е. в плотности $f_{x|z, \tilde{x}}(\xi | \zeta, \tilde{\xi})$ распределения вероятностей неизвестного вектора x при условии, что известны: (1) результат ζ для вектора наблюдений z и (2) сообщенное значение $\tilde{\xi}$ априорного данного вектора \tilde{x} с ковариацией \tilde{P} , что отражено принятием априорной плотности (11.39). Если удастся найти эту $f_{x|z, \tilde{x}}(\xi | \zeta, \tilde{\xi})$, то по ней можно будет принять решение, поскольку именно плотность распределения для любой случайной величины служит наиболее полной характеристикой.

Найдем $f_{x|z, \tilde{x}}(\xi | \zeta, \tilde{\xi})$. По формуле Байеса для условных плотностей

$$f_{x|z, \tilde{x}}(\xi | \zeta, \tilde{\xi}) = \frac{f_{x, z, \tilde{x}}(\xi, \zeta, \tilde{\xi})}{f_{z, \tilde{x}}(\zeta, \tilde{\xi})}. \quad (11.40)$$

Преобразуем (11.40) по известным из теории вероятностей законам, опуская, для простоты промежуточных записей, аргументы. Получаем

$$f_{x|z, \tilde{x}}(\xi | \zeta, \tilde{\xi}) = \frac{f_{z|x, \tilde{x}} \cdot f_{x|\tilde{x}} \cdot f_{\tilde{x}}}{f_{z|\tilde{x}} \cdot f_{\tilde{x}}} = \frac{f_{z|x, \tilde{x}}(\zeta | \xi, \tilde{\xi}) f_{x|\tilde{x}}(\xi | \tilde{\xi})}{f_{z|\tilde{x}}(\zeta | \tilde{\xi})}. \quad (11.41)$$

Здесь плотность $f_{x|\tilde{x}}(\xi | \tilde{\xi})$ известна как априорная и задана выражением (11.39). Остается найти другие две. Для их нахождения опираемся на (11.38). Имеем

$$z = Ax + v \left| \begin{array}{l} x = \xi \\ \tilde{x} = \tilde{\xi} \end{array} \right. = A\xi + v \left| \begin{array}{l} \\ \tilde{x} = \tilde{\xi} \end{array} \right. = A\xi + v,$$

так как v от $\tilde{\xi}$ не зависит. Отсюда и из (11.37) получаем

$$f_{z|x, \tilde{x}}(\zeta | \xi, \tilde{\xi}) = [(2\pi)^m |R|]^{-1/2} \exp \left\{ -\frac{1}{2} (\zeta - A\xi)^T R^{-1} (\zeta - A\xi) \right\}. \quad (11.42)$$

Для другой плотности, находящейся в знаменателе (11.41), снова рассматриваем выражение (11.38), но теперь при условии $\tilde{x} = \tilde{\xi}$. Найдем совместную плотность векторов x и v при этом условии:

$$f_{x, v|\tilde{x}}(\xi, \rho | \tilde{\xi}) = \frac{f_{x, v, \tilde{x}}(\xi, \rho, \tilde{\xi})}{f_{\tilde{x}}(\tilde{\xi})} = \frac{f_{v|x, \tilde{x}}(\rho | \xi, \tilde{\xi}) f_{x|\tilde{x}}(\xi | \tilde{\xi}) f_{\tilde{x}}(\tilde{\xi})}{f_{\tilde{x}}(\tilde{\xi})} = f_v(\rho) \cdot f_{x|\tilde{x}}(\xi | \tilde{\xi}). \quad (11.43)$$

Последнее равенство получено вследствие принятого свойства ошибок v в эксперименте (11.38): они не зависят ни от x , ни от \tilde{x} . В выражении (11.43) перемножаются две гауссовы плотности: (11.37) и (11.39), поэтому результат — тоже гауссова плотность. Вектор z , (11.38), образован как взвешенная сумма двух векторов, x и v , совместная плотность которых гауссова. Поэтому плотность в знаменателе (11.41) — гауссова. Чтобы ее записать, достаточно найти по (11.38) первые два условные момента. Первый момент, учитывая (11.39), равен

$$\mathbf{E} \left\{ z \mid \tilde{x} = \tilde{\xi} \right\} = A \mathbf{E} \left\{ x \mid \tilde{x} = \tilde{\xi} \right\} + 0 = A\tilde{\xi},$$

так как $\mathbf{E} \left\{ v \mid \tilde{x} = \tilde{\xi} \right\} = 0$. Находим второй момент, используя (11.39) и (11.37):

$$\mathbf{E} \left\{ \left[(z - A\tilde{\xi})(z - A\tilde{\xi})^T \mid \tilde{x} = \tilde{\xi} \right] \right\} = A\tilde{P}A^T + R, \quad R = I_m$$

(последнее — в силу нормализации, см. замечание 11.4). Теперь имеем возможность записать

$$f_{z|\tilde{x}}(\zeta \mid \tilde{\xi}) = \left[(2\pi)^m |A\tilde{P}A^T + I_m| \right]^{-1/2} \exp\{\cdot\}, \quad (11.44)$$

где $\{\cdot\} = \left\{ -\frac{1}{2}(\zeta - A\tilde{\xi})^T (A\tilde{P}A^T + I_m)^{-1} (\zeta - A\tilde{\xi}) \right\}$.

Все три плотности для (11.41) найдены: (11.42), (11.39) и (11.44). Подставляя их в (11.41), получаем

$$f_{x|z,\tilde{x}}(\xi \mid \zeta, \tilde{\xi}) = \left[(2\pi)^n \frac{|R| \cdot |\tilde{P}|}{|A\tilde{P}A^T + R|} \right]^{-1/2} \exp \left\{ -\frac{1}{2}(\alpha + \beta - \gamma) \right\}, \quad (11.45)$$

$$\left. \begin{aligned} \alpha &= (\zeta - A\tilde{\xi})^T R^{-1} (\zeta - A\tilde{\xi}), \\ \beta &= (\xi - \tilde{\xi})^T \tilde{P}^{-1} (\xi - \tilde{\xi}), \\ \gamma &= (\zeta - A\tilde{\xi})^T (A\tilde{P}A^T + R)^{-1} (\zeta - A\tilde{\xi}). \end{aligned} \right\} \quad (11.46)$$

ЗАМЕЧАНИЕ 11.5. Здесь и далее для удобства (прослеживания местоположения матрицы ковариаций ошибки v) вместо I_m , (11.36), сохранено общее обозначение R , как и в (11.37), которое в любой момент можно заменить на I_m , если будет принято предположение о нормализации наблюдений.

Очевидно, (11.45), (11.46) определяют нормальную плотность распределения, однако, ее явное определение требует двух действий: (1) приведение суммы трех квадратичных форм (11.46) к одной квадратичной форме и (2) приведение отношения трех определителей к одному определителю. Проведем эти преобразования. При первом действии активно используем лемму 11.1 (см. п. 11.7), беря ее в виде:

$$\left(\tilde{P}^{-1} + A^T R^{-1} H\right)^{-1} = \tilde{P} - \tilde{P} A^T \left(A \tilde{P} H^T + R\right)^{-1} A \tilde{P} \quad (11.47)$$

с обозначением $\tilde{\Lambda} = \tilde{P}^{-1}$ для априорной информационной матрицы. Перепишем (11.47) в нескольких эквивалентных формах. Имеем

$$\begin{aligned} \left(\tilde{P}^{-1} + A^T R^{-1} A\right)^{-1} A^T &= \tilde{P} A^T - \tilde{P} A^T \left(A \tilde{P} A^T + R\right)^{-1} A \tilde{P} A^T = \\ &= \tilde{P} A^T \left(A \tilde{P} A^T + R\right)^{-1} \left[\left(A \tilde{P} A^T + R\right) - A \tilde{P} A^T\right] = \tilde{P} A^T \left(A \tilde{P} A^T + R\right)^{-1} R. \end{aligned}$$

Отсюда

$$\tilde{P}^{-1} \left(\tilde{P}^{-1} + A^T R^{-1} A\right)^{-1} A^T R^{-1} = A^T \left(A \tilde{P} A^T + R\right)^{-1}. \quad (11.48)$$

Умножая (11.47) слева и справа на \tilde{P}^{-1} , получаем

$$\tilde{P}^{-1} \left(\tilde{P}^{-1} + A^T R^{-1} A\right)^{-1} \tilde{P}^{-1} = \tilde{P}^{-1} - A^T \left(A \tilde{P} A^T + R\right)^{-1} A. \quad (11.49)$$

Умножая (11.47) слева на A и справа на A^T и обозначая $C = A \tilde{P} A^T + R$, находим

$$\begin{aligned} A \left(\tilde{P}^{-1} + A^T R^{-1} A\right)^{-1} A^T &= A \tilde{P} A^T - A \tilde{P} A^T C^{-1} A \tilde{P} A^T = \\ &= (C - R) - (C - R) C^{-1} (C - R) = R - R \left(A \tilde{P} A^T + R\right)^{-1} R. \end{aligned}$$

Отсюда

$$R^{-1} A \left(\tilde{P}^{-1} + A^T R^{-1} A\right)^{-1} A^T R^{-1} = R^{-1} - \left(A \tilde{P} A^T + R\right)^{-1}. \quad (11.50)$$

Вычисляя квадратичную форму в (11.45) как $(\alpha + \beta - \gamma)$, введем промежуточные обозначения:

$$\left. \begin{aligned} \hat{P} &= \left(\tilde{P}^{-1} + A^T R^{-1} A\right)^{-1}, \\ a &= A^T R^{-1} \zeta, \\ b &= \tilde{P}^{-1} \tilde{\xi}, \\ c &= a + b. \end{aligned} \right\} \quad (11.51)$$

Теперь после раскрытия скобок в (11.46), приведения подобных членов в $(\alpha + \beta - \gamma)$ и при подстановках (11.48), (11.49) и (11.50) получим

$$\alpha + \beta - \gamma = \xi^T \hat{P}^{-1} \xi - 2\xi^T c + c^T \hat{P} c = (\xi - \hat{P} c)^T \hat{P}^{-1} (\xi - \hat{P} c). \quad (11.52)$$

Кроме того,

$$\begin{aligned} \hat{P} c &= \hat{P} \left(A^T R^{-1} \zeta + \tilde{P}^{-1} \tilde{\xi} \right) = \\ &= \hat{P} \left(A^T R^{-1} \zeta + \tilde{P}^{-1} \tilde{\xi} + A^T R^{-1} A \tilde{\xi} - A^T R^{-1} A \tilde{\xi} \right) = \\ &= \hat{P} \left[\left(\tilde{P}^{-1} + A^T R^{-1} A \right) \tilde{\xi} + A^T R^{-1} (\zeta - A \tilde{\xi}) \right] = \\ &= \tilde{\xi} + \hat{P} A^T R^{-1} (\zeta - A \tilde{\xi}). \end{aligned} \quad (11.53)$$

Из (11.48) и (11.51) видно, что

$$\hat{P} A^T R^{-1} = K = \tilde{P} A^T \left(A \tilde{P} A^T + R \right)^{-1} \quad (11.54)$$

есть матрица Калмана, определенная выражением (11.32) (см. п. 11.7). В свою очередь, величина (11.53), входящая в квадратичную форму (11.52) плотности распределения (11.45), есть не что иное как среднее значение вектора x , обусловленное двумя событиями: $z = \zeta$ и $\tilde{x} = \tilde{\xi}$ (первое — результатом ζ измерения и второе — результатом $\tilde{\xi}$ априорной оценки). Обозначим это условное среднее как \hat{x} , тогда (11.53) перепишется в виде:

$$\hat{x} = \tilde{\xi} + K(\zeta - A\tilde{\xi}). \quad (11.55)$$

Кроме того, из (11.52) видно, что матрица \hat{P} в (11.51) есть ковариация апостериорного распределения (11.45), и по лемме (11.47) она дается выражением

$$\hat{P} = \tilde{P} - \tilde{P} \left(A \tilde{P} A^T + R \right)^{-1} A \tilde{P} = \tilde{P} - K A \tilde{P}. \quad (11.56)$$

Сравнивая (11.54), (11.55) и (11.56) с подразд. 11.7, убеждаемся в том, что исходя из другой — чисто статистической задачи оценивания вектора x по зашумленным наблюдениям его компонент через матрицу A (см. рис. 11.4), мы получаем полное алгебраическое совпадение с рекурсией для МНК в стандартной ковариационной форме, если при этом принято во внимание, что $z = \zeta$ и $\tilde{x} = \tilde{\xi}$, а в качестве искомой наилучшей оценки \hat{x} для вектора x мы приняли апостериорное среднее значение (11.55), что соответствует критерию максимума апостериорной вероятности (МАН) (11.40), (11.41), (11.45).

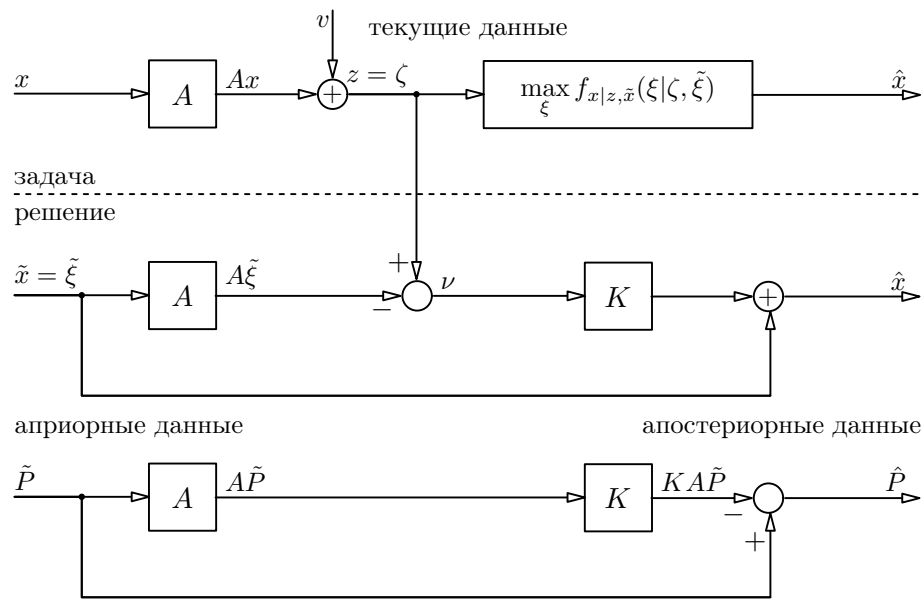


Рис. 11.5. Статистическая задача получения оценки \hat{x} по критерию МАВ и ее решение (полная статистическая интерпретация МНК-решения). Между моментами получения текущих данных апостериорные данные занимают место априорных данных, и процесс решения повторяется

Таким образом, при нормальных законах распределения и независимых ошибках наблюдения v МНК-решение \hat{x} интерпретируется как МАВ-оценка вектора x (рис. 11.5). Именно МАВ-оценка отвечает на поставленный выше вопрос (см. между (11.39) и (11.40)): как наилучшим образом скомбинировать априорные данные $(\tilde{\xi}, \tilde{P})$ с текущими данными ζ .

Для решения этой задачи, как видно из изложенного, пришлось вычислить апостериорную плотность (11.41) и затем найти максимизирующий ее аргумент ξ , — он оказался равен (11.55). Однако потребовались сложные вычисления, которые пока еще не коснулись отмеченного выше приведения отношения определителей в (11.45) к одному определителю. Из условия нормировки плотности видно, что он должен оказаться равен $|\hat{P}|$, но это еще предстоит доказать. Отложим это доказательство на окончание этого подразд. 11.9 (см. стр. 260), а сейчас рассмотрим, что будет, если изменить критерий, то есть ради простоты максимизировать по ξ не (11.41), а только одну из плотностей в числителе (11.41), — именно, (11.42), что означает *критерий максимального правдоподобия*, МП. Очевидно, такое решение получается моментально: максимум (11.42) по ξ совпадает с минимумом квадратичной формы в показателе экспоненты. При $R = I$ (что уже обсуждалось как при-

нятое неограничительное условие нормализации зашумленных наблюдений) имеем

$$\hat{x} = \arg \min_{\xi} (\zeta - A\xi)^T (\zeta - A\xi) = \hat{x}_{\text{МНК}} = \hat{x}_{\text{МП}},$$

то есть \hat{x} есть МНК-решение, $\hat{x}_{\text{МНК}}$, системы (11.1), переписанной здесь в виде $\zeta = A\xi + v$ с $R = I$, и одновременно, это \hat{x} есть оценка максимума правдоподобия, $\hat{x}_{\text{МП}}$.

Тем самым, получена полная статистическая интерпретация МНК-решения с точки зрения теории оценок:

1. Если в МНК-решении учтена априорная информация, то полученное решение совпадает с оценкой МАВ, $\hat{x}_{\text{МАВ}}$ (11.55). Эту оценку называют также *байесовской*, поскольку ее вывод основан на формуле Байеса (11.40). Любой алгоритм (т.е. инструмент) вычисления оценки называют *оценивателем*. Байесовский оцениватель в линейной задаче оценивания с нормальными законами распределения случайных величин дается формулами (11.55) и (11.56). Он совпадает с формулами фильтра Калмана — на этапе обновления оценок по измерениям, и он имеет наглядную схемную интерпретацию на рис. 11.5.
2. Если в МНК-решении не учтена априорная информация, то полученное решение имеет смысл оценки МП, $\hat{x}_{\text{МП}}$. Пренебрежение априорной информацией может быть вызвано тем, что она очень мала (недостоверна). Этот факт выражается простой записью: $\tilde{P} = \varepsilon^{-2}I$ при $\varepsilon \rightarrow 0$, как уже отмечалось на стр. 248.
3. Поскольку недовершенство или скудность априорной информации выражается большим ростом диагональных элементов матрицы \tilde{P} по сравнению с внедиагональными элементами: $\tilde{P}^{-1} \rightarrow 0$ при $\tilde{P}^{-1} = \varepsilon^2 I$, $\varepsilon \rightarrow 0$, то $\hat{P}^{-1} \rightarrow A^T A$, что формально видно из (11.51) при $R = I$. Оцениватель, который пренебрегает априорной информацией, называют *фишеровским оценивателем* в связи с именем R. A. Fisher — известного статистика. В этом оценивателе

$$\hat{x}_{\text{МАВ}} \rightarrow (A^T A)^{-1} A^T \zeta = \hat{x}_{\text{МНК}}, \quad (11.57)$$

что следует из преобразований выражения (11.53) для (11.55) при использовании (11.51) в пределе при $\tilde{P}^{-1} \rightarrow 0$ (и с неограничительным условием нормализации наблюдений $R = I$), — если, конечно, матрица

$(A^T A)$ обратима. Если же информационная матрица $(A^T A)$ полученных измерений $z \equiv \zeta$ не обратима, тогда вместо (11.57) предельное соотношение записывается в виде

$$\hat{x}_{\text{МAB}} \rightarrow A^+ \zeta + (I - A^+ A) \tilde{\xi} = \hat{x}_{\text{МНК}}, \quad (11.58)$$

что вытекает из теоремы 10.16 (см. подразд. 10.1) и означает сходимость $\hat{x}_{\text{МAB}}$ к общему решению нормальных уравнений МНК.

ЗАМЕЧАНИЕ 11.6. В (11.58) первое слагаемое есть нормальное псевдорешение при минимизации $\|z - Ax\|^2$, где $z = \zeta$, а второе — проекция *любого* $\tilde{\xi}$ на $\mathcal{N}(A)$.

Теперь, в завершение необходимых действий по приведению (11.45) к стандартному виду плотности для нормального закона распределения, кроме (11.52), докажем, что

$$\frac{|R| \cdot |\tilde{P}|}{|A\tilde{P}A^T + R|} = |\hat{P}|, \quad (11.59)$$

где \hat{P} определено формулой (11.56). Для этого введем вспомогательную матрицу

$$P^* = \left[\begin{array}{c|c} \tilde{P} & \tilde{P}A^T \\ \hline A\tilde{P} & A\tilde{P}A^T + R \end{array} \right].$$

Выполним ее блочное UL -разложение, где U — верхняя треугольная матрица с положительно определенными блоками на диагонали, а L — нижняя треугольная матрица с единичной диагональю. Получаем

$$P^* = \left[\begin{array}{cc} \tilde{P} - K A \tilde{P} & \tilde{P} A^T \\ 0 & A \tilde{P} A^T + R \end{array} \right] \left[\begin{array}{cc} I & 0 \\ K^T & I \end{array} \right].$$

Это позволяет найти определитель для P^* как произведение определителей диагональных блоков. С учетом (11.56), имеем

$$|P^*| = |\hat{P}| \cdot |A\tilde{P}A^T + R|. \quad (11.60)$$

С другой стороны, воспользуемся следующей леммой обращения симметричной положительно определенной матрицы [115].

ЛЕММА 11.2. При симметричных $X_1 > 0$, $X_2 > 0$ и произвольной $X_{21} = X_{21}^T$ справедливо равенство

$$\begin{aligned} \left[\begin{array}{cc} X_1 & X_{12} \\ X_{21} & X_2 \end{array} \right]^{-1} &= \left[\begin{array}{cc} I & -X_1^{-1}X_{12} \\ 0 & I \end{array} \right] \left[\begin{array}{cc} X_1^{-1} & 0 \\ 0 & (X_2 - X_{12}^T X_1^{-1} X_{12})^{-1} \end{array} \right] \times \\ &\times \left[\begin{array}{cc} I & 0 \\ -X_{12}^T X_1^{-1} & I \end{array} \right]. \end{aligned}$$

ДОКАЗАТЕЛЬСТВО. Проведите самостоятельно прямой проверкой. \square

Беря определитель в этом выражении, получаем

$$\begin{vmatrix} X_1 & X_{12} \\ X_{12}^T & X_2 \end{vmatrix}^{-1} = |X_1|^{-1} \cdot |X_2 - X_{12}^T X_1^{-1} X_{12}|^{-1}.$$

Применим этот результат к P^* , обозначая $X_1 = \tilde{P}$, $X_2 = A\tilde{P}A^T + R$, $X_{12} = \tilde{P}A$. Находим

$$|P^*| = |\tilde{P}| \cdot |R|. \quad (11.61)$$

Сопоставляя (11.60) и (11.61), получаем (11.59). \square

11.10 Основные результаты

1. Критерий наименьших квадратов

$$\min_x (z - Ax)^T (z - Ax) \quad (11.62)$$

для решения алгебраических линейных систем общего вида

$$z = Ax + v \quad (11.63)$$

не является изначально статистическим критерием. Решение \bar{x} , отвечающее этому критерию, дается *нормальными уравнениями*

$$(A^T A) \bar{x} = A^T z \quad (11.64)$$

и имеет вид суммы взаимно ортогональных векторов:

$$\bar{x} = A^+ z + (I - A^+ A) y \quad (11.65)$$

для некоторого y . Это решение единственно тогда и только тогда, когда $A^+ A = I$, что означает, что только нулевой вектор составляет ядро $\mathcal{N}(A)$ матрицы A (т.е. A имеет полный столбцовый ранг). *Нормальное псевдорешение* \bar{x}_0 есть единственный вектор из (11.65), имеющий минимальную норму, т.е. $\bar{x}_0 = A^+ z$.

2. Статистическая интерпретация задачи (11.62) означает следующее:

- (а) x есть постоянный вектор (случайный или нет), который необходимо оценить по результатам наблюдения z в виде (11.63).

- (б) A есть матрица наблюдений, которая показывает, какие линейные комбинации элементов вектора x включены, по условиям задачи, в вектор наблюдения z .
- (в) При этом v есть случайная ошибка наблюдения, ковариация которой R задана как положительно определенная матрица. Однако, если задача (11.62), (11.63) чисто алгебраическая, $R = I$.
- (г) При статистической интерпретации, матрица $(A^T A)$ в (11.64) называется *информационной матрицей* Λ . Она обратима тогда и только тогда, когда $\text{rank } A = n$ (матрица A имеет полный столбцовый ранг). Обратная к Λ матрица называется *ковариационной матрицей* ошибки оценивания вектора x , $P = \Lambda^{-1}$, и она характеризует степень неопределенности в решении задачи оценки (в скалярном случае она равна дисперсии).
3. Статистическая задача оценивания, указанная в предыдущем пункте, приводит к таким же (алгебраически эквивалентным) результатам, как решение алгебраической задачи (11.62), (11.63).
4. Решение задачи (11.62), (11.63), так же как и решение ее статистического эквивалента, может быть получено в рекуррентной форме, удобной с вычислительной точки зрения. Возможны две базовые эквивалентные рекуррентные формы (взаимно инверсные): *информационная схема оценивания* (неявная относительно апостериорной оценки \hat{x}) и *ковариационная схема оценивания* (явная относительно \hat{x}).
5. *Оценщик максимального правдоподобия*, выбирает оценку $\hat{x}_{\text{МП}}$ так, чтобы максимизировать функцию правдоподобия (11.42). При гауссовых независимых ошибках она является решением тех же самых нормальных уравнений

$$A^T R^{-1} A \bar{x} = A^T R^{-1} \zeta, \quad (11.66)$$

которые отвечают *критерию взвешенных наименьших квадратов*

$$\min_x (z - Ax)^T R^{-1} (z - Ax), \quad z = \zeta. \quad (11.67)$$

По смыслу, $\hat{x}_{\text{МП}}$ есть значение неизвестного вектора x , которое с наибольшей вероятностью обеспечивает то событие, что при случайном наблюдении (11.63) результат окажется равным ζ ($\hat{x}_{\text{МП}}$ обеспечивает наибольшую вероятность события $z = \zeta$). Эта оценка не учитывает никакой априорной информации о векторе x . С другой стороны, оценка,

удовлетворяющая «взвешенному» критерию (11.67) и даваемая, соответственно, решением «взвешенных» нормальных уравнений (11.66), называется *марковской оценкой* в связи с *теоремой Гаусса–Маркова*, которая утверждает, что эта оценка $\hat{x}_{МП}$, удовлетворяющая уравнениям (11.66), — *эффективная в классе всех линейных несмещенных оценок*, см., например, [49, 17].

6. *Оценщик максимума апостериорной вероятности* выбирает оценку $\hat{x}_{МАВ}$, в отличие от $\hat{x}_{МП}$, принципиально в рекуррентной форме, т. е. учитывает априорную информацию и наилучшим образом комбинирует ее с текущими наблюдениями. Оценка $\hat{x}_{МАВ}$ вычисляется в явном виде в алгоритме Калмана (11.54), (11.55), (11.56) (см. подразд. 11.7) или в неявном виде в информационном алгоритме (см. подразд. 11.6). При недостаточной (т. е. крайне неопределенной, скудной) априорной информации, оценка $\hat{x}_{МАВ}$ будет такой же, как $\hat{x}_{МП}$, или, что то же самое, $\hat{x}_{МНК}$ в задаче (11.67) с решением из нормальной системы (11.66).
7. Рекурсия (рекуррентная схема вычислений) в алгоритмах МНК и оптимального оценивания является краеугольным камнем теории и практики эффективных вычислительных алгоритмов, рассматриваемых ниже в разд. 13 и 14.

Следующий разд. 12, занимающий промежуточное положение, рассматривает случай, когда рекуррентная схема вычислений в задачах МНК не используется, т. е. когда нормальные уравнения решаются для всех накопленных экспериментальных данных целиком — после того, как все они накоплены и сохранены для обработки.

12

ОДНОВРЕМЕННОЕ РЕШЕНИЕ НОРМАЛЬНЫХ УРАВНЕНИЙ

12.1 Метод нормальных уравнений

Нормальные уравнения (11.3) показаны выше (см. стр. 238). Довольно часто их используют для отыскания МНК-решения \bar{x} [97]. Этот подход противоположен последовательным методам, берущим начало от идеи расщепления исходной переопределенной системы на априорную и текущую части (см. подразд. 11.5). Поэтому иногда его называют «одновременным решением» всех нормальных уравнений, характеризующих всю исходную переопределенную систему $Ax \approx z$ (см. стр. 238 и выражение (11.1)).

Метод Нормальных Уравнений. Алгоритм использует полный вектор наблюдений $z \in \mathbb{R}^m$ и всю матрицу плана эксперимента $A \in \mathbb{R}^{m \times n}$, имеющую $\text{rank}(A) = n$. По матрице A и вектору z алгоритм вычисляет решение \bar{x} задачи наименьших квадратов, доставляющее $\min \|z - Ax\|^2$. Алгоритм:

1. Вычисляет нижний треугольник матрицы $\Lambda = A^T A$.
2. Вычисляет $d = A^T z$.
3. Вычисляет разложение Холецкого $\Lambda = SS^T$.
4. Решает сначала систему $Sy = d$ и затем систему $S^T \bar{x} = y$.

Здесь в качестве разложения Холецкого может быть взято либо нижнее треугольное разложение ($S = L$), либо верхнее треугольное разложение (при $S = U$) — см. подразд. 6.2, стр. 91. Оба эти варианта требуют многократного применения операции извлечения квадратного корня. Если в п. 3 алгоритма вычислять разложение Холецкого без операции квадратного корня: либо $\Lambda = \bar{S}D\bar{S}^T$ с $\bar{S} = \bar{L}$, либо $\bar{S} = \bar{U}$, то п. 4 заменится на последовательное решение трех систем: $\bar{S}w = d \Rightarrow Dy = w \Rightarrow \bar{S}^T \bar{x} = y$.

12.2 Формирование матрицы A

Используйте следующие четыре варианта формирования матрицы A для оценки скорости и точности двух методов решения переопределенных систем линейных алгебраических уравнений, как указано в пп. В и Г задания (см. Задание ниже в подразд. 12.3).

Для $n = 2$ до 12 с шагом 5 выполнять

Для $m = n + 1$ до 26 с шагом 3 выполнять

Для $i = 1$ до m выполнять

Для $j = 1$ до n выполнять

$$\text{Вариант 1: } a_{ij} = \sin \left(\frac{(i-1)j}{m} \right),$$

$$\text{Вариант 2: } a_{ij} = \frac{1}{1 + 66 \left[\frac{(i-1)j}{m} \right]^4},$$

$$\text{Вариант 3: } a_{ij} = 1/(i+j),$$

$$\text{Вариант 4: } a_{ij} = 100(RAN - 1/2),$$

где RAN — равномерно распределенная в интервале $[0, 1]$ случайная величина, получаемая независимо для каждого элемента a_{ij} матрицы A .

12.3 Задание на лабораторный проект № 8

А. Спроектировать и отладить подпрограмму решения несовместной системы $Ax = z$, $A = A(m, n)$, $m > n$, $\text{rank}(A) = n$, в смысле наименьших квадратов при помощи заданного метода ортогонального приведения. Обосновать проект и дать набор инструкций для пользователей подпрограммы. Сделать подсчет операций (отдельно сложения, умножения, деления и извлечения квадратного корня) в зависимости от m и n , где m — число строк матрицы A , а n — число столбцов. Рекомендуется в качестве основы вашего проекта использовать ту программу, которая была вами написана и отлажена в рамках лабораторного проекта № 6 для решения совместной системы уравнений $Ax = f$ с квадратной матрицей A методом ортогонального приведения. Для этого в указанной программе достаточно осуществить незначительные изменения.

Б. Повторить п. А задания на основе метода нормальных уравнений $A^T A \bar{x} = A^T z$, которым удовлетворяет искомое решение \bar{x} в смысле наименьших квадратов, называемое нормальным псевдорешением несовместной системы $Ax \approx z$. Для этого применить вашу программу решения системы уравнений с симметричной положительно определенной матрицей методом квадратного корня (разложение Холецкого) из лабораторной работы № 5.

В. Спроектировать и провести вычислительный эксперимент для сравнения скорости выполнения двух программ по пп. А. и Б. Использовать четыре различных варианта генерации n векторов длины m для формирования матрицы A (см. подразд. 12.2) при $2 \leq n \leq 12$, $n + 1 \leq m \leq 26$. Результаты представить в виде таблиц и графиков, которые иллюстрируют поведение каждого метода на каждом варианте генерации матрицы A . Дать обобщенную (по вариантам матрицы) картину зависимости времени выполнения от значений параметров m и n матрицы. Проанализировать соотношение между фактическим временем выполнения и числом операций, рассчитанным по пп. А и Б. Правые части уравнений формировать, как указано в следующем пункте задания.

Г. Подобно п. В, сравнить точность нахождения нормального псевдорешения переопределенной системы линейных уравнений для методов из пп. А и Б. Для этого также четырьмя способами сгенерировать матрицу A (см. подразд. 12.2), выбрать (принудительно задать) точное нормальное псевдорешение x^* и образовать вектор $z^* = Ax^*$. К элементам этого вектора добавить случайные числа v_i , чтобы образовать правые части $z_i = z_i^* + av_i$, $i = 1, 2, \dots, m$. Написать подпрограмму генерации псевдослучайных чисел v_i так, чтобы каждое v_i , имело стандартное нормальное распределение (с нулевым средним значением и единичной дисперсией). При этом любые случайные величины v_i, v_j ($i \neq j$) должны моделироваться как попарно независимые. Предусмотреть множитель-переключатель a , чтобы по желанию включать или отключать добавление случайных чисел v_i , или же регулировать их уровень.

В качестве точного нормального псевдорешения взять вектор $x^* = [1, 2, \dots, m]^T$. Для оценки точности оценивания использовать норму вектора

$$\|x\|_\infty = \max_i (|x_i|).$$

При использовании программы, где выполняется ортогональное приведение $QA = B$, для проверки правильности метода убедиться в справедливо-

сти равенства $A^T A - B^T B = 0$. Для этого использовать норму матрицы

$$\|A\|_\infty = \max_i \left(\sum_{j=1}^n |a_{ij}| \right).$$

Д. Представить обобщенную аттестацию двух подпрограмм и соответствующих методов, основанную на проведенных наблюдениях. Обсудить любые сравнительные достоинства и недостатки, поддающиеся количественной оценке, и предложить план дальнейших вычислительных экспериментов, которые могли бы помочь уточнить различия между рассмотренными выше методами решения переопределенных систем уравнений.

Е. Решить следующую прикладную задачу [82]. Для $i = 1, 2, \dots, m$ (m кратно четырем) имеем

$$y_i = x_1 w_i + x_2 w_{i-1}, \quad w_i = \sin(2\pi i/m), \quad d_i = 2 \cos(2\pi i/m).$$

Найти оптимальное значение $\bar{x} = (\bar{x}_1, \bar{x}_2)^T$ вектора коэффициентов $x = (x_1, x_2)^T$, доставляющее минимум средней квадратической ошибке

$$J(x) = \frac{1}{m} \sum_{i=1}^m (y_i - d_i)^2.$$

Решение выполнить двумя способами: аналитически и численно. Аналитическое решение должно включать:

- 1) эквивалентную постановку задачи в виде переопределенной системы $Ax \approx z$,
- 2) решение для нее нормальных уравнений, дающее

$$\bar{x} = 2 \left[\begin{array}{c|c} \operatorname{ctg}(2\pi/m) & -\operatorname{cosec}(2\pi/m) \end{array} \right]^T,$$

- 3) представление критерия качества для общего случая в виде

$$J(x) = J_{\min} + (x - \bar{x})^T \Lambda (x - \bar{x}),$$

где $\Lambda = A^T A$ — информационная матрица, $\bar{x} = (A^T A)^{-1} A^T z$ — нормальное псевдорешение,

- 4) доказательство того, что в данном конкретном случае

$$J_{\min} = \min_x (J(x)) = J(\bar{x}) = 0,$$

5) вычисление собственных значений (λ_1, λ_2) матрицы Λ , дающее

$$\lambda_1 = \frac{1}{2} \left[1 - \cos(2\pi/m) \right], \quad \lambda_2 = \frac{1}{2} \left[1 + \cos(2\pi/m) \right],$$

6) вычисление соответствующих собственных векторов (v_1, v_2) матрицы Λ ,

7) представление критерия качества для общего случая в виде

$$J(x) = J_{\min} + e^T Q \bar{\Lambda} Q^T e,$$

где $e = x - \bar{x}$ — отклонение x от оптимального значения \bar{x} , Q — матрица ортонормированных собственных векторов матрицы Λ , диагональная матрица $\bar{\Lambda} = \text{diag} [\lambda_1, \lambda_2]$ составлена из собственных значений матрицы Λ , так что $\Lambda = Q \bar{\Lambda} Q^{-1} = Q \bar{\Lambda} Q^T$.

Изобразить на экране в системе координат $[x_1, x_2] = x^T$ линии постоянных уровней критерия $J(x) = \text{const}$ для шести значений $\text{const} = \{1, 2, 3, 4, 5, 6\}$ в окрестности точки минимума критерия для одного из значений $m = 4, 8, 12, 16, 20, 24, 28$ или 32 (по выбору). Объяснить геометрический смысл матриц Q и $\bar{\Lambda}$ в последнем представлении критерия.

Численное решение должно включать вычисление решения \bar{x} с помощью двух методов (по пп. А и Б) и сравнительную оценку точности двух решений для нормы вектора $\|\bar{x}\|_\infty$ (в зависимости от $m = 4, 8, 12, 16, 20, 24, 28, 32$). Эту зависимость необходимо представить таблицей и графиком.

12.4 Варианты задания на лабораторный проект № 8

По теме «Одновременное решение нормальных уравнений» студентам предлагается выполнение лабораторной работы — проекта № 8.

Задание на этот проект содержит 28 вариантов, которые аналогичны вариантам, приведенным в табл. 7.2 (см. стр. 139) для проекта № 6. Все варианты различаются по следующим признакам:

- четыре варианта заполнения треугольной матрицы R ;
- три вида ортогональных преобразований:
 - 1) отражения Хаусхолдера,
 - 2) вращения Гивенса,
 - 3) ортогонализация Грама–Шмидта,

Таблица 12.1. Варианты задания на лабораторный проект № 8

Вариант заполнения матрицы R	Отражения Хаусхолдера		Вращения Гивенса		Ортогонализация Грама-Шмидта		
	a	b	a	b	c	d	e
 $\triangleq R_{\text{ne}}$	1	2	3	4	5	6	7
 $\triangleq R_{\text{nw}}$	8	9	10	11	12	13	14
 $\triangleq R_{\text{se}}$	15	16	17	18	19	20	21
 $\triangleq R_{\text{sw}}$	22	23	24	25	26	27	28

a — столбцово-ориентированный алгоритм;

b — строчно-ориентированный алгоритм;

c — классическая схема;

d — модифицированная схема;

e — модифицированная схема с выбором ведущего вектора.

- две разновидности алгоритма ортогонализации по методу Хаусхолдера и по методу Гивенса:

- 1) столбцово-ориентированный алгоритм,
- 2) строчно-ориентированный алгоритм,

- три разновидности ортогонализации по методу Грама–Шмидта:

- 1) классическая схема,
- 2) модифицированная схема,
- 3) модифицированная схема с выбором ведущего вектора.

Если нет других указаний преподавателя, выбирайте ваш вариант в табл. 12.1 по вашему номеру в журнале студенческой группы.

УСТОЙЧИВЫЕ АЛГОРИТМЫ ФИЛЬТРАЦИИ

13.1 Фильтрация Калмана в историческом аспекте

Вскоре после открытия Калманом своего ставшего знаменитым нового подхода к задачам фильтрации и предсказания [101] выяснилось, что это изящное решение, которое теперь принято называть «*обыкновенным*» или *стандартным фильтром Калмана* (the conventional Kalman filter, СКФ), работает хорошо лишь в исключительных — так называемых хорошо обусловленных задачах — и расходится в большинстве практических задач. *Явление расходимости* теоретического алгоритма СКФ, детально исследованное 10 лет спустя [98], породило поиски альтернатив, алгебраически эквивалентных СКФ, но в вычислениях значительно более устойчивых.

В 1963 году Поттер нашел первое решение [125] в связи с разработкой американского лунохода (Lunar Excursion Module, LEM) для Программы Аполлон. По природе, это квадратно-корневой алгоритм, названный *квадратно-корневым фильтром Поттера*, PSRF (Potter Square Root Filter). Будучи приспособлен лишь для ограниченных случаев (некоррелированность скалярных измерений и отсутствие шума в уравнении состояния), PSRF породил две ключевые идеи: (1) *факторизация* (разложение на множители) ковариационных матриц, чтобы устранить опасность потери положительной определенности, и (2) *декорреляция и последующая скаляризация* векторных измерений, чтобы устранить операцию вычисления обратной матрицы в алгоритме СКФ.

Любой квадратно-корневой алгоритм фильтрации численно более устойчив, чем СКФ, так как число обусловленности квадратного корня из матрицы есть квадратный корень из числа обусловленности соответствующей матрицы. Этот факт широко известен и вошел в учебные курсы численных методов алгебры (разложение Холесского) — см. выше разд. 6 и [74].

Другая идея, также вошедшая в учебные курсы, — *ортогонализация* матриц (см. выше разд. 7 и [74]) — развивалась параллельно для численного решения плохо обусловленных и переопределенных систем по методу наименьших квадратов (МНК) [112]. Важный промежуточный результат тех лет — опубликование обзора [104] и монографии Бирмана [15].

Кроме исходных аэрокосмических приложений, эти и другие современные методы калмановской фильтрации вошли в другие области: эконометрика [93], метеорология [110], спутниковая геодезия [91], телекоммуникационные сети [121], высоконадежная спутниковая навигация [111], обработка изображений в реальном времени [124] и многое другое.

Казалось бы, основные идеи исчерпаны и все полезные результаты в этой области получены. Однако широта приложений заставила использовать новую вычислительную среду реализации этих алгоритмов — параллельные компьютеры [8], в результате чего они оказались представлены в новой, «блочной» форме с ортогонализацией [17] (эти инновации изложены ниже в разд. 14). Появилось учебное пособие, рассчитанное на широкий круг пользователей и отражающее почти все достижения в этой области к концу 20 столетия [16].

Указанные исследования велись за пределами СССР. По настоящее время они остаются в России малоизвестны, применены в немногих сложных проектах (например, в авиационном приборостроении [80] или судостроении [78, 79]) и преподаются в немногих специальных курсах [77]. В российской научной литературе квадратно-корневые алгоритмы фильтрации совсем не изложены [27] или освещены слабо [64]. Так, в книге Огаркова, 1990 г. [64] этой теме посвящен лишь один параграф 3.7. Нехватка литературы сказывается в том, что многие отечественные специалисты, работающие в прикладных областях регрессионного моделирования или эконометрики, продолжают использовать алгоритмы, которые можно считать устаревшими, несмотря на то, что они испытывают значительные трудности в случае плохо обусловленной схемы наблюдения или при мультиколлинеарности регрессоров [26]. Даже в публикациях последних лет [33] авторы пользуются уравнениями фильтра Калмана в их классическом виде, а не в форме современных вычислительно эффективных алгоритмов. Ситуация в России такова, что здесь лишь немногие специалисты уделяли внимание квадратно-корневым, а также ортогонализированным блочным алгоритмам, принадлежащим новому классу параллельно-ориентированных реализаций [64, 73, 78, 79, 86, 107, 108, 132, 135, 138].

Первая цель данного раздела — привлечь внимание студентов, магистрантов, аспирантов и специалистов к этому направлению исследований. Здесь мы ограничиваемся лишь кратким обзором алгоритмов нового класса, поскольку более подробные обзоры доступны. Коллективная монография [73] и книга [15] содержат подробный анализ и сравнительные таблицы эффективности стандартных и квадратно-корневых алгоритмов дискретной фильтрации. Прекрасный обзор, — начиная от зарождения идей у *Галилео Галилея* (1564–1642), через работы *Карла Фридриха Гаусса* (1777–1855), *Норберта Винера* (1894–1964) и многих других (список огромен) до *Рудольфа Эмиля Калмана* (1930–) и последних достижений, датируемых до 2001 года, содержится в учебном пособии [16].

Вторая цель данного раздела — показать прикладные возможности квадратно-корневых алгоритмов. Для примера рассмотрено их использование в актуальной задаче обнаружения момента вхождения судна в маневр, новое решение которой содержится в [41]. Соответствующий материал с опорой на достижения в этой области [60, 90, 95, 114, 128] помещен в подразд. 13.12. Работа [95], решающая ту же задачу сопровождения маневрирующих целей, также использует квадратно-корневую реализацию фильтров. В отличие от [95], в [41] применен *расширенный фильтр* первого, а не второго порядка; его построение здесь включено для примера.

13.2 Стандартный фильтр Калмана

Рассмотрим *источник данных*, представимый в виде линейной динамической системы, возмущаемой дискретным белым шумом:

$$x(t_{i+1}) = \Phi(t_{i+1}, t_i)x(t_i) + B(t_i)u(t_i) + \Gamma(t_i)w(t_i), \quad (13.1)$$

$$z(t_i) = H(t_i)x(t_i) + v(t_i), \quad (13.2)$$

где $\Phi(t_{i+1}, t_i)$, $B(t_i)$, $\Gamma(t_i)$, $H(t_i)$ — известные матрицы-параметры системы; $x(t_i)$ — n -мерный вектор состояния системы, $u(t_i)$ — r -мерный вектор входного воздействия, $z(t_i)$ — m -мерный вектор измерений, $w(t_i)$ и $v(t_i)$ — независимые нормально распределенные векторы шумов с нулевым средним и известными ковариационными матрицами $Q(t_i)$ и $R(t_i)$ соответственно, причём $Q(t_i) \geq 0$, $R(t_i) > 0$. Начальный вектор состояния системы x_0 распределен по нормальному закону с математическим ожиданием \bar{x}_0 и ковариационной матрицей P_0 .

При заданных матрицах-параметрах системы $\Phi(t_{i+1}, t_i)$, $B(t_i)$, $\Gamma(t_i)$, $H(t_i)$

и начальных условиях \bar{x}_0 , P_0 решение задачи оптимального оценивания вектора состояния $x(t_i)$ системы (13.1), (13.2) дается *фильтром Калмана*. Алгоритм этой рекуррентной обработки данных $\{z(t_i); \forall i \geq 1\}$ определен следующими уравнениями:

$$\left. \begin{aligned} \hat{x}(t_i^-) &= \Phi(t_i, t_{i-1})\hat{x}(t_{i-1}^+) + B(t_{i-1})u(t_{i-1}), \\ P(t_i^-) &= \Phi(t_i, t_{i-1})P(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + \Gamma(t_{i-1})Q(t_i)\Gamma^T(t_{i-1}), \\ K(t_i) &= P(t_i^-)H^T(t_i)[H(t_i)P(t_i^-)H^T(t_i) + R(t_i)]^{-1}, \\ P(t_i^+) &= P(t_i^-) - K(t_i)H(t_i)P(t_i^-), \\ \nu(t_i) &= z(t_i) - H(t_i)\hat{x}(t_i^-), \\ \hat{x}(t_i^+) &= \hat{x}(t_i^-) + K(t_i)\nu(t_i). \end{aligned} \right\} \quad (13.3)$$

ЗАМЕЧАНИЕ 13.1. Уравнения (13.1), (13.2) описывают широкий класс моделей систем, например, систем с управлением. Это могут быть и системы с возможными параметрическими нарушениями, — если уравнения (13.1), (13.2) рассматривать для каждого режима функционирования (с нарушением или нет) отдельно. Это могут быть и адаптивные системы с идентификацией, если матрицы, входящие в (13.1), (13.2), содержат неизвестные параметры, которые подлежат определению в процессе функционирования системы для целей слежения за состоянием системы или для целей управления. В частности, управление $u(t_i)$ может отсутствовать. Его наличие или отсутствие, так же как и наличие или отсутствие нарушений, не влияет на принципы эффективной численной реализации фильтров. Чтобы показать универсальность таких реализаций, управление не исключено из уравнений (13.1), (13.2). Наличие или отсутствие нарушений здесь не отмечено индексом режима возле обозначений матриц исключительно для упрощения записей.

Общность исходной модели (13.1), (13.2) и, соответственно, уравнений фильтра Калмана (13.3), объясняется также следующим.

ЗАМЕЧАНИЕ 13.2. Если $\Phi(t_{i+1}, t_i) = I$, $u(t_i) \equiv 0$ и $w(t_i) \equiv 0$, то модель (13.1), (13.2) совпадает с той моделью, которая принята в теории *регрессионного моделирования*, когда $x(t_i) = \text{const}$ — параметр, подлежащий оцениванию по наблюдениям $z(t_i)$, $i = 1, 2, \dots, N$. Алгоритм (13.3) осуществляет оптимальное оценивание этой регрессионной модели. Здесь принципиальная особенность заключается в том, что регрессионная модель и процесс ее оценивания — последовательные. Полная матрица «регрессоров» $H = [H(t_i); i = 1, 2, \dots, N]^T$ расщеплена на отдельные «порции» $H(t_i)$ соответственно тем «порциям» наблюдений $z(t_i)$, которые поступают в обработку

в последовательные моменты времени $i = 1, 2, \dots, N$. Из теории наименьших квадратов, изложенной выше (см. подразд. 11.6), известно, что для значения текущей оптимальной оценки по экспериментальным данным безразлично, поступили эти данные в обработку целиком или порциями. Последовательная (рекуррентная) форма алгоритма метода наименьших квадратов в форме (13.3) предпочтительнее, чем единовременная обработка всех данных «целиком» — методом решения нормальных уравнений (см. разд. 12), хотя последнее все еще (к сожалению) остается традицией в эконометрике или при обработке данных наблюдений в астрономии или спутниковой геодезии [26].

ЗАМЕЧАНИЕ 13.3. Еще раз отметим важное обобщение, присутствующее в модели (13.1), (13.2) и также в алгоритме фильтра (13.3). В отличие от предыдущих разделов этой книги, здесь источник обрабатываемых данных $\{z(t_i); \forall i \geq 1\}$ — не статический, а динамический. Он содержит информацию не о неизменном по своему значению неизвестном (возможно, случайном) векторе $x(t_i) = \text{const}$, — эта ситуация выделена в Замечании 13.2, — а об изменяющемся векторе $x(t_i) = \text{var}$ состояния динамического объекта (13.1), находящегося (в общем случае) под воздействием детерминистского управления $u(t_i)$ и/или случайного возмущения $w(t_i)$.

13.3 Скаляризованная форма фильтра Калмана

Когда матрица $R(t_i)$ диагональная, $R(t_i) = \text{diag} [r_1(t_i), r_2(t_i), \dots, r_m(t_i)]$, возможно произвольное расщепление системы наблюдений (вектора $z(t_i)$) на априорную и текущую части, — см. выше подразд. 10.8, 11.6 или [15]. Тогда предпочтительно вместо (13.3) использовать *фильтр Калмана со скалярной обработкой измерений*. Скаляризуя обработку измерений в (13.3), получаем:

I. Экстраполяция:

$$\left. \begin{aligned} \hat{x}(t_i^-) &= \Phi(t_i, t_{i-1})\hat{x}(t_{i-1}^+) + B(t_{i-1})u(t_{i-1}), \\ P(t_i^-) &= \Phi(t_i, t_{i-1})P(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + \Gamma(t_{i-1})Q(t_{i-1})\Gamma^T(t_{i-1}), \\ \hat{x}(t_0^+) &= \bar{x}_0, \quad P(t_0^+) = P_0. \end{aligned} \right\} \quad (13.4)$$

II. Обработка измерения:

A. Начальное присваивание: $\tilde{P} = P(t_i^-)$, $\tilde{x} = \hat{x}(t_i^-)$.

Б. m -кратное повторение процедуры скалярного обновления.

Для $j = 1, 2, \dots, m$ выполнять:

$$\alpha := h^T \tilde{P} h + r_j(t_i), \quad K := \tilde{P} h / \alpha, \quad \hat{P} := \tilde{P} - K h^T \tilde{P}, \quad (13.5)$$

$$\hat{x} := \tilde{x} + K(z - h^T \tilde{x}) \quad (13.6)$$

с экстраполяцией между повторениями: $\tilde{P} := \hat{P}$, $\tilde{x} := \hat{x}$.

В. Завершающее присваивание: $P(t_i^+) := \hat{P}$, $\hat{x}(t_i^+) := \hat{x}$,

где h — j -й столбец матрицы $H^T(t_i)$; z — j -й элемент вектора $z(t_i)$, $j = 1, 2, \dots, m$.

13.4 Стабилизированный фильтр Калмана–Джозефа

Этап экстраполяции *стабилизированного алгоритма* совпадает с этапом экстраполяции стандартного алгоритма Калмана. Поэтому приведем лишь этап обработки измерения. Джозеф [15] предложил для алгоритма (13.3) использовать общую формулу, справедливую для матрицы \hat{P} при любом, не обязательно оптимальном K : $\hat{P} = (I - KH)\tilde{P}(I - KH)^T + K R K^T$. (Смысл применяемых здесь обозначений матриц виден из подразд. 13.3.)

ЗАМЕЧАНИЕ 13.4. При оптимальном $K_{\text{opt}} = \tilde{P} H^T (H \tilde{P} H^T + R)^{-1}$ эта формула превращается в $\hat{P} = \tilde{P} - K_{\text{opt}} H \tilde{P}$. При подстановке сюда формулы для K_{opt} получаем *дискретное алгебраическое уравнение Риккати* (Discrete Algebraic Riccati Equation, DARE) [109]. Поэтому этот алгоритм может также рассматриваться как процесс решения DARE.

При таких вычислениях результирующая матрица сохраняет симметричность; кроме того, исчезает опасная для потери положительной определенности операция вычитания матриц, однако, сложность вычислений возрастает почти вдвое. Второй этап приобретает вид:

II. Обработка измерения:

А. Начальное присваивание: $\tilde{P} = P(t_i^-)$, $\tilde{x} = \hat{x}(t_i^-)$.

Б. m -кратное повторение процедуры скалярного обновления.

Для $j = 1, 2, \dots, m$ выполнять:

$$\begin{aligned} \alpha &:= h^T \tilde{P} h + r_j(t_i), \quad v := \tilde{P} h, \quad K := v / \alpha, \\ \hat{P} &:= \tilde{P} - K v^T, \quad v := \hat{P} h, \quad \hat{P} := \hat{P} - v K^T + K R K^T, \\ \hat{x} &:= \tilde{x} + K(z - h^T \tilde{x}) \end{aligned}$$

с экстраполяцией между повторениями: $\tilde{P} := \hat{P}$, $\tilde{x} := \hat{x}$.

- В. Завершающее присваивание: $P(t_i^+) := \hat{P}$, $\hat{x}(t_i^+) := \hat{x}$,
 где h — j -й столбец матрицы $H^T(t_i)$ z — j -й элемент вектора $z(t_i)$,
 $j = 1, 2, \dots, m$.

13.5 Квадратно-корневой фильтр Поттера

Как отмечалось в подразд. 11.8, здесь вместо матриц $P(t_i^\pm)$, по своей природе положительно определенных, оперируют с их квадратными корнями $S(t_i^\pm)$, отвечающими равенствам $S(t_i^\pm)S^T(t_i^\pm) = P(t_i^\pm)$. Ясно, что эти соотношения не дают однозначного определения квадратных корней. Однако это обстоятельство в общем случае не вызывает беспокойства, поскольку эти корни однозначно определяются по разложению Холецкого (например, LL^T -разложение или другие, см. разд. 6).

Основная идея метода фильтрации с использованием квадратного корня состоит в замене уравнений алгоритма Калмана на аналогичные, предназначенные для последовательного расчета $S(t_i^\pm)$. Такой подход оправдывается тем, что произведение $S(t_i^\pm)S^T(t_i^\pm)$ не теряет свойство положительной определенности (при условии полноты ранга) даже с учетом ошибок округления, тогда как ошибки округления могут приводить к потере этого свойства для матрицы $P(t^+)$, если она вычисляется по стандартному алгоритму (13.3).

На этапе экстраполяции, с учетом разложения матрицы $P(t_i^-)$ и также $Q(t_{i-1}) = Q^{1/2}(t_{i-1})Q^{T/2}(t_{i-1})$, уравнение для нее принимает следующий вид:

$$\begin{aligned} S(t_i^-)S^T(t_i^-) &= \Phi(t_i, t_{i-1})S(t_{i-1}^+)S^T(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + \\ &+ \Gamma(t_{i-1})Q^{1/2}(t_{i-1})Q^{T/2}(t_{i-1})\Gamma^T(t_{i-1}). \end{aligned}$$

Непосредственно вычислять матрицу $S(t_i^-)$ можно путем построения ортогональной матрицы T размера $(n + s) \times (n + s)$, такой что

$$\begin{bmatrix} S^T(t_i^-) \\ 0 \end{bmatrix} = T \begin{bmatrix} S^T(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) \\ Q^{T/2}(t_{i-1})\Gamma^T(t_{i-1}) \end{bmatrix}.$$

Это обеспечивает любой алгоритм ортогональных преобразований. Хороший результат дает [80] модифицированный алгоритм Грама–Шмидта. В результате после этапа экстраполяции матрица $S(t_i^-)$ всегда будет получаться треугольной.

На этапе обработки измерения требуем иметь уравнение $\hat{P} := \tilde{P} - Kh\tilde{P}$ в виде $\hat{S}\hat{S}^T := \tilde{S}(I_n - f f^T/\alpha)\tilde{S}^T$. Определим число β так, чтобы обеспечить тождественность: $I_n - f f^T/\alpha = (I_n - \beta f f^T)(I_n - \beta f f^T)$.

Из получающегося отсюда квадратного уравнения, с учетом диагональности $R(t_i) = \text{diag} [r_1(t_i), r_2(t_i), \dots, r_m(t_i)]$ и равенства $\alpha = f^T f + r_j(t_i)$, выбираем одно решение

$$\beta = (1/\alpha)/(1 + \sqrt{1/\alpha}),$$

защищенное от операции вычитания положительной величины $\sqrt{1/\alpha}$ в знаменателе.

С учетом вышеизложенного получим *алгоритм фильтра Поттера*:

I. Экстраполяция:

$$\hat{x}(t_i^-) = \Phi(t_i, t_{i+1})\hat{x}(t_{i-1}^+); \quad \hat{x}(t_0^+) = \bar{x}_0;$$

$$\begin{bmatrix} S^T(t_i^-) \\ 0 \end{bmatrix} = T \begin{bmatrix} S^T(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) \\ Q^{T/2}(t_{i-1})\Gamma^T(t_{i-1}) \end{bmatrix}.$$

II. Обработка измерения:

A. Начальное присваивание: $\tilde{S} = S(t_i^-)$; $\tilde{x} = \hat{x}(t_i^-)$.

B. m -кратное повторение процедуры скалярного обновления.
Для $j = 1, 2, \dots, m$ выполнять:

$$\begin{aligned} f &:= \tilde{S}^T h; \quad \alpha := f^T f + r_j(t_i); \quad \gamma := 1/(1 + \sqrt{1/\alpha}); \\ K &:= \tilde{S} f / \alpha; \quad \hat{S} := \tilde{S} - \gamma K f^T; \\ \hat{x} &:= \tilde{x} + K(z - h^T \tilde{x}) \end{aligned}$$

с экстраполяцией между повторениями $\tilde{S} := \hat{S}$; $\tilde{x} := \hat{x}$.

B. Завершающее присваивание: $S(t_i^+) := \hat{S}$; $\hat{x}(t_i^+) := \hat{x}$,
где h — j -й столбец матрицы $H^T(t_i)$; z — j -й элемент вектора $z(t_i)$,
 $j = 1, 2, \dots, m$.

Теперь вычисление S равносильно счету с двойной точностью P в стандартном алгоритме Калмана. Кроме того, устранена опасность утраты матрицей $\hat{P} = \hat{S}\hat{S}^T$ свойства положительной определенности, что влекло бы расходимость оценок вектора состояния. Недостатком данного алгоритма является для каждого момента t_i m -кратное наличие операции извлечения квадратного корня $\sqrt{1/\alpha}$.

13.6 Одноранговое обновление ПО-матриц

Рассмотрим теорему об *одноранговом обновлении* положительно определенной матрицы [15], которую доказали Agee и Turner (1972) для версии UDU^T разложения ПО-матрицы. Переформулируем ее для варианта LDL^T разложения.

ТЕОРЕМА 13.1 (Agee–Turner, 1972). Пусть матрица L — нижняя треугольная с единичной диагональю, матрица $\tilde{D} = \text{diag} [\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_n]$ — диагональная и $P = L\tilde{D}L^T$. Пусть также заданы некоторый скаляр c и вектор–столбец $a = [a_1, a_2, \dots, a_n]^T$, такие что $0 < \bar{P} = P + caa^T$. Тогда разложение $\bar{P} = \bar{L}\bar{D}\bar{L}^T$, где \bar{L} — нижняя треугольная с единичной диагональю матрица и $\bar{D} = \text{diag} [\bar{d}_1, \bar{d}_2, \dots, \bar{d}_n]$, существует и дается следующим алгоритмом:

А. Начальное присваивание: $c_1 = c$.

Б. Для $i = 1, 2, \dots, n - 1$ выполнять:

$$1) \bar{d}_i = \tilde{d}_i + c_i a_i^2;$$

2) Для $k = i + 1, i + 2, \dots, n$ выполнять:

$$\text{i. } a_k := a_k - a_i l_{ki};$$

$$\text{ii. } \bar{l}_{ki} = l_{ki} + c_i a_i a_k / \bar{d}_i; \quad \star \text{ В матрицах } L \text{ нетривиальные элементы находятся только ниже диагонали.}$$

$$3) c_{i+1} = c_i \tilde{d}_i / \bar{d}_i.$$

В. Завершающее присваивание: $\bar{d}_n = \tilde{d}_n + c_n a_n^2$.

ДОКАЗАТЕЛЬСТВО. Запишем квадратичную форму $0 < x^T \bar{P} x$ в виде суммы полных квадратов с подстановкой в нее левой и правой частей равенства $\bar{P} = P + caa^T$. Продолжим доказательство в виде несложного упражнения, последовательно выделяя эти полные квадраты с правой стороны полученного равенства квадратичных форм (см. [78]). Проделайте это самостоятельно в виде упражнения. \square

УПРАЖНЕНИЕ 13.1. Аналогично предыдущему, докажите следующую версию Теоремы 13.1.

ТЕОРЕМА 13.2. Пусть \tilde{L} — нижняя треугольная матрица и $\tilde{P} = \tilde{L}\tilde{L}^T$. Скаляр c и вектор–столбец $a = [a_1, a_2, \dots, a_n]^T$ такие что $0 < \bar{P} = \tilde{P} + caa^T$. Тогда разложение $\bar{P} = \bar{L}\bar{L}^T$ существует и дается следующим алгоритмом:

А. Начальное присваивание: $c_1 = c$.

Б. Для $i = 1, 2, \dots, n$ выполнять:

$$1) \bar{l}_{ii} = \sqrt{\tilde{l}_{ii}^2 + c_i a_i^2};$$

2) Для $k = i + 1, i + 2, \dots, n$ выполнять:

$$\text{i. } a_k := a_k - a_i \left(\tilde{l}_{ki} / \tilde{l}_{ii} \right);$$

$$\text{ii. } \bar{l}_{ki} = \left(\tilde{l}_{ki} / \tilde{l}_{ii} \right) \bar{l}_{ii} + \left(c_i a_i / \bar{l}_{ii} \right) a_k;$$

$$3) c_{i+1} = \left(c_i \tilde{l}_{ii}^2 \right) / \bar{l}_{ii}^2.$$

□

УПРАЖНЕНИЕ 13.2.

$$\tilde{L} = \begin{bmatrix} 2 & 0 \\ -1 & 3 \end{bmatrix}, \quad a = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad c = -1.$$

Найдите $\bar{P} = \tilde{L}\tilde{L}^T + caa^T = \begin{bmatrix} 3 & 0 \\ 0 & 6 \end{bmatrix}$, $\bar{L} = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{6} \end{bmatrix}$. Алгоритм теоремы 13.2 дает этот же результат. Проверьте. Измените исходные данные, например, возьмите $c = 1$. Проверьте, что алгоритм дает правильный результат: $\bar{L} = \begin{bmatrix} \sqrt{5} & 0 \\ -4/\sqrt{5} & 3\sqrt{6/5} \end{bmatrix}$.

УПРАЖНЕНИЕ 13.3. Сформулируйте и докажите еще две версии теоремы об одноранговом обновлении подобно двум предыдущим теоремам, опираясь на другие разложения Холецкого: $P = UU^T$ и $P = UDU^T$ [15].

13.7 Факторизованный фильтр Бирмана

Основная идея этого алгоритма состоит в разложении ковариационной матрицы P в произведение двух треугольных матриц и диагональной матрицы между ними. Можно рассматривать два варианта алгоритма: LD -алгоритм или UD -алгоритм Бирмана. Последний изложен в [15]. Здесь рассмотрим LD -алгоритм, т. е. используем разложение $P(t_i^\pm) = L(t_i^\pm)D(t_i^\pm)L^T(t_i^\pm)$, где $L(t_i^\pm)$ — нижние треугольные матрицы с единичной диагональю, $D(t_i^\pm)$ — диагональные матрицы.

На этапе экстраполяции с учетом разложения матрицы $Q(t_{i-1}) = L_q(t_{i-1})D_q(t_{i-1})L_q^T(t_{i-1})$ и аналогичного разложения матрицы $P(t_i^-)$ уравнение для $P(t_i^-)$ принимает следующий вид:

$$P(t_i^-) = L(t_i^-)D(t_i^-)L^T(t_i^-) = \Phi(t_i, t_{i-1})L(t_{i-1}^+)D(t_{i-1}^+)L^T(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + \Gamma(t_{i-1})L_q(t_{i-1})D_q(t_{i-1})L_q^T(t_{i-1})\Gamma^T(t_{i-1}).$$

Представим матрицу $P(t_i^-)$ в следующем виде:

$$P(t_i^-) = \begin{bmatrix} \Phi(t_i, t_{i-1})L(t_{i-1}^+) & \Gamma(t_{i-1})L_q(t_{i-1}) \end{bmatrix} \times \\ \times \text{Diag}[D(t_{i-1}^+), D_q(t_{i-1})] \begin{bmatrix} L^T(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) \\ L_q^T(t_{i-1})\Gamma^T(t_{i-1}) \end{bmatrix},$$

т. е. $P(t_i^-) = W(t_i)D(t_i)W^T(t_i)$,

$$W(t_i) = \begin{bmatrix} \Phi(t_i, t_{i-1})L(t_{i-1}^+) & \Gamma(t_{i-1})L_q(t_{i-1}) \end{bmatrix} = \begin{bmatrix} w_1^T(t_i) \\ \vdots \\ w_n^T(t_i) \end{bmatrix},$$

$$D(t_i) = \text{Diag}[D(t_{i-1}^+), D_q(t_{i-1})] = \text{Diag}[D_1(t_i), \dots, D_N(t_i)],$$

где $W(t_i)$ – матрица размера $(n \times (n+s))$ и $N = n+s$, s – размерность шума $w(t_i)$ в уравнении состояния (13.1). Факторы $L(t_{i-1}^-)$ и $D(t_{i-1}^-)$ вычисляются методом модифицированного взвешенного алгоритма Грама–Шмидта [15] на этапе экстраполяции.

Перефразируя его для нижних треугольных факторов, запишем следующий результат (для удобства изложения будем писать \tilde{L} , \hat{L} , \tilde{D} , \hat{D} , D , W вместо $L(t_i^-)$, $L(t_i^+)$, $D(t_i^-)$, $D(t_i^+)$, $D(t_i)$, $W(t_i)$ соответственно):

ТЕОРЕМА 13.3 (*Взвешенная ортогонализация Грама–Шмидта и факторизация матрицы – аналог теоремы VI.4.1 из [15]*). Пусть векторы w_1, \dots, w_n образуют линейно независимую систему, $w_i \in \mathbb{R}^N$, $N \geq n$ и $D = \text{Diag}[D_1, \dots, D_n] > 0$. Применим следующий алгоритм (нижний индекс указывает номер нетривиального элемента матрицы / вектора):

А. Начальное присваивание: $v_k^{(1)} = w_k$, $k = 1, \dots, n$.

Б. Для $j = 2, \dots, n$ выполнять:

$$\begin{aligned} \tilde{d}_{j-1} &= (v_{j-1}^{(j-1)})^T D v_{j-1}^{(j-1)}; \\ \tilde{l}_{k,j-1} &= (v_k^{(j-1)})^T D v_{j-1}^{(j-1)} / \tilde{d}_{j-1}, \quad k = j, \dots, n; \\ v_k^{(j)} &= v_k^{(j-1)} - \tilde{l}_{k,j-1} v_{j-1}^{(j-1)}, \quad k = j, \dots, n. \end{aligned}$$

В. Завершающее присваивание: $\tilde{d}_n = (v_n^{(n)})^T D v_n^{(n)}$.

Тогда $v_j^{(j)} = v_j$, $j = 1, \dots, n$, где v_j — взвешенные взаимно ортогональные векторы: $v_i^T D v_j = \tilde{d}_j \delta_{i,j}$ ($\delta_{i,j}$ — символ Кронекера) и

$$\begin{bmatrix} w_1^T \\ \dots \\ w_n^T \end{bmatrix} D \begin{bmatrix} w_1 & \dots & w_n \end{bmatrix} = \tilde{L} \begin{bmatrix} v_1^T \\ \dots \\ v_n^T \end{bmatrix} D \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix} \tilde{L}^T = \tilde{L} \tilde{D} \tilde{L}^T.$$

Из этого алгоритма имеем результат: $L(t_i^-) = \tilde{L}$, $D(t_i^-) = \tilde{D}$.

Этап обработки измерений представим в виде теоремы, формулируя ее для нижних треугольных сомножителей (LD -версия) [78] вместо UD -версии теоремы Бирмана из [15].

ТЕОРЕМА 13.4 (*Алгоритм Бирмана*). Пусть калмановская процедура скалярного обновления (13.5), (13.6) использует разложения $\hat{P} = \hat{L} \hat{D} \hat{L}^T$ и $\tilde{P} = \tilde{L} \tilde{D} \tilde{L}^T$, где L — нижние треугольные с единичной диагональю матрицы, D — диагональные (с положительными элементами) матрицы. Тогда данная процедура (13.5), (13.6) эквивалентна пунктам **В**, **Г**, **Д** и **Е**, следующего алгоритма.

II. Обработка измерения:

А. Начальное присваивание: $\tilde{L} = L(t_i^-)$, $\tilde{D} = D(t_i^-)$, $\tilde{x} = x(t_i^-)$.

Б. m -кратное повторение процедуры скалярного обновления.

Для $j = 1, 2, \dots, m$ выполнять:

В. Вычислить векторы $f = [f_1, f_2, \dots, f_n]^T = \tilde{L}^T h$;

$$v = [v_1, v_2, \dots, v_n]^T = \tilde{D} f.$$

Г. Задать начальные значения $\alpha' = r$; $K = [0 \dots 0 \mid v_n]^T$.

Д. Для $i = n, n-1, \dots, 2, 1$ выполнять:

начало

$$\alpha := \alpha' + v_i f_i; \quad \gamma := 1/\alpha;$$

$$\hat{d}_i := \tilde{d}_i \alpha' \gamma; \quad \lambda := -f_i \gamma;$$

$$\hat{l}_i := \tilde{l}_i + \lambda K; \quad K := K + \tilde{l}_i v_i; \quad (\star)$$

$$\alpha' := \alpha.$$

конец

Е. Вычислить векторы $\nu := \gamma(z - h^T \tilde{x})$; $\hat{x} := \tilde{x} + K \nu$

с экстраполяцией между повторениями:

$$\tilde{L} := \hat{L}; \quad \tilde{D} := \hat{D}; \quad \tilde{x} := \hat{x}.$$

Ж. Завершающее присваивание:

$$L(t_i^+) := \widehat{L}; \quad D(t_i^+) := \widehat{D}; \quad \widehat{x}(t_i^+) := \widehat{x}.$$

Здесь h — j -й столбец матрицы $H^T(t_i)$; z — j -й элемент вектора $z(t_i)$; r — j -й элемент $r_j(t_i)$ диагональной матрицы ковариаций шума измерений $R(t_i)$, $j = 1, 2, \dots, m$ — номер скалярного измерения в составе вектора измерений $z(t_i)$ в момент t_i .

ЗАМЕЧАНИЕ 13.5. Строка \star в алгоритме на стр. 281 выглядит так:

Для $k = i + 1$ до n выполнять
 начало
 $\widehat{l}_{ki} := \widetilde{l}_{ki} + \lambda K_k, \quad K_k := K_k + \widetilde{l}_{ki} v_i$
 конец

Она пропускается при $i = n$. Здесь K_k есть k -й элемент того вектора K , который существует в цикле \mathbf{D} алгоритма на стр. 281.

ДОКАЗАТЕЛЬСТВО. Применим для (13.5) факторизацию Холецкого вида $\widehat{P} = \widehat{L}\widehat{D}\widehat{L}^T$ и $\widetilde{P} = \widetilde{L}\widetilde{D}\widetilde{L}^T$. Получим

$$\widehat{L}\widehat{D}\widehat{L}^T = \widetilde{L}(\widetilde{D} + cvv^T)\widetilde{L}^T$$

с обозначениями

$$c = -1/\alpha, \quad f = \widetilde{L}^T h, \quad v = \widetilde{D} f.$$

Применяя теорему 13.1 к выражению $\widetilde{D} + cvv^T$, в ней следует считать, что: $P \equiv D$, $c \equiv -1/\alpha$ и $a \equiv v$. Отсюда получаем $\widetilde{L}\widetilde{D}\widetilde{L}^T = \widetilde{L}\widetilde{D}\widetilde{L}^T + cvv^T$ при том, что в этом выражении имеем $L = I$. Когда \bar{L} и \bar{D} будут найдены, мы получим

$$\widehat{L}\widehat{D}\widehat{L}^T = \widetilde{L}\bar{L}\bar{D}\bar{L}^T\widetilde{L}^T \implies \widehat{D} = \bar{D}, \quad \widehat{L} = \widetilde{L}\bar{L}. \quad (13.7)$$

Запишем теорему 13.1 применительно к выражению $\widetilde{D} + cvv^T$. Получим:

А. Начальное присваивание: $c_1 = c$.

Б. Для $i = 1, 2, \dots, n - 1$ выполнять:

1) $\widehat{d}_i = \widetilde{d}_i + c_i v_i^2$. ①

2) Для $k = i + 1, i + 2, \dots, n$ выполнять:

i. $v_k := v_k - v_i l_{ki} = v_k$. ○ Здесь $L = I$, т. е. $l_{ki} = 0$. Это действие исчезает.

ii. $\bar{l}_{ki} = l_{ki} + (c_i v_i / \widehat{d}_i) v_k = (c_i v_i / \widehat{d}_i) v_k$. ②

$$3) \ c_{i+1} = c_i \tilde{d}_i / \hat{d}_i. \quad \textcircled{3}$$

$$\text{В. Завершающее присваивание: } \hat{d}_n = \tilde{d}_n + c_n v_n^2. \quad \textcircled{4}$$

Этот алгоритм теоретически верен, но численно неустойчив из-за того, что должно выполняться условие: $\forall i : c_i < 0$. В силу этого величины \hat{d}_i в $\textcircled{1}$ могут стать отрицательны, что противоречит требованию положительной определенности $\hat{P} > 0$. Устраним этот недостаток, эквивалентно преобразуя алгоритм на стр. 282–283 по пунктам $\textcircled{1}$, $\textcircled{2}$, $\textcircled{3}$, $\textcircled{4}$. Преобразуем $\textcircled{1}$: $\hat{d}_i/c_i = \tilde{d}_i/c_i + v_i^2$. С учетом $\textcircled{3}$: $\tilde{d}_i/c_{i+1} = \tilde{d}_i/c_i$, поэтому $\tilde{d}_i/c_{i+1} = \tilde{d}_i/c_i + v_i^2$, следовательно, $1/c_{i+1} = 1/c_i + v_i^2/\tilde{d}_i$. Однако $v_i = \tilde{d}_i f_i$. Поэтому $-1/c_i = -1/c_{i+1} + v_i f_i$. Введем обозначение: $\forall i : \alpha_i = -1/c_i$. Это позволяет записать:

$$\alpha_i = \alpha_{i+1} + v_i f_i, \quad i = n-1, n-2, \dots, 2, 1. \quad (13.8)$$

Финальное значение, т. е. α_1 , согласно введенному обозначению, должно совпасть с $\alpha = r + v^T f$, где $r = r(t_j)$, так как $c_1 = c$ и $c = -1/\alpha$. Чтобы это произошло, в алгоритме необходимо стартовать от значения $\alpha_n = r + v_n f_n$. Таким образом, из $\textcircled{1}$ получили алгоритм (13.8), заменяющий $\textcircled{3}$.

Теперь выведем из $\textcircled{3}$ алгоритм, заменяющий $\textcircled{1}$:

$$\hat{d}_i = \tilde{d}_i (c_i/c_{i+1}) = \tilde{d}_i (\alpha_{i+1}/\alpha_i), \quad i = n-1, n-2, \dots, 2, 1.$$

Осталось преобразовать $\textcircled{2}$. Из $\textcircled{2}$ имеем $\bar{l}_{ki} = \lambda_i v_k$, $\lambda_i = (c_i v_i / \hat{d}_i)$, а из $\textcircled{3}$ $c_i / \hat{d}_i = c_{i+1} / \tilde{d}_i$, то есть $\lambda_i = -f_i / \alpha_{i+1}$, так как $v_i = \tilde{d}_i f_i$. Следовательно, алгоритм для \bar{l}_{ki} вместо $\textcircled{2}$ приобретает вид:

Для $i = n-1, n-2, \dots, 2, 1$ вычислять
 $\lambda_i = -f_i / \alpha_{i+1}$.
 Для $k = i+1$ до n вычислять
 $\bar{l}_{ki} = \lambda_i v_k$.

Наконец, преобразуем последнее действие $\textcircled{4}$ на стр. 283. Последовательно находим:

$$\begin{aligned} \hat{d}_n/c_n &= \tilde{d}_n/c_n + v_n^2, & (\hat{d}_n/\tilde{d}_n)(1/c_n) &= (1/c_n) + (v_n^2/\tilde{d}_n), \\ (v_n^2/\tilde{d}_n) &= (\tilde{d}_n f_n)^2/\tilde{d}_n = v_n f_n. \end{aligned}$$

Из обозначения $(-1/c_{i+1}) = \alpha_{i+1}$ при $i = n-1$ следует $(-1/c_n) = \alpha_n$, т. е. $\alpha_n = -(\hat{d}_n/\tilde{d}_n)(1/c_n) + v_n f_n$. С другой стороны, известно стартовое значение $\alpha_n = r + v_n f_n$, следовательно, $\hat{d}_n = \tilde{d}_n r / \alpha_n$. Запишем результирующий эквивалентный алгоритм:

Задать начальные значения: $\alpha_n = r + v_n f_n$, $\widehat{d}_n = \widetilde{d}_n r / \alpha_n$.
 Для $i = n - 1, n - 2, \dots, 2, 1$ вычислять
 $\alpha_i = \alpha_{i+1} + v_i f_i$;
 $\widehat{d}_i = \widetilde{d}_i (\alpha_{i+1} / \alpha_i)$;
 $\lambda_i = -f_i / \alpha_{i+1}$.
 Для $k = i + 1$ до n вычислять
 $\bar{l}_{ki} = \lambda_i v_k$.

Отмеченный выше недостаток устранен: теперь $\forall i : \widehat{d}_i > 0$. Проведенное эквивалентное преобразование привело к инверсии направления вычислений. В исходном массиве для нетривиальных элементов матриц \widetilde{L} и \widetilde{D} исходные столбцы сначала имеют (начиная с диагонали вниз) вид:

$$\begin{bmatrix} \widetilde{d}_1 \\ \widetilde{l}_1 \end{bmatrix}, \quad \begin{bmatrix} \widetilde{d}_2 \\ \widetilde{l}_2 \end{bmatrix}, \quad \dots, \quad \begin{bmatrix} \widetilde{d}_{n-1} \\ \widetilde{l}_{n-1} \end{bmatrix}, \quad \begin{bmatrix} \widetilde{d}_n \end{bmatrix}.$$

После выполнения этого алгоритма они имеют следующее наполнение:

$$\begin{bmatrix} \widehat{d}_1 \\ \bar{l}_1 \end{bmatrix}, \quad \begin{bmatrix} \widehat{d}_2 \\ \bar{l}_2 \end{bmatrix}, \quad \dots, \quad \begin{bmatrix} \widehat{d}_{n-1} \\ \bar{l}_{n-1} \end{bmatrix}, \quad \begin{bmatrix} \widehat{d}_n \end{bmatrix}.$$

Вернемся к последней формуле в (13.7), чтобы в согласии с ней перейти от «промежуточной» матрицы \bar{L} к «окончательной» матрице \widehat{L} .

Матрица \bar{L} — нижняя треугольная с единицами на главной диагонали. В ней поддиагональная часть i -го столбца, согласно нижней строке алгоритма на стр. 284, найдена в виде $\bar{l}_i^T = \lambda_i [v_{i+1}, v_{i+2}, \dots, v_n]$, где использованы элементы из вектора $v^T = [v_1, v_2, \dots, v_n]$, начиная от элемента v_{i+1} и далее. Введем вспомогательные обозначения n -мерных векторов:

$$v^{(n)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ v_n \end{bmatrix}, \quad v^{(n-1)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ v_{n-1} \\ v_n \end{bmatrix}, \quad \dots, \quad v^{(2)} = \begin{bmatrix} 0 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{bmatrix}, \quad v^{(1)} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{bmatrix} = v.$$

Теперь видно, что

$$\bar{L} = I + \left[\lambda_1 v^{(2)} \mid \lambda_2 v^{(3)} \mid \dots \mid \lambda_{n-1} v^{(n)} \mid \mathbf{0} \right].$$

Отсюда

$$\widehat{L} = \widetilde{L}\bar{L} = \widetilde{L} + \left[\lambda_1 \widetilde{L}v^{(2)} \mid \lambda_2 \widetilde{L}v^{(3)} \mid \cdots \mid \lambda_{n-1} \widetilde{L}v^{(n)} \mid \mathbf{0} \right].$$

Введем обозначения возникающих здесь матриц-столбцов:

$$K_2 = \widetilde{L}v^{(2)}, \quad K_3 = \widetilde{L}v^{(3)}, \quad \dots, \quad K_n = \widetilde{L}v^{(n)}.$$

В качестве начального столбца имеем $K_n = \widetilde{L}v^{(n)}$. Далее

$$\begin{aligned} K_{n-1} &= \widetilde{L}v^{(n-1)} = K_n + \widetilde{l}_{n-1}v_{n-1}, \\ K_{n-2} &= \widetilde{L}v^{(n-2)} = K_{n-1} + \widetilde{l}_{n-2}v_{n-2}, \\ &\quad \dots, \\ K_i &= K_{i+1} + \widetilde{l}_i v_i, \quad i = n-1, n-2, \dots, 2, 1, \end{aligned}$$

и в конце получаем

$$K_1 = \widetilde{L}v^{(1)} = \widetilde{L}v = \bar{K},$$

\bar{K} — вспомогательное обозначение, причем здесь для строгости записей надо понимать, что $\widetilde{l}_i, i = n-1, n-2, \dots, 2, 1$, обозначает весь j -й столбец матрицы \widetilde{L} — вместе с тривиальными элементами 0 и 1. \square

Данный в теореме 13.4 алгоритм не содержит операции извлечения квадратного корня, а работа с треугольными матрицами требует меньшего числа арифметических операций по сравнению с обычными.

13.8 Квадратно-корневой фильтр Карлсона

Этап экстраполяции здесь в точности совпадает с этим этапом в алгоритме Поттера. Выведем только алгоритм этапа обработки измерения.

УПРАЖНЕНИЕ 13.4. Выведите *алгоритм Карлсона обработки измерения*. Вывод можно сделать в полном соответствии с тем, как выше была доказана теорема 13.4 (см. стр. 281). Другой способ вывода — как следствие теоремы 13.4: за \widetilde{L} принять формально то, что там было $\widetilde{L}\widetilde{D}^{1/2}$. Первый способ предпочтительнее для понимания доказательства, второй полезен для освоения техники вычислений. Получите следующий результат.

ТЕОРЕМА 13.5. Пусть калмановская процедура скалярного обновления (13.5), (13.6) использует разложения $\widehat{P} = \widehat{L}\widehat{L}^T$ и $\widetilde{P} = \widetilde{L}\widetilde{L}^T$, где \widehat{L} и \widetilde{L} — нижние треугольные матрицы. Тогда данная процедура (13.5), (13.6) эквивалентна пунктам **В**, **Г**, **Д** и **Е**, следующего алгоритма.

II. Обработка измерения:

А. Начальное присваивание: $\tilde{L} = L(t_i^-)$, $\tilde{x} = x(t_i^-)$.

Б. m -кратное повторение процедуры скалярного обновления.

Для $j = 1, 2, \dots, m$ выполнять:

В. Вычислить векторы $f = [f_1, f_2, \dots, f_n]^T = \tilde{L}^T h$.

Г. Задать начальные значения: $\alpha' = r$; $K = [0 \dots 0 \mid \tilde{l}_{nn} f_n]^T$.

Д. Для $i = n, n-1, \dots, 2, 1$ выполнять:

начало

$$\begin{aligned} \alpha &:= \alpha' + f_i^2; & \beta &:= \sqrt{\alpha'/\alpha}; \\ \hat{l}_{ii} &:= \beta \tilde{l}_{ii}; & \lambda &:= -f_i/(\alpha\beta); \\ \hat{l}_i &:= \beta \tilde{l}_i + \lambda K; & K &:= K + \tilde{l}_i f_i; \\ \alpha' &:= \alpha. \end{aligned} \quad (\star)$$

конец

Е. Вычислить векторы $\nu := (z - h^T \tilde{x})/\alpha$; $\hat{x} := \tilde{x} + K\nu$

с экстраполяцией между повторениями:

$$\tilde{L} := \hat{L}; \quad \tilde{x} := \hat{x}.$$

Ж. Завершающее присваивание:

$$L(t_i^+) := \hat{L}; \quad \hat{x}(t_i^+) := \hat{x}.$$

Здесь h — j -й столбец матрицы $H^T(t_i)$; z — j -й элемент вектора $z(t_i)$; r — j -й элемент $r_j(t_i)$ диагональной матрицы ковариаций шума измерений $R(t_i)$, $j = 1, 2, \dots, m$ — номер скалярного измерения в составе вектора измерений $z(t_i)$ в момент t_i .

ЗАМЕЧАНИЕ 13.6. Строка \star в алгоритме на стр. 286 выглядит так:

Для $k = i + 1$ до n выполнять

начало

$$\hat{l}_{ki} := \beta \tilde{l}_{ki} + \lambda K_k, \quad K_k := K_k + \tilde{l}_{ki} f_i$$

конец

Она пропускается при $i = n$. Здесь K_k есть k -й элемент того вектора K , который существует в цикле **Д** алгоритма Карлсона на стр. 286.

13.9 Редуцированный фильтр Бирмана

В приложениях выбор измерительных средств бывает ограничен так, что в измерение z попадает лишь часть (заранее известная) элементов оцениваемого вектора x . Пусть эта часть — первые $q_j < n$ элементов для j -й строки матрицы наблюдений:

$$h_j^T = [\underbrace{\star \star \cdots \star}_{q_j < n, h_{q_j} \neq 0} \mid 0 \cdots 0] = [\star \star \cdots \star \mid 0 \cdots 0]. \quad (13.9)$$

ТЕОРЕМА 13.6 ([78]). Пусть калмановская процедура скалярного обновления (13.5), (13.6) использует разложения $\hat{P} = \hat{L}\hat{D}\hat{L}^T$ и $\tilde{P} = \tilde{L}\tilde{D}\tilde{L}^T$, где L — нижние треугольные с единичной диагональю матрицы, D — диагональные (с положительными элементами) матрицы. Тогда данная процедура (13.5), (13.6) эквивалентна пунктам **В**, **Г**, **Д** и **Е**, следующего алгоритма.

II. Обработка измерения:

А. Начальное присваивание: $\tilde{L} = L(t_i^-)$, $\tilde{D} = D(t_i^-)$, $\tilde{x} = x(t_i^-)$.

Б. m -кратное повторение процедуры скалярного обновления.
Для $j = 1, 2, \dots, m$ выполнять:

В. Вычислить векторы $f = \tilde{L}^T h$; $v = \tilde{D} f$. Их вид:

$$f = [\star \star \cdots \star \mid 0 \cdots 0]^T, \quad v = [\star \star \cdots \star \mid 0 \cdots 0]^T.$$

Г. Задать начальные значения $\alpha' = r$; $K = [0 \cdots 0]^T$.

Д. Для $i = q + 1, q + 2, \dots, n$ выполнять: $K_i := \tilde{l}_{iq} v_q$.

Е. Для $i = q, q - 1, \dots, 2, 1$ выполнять:

начало

$$\alpha := \alpha' + v_i f_i; \quad \hat{d}_i := \tilde{d}_i \alpha' / \alpha;$$

$$\lambda := -f_i / \alpha; \quad K_i := v_i;$$

$$\hat{l}_i := \tilde{l}_i + \lambda K; \quad K := K + \tilde{l}_i v_i; \quad (\star)$$

$$\alpha' := \alpha.$$

конец

Ж. Вычислить векторы $\nu := (z - h^T \tilde{x}) / \alpha$; $\hat{x} := \tilde{x} + K \nu$

с экстраполяцией между повторениями:

$$\tilde{L} := \hat{L}; \quad \tilde{D} := \hat{D}; \quad \tilde{x} := \hat{x}.$$

З. Завершающее присваивание:

$$L(t_i^+) := \hat{L}; \quad D(t_i^+) := \hat{D}; \quad \hat{x}(t_i^+) := \hat{x}.$$

Здесь h^T — j -я строка матрицы $H(t_i)$, имеющая вид (13.9); z — j -й элемент вектора $z(t_i)$; r — j -й элемент $r_j(t_i)$ диагональной матрицы ковариаций шума измерений $R(t_i)$, $j = 1, 2, \dots, m$ — номер скалярного измерения в составе вектора измерений $z(t_i)$ в момент времени t_i .

ЗАМЕЧАНИЕ 13.7. Строка \star в алгоритме на стр. 287 выглядит так же, как в замечании 13.5 на стр. 282.

ЗАМЕЧАНИЕ 13.8. Представление (13.9) мотивирует использование LD -версии фильтра Бирмана по подразд. 13.7. Если же

$$h_j^T = [0 \cdots 0 \mid \star \star \cdots \star],$$

то сокращенный объем вычислений будет получаться при использовании UD -версии фильтра Бирмана [15].

ЗАМЕЧАНИЕ 13.9. Сокращение объема вычислений в редуцированных таким образом версиях получается наибольшим среди алгоритмов этого класса, поскольку значение q берется как q_i — индивидуально для каждого из m повторений цикла **Е** в алгоритме на стр. 287.

ЗАМЕЧАНИЕ 13.10. Массив D может быть общим для \tilde{D} и \hat{D} , и в нем обновляются только первые q элементов. В массивах для матриц \tilde{L} и \hat{L} обновляются только первые $q - 1$ столбцов, а столбцы с q -го по n -й не обновляются и могут быть общими.

13.10 Редуцированный фильтр Бар-Ицхака

Если в исходном фильтра Калмана (13.3) свойство (13.9) рассматривать сразу для всех m строк матрицы наблюдений

$$H(t_i) = H = [H^{mq} \mid \mathbf{0}], \quad (13.10)$$

это может служить поводом для сокращения объема вычислений, как это произошло выше в подразд 13.9. В (13.10) H^{mq} — ненулевая подматрица, где mq указывает ее размер $(m \times q)$, $\mathbf{0}$ — нулевая подматрица, ее размер $(m \times s)$, $s = n - q$.

Свойство (13.10) означает, что оцениваемый вектор распадается на две части: $x = [x^q \mid x^s]$. Часть x^q попадает в вектор измерений $z = z(t_i)$, а часть

x^s — нет. Соответственно этому, каждую P -матрицу, т. е. \hat{P} и \tilde{P} , рассмотрим поблочно как матрицу следующего вида [89]:

$$P = \begin{bmatrix} P^{qq} & (P^{sq})^T \\ P^{sq} & P^{ss} \end{bmatrix}. \quad (13.11)$$

ТЕОРЕМА 13.7 (*Bar-Itzhack, 1980*). Если выполнено условие (13.10), алгоритм (13.3) распадается на независимый редуцированный фильтр размерности q для измеряемых компонент x^q вектора x (аргумент дискретного времени t_i для простоты опущен):

$$\left. \begin{aligned} K^{qm} &= \tilde{P}^{qq} (H^{mq})^T \left[H^{mq} \tilde{P}^{qq} (H^{mq})^T + R \right]^{-1}, \\ \hat{P}^{qq} &= \tilde{P}^{qq} - K^{qm} H^{mq} \tilde{P}^{qq}, \\ \hat{x}^q &= \tilde{x}^q + K^{qm} (z - H^{mq} \tilde{x}^q) \end{aligned} \right\} \quad (13.12)$$

и фильтр размерности $s = n - q$, зависящий от предыдущего фильтра (13.12), для неизмеряемых компонент x^s вектора x :

$$\left. \begin{aligned} K^{sq} &= \tilde{P}^{sq} \left(\tilde{P}^{qq} \right)^{-1}, \\ \hat{P}^{sq} &= K^{sq} \hat{P}^{qq}, \\ \hat{P}^{ss} &= \tilde{P}^{ss} - K^{sq} \left(\tilde{P}^{qq} - \hat{P}^{qq} \right) (K^{sq})^T, \\ \hat{x}^s &= \tilde{x}^s + K^{sq} (\hat{x}^q - \tilde{x}^q). \end{aligned} \right\} \quad (13.13)$$

ДОКАЗАТЕЛЬСТВО. Подстановка выражений (13.10) и (13.11) в уравнения стандартного фильтра Калмана (13.3) после несложных алгебраических преобразований приводит к (13.12) и (13.13). \square

Полученный алгоритм лишь выделяет редуцированный фильтр (13.12), но для него задача LD -факторизации остается актуальной. Ее решает следующий алгоритм (подразд. 13.11).

13.11 Редуцированный фильтр Бар-Ицхака–Медана

Этот результат представлен в 1983 году как новый LD -алгоритм обновления оценок по измерениям. Он является лучшим по сравнению с обычным LD -алгоритмом в случае, когда число элементов вектора состояния, непосредственно попадающих в вектор измерений, меньше, чем размерность вектора состояния. Такая ситуация типична для аэрокосмических приложений.

Так, в инерциальных навигационных системах число состояний может быть более 40, а число измеренных состояний — только 3.

ТЕОРЕМА 13.8 (*Bar-Itzhack–Medan, 1983 [89]*). Пусть при условии (13.10) алгоритм Калмана (13.3) использует разложения $\hat{P} = \hat{L}\hat{D}\hat{L}^T$ и $\tilde{P} = \tilde{L}\tilde{D}\tilde{L}^T$ с обозначениями $\hat{P} = P(t_i^-)$ и $\tilde{P} = P(t_i^+)$, причем \hat{P} и \tilde{P} рассмотрены поблочно, как в (13.11), и \hat{L} , \hat{D} , \tilde{L} , \tilde{D} разложены по типу выражений

$$L = \begin{bmatrix} L^{qq} & 0 \\ L^{sq} & L^{ss} \end{bmatrix}, \quad D = \begin{bmatrix} D^q & 0 \\ 0 & D^{ss} \end{bmatrix}. \quad (13.14)$$

Тогда этот алгоритм эквивалентен следующему алгоритму (излагаемое относится лишь к этапу **II** обработки измерения с матрицей вида (13.10)):

II. Обработка измерения, эквивалентная работе алгоритма (13.12):

А. Начальное присваивание: $\tilde{L} = L^{qq}(t_i^-)$, $\tilde{D} = D^q(t_i^-)$, $\tilde{x} = x^q(t_i^-)$.

Б. m -кратное повторение процедуры скалярного обновления.

Для $j = 1, 2, \dots, m$ выполнять:

В. Вычислить векторы $f = [f_1, f_2, \dots, f_q]^T = \tilde{L}^T h$;

$$v = [v_1, v_2, \dots, v_q]^T = \tilde{D}f.$$

Г. Задать начальные значения $\alpha' = r$; $K = [0 \dots 0 \mid v_q]^T$.

Д. Для $i = q, q-1, \dots, 2, 1$ выполнять:

начало

$$\begin{aligned} \alpha &:= \alpha' + v_i f_i; & \gamma &:= 1/\alpha; \\ \hat{d}_i &:= \tilde{d}_i \alpha' \gamma; & \lambda &:= -f_i \gamma; \\ \tilde{l}_i &:= \tilde{l}_i + \lambda K; & K &:= K + \tilde{l}_i v_i; \\ \alpha' &:= \alpha. \end{aligned} \quad (\star)$$

конец

Е. Вычислить векторы $\nu := \gamma(z - h^T \tilde{x})$; $\hat{x} := \tilde{x} + K\nu$

с экстраполяцией между повторениями:

$$\tilde{L} := \hat{L}; \quad \tilde{D} := \hat{D}; \quad \tilde{x} := \hat{x}.$$

Ж. Завершающее присваивание по п. **II**:

$$\hat{L}^{qq}(t_i^+) := \hat{L}; \quad \hat{D}^q(t_i^+) := \hat{D}; \quad \hat{x}^q(t_i^+) := \hat{x}.$$

Здесь h — j -й столбец подматрицы $(H^{mq})^T(t_i)$ из (13.10); z — j -й элемент вектора $z(t_i)$; r — j -й элемент $r_j(t_i)$ диагональной матрицы

ковариаций шума измерений $R(t_i)$, $j = 1, 2, \dots, m$ — номер скалярного измерения в составе вектора измерений $z(t_i)$ в момент t_i .

III. Обновление оценок для неизмеряемых компонент x^s вектора x , эквивалентное работе алгоритма (13.13):

3. Вычислить:

$$\left. \begin{aligned} K^{sq} &= \tilde{L}^{sq} \left(\tilde{L}^{qq} \right)^{-1}, \\ \hat{L}^{sq} &= K^{sq} \hat{L}^{qq}, \\ \hat{L}^{ss} &= \tilde{L}^{ss}, \quad \hat{D}^s = \tilde{D}^s, \\ \hat{x}^s &= \tilde{x}^s + K^{sq} (\hat{x}^q - \tilde{x}^q). \end{aligned} \right\} \quad (13.15)$$

И. Завершающее присваивание по п. **III**:

$$\hat{L}^{sq}(t_i^+) := \hat{L}^{sq}; \quad \hat{L}^{ss}(t_i^+) := \hat{L}^{ss}; \quad \hat{D}^s(t_i^+) := \hat{D}^s; \quad \hat{x}^s(t_i^+) := \hat{x}^s.$$

ЗАМЕЧАНИЕ 13.11. Строка ★ в алгоритме на стр. 290 выглядит так:

Для $k = i + 1$ до q выполнять
 начало
 $\hat{l}_{ki} := \tilde{l}_{ki} + \lambda K_k, \quad K_k := K_k + \tilde{l}_{ki} v_i$
 конец

Она пропускается при $i = q$. Здесь K_k есть k -й элемент того вектора K , который существует в цикле **Д** алгоритма на стр. 290.

ДОКАЗАТЕЛЬСТВО. Подстановка (13.14) в разложение $P = LDL^T$ дает

$$\left. \begin{aligned} P^{qq} &= L^{qq} D^q (L^{qq})^T, \\ P^{sq} &= L^{sq} D^q (L^{qq})^T, \\ P^{ss} &= L^{sq} D^q (L^{sq})^T + L^{ss} D^s (L^{ss})^T. \end{aligned} \right\} \quad (13.16)$$

Из первого уравнения (13.13) и из первых двух уравнений (13.16), учитывая добавление верхней тильды $\tilde{\cdot}$ над любым символом \cdot , получаем первое уравнение из (13.15). Из второго уравнения (13.13) и из первых двух уравнений (13.16), учитывая добавление верхней крышки $\hat{\cdot}$ над любым символом \cdot , получаем второе уравнение из (13.15), имея в виду, что $\hat{P} > 0$ влечет $\hat{P}^{qq} > 0$, $\hat{D}^q > 0$ и $\det \hat{L}^{qq} \neq 0$. Из третьего уравнения (13.13), из первых двух уравнений (13.15) и из первого и третьего уравнений (13.16) следует

$$\hat{L}^{ss} \hat{D}^s (\hat{L}^{ss})^T = \tilde{L}^{ss} \tilde{D}^s (\tilde{L}^{ss})^T.$$

Благодаря единственности LD -факторизации, это дает третью строку из выражений (13.15). \square

Данный алгоритм привлекателен своей простотой, но применять его надо с большой осторожностью, принимая во внимание следующие замечания.

ЗАМЕЧАНИЕ 13.12. Из первых двух уравнений (13.13) видно, что $\hat{P}^{sq}(\hat{P}^{qq})^{-1} = \tilde{P}^{sq}(\tilde{P}^{qq})^{-1}$, так же как первые два уравнения из (13.15) дают $\hat{L}^{sq}(\hat{L}^{qq})^{-1} = \tilde{L}^{sq}(\tilde{L}^{qq})^{-1}$. Если на этапе экстраполяции должно быть сохранение ковариаций, что соответствует операторам $\tilde{P}^{sq} := \hat{P}^{sq}$ и $\tilde{P}^{qq} := \hat{P}^{qq}$, то значение K^{sq} не будет вообще изменяться. Известно, что такое сохранение ковариаций возникает в задаче чистой регрессии — в предположении, что оцениваемый вектор x постоянен во времени. Поэтому данный алгоритм в задаче регрессионного моделирования неприменим. Это понятно также из того, что в этой задаче условие (13.10) означает полную ненаблюдаемость компонент x^s оцениваемого вектора $x = \text{const}$.

ЗАМЕЧАНИЕ 13.13. Для алгоритма Бар-Ицхака (см. подразд 13.10) из $\tilde{P}^{sq} = 0$, а здесь из $\tilde{L}^{sq} = 0$, следует $K^{sq} = 0$, $\hat{x}^s = \tilde{x}^s$, а также $\hat{P}^{sq} = 0$, $\hat{P}^{ss} = \tilde{P}^{ss}$ и $\hat{L}^{sq} = 0$. Поэтому, если в измерение z непосредственно попадает только часть x^q оцениваемого вектора x и если априори выбрана такая матрица \tilde{P} (для данного алгоритма \tilde{L}), что в ней $\tilde{P}^{sq} = 0$ (соответственно, $\tilde{L}^{sq} = 0$), то в принципе нельзя улучшить априорную оценку \tilde{x}^s . Отсюда видно, насколько важно задавать $\tilde{P}^{sq} \neq 0$ (соответственно, $\tilde{L}^{sq} \neq 0$) до запуска алгоритма. В динамических задачах оценивания это не столь критично, так как там на этапе экстраполяции вычисляют новое $\tilde{P} := \Phi \hat{P} \Phi^T + \Gamma Q \Gamma^T$, а не просто полагают $\tilde{P} := \hat{P}$.

ЗАМЕЧАНИЕ 13.14. Выбор LD -факторизации здесь не произволен, как отмечается в [89], а подчинен желанию обеспечить независимость редуцированного фильтра (13.12) (соответственно, этапа **II** алгоритма на стр. 290), а также правым, а не левым размещением нулевого блока в (13.10).

ЗАМЕЧАНИЕ 13.15. Первое выражение в (13.15) реализуется проще без вычисления обратной матрицы, если его переписать как линейную систему

$$(\tilde{L}^{qq})^T (K^{sq})^T = (\tilde{L}^{sq})^T$$

и решать ее обратной подстановкой относительно $(K^{sq})^T$.

СЛЕДСТВИЕ 13.1. В условиях теоремы 13.8 ее утверждение справед-

ливо для алгоритма, в котором (13.15) заменены следующими выражениями:

$$\left. \begin{aligned} \bar{L}^{qq} &= (\tilde{L}^{qq})^{-1} \hat{L}^{qq}, \\ \hat{L}^{sq} &= \tilde{L}^{sq} \bar{L}^{qq}, \\ \hat{L}^{ss} &= \tilde{L}^{ss}, \quad \hat{D}^s = \tilde{D}^s, \\ \delta^q &= (\tilde{L}^{qq})^{-1} (\hat{x}^q - \tilde{x}^q), \\ \hat{x}^s &= \tilde{x}^s + \tilde{L}^{sq} \delta^q. \end{aligned} \right\} \quad (13.17)$$

ЗАМЕЧАНИЕ 13.16. Первое выражение из (13.17) реализуется не вычислением обратной матрицы, а применением процедуры обратной подстановки к системе $\tilde{L}^{qq} \bar{L}^{qq} = \hat{L}^{qq}$ относительно \bar{L}^{qq} . Аналогично, четвертое выражение из (13.17) реализуется применением обратной подстановки к системе $\tilde{L}^{qq} \delta^q = \hat{x}^q - \tilde{x}^q$ относительно δ^q . Следствие 13.1 дает более эффективный алгоритм, чем теорема 13.8, когда $q + 1 < s = n - q$, т. е. при $q < (n - 1)/2$.

13.12 Задача сопровождения судна на траектории

Обработка данных, поступающих от некоторого источника, с помощью алгоритмов фильтрации требует иметь отдельно модель движения (состояния) объекта и модель измерений. Продемонстрируем, как строить эти модели применительно к анализу движения надводного судна.

Модель движения судна

Текущее состояние судна на плоской траектории с точки зрения кинематики (а не динамики) движения, характеризуется пятью параметрами: две координаты, скорость, направление и угловая скорость движения. Модель пространственного движения включает 12 уравнений для координат центра масс, углов Эйлера, составляющих линейной и угловой скоростей в связанной системе координат [60]. Представим движение судна как дискретный стохастический процесс s_k , подчиняющийся следующему уравнению [114]:

$$s_k = \begin{bmatrix} x + (2/\omega) v \sin(\omega \Delta t/2) \cos(\phi + \omega \Delta t/2) \\ y + (2/\omega) v \sin(\omega \Delta t/2) \sin(\phi + \omega \Delta t/2) \\ v \\ \phi + \omega \Delta t \\ \omega \end{bmatrix}_{k-1} + \Gamma w_{k-1}. \quad (13.18)$$

Модель (13.18) есть модель состояния s_k , что позволяет применять теорию фильтрации Калмана. Вектор $s_k = (x, y, v, \phi, \omega)^T$ содержит пять величин: x, y — географические координаты цели (широта и долгота в прямоугольной системе координат в метрах), v — линейная скорость (м/с), ϕ — направление движения (курсовой угол в радианах от северного направления по часовой стрелке), ω — угловая скорость (рад/с). Аддитивно действующий белый шум с единичной ковариационной матрицей $Q(t_i) = I$ обозначен через w_{k-1} , а Δt — период дискретизации времени. Ковариация шума Γw равна [114]

$$\text{cov}(\Gamma w) \triangleq \mathbf{E} \{ \Gamma w w^T \Gamma^T \} = \Gamma \Gamma^T = \text{diag} \left[\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \Delta t^2 \sigma_v^2, \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t^2 \end{bmatrix} \sigma_{\dot{\omega}}^2 \right],$$

где σ_v^2 и $\sigma_{\dot{\omega}}^2$ — дисперсии случайных гауссовских процессов, аддитивно воздействующих на скорость корабля и его угловую скорость, соответственно.

Модель движения (13.18) включает угловую скорость и потому пригодна для анализа траектории как маневрирующего судна, так и судна, движущегося равномерно и прямолинейно (в последнем случае угловая скорость судна-цели равна нулю).

Модель измерений

Будем считать, что наблюдения за судном ведутся при помощи радиолокационной станции (РЛС), возвращающей измерения сферических (на плоскости — полярных) координат цели: азимут z_θ и дальность до цели z_ρ .

Построим модель измерения в декартовых координатах. Будем называть такие преобразованные измерения *псевдоизмерениями*. Первичные измерения, приходящие от РЛС, обозначим $z_{\text{sph}} = \begin{bmatrix} z_\rho \\ z_\theta \end{bmatrix}$. Тогда $z_\rho = \rho + \delta_\rho$,

$z_\theta = \theta + \delta_\theta$, где ρ и θ — истинные значения дальности и пеленга, а δ_ρ и δ_θ — соответствующие погрешности измерения. Обозначим шум измерений полярных координат $v_{\text{sph}} = \begin{bmatrix} \delta_\rho \\ \delta_\theta \end{bmatrix}$. Его ковариационная матрица имеет вид

$R_{\text{sph}} \triangleq \mathbf{E} \{ v_{\text{sph}} v_{\text{sph}}^T \} = \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$, т. е. матрица диагональна, а шумы по дальности и азимуту взаимно не коррелированы (ошибки измерения дальности не связаны с ошибками измерения направления на цель).

Перейдем от измерений в сферической (точнее, полярной) системе координат к псевдоизмерениям в декартовых координатах:

$$z_{\text{dec}} = \begin{bmatrix} z_x \\ z_y \end{bmatrix} = \begin{bmatrix} z_\rho \cos(z_\theta) \\ z_\rho \sin(z_\theta) \end{bmatrix}.$$

Для удобства используем здесь и далее индекс «dec» в случаях работы с декартовыми координатами. Малые приращения координат в сферической и декартовой системе координат связаны соотношением:

$$dz_{\text{dec}} = \begin{bmatrix} dz_x \\ dz_y \end{bmatrix} = \begin{bmatrix} \cos(z_\theta) & -z_\rho \sin(z_\theta) \\ \sin(z_\theta) & z_\rho \cos(z_\theta) \end{bmatrix} \begin{bmatrix} dz_\rho \\ dz_\theta \end{bmatrix}.$$

Обозначим $T = \begin{bmatrix} \cos(z_\theta) & -z_\rho \sin(z_\theta) \\ \sin(z_\theta) & z_\rho \cos(z_\theta) \end{bmatrix}$. Перейдем от дифференциалов к конечным разностям и получим закон преобразования случайных погрешностей измерений из полярных координат в декартовы:

$$\Delta z_{\text{dec}} = \begin{bmatrix} \Delta z_x \\ \Delta z_y \end{bmatrix} = \begin{bmatrix} \cos(z_\theta) & -z_\rho \sin(z_\theta) \\ \sin(z_\theta) & z_\rho \cos(z_\theta) \end{bmatrix} \begin{bmatrix} \Delta z_\rho \\ \Delta z_\theta \end{bmatrix} = T \Delta z_{\text{sph}}.$$

Ковариационная матрица погрешностей измерений в декартовых координатах равна

$$R_{\text{dec}} \triangleq \mathbf{E} \{ \Delta z_{\text{dec}} \Delta z_{\text{dec}}^T \} = \mathbf{E} \{ T \Delta z_{\text{sph}} \Delta z_{\text{sph}}^T T^T \} = T R_{\text{sph}} T^T.$$

Матрица T здесь зависит от времени, однако для простоты изложения индекс времени опускаем.

Выразим псевдоизмерения через истинные декартовы координаты цели и погрешности псевдоизмерений:

$$\begin{aligned} z_{\text{dec}} &= \begin{bmatrix} z_\rho \cos(z_\theta) \\ z_\rho \sin(z_\theta) \end{bmatrix} = \begin{bmatrix} (\rho + \Delta\rho) \cos(\theta + \Delta\theta) \\ (\rho + \Delta\rho) \sin(\theta + \Delta\theta) \end{bmatrix} \approx \\ &\approx \begin{bmatrix} \rho \cos(\theta) + \Delta z_x \\ \rho \sin(\theta) + \Delta z_y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta z_x \\ \Delta z_y \end{bmatrix}. \end{aligned}$$

Здесь обозначим: $\bar{s} = \begin{bmatrix} x \\ y \end{bmatrix}$ — точные значения декартовых координат цели,

$\bar{v} = \begin{bmatrix} \Delta z_x \\ \Delta z_y \end{bmatrix}$ — погрешности псевдоизмерений декартовых координат. Тогда $z_{\text{dec}} = \bar{s} + \bar{v}$. Вследствие нелинейности преобразования координат ковариационная матрица погрешностей псевдоизмерений R_{dec} не является диагональной. Приведем ковариационную матрицу к единичному виду, т. е. выполним декорреляцию погрешностей псевдоизмерений.

Представим матрицу R_{dec} как произведение $R_{\text{dec}} = S_{\text{dec}} S_{\text{dec}}^T$, где $S = TR_{\text{sph}}^{1/2}$. Обозначим $\tilde{z}_{\text{dec}} = S_{\text{dec}}^{-1} z_{\text{dec}} = S_{\text{dec}}^{-1}(\bar{s} + \bar{v}) = S_{\text{dec}}^{-1}\bar{s} + S_{\text{dec}}^{-1}\bar{v} = S_{\text{dec}}^{-1}\bar{s} + \tilde{v}$. Найдем ковариацию \tilde{v} : $\text{cov}(\tilde{v}) = \tilde{R}_{\text{dec}} \triangleq \mathbf{E}\{\tilde{v}\tilde{v}^T\} = \mathbf{E}\{S_{\text{dec}}^{-1}\bar{v}\bar{v}^T(S_{\text{dec}}^{-1})^T\} = S_{\text{dec}}^{-1}R_{\text{dec}}S_{\text{dec}}^{-T} = I$. Будем называть \tilde{z}_{dec} *нормализованными псевдоизмерениями*, а \tilde{v} — вектором погрешностей нормализованных псевдоизмерений (его компоненты взаимно не коррелированы, так как $\tilde{R}_{\text{dec}} = I$). Отсутствие корреляции компонентов вектора погрешностей позволяет применять скаляризованную обработку измерений.

Поскольку уравнения в системе (13.18) обладают значительной нелинейностью, непосредственное применение линейного фильтра Калмана, СКФ, не представляется возможным. Одним из способов обойти это ограничение является использование *расширенного фильтра Калмана* (Extended Kalman Filter, EKF).

Стандартный алгоритм расширенного фильтра Калмана

Расширенный фильтр Калмана является субоптимальным алгоритмом [90]. В области фильтрации и идентификации систем этот термин не обладает уникальностью смысла. В терминологии данного подраздела «расширенность» в сравнении с линейным фильтром состоит в возможности принятия нелинейной модели движения цели и/или модели измерений. В данном контексте это — общепринятая терминология. Формулы расширенной фильтрации получены в работе [128] методом инвариантного погружения. Матрицы Φ и H , задающие линейные преобразования, заменены функциями, которые в общем случае могут быть нелинейными. Это означает, что функционирование системы описывается следующими уравнениями:

$$\begin{aligned} x(t_{i+1}) &= f[x(t_i)] + w(t_i), \\ z(t_i) &= h[x(t_i)] + v(t_i). \end{aligned}$$

Функция $f[\cdot]$ вычисляет состояние системы в момент времени t_{i+1} по состоянию в момент времени t_i , функция $h[\cdot]$ преобразует вектор состояния к виду, в котором измерения поступают на вход фильтра, $w(t_i)$ и $v(t_i)$ — шум процесса и шум измерения в момент t_i , соответственно. Когда обе функции $f[\cdot]$ и $h[\cdot]$ — линейные, расширенный фильтр превращается в стандартный фильтр Калмана.

Расширенный фильтр Калмана, как и стандартный, содержит два этапа: этап 1 — экстраполяция и этап 2 — обработка измерения.

Экстраполяция (здесь учтено, что $Q(t_i) = I$):

$$\begin{aligned}\hat{x}(t_i) &= f[\hat{x}(t_{i-1}^+)], \\ F(t_{i-1}) &= \left. \frac{\partial f}{\partial x} \right|_{x=\hat{x}(t_{i-1}^+)}, \\ P(t_i^-) &= F(t_{i-1})P(t_{i-1}^+)F^T(t_{i-1}) + \Gamma(t_{i-1})\Gamma^T(t_{i-1}).\end{aligned}$$

Обработка измерения (здесь учтено, что $R(t_i) = \tilde{R}_{\text{dec}} = I_m$):

$$\begin{aligned}\nu(t_i) &= z(t_i) - h[\hat{x}_k(t_i^-)], \\ H(t_i) &= \left. \frac{\partial h}{\partial x} \right|_{x=\hat{x}(t_{i-1}^+)}, \\ K(t_i) &= P(t_i^-)H^T(t_i)[H(t_i)P(t_i^-)H^T(t_i) + I_m]^{-1}, \\ P(t_i^+) &= P(t_i^-) - K(t_i)H(t_i)P(t_i^-), \\ \hat{x}(t_i^+) &= \hat{x}(t_i^-) + K(t_i)\nu(t_i).\end{aligned}$$

Приведенные уравнения расширенной фильтрации по структуре совпадают с уравнениями обычного, линейного фильтра. Это — следствие того, что в них сохранены только первые (линейные) члены разложения функций $f[\cdot]$ и $h[\cdot]$ в ряды Тейлора. В силу этого, рассматриваемый расширенный фильтр — линейный (так называемый *расширенный фильтр первого порядка*).

Линеаризация моделей процесса и измерений производится относительно последней (текущей) оценки вектора состояния. Для этого вычисляют две матрицы Якоби: $F(t_{i-1})$ и $H(t_i)$, которые затем подставляют в уравнения стандартной калмановской фильтрации.

Эффективная реализация расширенного фильтра

Покажем, как применять рассмотренные выше квадратно-корневые реализации фильтра Калмана к задаче фильтрации траектории судна-цели в расширенном фильтре. Прделаем это на примере одной из реализаций — на примере фильтра Поттера, хотя могут быть использованы другие.

Пусть на вход фильтра поступают нормализованные псевдоизмерения (обладающие, как показано выше, единичной ковариационной матрицей ошибок). Для модели процесса, описываемой уравнениями (13.18), функции $f[\cdot]$ и $h[\cdot]$ имеют следующий вид:

$$f[s(t_i)] = \begin{bmatrix} x + (2/\omega) v \sin(\omega \Delta t/2) \cos(\phi + \omega \Delta t/2) \\ y + (2/\omega) v \sin(\omega \Delta t/2) \sin(\phi + \omega \Delta t/2) \\ v \\ \phi + \omega \Delta t \\ \omega \end{bmatrix}_{t_i},$$

$$h[s(t_i)] = \begin{bmatrix} x \\ y \end{bmatrix}_{t_i}.$$

Обозначим: $\varphi = \phi + \omega \Delta t$. Вычислим матрицы Якоби:

$$F(t_{i-1}) \triangleq \frac{\partial f}{\partial x} \Big|_{x=\hat{x}(t_{i-1}^+)} \triangleq F = \begin{bmatrix} 1 & 0 & \frac{\sin(\varphi) - \sin(\phi)}{\omega} & \frac{v(\cos(\varphi) - \cos(\phi))}{\omega} & \frac{v(\Delta t \omega \cos(\varphi) + \sin(\phi) - \sin(\varphi))}{\omega^2} \\ 0 & 1 & \frac{\cos(\phi) - \cos(\varphi)}{\omega} & \frac{v(\sin(\varphi) - \sin(\phi))}{\omega} & \frac{v(\Delta t \omega \sin(\varphi) + \cos(\varphi) - \cos(\phi))}{\omega^2} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{t_{i-1}^+},$$

$$H(t_i) \triangleq \frac{\partial h}{\partial x} \Big|_{x=\hat{x}(t_{i-1}^+)} \approx \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} = H.$$

Поскольку нормализованные псевдоизмерения на входе фильтра «производятся» (т. е. формально представлены) в той же системе координат, в которой записаны координаты объекта в векторе состояния, функция $h[\cdot]$ выполняет линейное преобразование, и матрица $H(t_i) = H$ не зависит от времени. Приближенность записи $H(t_i) = H$ здесь состоит в том, что в условиях малости величин Δz_{sph} косинус угловой погрешности заменен на 1.

При прямолинейном равномерном движении ($\omega = 0$) функции $f[\cdot]$ и F принимают следующую форму:

$$\begin{aligned}
f_{\omega=0}[s(t_i)] &= \lim_{\omega \rightarrow 0} \begin{bmatrix} x + (2/\omega) v \sin(\omega \Delta t/2) \cos(\phi + \omega \Delta t/2) \\ y + (2/\omega) v \sin(\omega \Delta t/2) \sin(\phi + \omega \Delta t/2) \\ v \\ \phi + \omega \Delta t \\ \omega \end{bmatrix}_{t_i} = \\
&= \begin{bmatrix} x + v \Delta t \cos(\phi) \\ y + v \Delta t \sin(\phi) \\ v \\ \phi \\ 0 \end{bmatrix}_{t_i}, \\
F_{\omega=0}(t_{i-1}) &= \begin{bmatrix} 1 & 0 & \Delta t \cos(\phi) & -v \Delta t \sin(\phi) & -\frac{v \Delta t^2 \sin(\phi)}{2} \\ 0 & 1 & \Delta t \sin(\phi) & v \Delta t \cos(\phi) & -\frac{v \Delta t^2 \cos(\phi)}{2} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{t_{i-1}^+}.
\end{aligned}$$

Конкретные значения элементов F во время фильтрации получаются подстановкой сюда элементов соответствующего вектора оценки состояния, однако для упрощения записей в последующих формулах символы экстраполяционной оценки (тильда \sim) и отфильтрованной оценки («крышка» $\hat{}$) опущены над всеми входящими величинами.

Учывая изложенное, скаляризованный алгоритм Поттера для нелинейной фильтрации данных о траектории судна-цели получим в следующем виде:

I. Экстраполяция:

$$\begin{aligned}
\hat{s}(t_i^-) &= \begin{bmatrix} x + (2/\omega) v \sin(\omega \Delta t/2) \cos(\phi + \omega \Delta t/2) \\ y + (2/\omega) v \sin(\omega \Delta t/2) \sin(\phi + \omega \Delta t/2) \\ v \\ \phi + \omega \Delta t \\ \omega \end{bmatrix}_{t_i^+}, \quad \hat{s}(t_0^+) = \bar{s}_0, \\
F(t_i, t_{i-1}) &= \begin{bmatrix} 1 & 0 & \frac{\sin(\varphi) - \sin(\phi)}{\omega} & \frac{v(\cos(\varphi) - \cos(\phi))}{\omega} & \frac{v(\Delta t \omega \cos(\varphi) + \sin(\phi) - \sin(\varphi))}{\omega^2} \\ 0 & 1 & \frac{\cos(\phi) - \cos(\varphi)}{\omega} & \frac{v(\sin(\varphi) - \sin(\phi))}{\omega} & \frac{v(\Delta t \omega \sin(\varphi) + \cos(\varphi) - \cos(\phi))}{\omega^2} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{t_{i-1}^+},
\end{aligned}$$

$$\begin{bmatrix} S^T(t_i^-) \\ 0 \end{bmatrix} = T \begin{bmatrix} S^T(t_{i-1}^+) F^T(t_i, t_{i-1}) \\ \Gamma^T(t_i) \end{bmatrix}.$$

II. Обработка измерения:

А. Начальное присваивание: $\tilde{S} = S(t_i^-)$; $\tilde{s} = \hat{s}(t_i^-)$.

Б. m -кратное повторение процедуры «скалярного» обновления:

$$\begin{aligned} q &:= \tilde{S}^T h, \quad \alpha := q^T q + 1, \quad \gamma := 1/(1 + \sqrt{1/\alpha}), \\ K &:= \tilde{S} q / \alpha, \quad \hat{S} := \tilde{S} - \gamma K q^T, \\ \hat{s} &:= \tilde{s} + K(z - h^T \tilde{s}) \end{aligned}$$

с экстраполяцией между повторениями $\tilde{S} := \hat{S}$, $\tilde{s} := \hat{s}$.

В. Завершающее присваивание: $S(t_i^+) := \hat{S}$, $\hat{s}(t_i^+) := \hat{s}$,
где h — j -й столбец матрицы $H^T(t_i)$; z — j -й элемент вектора $z(t_i)$
($j = 1, 2, \dots, m$).

ЗАМЕЧАНИЕ 13.17. В рассматриваемой прикладной задаче может быть использован также *расширенный фильтр второго порядка*, как это сделано в работе [95]. Важно отметить, что для придания своему фильтру численной устойчивости авторы [95] также приводят его к квадратно-корневой форме.

13.13 Пример задачи с мультиколлинеарностью

Пусть требуется решить несовместную систему $Ax \approx z$ при

$$A = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ N \end{matrix} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix} \end{matrix}, \quad z = \begin{matrix} & 1 \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ N \end{matrix} & \begin{pmatrix} 1 \\ 2 \\ \vdots \\ N \end{pmatrix} \end{matrix} \quad (13.19)$$

по критерию наименьших квадратов: $\|z - Ax\|^2 = (z - Ax)^T(z - Ax) \rightarrow \min_x$.

Этот пример здесь выбран потому, что он иллюстрирует «тяжелый» случай *мультиколлинеарности*: столбцы матрицы A в (13.19) не просто коллинеарны, они совпадают! Данный пример в его статистической интерпретации эквивалентен решению задачи оптимального оценивания неизвестного параметра \hat{x}_N° по экспериментальным данным

$$z = A\hat{x}_N^\circ + v, \quad v \sim \mathcal{N}(0, I), \quad (13.20)$$

где $\mathcal{N}(0, I)$ — нормальное распределение с нулевым средним значением и единичной матрицей ковариаций: $\mathbf{E} \{vv^T\} = I$.

УПРАЖНЕНИЕ 13.5. Докажите, что в данном примере *нормальная система* $A^T A \hat{x}_N = A^T z$ относительно *МНК-решения* $\hat{x}_N \in \mathbb{R}^n$ (здесь $n = 2$) имеет вид

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \hat{x}_N = \frac{N+1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (13.21)$$

с общим решением

$$\hat{x}_N = \hat{x}_N^\circ + \gamma \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \hat{x}_N^\circ = \frac{N+1}{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \forall \gamma \in \mathbb{R}, \quad (13.22)$$

где \hat{x}_N° есть *нормальное псевдорешение*, — единственное *МНК-решение* с минимальной нормой среди всех *МНК-решений* \hat{x}_N , γ — любое число.

Пригоден ли здесь метод нормальных уравнений? Очевидно — нет, так как нормальная система (13.21) вырождена.

1. *По поводу метода нормальных уравнений.* Даже если нормальная система не является вырожденной, она может оказаться плохо обусловленной. В данном примере она вырождена: $\Lambda = N \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. Кроме того, формирование нормальной системы может потребовать хранения и затем одновременной обработки большого массива данных. Например, при обработке данных в астрономии или спутниковой геодезии N (число экспериментальных данных в векторе z) очень велико — десятки или сотни тысяч — при том, что число оцениваемых параметров n в векторе x сравнительно мало — несколько десятков [68]. Единовременная обработка сверхбольших массивов данных $[A \mid z]$ в таких задачах, очевидно, нецелесообразна. Главное, однако, не в этом, а в особенностях численного решения задач с мультиколлинеарностью.
2. *По поводу численного решения задачи.* Решение нормальной системы в ситуациях, близких к мультиколлинеарности, любым одновременным методом (см. разд. 12, стр. 264) не дает результатов. Так, в данном примере (13.19) метод Хаусхолдера, примененный к нормальным уравнениям (13.21), приводит их к эквивалентному виду

$$\begin{bmatrix} -\sqrt{2} & -\sqrt{2} \\ 0 & 0 \end{bmatrix} \hat{x}_N = \frac{N+1}{2} \begin{bmatrix} -\sqrt{2} \\ 0 \end{bmatrix}, \quad (13.23)$$

с общим решением (13.22), но не позволяет выделить \hat{x}_N° .

Может ли в подобных задачах помочь последовательная обработка, изложенная выше и принципиально возможная как в информационной форме (см. подразд. 11.6, стр. 244), так и в ковариационной форме (см. подразд. 11.7–11.8 и далее подразд. 13.3–13.8)?

Введем обозначения: $A_1 = [1 \mid 1]$, $z_1 = [1]$, $a_i^T = [1 \mid 1]$, $A_{i+1} = \begin{bmatrix} A_i \\ a_i^T \end{bmatrix}$, $z_{i+1} = \begin{bmatrix} z_i \\ \zeta_{i+1} \end{bmatrix}$, $\zeta_i = i$, $i = 1, 2, \dots$. Тогда $A_N = A$, $z_N = z$. Информационный последовательный алгоритм для $i = 1, 2, \dots$ имеет вид:

$$\Lambda_i = \Lambda_{i-1} + a_i a_i^T, \quad \Lambda_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad d_i = d_{i-1} + a_i \zeta_i, \quad d_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Тогда $\Lambda_N = \Lambda$, $d_N = \frac{N(N+1)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, и при $i = N$ получаем нормальную систему (13.21) в виде $\Lambda_N \hat{x}_N = d_N$. Отсюда видно, что при таком Λ_0 этот информационный последовательный алгоритм также не решает проблему.

УПРАЖНЕНИЕ 13.6. Докажите, что если взять $\Lambda_0 = \varepsilon^2 I$, то $\Lambda_i = \begin{bmatrix} i + \varepsilon^2 & i \\ i & i + \varepsilon^2 \end{bmatrix}$, $d_i = \frac{i(i+1)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, и нормальная система имеет вид $\Lambda_i \hat{x}_i = d_i$, следовательно, $\hat{x}_i = \Lambda_i^{-1} d_i = \frac{i(i+1)}{2(2i + \varepsilon^2)} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \xrightarrow{\varepsilon \rightarrow 0} \frac{i+1}{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \hat{x}_i^\circ$.

Убедитесь, что прием, примененный в этом упражнении 13.6, фактически совпадает с *ридж-оцениванием* [18] (гребневое оценивание), т. е. с *методом регуляризации Тихонова* [67, 81]. Проблема, которая здесь остается, — выбор малого ридж-параметра ε^2 . Из этого упражнения видно, что хотя рекуррентный информационный алгоритм и прост в реализации, и формально дает решение, стремящееся к нормальному псевдорешению \hat{x}_i° при $\varepsilon \rightarrow 0$, для явного его получения не избежать решения нормальной системы. При малом ε^2 это снова создает проблему. Ридж-оценивание ее не решает. Выход можно искать в рекуррентном ковариационном алгоритме МНК.

УПРАЖНЕНИЕ 13.7. Возьмите к качестве первого рекуррентного ковариационного алгоритма МНК скаляризованную форму фильтра Калмана (см. подразд. 13.3) с начальной матрицей ковариации $P_0 = \Lambda_0^{-1} = \varepsilon^{-2} I$ при малом ε (в пределе $\varepsilon \rightarrow 0$). Получите этот алгоритм в виде:

Таблица 13.1. Скаляризованный фильтр Калмана в задаче регрессионного моделирования с мультиколлинеарностью регрессоров

Величина	Зависимость от ε	Предел при $\varepsilon \rightarrow 0$	То же при $i = N$
α_i	$\frac{2i + \varepsilon^2}{2(i-1) + \varepsilon^2}$	$\frac{i}{i-1}, \quad i \geq 2$	$\frac{N}{N-1}$
K_i	$\frac{1}{2i + \varepsilon^2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\frac{1}{2i} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\frac{1}{2N} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
P_i	$\frac{P_0}{2i + \varepsilon^2} \begin{bmatrix} i + \varepsilon^2 & -i \\ -i & i + \varepsilon^2 \end{bmatrix}$	$\frac{P_0}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$	$\frac{P_0}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$
\hat{x}_i	$\frac{i(i+1)}{2(2i + \varepsilon^2)} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\frac{i+1}{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\frac{N+1}{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \hat{x}_N^\circ$

Для $i = 1, 2, \dots$ выполнять:

$$\alpha_i = a_i^T P_{i-1} a_i + 1, \quad K_i = P_{i-1} a_i / \alpha_i, \quad (13.24)$$

$$P_i = P_{i-1} - K_i a_i^T P_{i-1}, \quad \hat{x}_i = \hat{x}_{i-1} + K_i (\zeta_i - a_i^T \hat{x}_{i-1}). \quad (13.25)$$

УПРАЖНЕНИЕ 13.8. Докажите результаты работы этого алгоритма, сведенные в табл. 13.1.

Таким образом, алгоритм (13.24), (13.25) не требует ни решения систем, ни обращения матриц для задач вида (13.19), (13.20), — независимо от размерностей и конкретных данных, — и при малом ε^2 он гарантирует выделение в явном виде решения \hat{x}_N° в составе общего МНК-решения (13.22). Проблемы выбора малого ε^2 не возникает; ε^{-2} нужно выбирать настолько большим, насколько позволяет диапазон вещественных чисел на компьютере, поскольку параметры алгоритма K_i , P_i обычно быстро убывают и стабилизируются [61].

УПРАЖНЕНИЕ 13.9. Наилучшие результаты дадут в этой задаче не формулы (13.24), (13.25), а их алгебраический эквивалент, получаемый при переходе от P_i к квадратному корню из P_i . Аналогично упражнению 13.8, испытайте другие алгоритмы, а именно: стабилизированный фильтр Калмана–Джозефа из подразд. 13.4, фильтр Поттера из подразд. 13.5, LDL^T -фильтр Бирмана из подразд. 13.7, UDU^T -фильтр Бирмана из задания на стр. 311, LL^T -фильтр Карлсона из подразд. 13.8 или UU^T -фильтр Карлсона из задания на стр. 311.

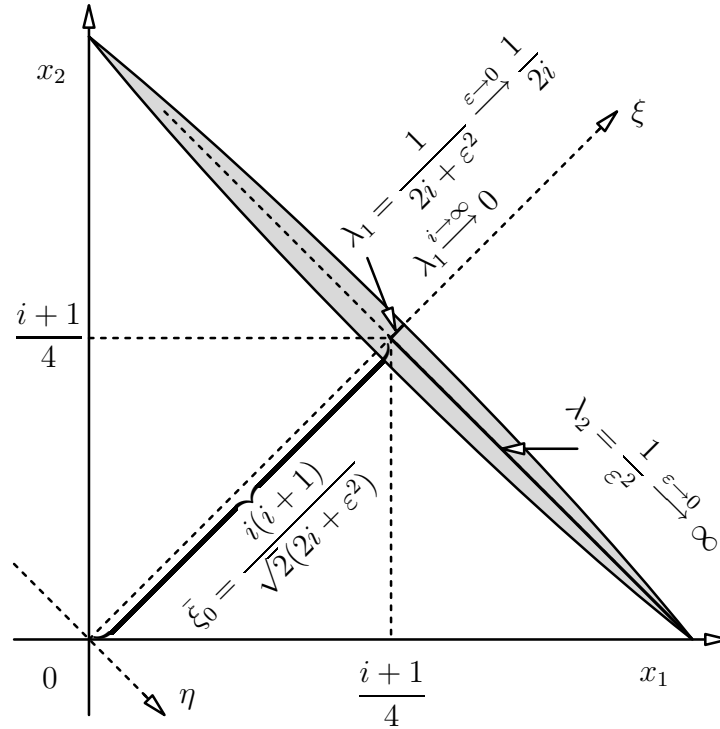


Рис. 13.1. Сечение поверхности критерия качества $J(x) = (z - Ax)^T(z - Ax)$ (для упражнения 13.10) плоскостью уровня

Исследование задачи (13.19) будет неполным, если вы не изучите поверхность критерия качества $J(x) = (z - Ax)^T(z - Ax)$ и не используете возможность преобразования базиса в пространстве \mathbb{R}^n , здесь $n = 2$ — размерность оцениваемого вектора x .

УПРАЖНЕНИЕ 13.10. Задайте начальное значение ковариации $P_0 = \Lambda_0^{-1} = \varepsilon^{-2}I$. Возьмите зависимость P_i от ε из табл. 13.1. Найдите собственные значения матрицы P_i . Убедитесь, что они равны:

$$\lambda_1 = \frac{1}{2i + \varepsilon^2}, \quad \lambda_2 = \frac{1}{\varepsilon^2}.$$

Найдите соответствующие собственные векторы. Убедитесь, что они равны:

$$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Постройте график на плоскости $x = (x_1, x_2)$ и дайте необходимые обоснования к его построению (рис. 13.1).

Введите преобразование базиса по формуле $x_\star \triangleq \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} x$. Убедитесь, что при этом $P_0^\star = \varepsilon^{-2}I$ выводится из $P_0 = \varepsilon^{-2}I$. Докажите, что стандартный ковариационный алгоритм дает на i -м шаге значения

$$\alpha_i^\star = \alpha_i, \quad K_i^\star = \frac{\sqrt{2}}{2i + \varepsilon^2} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad P_i^\star = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad \bar{x}_i^\star = \frac{i(i+1)}{\sqrt{2}(2i + \varepsilon^2)} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

вместо значений во втором столбце табл. 13.1 на стр. 303. Переход к координатам $x_\star \triangleq \begin{bmatrix} \xi & \eta \end{bmatrix}^T$ соответствует *методу главных компонент* [18]. Видно, что первая компонента выделяется (оценивается) со все возрастающей точностью (по мере $i \rightarrow \infty, \varepsilon \rightarrow 0$): $\bar{\xi} = i(i+1)/(\sqrt{2}(2i + \varepsilon^2)) \rightarrow (i+1)/(2\sqrt{2})$, а оценка второй компоненты остается равна нулю: $\bar{\eta} = 0$. Проверьте, что обратное преобразование дает правильный результат: $\bar{x}_1 \triangleq (\bar{\xi} + \bar{\eta})/\sqrt{2} \rightarrow (i+1)/4$ и $\bar{x}_2 \triangleq (\bar{\xi} - \bar{\eta})/\sqrt{2} \rightarrow (i+1)/4$.

13.14 Задание на лабораторный проект № 9

Введение

В данном лабораторном проекте мы предлагаем к изучению современные методы построения численно устойчивых и экономичных по затратам ресурсов ЭВМ алгоритмов метода наименьших квадратов (МНК), решающих актуальную задачу *последовательного* обновления оценок по измерениям. Как уже отмечалось, эта задача возникает во многих приложениях, например, при оценке состояния (элементов движения) объекта по последовательно поступающим данным наблюдения (см. выше подразд 13.12), при подгонке параметров модели под результаты продолжительных экспериментов, проводимых в физике, астрономии, экономике, бизнесе, социологии, психологии и в других областях, где выявление регрессий, анализ и прогнозирование тенденций опирается на включение в обработку данных наблюдения по мере их поступления, для того, чтобы постепенно идентифицировать модель объекта (процесса) или уточнять параметры модели.

1. Задание

А. Спроектировать, отладить и продемонстрировать в действии программу решения несовместной системы уравнений $Ax \approx b$, $A = A(m, n)$,

$m > n$, $\text{rank } A = n$, в смысле наименьших квадратов в соответствии с вашим вариантом последовательного алгоритма (см. ниже подразд. 13.15).

Б. Оценить результаты решения по трем показателям:

- (1) погрешность (абсолютная и относительная) решения;
- (2) затраты основной памяти компьютера на хранение данных, необходимых только для заданного алгоритма;
- (3) теоретическое и реальное число основных операций компьютера для выполнения заданного алгоритма.

Эти показатели определить в зависимости от следующих параметров задачи:

- (1) размерность задачи, т. е. число неизвестных n ;
- (2) степень переопределенности задачи, т. е. число p , указывающее, во сколько раз число уравнений m больше числа неизвестных, $m = pn$;
- (3) степень несовместности системы, т. е. вещественное положительное число c , показывающее среднеквадратическое значение элементов случайного разностного вектора $d = b - A\bar{x}$, где \bar{x} — нормальное псевдорешение системы $Ax \approx b$, относительно среднего (единичного) значения;
- (4) способ генерации матрицы A .

В. Провести вычислительный эксперимент, используя в нем:

- (1) десять значений n , $n = 1, 2, \dots, 10$;
- (2) три значения p , $p = 10, 100$ и 1000 ;
- (3) три значения c , $c = 1/10, 1$ и 10 ;
- (4) четыре способа генерации матрицы A (см. п. 2 в подразд. 13.14).

Результаты эксперимента вывести на экран в виде следующей таблицы:

Вычислительный эксперимент

$p=(\text{значение}), \quad c=(\text{значение}), \quad A=(\text{способ})$					
n	Погрешность		Память КБайт	Число операций	
	абсолютная	относительная		теоретическое	реальное

При подсчете числа операций учитывать: сложение, умножение, деление и извлечение квадратного корня. К отчету о работе приложить расчетные

формулы числа операций отдельно по этим видам операций и их сумму. В таблицу выводить только суммарное число операций.

Г. Выполнить отладку программы и продемонстрировать результаты отладки на решении следующей тестовой задачи [82]:

$$Ax \approx b, \quad A = A(m, 2) = \begin{bmatrix} a_{i1} & a_{i2} \end{bmatrix}, \quad i = 1, 2, \dots, m;$$

$$m = 4, 8, 12, 16, 20, 24, 28, 32, 36, 40;$$

$$a_{i1} = w_i = \sin(2\pi i/m), \quad a_{i2} = w_{i-1};$$

$$b^T = (b_1, \dots, b_m), \quad b_i = 2 \cos(2\pi i/m).$$

Известно (стр. 267), что решением этой задачи является вектор $\bar{x}^T = (\bar{x}_1, \bar{x}_2)$,

$$\bar{x}_1 = 2 \operatorname{ctg}(2\pi/m), \quad \bar{x}_2 = -2 \operatorname{cosec}(2\pi/m). \quad (13.26)$$

Для демонстрации процесса отладки вывести на экран еще одну таблицу:

Отладка программы		
m	Погрешность	
	абсолютная	относительная

Д. Во всех случаях для оценки абсолютной погрешности использовать норму вектора $e = x - \bar{x}$ вида

$$\|e\|_\infty = \max_i |e_i|,$$

где x — вычисленное решение, а относительную погрешность определять по формуле

$$\delta = \|e\|_\infty / \|\bar{x}\|_\infty,$$

в которой \bar{x} — точное МНК-решение (нормальное псевдорешение задачи). В отладочной (тестовой) задаче по п. Г решением является вектор (13.26), а в задачах вычислительного эксперимента по п. В — вектор $\bar{x} = (1, 2, \dots, n)$.

2. Генерация задач для вычислительного эксперимента

Как отмечено, для этих задач точное МНК-решение следует задать в виде вектора $\bar{x} = (1, 2, \dots, n)$. Затем следует сгенерировать матрицу A (см. ниже) и образовать вектор $\hat{b} = A\bar{x}$. К нему нужно добавить случайный вектор $d = c\xi$, где c — число (см. выше п. 1), а $\xi \sim \mathcal{N}(0, 1)$ — вектор случайных чисел (от подпрограммы псевдослучайных чисел), взятых из стандартного

нормального распределения с нулевым средним значением и единичной дисперсией. В результате получаем вектор $b = A\bar{x} + d$ для системы уравнений $Ax \approx b$.

Для генерации матрицы A необходимо предусмотреть один из четырех способов:

Способ 1 — матрица $A = A(m, n)$ заполняется случайными числами из равномерного распределения в диапазоне $[-100, +100]$. Условно запишем это так:

$$A = [\text{Random}(m \times n)].$$

Способ 2 — верхняя часть матрицы A , а именно, ее первые n строк, заполняются по способу 1, а остальные строки образуют подматрицу, в которой только первые q столбцов, где $q = \lfloor n/2 \rfloor$ — ближайшее снизу целое число к $n/2$, заполняются как в способе 1, а все остальные столбцы — нулевые. Таким образом, матрица имеет вид:

$$A = \begin{array}{|c|c|} \hline \text{Random}(n \times n) & \\ \hline \text{Random} & 0 \\ \hline \end{array}$$

Способ 3 — первая часть матрицы A должна формироваться как в способе 2, а оставшаяся часть образуется располагающимися последовательно вниз блоками из единичных матриц I размера $n \times n$:

$$A = \begin{array}{|c|} \hline \text{Random}(n \times n) \\ \hline I(n \times n) \\ \dots \\ I(n \times n) \\ \hline \end{array}$$

Способ 4 — верхняя часть матрицы A должна формироваться как в способе 2, оставшая же часть строится подобно способу 2, но ненулевые q столбцов нижней подматрицы заполняются не случайными числами, а располагающимися последовательно вниз блоками из единичных матриц I размера $(q \times q)$:

$$A = \begin{array}{|c|c|} \hline \text{Random}(n \times n) & \\ \hline I(q \times q) & \\ \dots & \\ I(q \times q) & 0 \\ \hline \end{array}$$

ЗАМЕЧАНИЕ 13.18. Не нужно генерировать всю матрицу A , так же как и весь вектор b и весь вектор d , одновременно. Матрицу A нужно

генерировать построчно, а векторы b и d — поэлементно. Например, если $a^T = (a_1, a_2, \dots, a_n)$ есть текущая строка матрицы A , то $z = a^T \bar{x} + v$, где z — текущий элемент вектора b , v — текущий элемент вектора d , $v = c\xi$, $\xi \sim \mathcal{N}(0, 1)$ — текущее случайное число из стандартного нормального распределения. Таким образом, последовательно генерируемые данные (a^T, z) нужно вводить в алгоритм решения, а также использовать в нем значение $r = c^2$, имеющее смысл дисперсии ошибки измерения вектора $\hat{b} = A\bar{x}$, исключительно последовательно.

13.15 Варианты задания на лабораторный проект №9

Общее число вариантов составляет 26 (учитывая подварианты — различия в методах ортогонализации в некоторых из вариантов).

ЗАМЕЧАНИЕ 13.19. Для всех ковариационных алгоритмов (варианты 1–13) в качестве начальных значений можно взять: $x_0 = 0$, $P_0 = (1/\varepsilon^2)I$, $\varepsilon \rightarrow 0$.

Вариант 1. Стандартный ковариационный алгоритм (Калмана). Найдите его на стр. 274, подразд. 13.3.

Вариант 2. Стабилизированный ковариационный алгоритм (Джозефа). Найдите его на стр. 275, подразд. 13.4:

(ii) Обработка наблюдений (очередные данные $z = a^T \bar{x} + v$):

$$\alpha = a^T \tilde{P}a + r, \quad K = \tilde{P}a/\alpha,$$

$$\hat{P} = (I - Ka^T)\tilde{P}(I - aK^T) + rKK^T. \quad (13.27)$$

ЗАМЕЧАНИЕ 13.20. Вычислительные затраты существенно зависят от способа программирования выражений. Например, выражение (13.27) для \hat{P} может быть запрограммировано в следующей последовательности:

$$\begin{aligned} W_1 &= I - Ka^T, & (n \times n)\text{-матрица} \\ W_2 &= W_1 \tilde{P}, & (n \times n)\text{-матрица} \\ \hat{P} &= W_2 W_1^T + r(KK^T) \end{aligned}$$

или, эквивалентно, в виде:

$$\begin{aligned}
 v_1 &= \tilde{P}a, & n\text{-вектор} \\
 P_1 &= \tilde{P} - Kv_1^T, & (n \times n)\text{-матрица} \\
 v_2 &= P_1a, & n\text{-вектор} \\
 \hat{P} &= (P_1 - v_2K^T) + (rK)K^T,
 \end{aligned}$$

и в обоих способах можно экономить вычисления, благодаря симметрии матрицы \hat{P} . Однако второй способ имеет на порядок меньше вычислений: в первом способе выполняется $(1, 5n^3 + 3n^2 + n)$ умножений, а во втором только $(4n^2 + 2n)$ умножений.

Вариант 3. Квадратно-корневой ковариационный алгоритм Поттера. Найдите его на стр. 276, подразд. 13.5, в следующем виде.

(i) Инициализация (начальные значения x_0, P_0):

$$\tilde{x} := x_0, \quad \tilde{S} := P_0^{1/2}.$$

(ii) Обработка наблюдений (очередные данные $z = a^T\tilde{x} + v$):

$$\begin{aligned}
 f &= \tilde{S}^T a, \quad \alpha = f^T f + r, \quad \gamma = 1/(1 + \sqrt{r/\alpha}), \\
 K &= \tilde{S}f/\alpha, \quad \hat{S} = \tilde{S} - \gamma Kf^T, \\
 \hat{x} &= \tilde{x} + K(z - a^T\tilde{x}).
 \end{aligned}$$

(iii) Экстраполяция (между повторениями этапа (ii)):

$$\tilde{S} := \hat{S}, \quad \tilde{x} := \hat{x}.$$

ЗАМЕЧАНИЕ 13.21. Вариант 3, в котором вместо $\tilde{S} := \hat{S}$ предусмотрена процедура триангуляризации $\tilde{S} := \text{triang } \hat{S}$, дает следующие версии:

- Версия 3.1: обе матрицы, \tilde{S} и \hat{S} , — нижние треугольные ($S \equiv L$), или
- Версия 3.2: обе матрицы, \tilde{S} и \hat{S} , — верхние треугольные ($S \equiv U$).

Именно для этого этап (iii) должен содержать, вместо $\tilde{S} := \hat{S}$, процедуру триангуляризации $\tilde{S} := \text{triang } \hat{S}$, матрицы \hat{S} . Возможны четыре алгоритма этой процедуры: (1) отражения Хаусхолдера, (2) вращения Гивенса, (3) классическая Грама–Шмидта ортогонализация и (4) модифицированная Грама–Шмидта ортогонализация (см. лабораторный проект № 6). Соответственно

этому, всего имеем 8 подвариантов для указанного варианта 3, сведенных в следующую таблицу:

triang	$S \equiv L$	$S \equiv U$
Хаусхолдер	3.1.1	3.2.1
Гивенс	3.1.2	3.2.2
ГШО	3.1.3	3.2.3
МГШО	3.1.4	3.2.4

Вариант 4. Факторизованный LDL^T ковариационный алгоритм (Бирмана). Найдите его на стр. 279, подразд. 13.7.

ЗАМЕЧАНИЕ 13.22. Рациональное программирование этого LDL^T факторизованного алгоритма должно экономить память компьютера. Здесь можно записывать \hat{D} поверх \tilde{D} и столбцы \hat{L} поверх столбцов \tilde{L} (при этом нужна только поддиагональная часть этих столбцов).

Вариант 5. Факторизованный UDU^T ковариационный алгоритм (Бирмана). Выведите его самостоятельно аналогично алгоритму варианта 4 [15].

Вариант 6. Факторизованный UU^T ковариационный алгоритм Карлсона. Выведите его самостоятельно, опираясь на вывод LL^T алгоритма, данный в подразд. 13.8. Получите следующий результат [15]:

(i) Инициализация (начальные значения x_0, P_0):

$$\tilde{x}_0 := x_0, \quad \tilde{U} := P_0^{1/2}.$$

(ii) Обработка наблюдений (очередные данные $z = a^T \hat{x} + v$):

$$f = \tilde{U}^T a, \quad f^T = (f_1, \dots, f_n),$$

$$\alpha_0 = r, \quad K_1^T = \begin{pmatrix} \tilde{U}_{11} f_1 & 0 & \dots & 0 \end{pmatrix}.$$

Для $j = 1, \dots, n$ выполнять

$$\alpha_j = \alpha_{j-1} + f_j^2,$$

$$\beta_j = (\alpha_{j-1}/\alpha_j)^{1/2}, \quad \gamma_j = f_j/(\beta_j \alpha_j),$$

$$\hat{U}_{jj} = \beta_j \tilde{U}_{jj},$$

$$\hat{U}_{ij} = \beta_j \tilde{U}_{ij} - \gamma_j K_j(i), \quad i = 1, 2, \dots, j-1, j \neq 1,$$

$$K_{j+1}(i) = K_j(i) + f_j \tilde{U}_{ij}, \quad i = 1, 2, \dots, j.$$

По завершении цикла выполнить

$$K = K_{n+1}/\alpha_n, \quad \hat{x} := \tilde{x} + K(z - a^T \tilde{x}).$$

Вариант 7. Факторизованный LL^T ковариационный алгоритм Карлсона. Найдите его на стр. 285, подразд. 13.8.

Вариант 8. Редуцированный фильтр Бирмана. Найдите его на стр. 287, подразд. 13.9.

Вариант 9. Редуцированный стандартный ковариационный алгоритм Бар-Ицхака–Калмана. Найдите его на стр. 288, подразд. 13.10.

ЗАМЕЧАНИЕ 13.23. В этом варианте используются только способы 2 и 4 генерации матрицы A (см. выше п. 2 в подразд. 13.14). Инициализация и обработка первых n наблюдений выполняются, как в варианте 1. После этого каждое очередное наблюдение имеет вид $z = (a^{(q)})^T \bar{x}^{(q)} + v$, так как $a^T = \begin{bmatrix} a^{(q)T} & 0 \end{bmatrix}$ и $\bar{x}^T = \begin{bmatrix} \bar{x}^{(q)T} & \bar{x}^{(s)T} \end{bmatrix}$, где $s = n - q$. Вектор \bar{x} , так же как и соответствующие ему векторы x , \tilde{x} , \hat{x} , разбит на две части вида: $\bar{x}^{(q)}$ размерности q и $\bar{x}^{(s)}$ размерности s . Запишем алгоритм Бар-Ицхака–Калмана для варианта 9:

(ii) Обработка наблюдений (очередные данные) $z = a^{(q)T} \bar{x}^{(q)} + v$:

$$\alpha = a^{(q)T} \tilde{P}^{(qq)} a^{(q)} + r, \quad K^{(q)} = \tilde{P}^{(qq)} a^{(q)} / \alpha,$$

$$\hat{P}^{(qq)} = \tilde{P}^{(qq)} - K^{(q)} a^{(q)T} \tilde{P}^{(q)},$$

$$\hat{x}^{(q)} = \tilde{x}^{(q)} + K^{(q)} (z - a^{(q)T} \tilde{x}^{(q)}),$$

$$K^{(sq)} = \tilde{P}^{(sq)} (\tilde{P}^{(qq)})^{-1},$$

$$\hat{P}^{(sq)} = K^{(sq)} \hat{P}^{(qq)},$$

$$\hat{P}^{(ss)} = \tilde{P}^{(ss)} - K^{(sq)} (\tilde{P}^{(qq)} - \hat{P}^{(qq)}) (K^{(sq)})^T,$$

$$\hat{x}^{(s)} = \tilde{x}^{(s)} + K^{(sq)} (\hat{x}^{(q)} - \tilde{x}^{(q)}).$$

Здесь одинарный верхний индекс в скобках указывает размерность вектора, а двойной верхний индекс в скобках указывает размер матрицы. Матрицы \hat{P} и \tilde{P} из варианта 1 здесь разбиты на блоки по схеме:

$$P = \begin{bmatrix} P^{(qq)} & P^{(qs)} \\ P^{(sq)} & P^{(ss)} \end{bmatrix}, \quad P = P^T.$$

Вариант 10. Редуцированный стабилизированный ковариационный алгоритм (Бар-Ицхака–Джозефа). Как и в варианте 9, здесь применяется декомпозиция, т. е. разбиение на блоки векторов и матриц, что выделяет

редуцированную часть алгоритма вида (13.12) и специфическую часть, наподобие (13.13). Для инициализации и обработки первых n измерений используется алгоритм варианта 2.

Вариант 11. Редуцированный квадратно-корневой алгоритм Бар-Ицхака–Поттера, с нижнетреугольным разложением, $S \equiv L$. За основу берется алгоритм варианта 9, но в нем редуцированная стандартная часть заменяется на редуцированную часть размера $(q \times q)$. Формулы специфической части заменяются, соответственно, на следующие выражения:

$$K^{(sq)} = \tilde{L}^{(sq)}(\tilde{L}^{(qq)})^{-1},$$

$$\hat{L}^{(sq)} = K^{(sq)}\hat{L}^{(qq)},$$

$$\hat{L}^{(ss)} = \tilde{L}^{(ss)}.$$

Этап экстраполяции выполняется, как в алгоритме Поттера, вариант 3, при этом

$$S \equiv L = \begin{bmatrix} L^{(qq)} & 0 \\ L^{(sq)} & L^{(ss)} \end{bmatrix}$$

и $L^{(qq)}$, $L^{(ss)}$ — нижние треугольные матрицы (с верхним знаком \sim или $\hat{}$). Для инициализации и обработки первых n измерений используется алгоритм варианта 3.

Вариант 12. Редуцированный квадратно-корневой алгоритм (Бар-Ицхака–Бирмана–Медана), с $P = LDL^T$ -разложением по типу алгоритма Бирмана (см. вариант 4).

Здесь используются декомпозиции вида:

$$P = \begin{bmatrix} P^{(qq)} & (P^{(sq)})^T \\ P^{(sq)} & P^{(ss)} \end{bmatrix}, \quad L = \begin{bmatrix} L^{(qq)} & 0 \\ L^{(sq)} & L^{(ss)} \end{bmatrix}, \quad D = \begin{bmatrix} D^q & 0 \\ 0 & D^s \end{bmatrix},$$

где L — нижние треугольные с единичной диагональю матрицы, D — диагональные матрицы. Действует замечание 13.23 после Варианта 9, но для инициализации и обработки первых n наблюдений используется алгоритм варианта 4. Для обработки каждого из остальных измерений применяется следующий алгоритм:

- (1) Выполнить алгоритм из варианта 4, но в применении к векторам размерности q и матрицам размера $(q \times q)$, т.е. по данным z , $a^{(q)}$, $\tilde{L}^{(qq)}$, $\tilde{D}^{(q)}$, $\tilde{x}^{(q)}$ найти $\hat{L}^{(qq)}$, $\hat{D}^{(q)}$, $\hat{x}^{(q)}$.

(2) Вычислить

$$\begin{aligned} K^{(sq)} &= \tilde{L}^{(sq)} (\tilde{L}^{(qq)})^{-1}, \\ \hat{L}^{(sq)} &= K^{(sq)} \hat{L}^{(qq)}, \\ \hat{L}^{(ss)} &= \tilde{L}^{(ss)}, \\ \hat{D}^{(s)} &= \tilde{D}^{(s)}, \\ \hat{x}^{(s)} &= \tilde{x}^{(s)} + K^{(sq)} (\hat{x}^{(q)} - \tilde{x}^{(q)}). \end{aligned}$$

Вариант 13. Редуцированный квадратно-корневой ковариационный алгоритм (Бар-Ицхака–Карлсона). Действует замечание 13.23 после Варианта 9, но для инициализации и обработки первых n измерений используется алгоритм варианта 7. Для обработки остальных измерений применяется следующий алгоритм:

(1) Выполнить алгоритм из варианта 7 (его следует вывести самостоятельно, наподобие алгоритма варианта 6), но в применении к векторам размерности q и матрицам размера $(q \times q)$, т. е. по данным z , $a^{(q)}$, $\tilde{L}^{(qq)}$, $\tilde{x}^{(q)}$ найти $\hat{L}^{(qq)}$ и $\hat{x}^{(q)}$.

(2) Вычислить все остальные матрицы и вектор $\hat{x}^{(s)}$.

Вариант 14. Стандартный информационный алгоритм (см. стр. 245).

(i) Инициализация (начальные значения x_0 , Λ_0):

$$d_0 = \Lambda_0 x_0, \quad \left(\Lambda \quad \vdots \quad d \right) := \left(\Lambda_0 \quad \vdots \quad d_0 \right).$$

(ii) Обработка наблюдений (очередные данные $z = a^T \bar{x} + v$):

$$\left(\Lambda \quad \vdots \quad d \right) := \left(\Lambda \quad \vdots \quad d \right) + a \left(a^T \quad \vdots \quad z \right) / r.$$

(iii) Выдача результата: $\hat{x} = \Lambda^{-1} d$.

В качестве начальных значений рекомендуется взять $x_0 = 0$, $\Lambda_0 = 0$.

Вариант 15. Квадратно-корневой информационный алгоритм. См. подразд. 7.3, стр. 110, формулу (7.7), здесь — ее рекуррентная версия (13.28).

(i) Инициализация (начальные значения \tilde{R}_0 , x_0):

$$\tilde{z}_0 = \tilde{R}_0 x_0; \quad \left[\hat{R}_0 \quad \mid \quad \hat{z}_0 \right] = \left[\tilde{R}_0 \quad \mid \quad \tilde{z}_0 \right].$$

(ii) Обработка наблюдений (очередные скалярные данные $z = a^T \bar{x} + v$):

$$\begin{bmatrix} \hat{R}_j & \hat{z}_j \\ 0 & e \end{bmatrix} = T_j \begin{bmatrix} \hat{R}_{j-1} & \hat{z}_{j-1} \\ a^T & z \end{bmatrix}, \quad j = 1, 2, \dots, m, \quad (13.28)$$

где T_j — ортогональная матрица, которая выбирается так, чтобы привести к верхнетреугольному виду расширенную матрицу $\begin{bmatrix} \hat{R}_{j-1} \\ a^T \end{bmatrix}$, j — номер очередного измерения, все матрицы R здесь имеют размер $(n \times n)$, при этом все R_j , $j \geq 1$, — верхние треугольные.

(iii) Выдача результата: $\hat{x} = \hat{R}_j^{-1} \hat{z}_j$.

Начальные значения рекомендуется взять в виде $x_0 = 0$, $\tilde{R}_0 = 0$. В качестве преобразований T_j возможны четыре процедуры, указанные в описании варианта 3. Таким образом, всего имеем 4 разновидности данного варианта:

T_j	вариант
Хаусхолдер	15.1
Гивенс	15.2
ГШО	15.3
МГШО	15.4

ЗАМЕЧАНИЕ 13.24. В каждом варианте лабораторного проекта №9 необходимо выполнить проверку всех положений для задачи с мультиколлинearностью, которые изложены в подразд. 13.13 на стр. 300–305.

ОРТОГОНАЛИЗОВАННЫЕ БЛОЧНЫЕ АЛГОРИТМЫ

В этом разделе представлены дальнейшие инновации в области вычислительных методов оценивания, основанные на соединении идей факторизации матриц, скаляризации процесса обработки данных и *блочной ортогонализации массивов данных*. В оригинале эти алгоритмы называются *array algorithms* [17]. Использован материал диссертации М. В. Куликовой [51] с систематизацией (переименованиями) некоторых алгоритмов.

14.1 Задача оценивания

Перепишем постановку задачи оценивания, определенную общими выражениями (13.1), с указанием дискретного времени не в скобках, а в индексе.

Пусть дискретная динамическая система описывается уравнениями:

$$x_{t+1} = \Phi_t x_t + B_t u_t + G_t w_t, \quad (14.1)$$

$$z_t = H_t x_t + v_t, \quad (14.2)$$

где x_t — n -мерный вектор состояния системы, z_t — доступный m -мерный вектор измерений, u_t — детерминистский r -мерный вектор (входное управляющее воздействие), $t = 0, 1, \dots$, — дискретное время. Матрицы $\Phi_t \in \mathbb{R}^{n \times n}$, $B_t \in \mathbb{R}^{n \times r}$, $G_t \in \mathbb{R}^{n \times q}$, $H_t \in \mathbb{R}^{m \times n}$, $Q_t \in \mathbb{R}^{q \times q}$, $Q_t \geq 0$ и $R_t \in \mathbb{R}^{m \times m}$, $R_t > 0$ известны, могут быть параметризованы и полностью характеризуют систему (14.1), (14.2). Последовательности $\{w_0, w_1, \dots\}$ и $\{v_1, v_2, \dots\}$ — независимые нормально распределенные последовательности шумов с нулевыми средними значениями и ковариационными матрицами Q_t и R_t , соответственно. Без потери общности считаем, что $\{w_0, w_1, \dots\}$ и $\{v_1, v_2, \dots\}$ не зависят от начального вектора состояния системы x_0 , распределенного

по нормальному закону с математическим ожиданием \bar{x}_0 и ковариацией P_0 , т. е. $x_0 \sim \mathcal{N}(\bar{x}_0, P_0)$.

Задача оценивания заключается в получении оценки неизвестного вектора x_t , $t = 0, 1, \dots$ по доступным наблюдениям $Z_1^N = \{z_t, t = 1, 2, \dots, N\}$, содержащим информацию о векторе x_t . Если $N = t - 1$, оценка \hat{x}_t вектора называется *экстраполяционной оценкой*, точнее, оценкой одношагового предсказания. Если $N = t$, оценка \hat{x}_t вектора называется *отфильтрованной оценкой*. Если $N > t$, оценка \hat{x}_t вектора называется *сглаженной оценкой*. Задачи сглаживания в этом пособии не рассматриваются. Они подробно представлены в [61], где можно видеть, что в составе алгоритмов сглаживания присутствуют оптимальные оценки от фильтра Калмана.

Оптимальные оценки — это те, которые наилучшим образом (в заранее определенном смысле) соответствуют истинному значению вектора x_t . Если *критерий качества оценивателя* J_t определен условным математическим ожиданием как $J_t \triangleq \mathbf{E} \{ \tilde{x}_t^T \tilde{x}_t \mid Z_1^N \} = \mathbf{E} \{ (x_t - \hat{x}_t)^T (x_t - \hat{x}_t) \mid Z_1^N \}$, где $\mathbf{E} \{ \cdot \}$ — оператор математического ожидания, $\tilde{x}_t \triangleq x_t - \hat{x}_t$ — погрешность, то оцениватель \hat{x}_t , минимизирующий критерий J_t , называется *оптимальным в среднеквадратическом смысле оценивателем*.

14.2 Блочные алгоритмы в исторической перспективе

Дискретный *стандартный ковариационный фильтр* (СКФ) Калмана дается уравнениями (13.3). Их вывод широко известен, см. например, [61, 113]. В обозначениях системы (14.1), (14.2) эти уравнения выглядят несколько более обозримо в следующей записи:

- **Этап экстраполяции:** $t = 0, 1, \dots$; $P_0^+ = P_0$,

$$\text{оценка: } \hat{x}_{t+1}^- = \Phi_t \hat{x}_t^+ + B_t u_t, \quad (14.3)$$

$$\text{ковариация ошибки: } P_{t+1}^- = \Phi_t P_t^+ \Phi_t^T + G_t Q_t G_t^T. \quad (14.4)$$

- **Этап обработки измерений (фильтрация):** $t = 1, 2, \dots$,

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1}, \quad (14.5)$$

$$\text{оценка: } \hat{x}_t^+ = \hat{x}_t^- + K_t (z_t - H_t \hat{x}_t^-), \quad (14.6)$$

$$\text{ковариация: } P_t^+ = P_t^- - K_t H_t P_t^-, \quad (14.7)$$

где все данные берутся из модели (14.1), (14.2): Φ_t , G_t , B_t , H_t , Q_t , R_t — известные матрицы-параметры линейной дискретной динамической

системы, u_t — вектор управления, z_t — доступный вектор наблюдений, x_t — вектор состояния системы с начальным значением $x_0 \sim \mathcal{N}(\bar{x}_0, P_0)$.

Соединяя в (14.3)–(14.7) два этапа в один, для $t = 0, 1, \dots$ получим (14.8) и (14.9) — так называемую *предиктивную форму фильтра Калмана*:

$$\text{оценка: } \hat{x}_{t+1}^- = \Phi_t \{ \hat{x}_t^- + P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1} (z_t - H_t \hat{x}_t^-) \} \quad (14.8)$$

с $\{\cdot\} = x_0$ при $t = 0$ в (14.8), поскольку $\{\cdot\}$ в (14.8) выражает оценку (14.6), отфильтрованную в результате обработки измерения, но измерений в момент $t = 0$ еще нет (они начинаются с момента $t = 1$);

$$\text{ковариация: } P_{t+1}^- = G_t Q_t G_t^T + \Phi_t \{ P_t^- - P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1} H_t P_t^- \} \Phi_t^T \quad (14.9)$$

с $\{\cdot\} = P_0$ при $t = 0$ в (14.9), поскольку $\{\cdot\}$ в (14.9) выражает ковариацию (14.7) оценки (14.6), отфильтрованной в результате обработки измерения, но измерений в момент $t = 0$ еще нет (они начинаются с момента $t = 1$).

Уравнение (14.9) есть уравнение Риккати [109] относительно P_{t+1}^- .

ЗАМЕЧАНИЕ 14.1. Видно, что в (14.8) отсутствует слагаемое $B_t u_t$. Оно опущено в силу предположения $u_t \equiv 0$. В любой момент оно может быть добавлено, если $u_t \neq 0$. Такой прием — убрать это слагаемое (для простоты записей) или добавить его (в нужный момент) — можно применять всегда.

Квадратно-корневая реализация уравнений (14.3)–(14.9) с использованием разложений вида $P_t^\pm = S_t^\pm (S_t^\pm)^T$ рассмотрена выше (разд. 13). Там уже возникала необходимость процедуры триангуляризации, т. е. процедуры приведения матрицы S_t^\pm к требуемому треугольному виду. Эту идею (применительно к задаче фильтрации) предложил и разработал Schmidt, S.F. [129] в 1970 г. В частности, Schmidt показал, что уравнение (14.4) для предсказания матрицы ковариации ошибки оценивания на этапе экстраполяции может быть заменено на эквивалентное уравнение

$$\begin{bmatrix} (S_{t+1}^-)^T \\ 0 \end{bmatrix} = T \begin{bmatrix} (S_t^+)^T \\ \tilde{Q}_t^T G_t^T \end{bmatrix}, \quad (14.10)$$

где S_{t+1}^- , S_t^+ — нижние треугольные матрицы в представлениях $P_{t+1}^- = S_{t+1}^- (S_{t+1}^-)^T$, $P_t^+ = S_t^+ (S_t^+)^T$, $Q_t = \tilde{Q}_t \tilde{Q}_t^T$. Он также разработал алгоритм построения ортогонального преобразования T , приводящего к требуемому треугольному виду матрицу, стоящую в правой части формулы (14.10). Эта процедура известна как процедура ортогонализации Грама–Шмидта.

В 1971 г. Kaminski, P.G. предложил новые модификации ковариационных и информационных типов квадратно-корневых методов фильтрации,

обладающие рядом преимуществ перед ранее известными алгоритмами [103]. Остановимся более подробно лишь на одном из них, который потребуется в дальнейшем. Исходя из хорошо известной дуальности ковариационных и информационных алгоритмов, Kaminski разработал новую схему [103]

$$\begin{bmatrix} R_t^{1/2} & H_t S_t^- \\ 0 & S_t^- \end{bmatrix} = T \begin{bmatrix} R_{e,t}^{1/2} & 0 \\ \tilde{K}_t & S_t^+ \end{bmatrix} \quad (14.11)$$

для осуществления этапа обработки измерений и фильтрации, где $R_{e,t}^{1/2}$ — квадратный корень матрицы $R_{e,t}$ ковариации невязки измерений $e_t = z_t - H\hat{x}_t^-$, $\tilde{K}_t = P_t^- H_t^T R_{e,t}^{-T/2}$ и где T — ортогональное преобразование, приводящее к нижнему треугольному виду матрицу, стоящую в правой части формулы (14.11). Он также предложил использовать в качестве алгоритма экстраполяции формулу (14.10), которую ввел Schmidt. Так возникла еще одна эффективная реализация алгоритма калмановской фильтрации.

Вслед за этими нововведениями в 1975 г. Morf, М. и Kailath, Т. разработали так называемые *быстрые квадратно-корневые алгоритмы фильтрации* для систем, матрицы-параметры Φ , G , B , H , Q , R которых не зависят от времени. Их основная идея — создание нового способа вычислений, основанного на работе с матрицами $\delta P_t^\pm = P_{t+1}^\pm - P_t^\pm$ вместо P_t^\pm . Это значительно улучшило численные характеристики квадратно-корневых алгоритмов для подобных систем, т. е. сократило общее количество арифметических действий на вычислениях и, следовательно, время работы [119]. Кроме того, ими была предложена идея комбинирования алгоритмов, которые разработал Schmidt для этапа экстраполяции фильтра Калмана (14.10), и алгоритмов, которые предложил Kaminski для этапа обработки измерений (14.11), с целью получения новой, более эффективной реализации фильтра в виде

$$\begin{bmatrix} R_{e,t}^{1/2} & \bar{K}_{p,t}^T \\ 0 & (P_{t+1}^-)^{1/2} \\ 0 & 0 \end{bmatrix} = T \begin{bmatrix} R_t^{1/2} & 0 \\ (P_t^-)^{1/2} H_t^T & (P_t^-)^{1/2} \Phi_t^T \\ 0 & Q_t^{1/2} G_t^T \end{bmatrix}, \quad (14.12)$$

где T — любое ортогональное преобразование, приводящее к верхнему треугольному виду матрицу, стоящую в правой части формулы (14.12) и $\bar{K}_{p,t} \triangleq \Phi_t \tilde{K}_t$. Уравнение (14.9) алгебраически эквивалентно формуле (14.12). Алгоритм (14.12) в настоящее время известен как *квадратно-корневой ковариационный фильтр Калмана* — КККФ.

14.3 Расширенный квадратно-корневой ковариационный фильтр

Для удобства введем следующие обозначения. Когда некоторая матрица $A > 0$, будем рассматривать для нее разложение Холесского вида: $A = A^{T/2} A^{1/2}$, где $A^{1/2}$ — верхняя треугольная матрица, являющаяся квадратным корнем из A . Тогда $A^{T/2} = (A^{1/2})^T$, $A^{-1/2} = (A^{1/2})^{-1}$ и $A^{-T/2} = (A^{-1/2})^T$. Для величин, вычисляемых в фильтре Калмана, примем обозначения: $K_{p,t} \triangleq \Phi_t \tilde{P}_t H_t^T R_{e,t}^{-1}$, $\bar{K}_{p,t} \triangleq \Phi_t \tilde{P}_t H_t^T R_{e,t}^{-1/2}$, \bar{e}_t — нормализованные невязки фильтра Калмана, т. е. $\bar{e}_t \triangleq R_{e,t}^{-T/2} e$, где $e \triangleq z_t - H \hat{x}_t^-$ — невязка измерений фильтра в момент времени t , характеризующаяся ковариационной матрицей $\mathbf{E}\{e_t e_t^T\} \triangleq R_{e,t}$, $R_{e,t} \triangleq H_t \tilde{P}_t H_t^T + R_t$. Кроме того, \hat{x}_t^- , \hat{x}_t^+ — предсказанная и отфильтрованная оценки вектора состояния системы (14.1), (14.2), соответственно; \tilde{P}_t , \hat{P}_t — матрицы ковариации ошибки предсказанной и отфильтрованной оценок, соответственно.

Следуя цели данного раздела, рассмотрим так называемый *расширенный квадратно-корневой ковариационный фильтр* (РКККФ), впервые предложенный в [123].

Алгоритм РКККФ

По данным $\hat{x}_0^- = \bar{x}_0$, $\tilde{P}_0 = P_0$ в каждый момент времени t , $t = 1, 2, \dots$, вычисляют:

$$\begin{bmatrix} R_{e,t}^{1/2} & \bar{K}_{p,t}^T \\ 0 & \tilde{P}_{t+1}^{1/2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\bar{e}_t \\ \tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \\ \gamma \end{bmatrix} = O_t \begin{bmatrix} R_t^{1/2} & 0 \\ \tilde{P}_t^{1/2} H_t^T & \tilde{P}_t^{1/2} \Phi_t^T \\ 0 & Q_t^{1/2} G_t^T \end{bmatrix} \begin{bmatrix} -R_t^{-T/2} z_t \\ \tilde{P}_t^{-T/2} \hat{x}_t^- \\ 0 \end{bmatrix}, \quad (14.13)$$

где O_t — матрица ортогонального преобразования, которая при умножении слева на матрицу в правой части формулы (14.13) приводит ее к блочному верхнему треугольному виду. Кроме того, $\gamma = -\tilde{Q}_t^{-1/2} K_{b,t} \hat{x}_{t+1}^-$, где $K_{b,t} = Q_t G_t \tilde{P}_{t+1}^{-1}$ и $\tilde{Q}_t = Q_t - Q_t G_t^T \tilde{P}_{t+1}^{-1} G_t Q_t$.

Теоретическое обоснование приведенного алгоритма можно найти в [123]. Отметим, что уравнения (14.8), (14.9) стандартного фильтра Калмана (СКФ) алгебраически эквивалентны формуле (14.13). Данный РКККФ есть не что иное как «расширение» стандартного квадратно-корневого ковариационного алгоритма фильтрации (см. формулу (14.12)) посредством добавления еще одного столбца — столбца данных:

$$\begin{bmatrix} -z_t^T R_t^{-1/2}, & (\hat{x}_t^-)^T \tilde{P}_t^{-1/2}, & 0 \end{bmatrix}^T. \quad (14.14)$$

При всей своей простоте, подобная модификация алгоритма обладает рядом преимуществ перед ранее известными ковариационными реализациями дискретного фильтра Калмана [123]:

- 1° Для отыскания предсказанной оценки вектора состояния системы \hat{x}_{t+1}^- с помощью стандартного квадратно-корневого ковариационного алгоритма фильтрации (СКККФ) (см. формулу (14.12)) существует необходимость проводить дополнительные вычисления, а именно: по доступным из (14.12) $\bar{K}_{p,t}$, $R_{e,t}^{1/2}$ находить $\hat{x}_{t+1}^- = \Phi_t \hat{x}_t^- + \bar{K}_{p,t} \left(R_{e,t}^{1/2} \right)^{-1} (z_t - H_t \hat{x}_t^-)$. Последнее, в свою очередь, требует обращения верхней треугольной матрицы $R_{e,t}^{1/2}$ размера $(m \times m)$. Теперь — при использовании РКККФ эта необходимость исчезает. По найденным из (14.13) величинам $\tilde{P}_{t+1}^{1/2}$ и $\tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^-$ сразу находим $\hat{x}_{t+1}^- = \tilde{P}_{t+1}^{T/2} \left(\tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \right)$.
- 2° То, что отыскание \hat{x}_{t+1}^- включено в алгоритм без дополнительных вычислений и что все данные, необходимые для продолжения работы фильтра, оказываются найдены одновременно и независимо друг от друга, делает РКККФ (14.13) более приспособленным к параллельным вычислениям, чем ранее известные методы.
- 3° Для каждого момента времени t единственной матрицей, для которой требуется вычисление обратной, является верхняя треугольная матрица $R_t^{1/2}$. (В случае, если $P_0 \neq I_n$, где I_n — единичная матрица размера $(n \times n)$, также необходимо вычислить $P_0^{-1/2}$). Таким образом, в случае $P_0 = I_n$ работоспособность РКККФ в каждый момент времени t существенно зависит от свойств обрабатываемой матрицы $R_t^{1/2}$.

Следуя работе [123], выделим в РКККФ (14.13) два этапа:

• **Этап обработки измерений (фильтрация):**

$$\left[\begin{array}{cc|c} R_{e,t}^{1/2} & \bar{K}_{p,t}^T & -\bar{e}_t \\ 0 & \hat{P}_t^{1/2} \Phi_t^T & \hat{P}_t^{-T/2} \hat{x}_t^+ \end{array} \right] = O_{t,1} \left[\begin{array}{cc|c} R_t^{1/2} & 0 & -R_t^{-T/2} z_t \\ \tilde{P}_t^{1/2} H_t^T & \tilde{P}_t^{1/2} \Phi_t^T & \tilde{P}_t^{-T/2} \hat{x}_t^- \end{array} \right], \quad (14.15)$$

где $O_{t,1}$ — ортогональное преобразование, приводящее к блочному верхнему треугольному виду первых два (блочных) столбца матрицы, стоящей в правой части формулы (14.15). Можно видеть, что формула (14.15) есть не что иное как «расширенное» уравнение (14.11), которое предложил Kaminski.

• **Этап экстраполяции:**

$$\left[\begin{array}{c|c} \tilde{P}_{t+1}^{1/2} & \tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \\ \hline 0 & -\tilde{Q}_t^{-T/2} K_{b,t} \hat{x}_{t+1} \end{array} \right] = O_{t,2} \left[\begin{array}{c|c} \hat{P}_t^{1/2} \Phi_t^T & \hat{P}_t^{-T/2} \hat{x}_t^+ \\ \hline Q_t^{1/2} G_t^T & 0 \end{array} \right], \quad (14.16)$$

где $O_{t,2}$ — ортогональное преобразование, приводящее к верхнему треугольному виду первый (блочный) столбец матрицы, стоящей в правой части формулы (14.16). Также видно, что формула (14.16) есть не что иное как «расширенное» уравнение (14.10), которое предложил Schmidt.

14.4 Расширенный квадратно-корневой информационный фильтр

Как видно из подразд. 11.6, информационные алгоритмы могут стартовать в условиях очень скудной априорной информации, — возможно, даже нулевой, когда начальная информационная матрица $\Lambda_0 = 0$. Говоря строго, применение ковариационных фильтров при $\Lambda_0 = 0$ невозможно, так как начальная ковариационная матрица, равная $P_0 = \Lambda_0^{-1}$, не имеет конечного значения при запуске алгоритма. Алгоритмы информационного типа помогают избежать таких осложнений на начальном участке фильтрации. В этом случае задают $\Lambda_0 = \varepsilon^2 I$, ε — очень малое число, возможно, 0.

Рассмотрим *расширенный квадратно-корневой информационный алгоритм фильтрации* (РККИФ), предложенный в [123].

Алгоритм РККИФ

Предполагая, что матрицы $P_0 > 0$, $R_t > 0$, по данным $\hat{x}_0^- = \bar{x}_0$, $\tilde{P}_0 = P_0$ в каждый момент времени t , $t = 1, 2, \dots$, вычисляют:

$$\left[\begin{array}{ccc|c} R_{e,t}^{-T/2} & 0 & 0 & -\bar{e}_t \\ -\tilde{P}_{t+1}^{-T/2} K_{p,t} & \tilde{P}_{t+1}^{-T/2} & 0 & \tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \\ * & * & * & * \end{array} \right] =$$

$$= O_t \left[\begin{array}{ccc|c} R_t^{-T/2} & -R_t^{-T/2} H_t \Phi_t^{-1} & R_t^{-T/2} H_t \Phi_t^{-1} G_t Q_t^{T/2} & -R_t^{-T/2} z_t \\ 0 & \tilde{P}_t^{-T/2} \Phi_t^{-1} & -\tilde{P}_t^{-T/2} \Phi_t^{-1} G_t Q_t^{T/2} & \tilde{P}_t^{-T/2} \hat{x}_t^- \\ 0 & 0 & I_q & 0 \end{array} \right], \quad (14.17)$$

где O_t — то же самое ортогональное преобразование, что и в СКККФ (14.12), или любое иное, приводящее к нижнему треугольному

виду первые три (блочных) столбца матрицы, стоящей в правой части формулы (14.17), и I_q — единичная матрица размера $(q \times q)$.

ЗАМЕЧАНИЕ 14.2. Требовать обратимость матрицы Φ_t излишне, так как известно, что переходная матрица состояния линейной динамической системы всегда невырождена [75].

Отметим, что РККИФ (14.17) получен так же, как и РКККФ, т. е. посредством добавления столбца данных (14.14) к квадратно-корневому информационному алгоритму фильтрации, т. е.

$$\begin{aligned} & \begin{bmatrix} R_{e,t}^{-T/2} & 0 & 0 \\ -\tilde{P}_{t+1}^{-T/2} K_{p,t} & \tilde{P}_{t+1}^{-T/2} & 0 \\ * & * & * \end{bmatrix} = \\ & = O_t \begin{bmatrix} R_t^{-T/2} & -R_t^{-T/2} H_t \Phi_t^{-1} & R_t^{-T/2} H_t \Phi_t^{-1} G_t Q_t^{T/2} \\ 0 & \tilde{P}_t^{-T/2} \Phi_t^{-1} & -\tilde{P}_t^{-T/2} \Phi_t^{-1} G_t Q_t^{T/2} \\ 0 & 0 & I_q \end{bmatrix}. \end{aligned}$$

Подобная модификация обладает рядом преимуществ перед ранее известными информационными типами реализаций фильтра Калмана [123]:

- 1° Поскольку теперь величины $\tilde{P}_{t+1}^{-T/2}$ и $\left(\tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^{-}\right)$ доступны непосредственно из алгоритма (см. РККИФ, формула (14.17)), предсказанная оценка вектора состояния \hat{x}_{t+1}^{-} может быть найдена с помощью решения линейной треугольной системы вида: $\left(\tilde{P}_{t+1}^{-T/2}\right) \left(\hat{x}_{t+1}^{-}\right) = \left(\tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^{-}\right)$.
- 2° Поскольку все необходимые для продолжения работы фильтра данные могут быть найдены одновременно и независимо друг от друга, это свойство РККИФ (14.17) делает его более приспособленным к параллельным вычислениям, чем ранее известные методы.

Алгоритм (14.17) также может быть разделен на два этапа [123]:

- **Этап обработки измерений (фильтрация):**

$$\begin{bmatrix} 0 & -\bar{e}_t \\ \hat{P}_t^{-T/2} & \hat{P}_t^{-T/2} \hat{x}_t^{+} \end{bmatrix} = O_{t,1} \begin{bmatrix} -R_t^{-T/2} H_t & -R_t^{-T/2} z_t \\ \tilde{P}_t^{-T/2} & \tilde{P}_t^{-T/2} \hat{x}_t^{-} \end{bmatrix}, \quad (14.18)$$

где $O_{t,1}$ — ортогональное преобразование, приводящее к нижнему треугольному виду первый (блочный) столбец матрицы, стоящей в правой части (14.18).

• **Этап экстраполяции:**

$$\begin{aligned} & \left[\begin{array}{cc|c} \tilde{P}_{t+1}^{-T/2} & 0 & \tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \\ -\tilde{Q}_t^{-T/2} K_{b,t} & \tilde{Q}_t^{-T/2} & -\tilde{Q}_t^{-T/2} K_{b,t} \hat{x}_{t+1}^- \end{array} \right] = \\ & = O_{t,2} \left[\begin{array}{cc|c} \hat{P}_t^{-T/2} \Phi_t^{-1} & -\hat{P}_t^{-T/2} \Phi_t^{-1} G_t & \hat{P}_t^{-T/2} \hat{x}_t^+ \\ 0 & Q_t^{-T/2} & 0 \end{array} \right], \end{aligned} \quad (14.19)$$

где $O_{t,2}$ — ортогональное преобразование, приводящее к блочному нижнему треугольному виду два первых (блочных) столбца матрицы, стоящей в правой части формулы (14.19).

14.5 Модифицированный квадратно-корневой информационный фильтр

Модифицируем изложенные ранее РКККФ (14.13) и РККИФ (14.17), соединяя их в одно целое на основе РККИФ (14.17). Полученный результат имеет название *модифицированный квадратно-корневой информационный фильтр* (МККИФ), и его теоретическое обоснование можно найти в [123].

Алгоритм МККИФ

Предполагая, что $P_0 > 0$, $R_t > 0$, по данным $\hat{x}_0^- = \bar{x}_0$, $\tilde{P}_0 = P_0$ в каждый момент времени t , $t = 1, 2, \dots$, вычисляют:

$$\begin{aligned} & \left[\begin{array}{ccc|c} R_{e,t}^{-T/2} & 0 & 0 & \bar{K}_{p,t}^T \\ -\tilde{P}_{t+1}^{-T/2} K_{p,t} & \tilde{P}_{t+1}^{-T/2} & 0 & \tilde{P}_{t+1}^{1/2} \\ * & * & * & 0 \end{array} \middle| \begin{array}{c} -\bar{e}_t \\ \tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \\ * \end{array} \right] = \\ & = O_t \left[\begin{array}{ccc|c} R_t^{-T/2} & -R_t^{-T/2} H_t \Phi_t^{-1} & R_t^{-T/2} H_t \Phi_t^{-1} G_t Q_t^{T/2} & 0 \\ 0 & \tilde{P}_t^{-T/2} \Phi_t^{-1} & -\tilde{P}_t^{-T/2} \Phi_t^{-1} G_t Q_t^{T/2} & \tilde{P}_t^{1/2} \Phi_t^T \\ 0 & 0 & I_q & Q_t^{1/2} G_t^T \end{array} \middle| \begin{array}{c} -R_t^{-T/2} z_t \\ \tilde{P}_t^{-T/2} \hat{x}_t^- \\ 0 \end{array} \right], \end{aligned} \quad (14.20)$$

где O_t — любое ортогональное преобразование, которое приводит к нижнему треугольному виду главный блок (т.е. первые три (блочных) столбца) матрицы, стоящей в правой части (14.20).

Подобная модификация обладает рядом преимуществ перед изложенным выше РККИФ и ранее известными информационными реализациями фильтра [123], а именно:

1° Поскольку теперь величина $\tilde{P}_{t+1}^{1/2}$ доступна непосредственно из алгоритма (см. МККИФ, формула (14.20)), предсказанная оценка вектора состояния системы \hat{x}_{t+1}^- может быть найдена непосредственно: $\tilde{P}_{t+1}^{T/2}(\tilde{P}_{t+1}^{-T/2}\hat{x}_{t+1}^-) = \hat{x}_{t+1}^-$. Таким образом, в отличие от РККИФ (14.17), алгоритм (14.20) позволяет избежать решения линейной системы для нахождения оценки \hat{x}_{t+1}^- .

2° Все данные, необходимые для продолжения работы фильтра, могут быть найдены одновременно и независимо друг от друга. Это свойство МККИФ (14.20) делает его более приспособленным к параллельным вычислениям, чем предшествующие известные методы.

МККИФ (14.20) может быть также разделен на два этапа следующим образом [123]:

• **Этап обработки измерений (фильтрация):**

$$\left[\begin{array}{c|c|c} 0 & \bar{K}_{p,t}^T & -\bar{e}_t \\ \hline \hat{P}_t^{-T/2} & \hat{P}_t^{1/2}\Phi_t^T & \hat{P}_t^{-T/2}\hat{x}_t^+ \end{array} \right] = O_{t,1} \left[\begin{array}{c|c|c} -R_t^{-T/2}H_t & 0 & -R_t^{-T/2}z_t \\ \hline \tilde{P}_t^{-T/2} & \tilde{P}_t^{1/2}\Phi_t^T & \tilde{P}_t^{-T/2}\hat{x}_t^- \end{array} \right], \quad (14.21)$$

где $O_{t,1}$ — ортогональное преобразование, приводящее к нижнему треугольному виду первый (блочный) столбец матрицы, стоящей в правой части (14.21).

• **Этап экстраполяции:**

$$\begin{aligned} & \left[\begin{array}{c|c|c} \tilde{P}_{t+1}^{-T/2} & 0 & \tilde{P}_{t+1}^{1/2} \\ \hline -\tilde{Q}_t^{-T/2}K_{b,t} & \tilde{Q}_t^{-T/2} & -\tilde{Q}_t^{-T/2}K_{b,t}\hat{x}_{t+1}^- \end{array} \right] = \\ & = O_{t,2} \left[\begin{array}{c|c|c} \hat{P}_t^{-T/2}\Phi_t^{-1} & -\hat{P}_t^{-T/2}\Phi_t^{-1}G_t & \hat{P}_t^{1/2}\Phi_t^T \\ \hline 0 & Q_t^{-T/2} & Q_t^{1/2}G_t^T \end{array} \right] \left[\begin{array}{c|c} \hat{P}_t^{-T/2}\hat{x}_t^+ \\ \hline 0 \end{array} \right], \end{aligned} \quad (14.22)$$

где $O_{t,2}$ — ортогональное преобразование, приводящее либо к блочному нижнему треугольному виду два первых (блочных) столбца, либо к верхнему треугольному виду третий (блочный) столбец матрицы, стоящей в правой части формулы (14.22).

Дальнейшие усилия разработчиков блочных ортогонализированных алгоритмов были направлены на комбинирование преимуществ схем двух взаимно инверсных типов — ковариационной и информационной. Это выразилось в следующем алгоритме.

14.6 Комбинированный квадратно-корневой фильтр

Алгоритм КоККФ

Предполагая, что $P_0 > 0$, $R_t > 0$, по данным $\hat{x}_0^- = \bar{x}_0$, $\tilde{P}_0 = P_0$ в каждый момент времени t , $t = 1, 2, \dots$, вычисляют:

$$\begin{bmatrix} R_{e,t}^{1/2} & \bar{K}_{p,t}^T & 0 & 0 & -\bar{e}_t \\ 0 & \tilde{P}_{t+1}^{1/2} & \tilde{P}_{t+1}^{-T/2} & 0 & \tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \\ 0 & 0 & -\tilde{Q}_t^{-T/2} K_{b,t} & \tilde{Q}_t^{-T/2} & -\tilde{Q}_t^{-T/2} K_{b,t} \hat{x}_{t+1}^- \end{bmatrix} =$$

$$= O_t \begin{bmatrix} R_t^{1/2} & 0 & -R_t^{-T/2} H_t \Phi_t^{-1} & R_t^{-T/2} H_t \Phi_t^{-1} G_t Q_t^{T/2} & -R_t^{-T/2} z_t \\ \tilde{P}_t^{1/2} H_t^T & \tilde{P}_t^{1/2} \Phi_t^T & \tilde{P}_t^{-T/2} \Phi_t^{-1} & -\tilde{P}_t^{-T/2} \Phi_t^{-1} G_t Q_t^{T/2} & \tilde{P}_t^{-T/2} \hat{x}_t^- \\ 0 & Q_t^{1/2} G_t^T & 0 & I_q & 0 \end{bmatrix}, \quad (14.23)$$

где O_t — ортогональное преобразование, приводящее либо к верхнему треугольному виду первые два блочных столбца, либо к нижнему треугольному виду третий и четвертый блочные столбцы матрицы, стоящей в правой части формулы (14.23).

Как и ранее, КоККФ (14.23) может быть разделен на два этапа [123]:

- **Этап обработки измерений (фильтрация):**

$$\begin{bmatrix} R_{e,t}^{1/2} & \bar{K}_{p,t}^T & 0 & -\bar{e}_t \\ 0 & \hat{P}_t^{1/2} \Phi_t^T & \hat{P}_t^{-T/2} & \hat{P}_t^{-T/2} \hat{x}_t^+ \end{bmatrix} =$$

$$= O_{t,1} \begin{bmatrix} R_t^{1/2} & 0 & -R_t^{-T/2} H_t & -R_t^{-T/2} z_t \\ \tilde{P}_t^{1/2} H_t^T & \tilde{P}_t^{1/2} \Phi_t^T & \tilde{P}_t^{-T/2} & \tilde{P}_t^{-T/2} \hat{x}_t^- \end{bmatrix}, \quad (14.24)$$

где $O_{t,1}$ — ортогональное преобразование, приводящее либо к верхнему треугольному виду первых два (блочных) столбца, либо к нижнему треугольному виду третий (блочный) столбец матрицы, стоящей в правой части (14.24).

- **Этап экстраполяции:**

$$\begin{bmatrix} \tilde{P}_{t+1}^{-T/2} & 0 & \tilde{P}_{t+1}^{1/2} & \tilde{P}_{t+1}^{-T/2} \hat{x}_{t+1}^- \\ -\tilde{Q}_t^{-T/2} K_{b,t} & \tilde{Q}_t^{-T/2} & 0 & -\tilde{Q}_t^{-T/2} K_{b,t} \hat{x}_{t+1}^- \end{bmatrix} =$$

$$= O_{t,2} \begin{bmatrix} \hat{P}_t^{-T/2} \Phi_t^{-1} & -\hat{P}_t^{-T/2} \Phi_t^{-1} G_t & \hat{P}_t^{1/2} \Phi_t^T & \hat{P}_t^{-T/2} \hat{x}_t^+ \\ 0 & Q_t^{-T/2} & Q_t^{1/2} G_t^T & 0 \end{bmatrix}, \quad (14.25)$$

где $O_{t,2}$ — ортогональное преобразование, приводящее либо к нижнему треугольному виду первых два (блочных) столбца, либо к верхнему треугольному виду третий (блочный) столбец матрицы, стоящей в правой части (14.25).

Таким образом, комбинированный квадратно-корневой алгоритм фильтрации (КоККФ) представляет собой полную комбинацию изложенных ранее алгоритмов РКККФ (14.13) и РККИФ (14.20) и получен путем их соединения в одно целое [123].

14.7 Скаляризованный квадратно-корневой ковариационный фильтр

Новые блочные алгоритмы рекуррентного оценивания выведены в диссертации [51] из алгоритмов РКККФ (14.15), (14.16), РККИФ (14.18), (14.19), МККИФ (14.21), (14.22) и КоККФ (14.24), (14.25) с переводом в них этапа обработки измерений в скаляризованную форму. Это дало основание уточнить названия этих алгоритмов, а именно: определяющий термин «скаляризованный» здесь поставлен на первое место. В остальном эти алгоритмы приведены в подразд. 14.7, 14.8, 14.9 и 14.10 в точном соответствии с работой [51].

Используем следующие обозначения: $\left(h_t^{(k)}\right)^T$ — k -я строка матрицы H_t , $z_t^{(k)}$ — k -й элемент вектора наблюдений z_t , $\left(\sigma_t^{(k)}\right)^2$ — k -й диагональный элемент матрицы R_t . Кроме того, $\left(r_{e,t}^{(k+1)}\right)^2 = \left(h_t^{(k+1)}\right)^T \tilde{P}_t^{(k)} h_t^{(k+1)} + \left(\sigma_t^{(k+1)}\right)^2$, $e_t^{(k+1)} = z_t^{(k+1)} - \left(h_t^{(k+1)}\right)^T \tilde{x}_t^{(k)}$, $\bar{e}_t^{(k)} = e_t^{(k)} / r_{e,t}^{(k)}$ и $\bar{K}_{p,t}^{(k+1)} = \Phi_t \tilde{P}_t^{(k)} h_t^{(k+1)} / r_{e,t}^{(k+1)}$.

Алгоритм СКККФ

I. Этап обработки измерений (фильтрация):

- Положить: $S_t^{(0)} = \tilde{P}_t^{1/2}$, $x_t^{(0)} = \hat{x}_t^-$.

- Для $k = 0, 1, \dots, m-1$ вычислять:

$$\begin{aligned} & \left[\begin{array}{c|c} r_{e,t}^{(k+1)} & \left(\bar{K}_{p,t}^{(k+1)} \right)^T \\ \hline 0 & S_t^{(k+1)} \Phi_t^T \end{array} \middle| \begin{array}{c} -\bar{e}_t^{(k+1)} \\ \left(S_t^{(k+1)} \right)^{-T} x_t^{(k+1)} \end{array} \right] = \\ & = O_{t,1}^{(k)} \left[\begin{array}{c|c} \sigma_t^{(k+1)} & 0 \\ \hline S_t^{(k)} h_t^{(k+1)} & S_t^{(k)} \Phi_t^T \end{array} \middle| \begin{array}{c} -z_t^{(k+1)} / \sigma_t^{(k+1)} \\ \left(S_t^{(k)} \right)^{-T} x_t^{(k)} \end{array} \right], \end{aligned} \quad (14.26)$$

где $O_{t,1}^{(k)}$ — ортогональное преобразование, приводящее к верхнему треугольному виду два первых (блочных) столбца матрицы, стоящей в правой части (14.26).

- Положить: $\hat{P}_t^{1/2} \Phi_t^T = S_t^{(m)} \Phi_t^T$ и $\hat{P}_t^{-T/2} \hat{x}_t^+ = \left(S_t^{(m)} \right)^{-T} x_t^{(m)}$.

II. Этап экстраполяции: совпадает с этапом экстраполяции базового метода, т. е. РКККФ, — см. формулу (14.16).

14.8 Скаляризованный квадратно-корневой информационный фильтр

Алгоритм СККИФ

I. Этап обработки измерений (фильтрация):

- Положить: $S_t^{(0)} = \tilde{P}_t^{-T/2}$, $x_t^{(0)} = \hat{x}_t^-$.
- Для $k = 0, 1, \dots, m-1$ вычислять:

$$\left[\begin{array}{c|c} 0 & -\bar{e}_t^{(k+1)} \\ \hline S_t^{(k+1)} & S_t^{(k+1)} x_t^{(k+1)} \end{array} \right] = O_{t,1}^{(k)} \left[\begin{array}{c|c} -\frac{\left(h_t^{(k+1)} \right)^T}{\sigma_t^{(k+1)}} & -\frac{z_t^{(k+1)}}{\sigma_t^{(k+1)}} \\ \hline S_t^{(k)} & S_t^{(k)} x_t^{(k)} \end{array} \right], \quad (14.27)$$

где $O_{t,1}^{(k)}$ — матрица ортогонального преобразования, которая при домножении слева на первый блочный столбец матрицы, стоящей в правой части формулы (14.27), приводит его к нижнему треугольному виду.

- Положить: $\hat{P}_t^{-T/2} = S_t^{(m)}$ и $\hat{P}_t^{-T/2} \hat{x}_t^+ = S_t^{(m)} x_t^{(m)}$.

II. Этап экстраполяции: совпадает с этапом экстраполяции базового метода, т. е. РККИФ см. формулу (14.19).

14.9 Скаляризованный модифицированный квадратно-корневой информационный фильтр

Алгоритм СМККИФ

I. Этап обработки измерений (фильтрация):

- Положить: $S_t^{(0)} = \tilde{P}_t^{-T/2}$, $x_t^{(0)} = \hat{x}_t^-$.
- Для $k = 0, 1, \dots, m-1$ вычислять:

$$\begin{aligned} & \left[\begin{array}{c|c|c} 0 & \left(\bar{K}_{p,t}^{(k+1)} \right)^T & -\bar{e}_t^{(k+1)} \\ S_t^{(k+1)} & \left(S_t^{(k+1)} \right)^{-T} \Phi_t^T & S_t^{(k+1)} x_t^{(k+1)} \end{array} \right] = \\ & = O_{t,1}^{(k)} \left[\begin{array}{c|c|c} -\left(h_t^{(k+1)} \right)^T / \sigma_t^{(k+1)} & 0 & -z_t^{(k+1)} / \sigma_t^{(k+1)} \\ S_t^{(k)} & \left(S_t^{(k)} \right)^{-T} \Phi_t^T & S_t^{(k)} x_t^{(k)} \end{array} \right], \end{aligned} \quad (14.28)$$

где $O_{t,1}^{(k)}$ — ортогональное преобразование, которое приводит либо к нижнему треугольному виду первый (блочный) столбец, либо к верхнему треугольному виду второй (блочный) столбец матрицы, стоящей в правой части формулы (14.28).

- Положить: $\hat{P}_t^{-T/2} = S_t^{(m)}$ и $\hat{P}_t^{-T/2} \hat{x}_t^+ = S_t^{(m)} x_t^{(m)}$.

II. Этап экстраполяции: совпадает с этапом экстраполяции базового метода, т. е. МККИФ, см. формулу (14.22).

14.10 Скаляризованный комбинированный квадратно-корневой фильтр

Алгоритм СКоККФ

I. Этап обработки измерений (фильтрация):

- Положить: $S_t^{(0)} = \tilde{P}_t^{1/2}$, $x_t^{(0)} = \hat{x}_t^-$.

- Для $k = 0, 1, \dots, m-1$ вычислять:

$$\begin{aligned} & \left[\begin{array}{cc|c} r_{e,t}^{(k+1)} & \left(\bar{K}_{p,t}^{(k+1)} \right)^T & 0 \\ 0 & S_t^{(k+1)} \Phi_t^T & \left(S_t^{(k+1)} \right)^{-T} \end{array} \left| \begin{array}{c} -\bar{e}_t^{(k+1)} \\ \left(S_t^{(k+1)} \right)^{-T} x_t^{(k+1)} \end{array} \right. \right] = \\ & = O_{t,1}^{(k)} \left[\begin{array}{cc|c} \sigma_t^{(k+1)} & 0 & -\left(h_t^{(k+1)} \right)^T / \sigma_t^{(k+1)} \\ S_t^{(k)} h_t^{(k+1)} & S_t^{(k)} \Phi_t^T & \left(S_t^{(k)} \right)^{-T} \end{array} \left| \begin{array}{c} -z_t^{(k+1)} / \sigma_t^{(k+1)} \\ \left(S_t^{(k)} \right)^{-T} x_t^{(k)} \end{array} \right. \right], \end{aligned} \quad (14.29)$$

где $O_{t,1}^{(k)}$ — ортогональное преобразование, приводящее либо к верхнему треугольному виду первых два (блочных) столбца, либо к нижнему треугольному виду третий (блочный) столбец матрицы, стоящей в правой части формулы (14.29).

- Положить: $\hat{P}_t^{1/2} = S_t^{(m)}$ и $\hat{x}_t^+ = x_t^{(m)}$.

II. Этап экстраполяции: совпадает с этапом экстраполяции базового метода, т. е. КоККФ, см. формулу (14.25).

Теоретические обоснования четырех алгоритмов, приведенных выше в подразд. 14.7, 14.8, 14.9 и 14.10, помещены в приложение А.1 (стр. 339) в авторской редакции М. В. Куликовой [51].

14.11 Задание на лабораторный проект № 10

Введение

В данном лабораторном проекте мы предлагаем к изучению изложенные выше ортогонализованные блочные алгоритмы калмановской фильтрации на материале реальной прикладной задачи. В качестве такой задачи возьмем задачу получения оценок высоты и вертикальной скорости летательного аппарата (ЛА) по показаниям двух приборов: инерциального датчика вертикального ускорения $a_y(t)$ и барометрического датчика высоты, который обладает собственной инерционностью в своих показаниях $h(t)$, характеризующейся постоянной времени τ .

Введем обозначения физических величин, участвующих в математической модели движения объекта (ЛА) по высоте и в модели наблюдения за

этим движением. В соответствии с принципиальной моделью — вторым законом Ньютона, движение центра масс объекта характеризуем тремя величинами:

- 1) $y = y(t)$ — вертикальная координата (высота над Землей),
- 2) $v_y = v_y(t)$ — вертикальная скорость,
- 3) $a_y = a_y(t)$ — вертикальное ускорение.

Будем рассматривать случай движения с постоянной силой тяги двигателя ЛА. В этом случае следует считать, что ускорение ЛА тоже постоянно (пренебрегая изменением массы ЛА), однако, оно неизвестно, то есть подлежит оцениванию по показаниям приборов. В роли оценивателя должен быть использован фильтр Калмана, который, как известно, оптимален, если обе модели — модель состояния и модель наблюдения — линейные.

Прочитанные показания приборов (измеренные величины) обозначим:

- 1) $z_a = z_a(t)$ — *измеренное* ускорение $a(t)$,
- 2) $z_h = z_h(t)$ — *измеренная* барометрическая высота $h(t)$.

Считывание этих показаний происходит в дискретные моменты времени t_i с аддитивными погрешностями $v_1(t_i)$ и $v_2(t_i)$, соответственно. Это означает, что

$$z_a(t_i) = a(t_i) + v_1(t_i), \quad z_h(t_i) = h(t_i) + v_2(t_i).$$

Погрешности датчиков между собой независимы и в отдельные моменты времени t_i являются гауссовыми случайными величинами с нулевыми средними значениями и с постоянными дисперсиями σ_1^2 , σ_2^2 , соответственно. Истинная высота $y(t)$, на которой находится ЛА, воздействует на барометрический датчик (БД) высоты так, что барометрическая высота $h(t)$ подчиняется уравнению

$$\tau \frac{d}{dt} h(t) + h(t) = y(t).$$

Теперь есть все сведения, чтобы записать уравнения состояния обобщенного динамического объекта, включающего ЛА и БД. Для этого введем основные обозначения: 1) вектор состояния

$$x = x(t) = [x_1 \equiv y(t) \mid x_2 \equiv v_y(t) \mid x_3 \equiv a_y(t) \mid x_4 \equiv h(t)]^T$$

и 2) вектор измерений

$$z = z(t) = [z_1 \equiv z_a(t) \mid z_2 \equiv z_h(t)]^T$$

с погрешностью $v(t) = [v_1(t) \mid v_2(t)]^T$. Из принятых предположений получаем непрерывную модель состояния

$$\frac{d}{dt}x(t) = Fx(t), \quad F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \alpha & 0 & 0 & -\alpha \end{bmatrix}, \quad \alpha \triangleq 1/\tau \quad (14.30)$$

и дискретную модель наблюдения

$$z(t_i) = Hx(t_i) + v(t_i), \quad H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (14.31)$$

1. Задание

А. Построить дискретную модель обобщенного объекта (14.30), (14.31). Для этого найти резольвентную матрицу $\Phi(s) \triangleq (Is - F)^{-1}$, где s — комплексная переменная преобразования Лапласа [75]. Доказать, что

$$\Phi(s) = \left[\begin{array}{c|c|c|c} s^{-1} & s^{-2} & s^{-3} & 0 \\ 0 & s^{-1} & s^{-2} & 0 \\ 0 & 0 & s^{-1} & 0 \\ \hline s^{-1}(s + \alpha)^{-1}\alpha & s^{-2}(s + \alpha)^{-1}\alpha & s^{-3}(s + \alpha)^{-1}\alpha & (s + \alpha)^{-1} \end{array} \right].$$

Совершить отсюда обратное преобразование Лапласа [75] и тем самым найти переходную матрицу состояния на интервале времени t :

$$\Phi(t) = \left[\begin{array}{c|c|c|c} 1 & t & t^2/2 & 0 \\ 0 & 1 & t & 0 \\ 0 & 0 & 1 & 0 \\ \hline a_1(t) & a_2(t) & a_3(t) & a_4(t) \end{array} \right], \quad \begin{aligned} a_1(t) &= 1 - e^{-\alpha t}, \\ a_2(t) &= (\alpha t - 1 + e^{-\alpha t})/\alpha, \\ a_3(t) &= (1 - \alpha t + \frac{(\alpha t)^2}{2} - e^{-\alpha t})/\alpha^2, \\ a_4(t) &= e^{-\alpha t}. \end{aligned}$$

Задать τ_s — постоянный интервал выборки (sampling interval), т. е. темп считывания данных с датчиков; при этом исходить из условия $\gamma \triangleq \tau_s/\tau < 1$, например, $\gamma = 1/10$ или меньше. Подстановкой $t = \tau_s$ в $\Phi(t)$ определить постоянную $\Phi \triangleq \Phi(\tau_s)$ — переходную матрицу состояния дискретной модели

$$\text{Модель А:} \quad x_{t+1} = \Phi x_t, \quad z_t = Hx_t + v_t, \quad (14.32)$$

которая является частным случаем общей модели (14.1), (14.2), где нижний индекс в записи $\{\cdot\}_t$ есть индекс (номер) дискретного момента времени для

переменной $\{\cdot\}$. Таким образом, исходная дискретная модель общего вида (14.1), (14.2) конкретизирована для исследуемого объекта в виде (14.32).

Б. Допуская некоторое усложнение модели (14.32), ввести в нее «шумовую» составляющую $G_t w_t$ по формуле

$$\text{Модель Б: } \left\{ \begin{array}{l} x_{t+1} = \Phi x_t + G_t w_t, \quad z_t = H x_t + v_t \\ G_t = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T = \text{const} \end{array} \right\} \quad (14.33)$$

с независимым дискретным белым шумом w_t : $w_t \sim \mathcal{N}(0, Q_t = q\tau_s = \text{const})$, где q — коэффициент диффузии соответствующего процесса броуновского движения (винеровского процесса), $\text{м}^2/\text{с}^5$. Таким образом может быть учтена нестационарность силы тяги двигателя ЛА или «скомпенсированы» другие неточности идеализированной математической модели А (14.32), но при этом потребуется задавать q . При $q = 0$ возвращаемся к модели А.

В. Спроектировать, отладить и продемонстрировать в действии программу решения задачи оптимального оценивания состояния применительно к модели А и также модели Б в соответствии с вашим номером и содержанием проекта по табл. 14.1 (стр. 334) и вариантом исходных данных для него по табл. 14.2 (стр. 334).

Г. Спланировать вычислительный эксперимент с целью сравнить поведение оценок состояния объекта в модели А и в модели Б. При планировании предусмотреть вариации параметров по табл. 14.2 на $\pm 50\%$.

Д. Получить от преподавателя восьмизначное число N , в котором каждая цифра $n_i \in \{1, 2, 3, 4, 5, 6\}$ обозначает номер варианта исходных данных по табл. 14.2. Начальный вектор состояния выбрать самостоятельно или спросить у преподавателя.

Е. Сделать выводы по выполненной работе. Выводы делаются на основании аналитического исследования и по результатам компьютерного моделирования. Они должны отражать результаты решения следующих вопросов:

- построение дискретной модели,
- оценка точности использованных алгоритмов оценивания,
- оценка времени сходимости к установившемуся режиму фильтрации,
- аналитическое определение установившегося решения \bar{P} алгебраического уравнения Риккати (14.9); для этого надо в уравнении (14.9) положить $\bar{P} \triangleq P_{t+1}^- = P_t^-$,
- оценка влияния вариаций параметров модели:

Таблица 14.1. Номер и содержание проекта № 10

Номер проекта							
1	2	3	4	5	6	7	8
Разновидность алгоритма по разд. 14							
РКККФ	РККИФ	МККИФ	КоККФ	СКККФ	СККИФ	СМККИФ	СКоККФ
Страница, где начинается описание алгоритма							
320	322	324	326	327	328	329	329

Таблица 14.2. Варианты исходных данных для проекта № 10

№	Параметр	Вариант исходных данных					
		1	2	3	4	5	6
1	Диффузия q , $\text{м}^2/\text{с}^5$	3000	340	400	300	3500	3350
2	Постоянная времени τ , с	0.05	0.65	0.80	0.90	0.10	0.12
3	Дисперсия σ_1^2 , $(\text{м}/\text{с}^2)^2$	1.00	2.25	4.00	6.25	6.00	5.50
4	Дисперсия σ_2^2 , м^2	40	30	25	35	45	50
5	Дисперсия $P_{11}(0)$, м^2	10	20	30	40	50	60
6	Дисперсия $P_{22}(0)$, $(\text{м}/\text{с})^2$	60	50	40	30	20	10
7	Дисперсия $P_{33}(0)$, $(\text{м}/\text{с}^2)^2$	15	20	25	30	35	40
8	Дисперсия $P_{44}(0)$, м^2	45	40	35	25	30	15

- диффузии q ,
- постоянной времени τ ,
- дисперсий σ_1^2 , σ_2^2 ,
- а также дисперсий $P_{11}(0)$, $P_{22}(0)$, $P_{33}(0)$, $P_{44}(0)$

на скорость сходимости параметров фильтра к значениям установившегося режима оценивания. Выводы обосновать протоколами компьютерного моделирования. Для сравнения рекомендуем использовать MATLAB.

14.12 Варианты задания на лабораторный проект № 10

Данный проект — групповой, т. е. рассчитан на выполнение одного проекта двумя (максимум тремя) студентами. Свое задание и содержание проекта студенты определяют по номеру назначенного проекта в табл. 14.1 и исходные данные для него берут из табл. 14.2.

ЗАКЛЮЧЕНИЕ

Данное учебное пособие разделено на две части, первая из которых содержит материал начального уровня, составляющий основу всех вычислительных методов алгебры, а вторая часть представляет собой курс повышенного типа и, соответственно, может использоваться как учебный материал для специальных разделов вычислительной математики. Вторая часть может использоваться также теми, кто занимается применением вычислительных методов оценивания либо в практических разработках, либо в научных исследованиях.

В Часть I — *Вычислительная линейная алгебра* — включены учебные материалы по следующим темам стандартного курса «Численные методы»:

- тема 1 — методы исключения в решении систем:

1) текст и лабораторная работа № 1, стр. 27–52 включают:

- алгоритмы метода Гаусса,
- выбор ведущего элемента,
- компактные схемы,
- алгоритмы метода Жордана,
- вычисление обратной матрицы,
- плохо обусловленные матрицы,
- задание на лабораторный проект № 1,
- 26 вариантов задания;

2) текст и лабораторная работа № 2, стр. 53–72 включают:

- гауссово исключение и ijk -алгоритмы,
- распараллеливание вычислений,
- параллельное умножение,
- параллельное LU -разложение,
- LU -разложение и его ijk -формы,
- треугольные системы,

- задание на лабораторный проект № 2,
- 40 вариантов задания;
- 3) текст и лабораторная работа № 3, стр. 73–81 включают:
 - метод окаймления,
 - окаймление известной части LU -разложения,
 - окаймление неизвестной части LU -разложения,
 - задание на лабораторный проект № 3,
 - 16 вариантов задания;
- 4) текст и лабораторная работа № 4, стр. 82–89 включают:
 - упакованные формы хранения матриц,
 - выбор ведущего элемента,
 - задание на лабораторный проект № 4,
 - 48 вариантов задания;
- тема 2 — разложения Холецкого положительно определенных матриц:
 - 5) текст и лабораторная работа № 5, стр. 90–106 включают:
 - положительно определенные матрицы,
 - квадратные корни из матриц и алгоритмы Холецкого,
 - программная реализация алгоритмов Холецкого,
 - разложение Холецкого: ijk -формы,
 - разложение Холецкого: алгоритмы окаймления,
 - задание на лабораторный проект № 5,
 - 40 вариантов задания;
- тема 3 — методы ортогональных преобразований:
 - 6) текст и лабораторная работа № 6, стр. 107–140 включают:
 - ортогональные матрицы и приложения,
 - линейная задача наименьших квадратов,
 - ортогональные матрицы в задаче о наименьших квадратах,
 - преобразование Хаусхолдера,
 - шаг триангуляризации матрицы преобразованием Хаусхолдера,
 - решение треугольной системы $Rx = z$ и обращение матрицы R ,
 - преобразование Гивенса,

- варианты заполнения матрицы R ,
 - правосторонние ортогональные преобразования и их применение,
 - двусторонние ортогональные преобразования и их применение,
 - ортогонализация Грама–Шмидта,
 - алгоритмы ортогонализации Грама–Шмидта,
 - решение системы линейных алгебраических уравнений после ортогонализации Грама–Шмидта,
 - вычисление обратной матрицы A^{-1} после ортогонализации Грама–Шмидта,
 - задание на лабораторный проект № 6,
 - 28 вариантов задания;
- тема 4 — итерационные методы решения линейных систем уравнений:
 - 7) текст и лабораторная работа № 7, стр. 141–159 включают:
 - метод Якоби,
 - метод Зейделя,
 - метод простой итерации,
 - метод Рундсона,
 - метод верхней релаксации (Юнга),
 - четыре метода вариационного типа,
 - задание на лабораторный проект № 7,
 - 15 вариантов задания.

В качестве дополнительного материала в Часть I включен:

- фонд задач, стр. 160–196.

Часть II — *Линейное оценивание* — содержит обширный материал от теоретических основ до эффективных и новейших вычислительных алгоритмов, а именно:

- необходимые теоретические сведения из линейной алгебры,
- основы оценивания по методу наименьших квадратов, включая последовательные алгоритмы и полную статистическую интерпретацию задачи МНК,
- методы одновременного решения нормальных уравнений, включая

- 8) текст и лабораторную работу № 8,
- устойчивые алгоритмы фильтрации, в том числе в задаче регрессионного моделирования, с акцентом на этап обработки измерения, включая
- 9) текст и лабораторную работу № 9,
- передовые блочные алгоритмы дискретной фильтрации, применяющие ортогональные преобразования для ковариационных или информационных алгоритмов, совмещающие или не совмещающие этапы экстраполяции оценок и обработки измерения (т. е. одностадийные или двухстадийные) и ориентированные на параллельные вычисления, включая
- 10) текст и лабораторную работу № 10.

В приложение А.1 (стр. 339) помещены формальные доказательства для новых скаляризованных блочных алгоритмов с ортогонализацией.

Практическое значение рассмотренных вычислительных алгоритмов оценивания общепризнано. Эти алгоритмы оправдали усилия на их разработку. Кроме этого, они имеют и более широкое — теоретическое значение.

Мы завершаем пособие указанием на одно из возможных направлений расширенного использования изложенных вычислительных методов оценивания. Учитывая известное свойство двойственности задачи оценивания и задачи управления, таким направлением может быть разработка и исследование вычислительных алгоритмов для задачи оптимального линейного управления. Для перехода к этой задаче в приложение В (стр. 343) помещен небольшой вводный материал.

Наконец, необходимо отметить, что мы считаем главным в изучении предмета «Вычислительная математика».

- Наш курс опирается на *Проектно-ориентированное изучение (ПОИ)* предмета. Квинтэссенция ПОИ выражена на стр. 23 в типичном диалоге между Студентом и Экзаменатором. Подробнее см. [134].
- Свои оценки студент зарабатывает в течение семестра (принцип распределенных по времени требований) в соответствии с теми целями, которые он ставит перед собой. Подробнее см. [130]
- Пока студент не ставит перед собой целей и не преодолевает препятствия, до тех пор он не понимает смысла слова «студент» — изучающий. Подробнее см. *Введение*, стр. 15–23.

Приложение А

Обоснования алгоритмов для подразд. 14.7–14.10

А.1 Построение новых скаляризованных алгоритмов

В подразделы 14.7, 14.8, 14.9 и 14.10 помещены результаты построения новых «скаляризованных» реализаций дискретного фильтра Калмана, принадлежащие М. В. Куликовой [51]. Приводимые ниже теоретические обоснования даются по тексту [51]. Они опираются на широко известные факты (леммы А.1 и А.2) из теории оптимального оценивания [113].

ЛЕММА А.1. Пусть матрицы ковариации шумов в измерителе (14.2) R_t имеют диагональный вид, т. е. $R_t = \text{diag} \left\{ \left(\sigma_t^{(1)} \right)^2, \left(\sigma_t^{(2)} \right)^2, \dots, \left(\sigma_t^{(m)} \right)^2 \right\}$. Представим матрицу H_t построчно:

$$H_t = \left[\left(h_t^{(1)} \right)^T, \left(h_t^{(2)} \right)^T, \dots, \left(h_t^{(k)} \right)^T, \dots, \left(h_t^{(m)} \right)^T \right]^T.$$

Тогда этап обработки измерений *стандартного ковариационного фильтра* (СКФ) Калмана (14.5)–(14.7) эквивалентен следующей последовательной процедуре:

$$\tilde{P}_t^{(k+1)} = \tilde{P}_t^{(k)} - \tilde{P}_t^{(k)} h_t^{(k+1)} \left[\left(h_t^{(k+1)} \right)^T \tilde{P}_t^{(k)} h_t^{(k+1)} + \left(\sigma_t^{(k+1)} \right)^2 \right]^{-1} \left(h_t^{(k+1)} \right)^T \tilde{P}_t^{(k)}, \quad (\text{А.1})$$

$$\tilde{x}_t^{(k+1)} = \tilde{x}_t^{(k)} + P_t^{(k)} \frac{h_t^{(k+1)}}{\left(r_{e,t}^{(k+1)} \right)^2} \left[z_t^{(k+1)} - \left(h_t^{(k+1)} \right)^T \tilde{x}_t^{(k)} \right]. \quad (\text{А.2})$$

ЛЕММА А.2. В условиях леммы А.1. этап обработки измерений *стандартного информационного фильтра* эквивалентен следующей последова-

тельной процедуре обновления оценок:

$$\left(\tilde{P}_t^{(k+1)}\right)^{-1} = \left(\tilde{P}_t^{(k)}\right)^{-1} + h_t^{(k+1)} \left(h_t^{(k+1)}\right)^T / \left(\sigma_t^{(k+1)}\right)^2, \quad (\text{A.3})$$

$$\left(\tilde{P}_t^{(k+1)}\right)^{-1} \tilde{x}_t^{(k+1)} = \left(\tilde{P}_t^{(k)}\right)^{-1} \tilde{x}_t^{(k)} + h_t^{(k+1)} z_t^{(k+1)} / \left(\sigma_t^{(k+1)}\right)^2. \quad (\text{A.4})$$

ТЕОРЕМА А.1 ([51]). Пусть ковариационная матрица дискретного белого шума в измерителе (14.2) R_t имеет диагональный вид, т.е. $R_t = \text{diag} \left\{ \left(\sigma_t^{(1)}\right)^2, \left(\sigma_t^{(2)}\right)^2, \dots, \left(\sigma_t^{(m)}\right)^2 \right\}$. Представим матрицу H_t построчно:

$$H_t = \left[\left(h_t^{(1)}\right)^T, \left(h_t^{(2)}\right)^T, \dots, \left(h_t^{(k)}\right)^T, \dots, \left(h_t^{(m)}\right)^T \right]^T.$$

Тогда этапы обработки измерений следующих четырех алгоритмов:

- 1) СКККФ (14.26) — см. подразд. 14.7, стр. 327,
- 2) СККИФ (14.27) — см. подразд. 14.8, стр. 328,
- 3) СМККИФ (14.28) — см. подразд. 14.9, стр. 329,
- 4) СКоККФ (14.29) — см. подразд. 14.10, стр. 329

алгебраически эквивалентны этапу обработки измерений *стандартного ковариационного фильтра* (СКФ) Калмана (14.5)–(14.7), где вектор измерений z_t в каждый момент времени t обрабатывается поэлементно:

$$z_t = \left[z_t^{(1)}, z_t^{(2)}, \dots, z_t^{(m)} \right].$$

ДОКАЗАТЕЛЬСТВО. Воспользуемся леммами А.1 и А.2. Покажем, что этапы обработки измерений последовательных ковариационных алгоритмов, т.е. СКККФ и СКоККФ, алгебраически эквивалентны уравнениям (А.1), (А.2). В то же время для информационных типов фильтров, т.е. для СККИФ и СМККИФ, справедливы формулы (А.3), (А.4).

Для удобства дальнейшего изложения обозначим матрицы, стоящие в правой и левой частях формул (14.26), (14.27), через A_t и B_t , соответственно. Представим дальнейшее доказательство по пунктам ①, ②, ③:

① Для СКККФ из формулы (14.26) непосредственно следует

$$A_t^T \left(O_{t,1}^{(k)} \right)^T O_{t,1}^{(k)} A_t = A_t^T A_t \stackrel{(14.26)}{=} B_t^T B_t.$$

Рассмотрим поэлементно матрицы $A_t^T A_t$ и $B_t^T B_t$. Для элемента $(2, 2)$ матрицы $A_t^T A_t$ справедлива следующая последовательность равенств:

$$\begin{aligned} (A_t^T A_t)_{2,2} &= \Phi_t \tilde{P}_t^{(k)} \Phi_t^T \stackrel{(14.26)}{=} (B_t^T B_t)_{2,2} = \bar{K}_{p,t}^{(k+1)} \left(\bar{K}_{p,t}^{(k+1)} \right)^T + \\ &+ \Phi_t \tilde{P}_t^{(k+1)} \Phi_t^T \implies \Phi_t \tilde{P}_t^{(k+1)} \Phi_t^T = \\ &= \Phi_t \tilde{P}_t^{(k)} \Phi_t^T - \bar{K}_{p,t}^{(k+1)} \left(\bar{K}_{p,t}^{(k+1)} \right)^T = \\ &= \Phi_t \tilde{P}_t^{(k)} \Phi_t^T - \Phi_t \tilde{P}_t^{(k)} h_t^{(k+1)} \left(r_{e,t}^{(k+1)} \right)^{-2} \left(h_t^{(k+1)} \right)^T \tilde{P}_t^{(k)} \Phi_t^T. \end{aligned}$$

Последнее есть не что иное как (A.1), умноженное слева и справа на матрицы Φ_t и Φ_t^T , соответственно. Далее имеем

$$\begin{aligned} (A_t^T A_t)_{2,3} &= \Phi_t \tilde{x}_t^{(k)} \stackrel{(14.26)}{=} (B_t^T B_t)_{2,3} = -\bar{e}_t^{(k+1)} \bar{K}_{p,t}^{(k+1)} + \Phi_t \tilde{x}_t^{(k+1)} \implies \\ &\implies \Phi_t \tilde{x}_t^{(k+1)} = \Phi_t \tilde{x}_t^{(k)} + \bar{e}_t^{(k+1)} \bar{K}_{p,t}^{(k+1)} = \\ &= \Phi_t \tilde{x}_t^{(k)} + \Phi_t \tilde{P}_t^{(k)} h_t^{(k+1)} \left(r_{e,t}^{(k+1)} \right)^{-2} \left[z_t^{(k+1)} - \left(h_t^{(k+1)} \right)^T \tilde{x}_t^{(k)} \right], \end{aligned}$$

а это есть уравнение (A.2), домноженное слева на матрицу Φ_t .

Поэтому для СКККФ из формулы (14.26) непосредственно следует справедливость (A.1), (A.2). Таким образом, согласно лемме A.1, этапы обработки измерений последовательного алгоритма СКККФ и стандартного фильтра Калмана эквивалентны друг другу.

② Аналогично для СККИФ из (14.27) непосредственно выводится

$$\begin{aligned} (A_t^T A_t)_{1,1} &= \left(\tilde{P}_t^{(k)} \right)^{-1} + h_t^{(k+1)} \left(h_t^{(k+1)} \right)^T / \left(\sigma_t^{(k+1)} \right)^2 \stackrel{(14.27)}{=} \\ &\stackrel{(14.27)}{=} (B_t^T B_t)_{1,1} = \left(\tilde{P}_t^{(k+1)} \right)^{-1}. \end{aligned}$$

Видно, что это совпадает с уравнением (A.3). Далее находим

$$\begin{aligned} (A_t^T A_t)_{1,2} &= \left(\tilde{P}_t^{(k)} \right)^{-1} \tilde{x}_t^{(k)} + h_t^{(k+1)} z_t^{(k+1)} / \left(\sigma_t^{(k+1)} \right)^2 \stackrel{(14.27)}{=} \\ &\stackrel{(14.27)}{=} (B_t^T B_t)_{1,2} = \left(\tilde{P}_t^{(k+1)} \right)^{-1} \tilde{x}_t^{(k+1)}. \end{aligned}$$

Последнее есть уравнение (А.4). Таким образом, этапы обработки измерений последовательного алгоритма СККИФ и стандартного информационного фильтра также взаимно эквивалентны.

- ③ Поскольку фильтры СМККИФ и СКоККФ являются модификациями алгоритмов СКККФ и СККИФ, то из ранее установленной справедливости последних следует правильность СМККИФ (14.28) и СКоККФ (14.29).

□

ЗАМЕЧАНИЕ А.1. От ковариационной матрицы R_t наблюдений всегда требуется невырожденность. Это — одно из условий существования фильтра Калмана. Для построения скаляризованных алгоритмов фильтрации (на этапе обработки наблюдений) дополнительно требуется свойство диагональности этой матрицы R_t . Очевидно, это требование не является существенным ограничением, поскольку в случае его невыполнения всегда можно использовать так называемые *псевдонаблюдения*

$$\bar{z}_t = L_t^{-1} z_t,$$

где матрицы L_t получены разложением Холецкого матриц $R_t = L_t L_t^T > 0$, так как обратная матрица L_t^{-1} существует. Для псевдонаблюдений уравнение измерителя

$$z_t = H_t x_t + v_t$$

заменяется на алгебраически эквивалентное уравнение

$$\bar{z}_t = \bar{H}_t x_t + \bar{v}_t,$$

где

$$\bar{H}_t = L_t^{-1} H_t, \quad \bar{v}_t = L_t^{-1} v_t.$$

Тогда

$$\bar{R}_t = \mathbf{E} \{ \bar{v}_t \bar{v}_t^T \} = L_t^{-1} \mathbf{E} \{ v_t v_t^T \} L_t^{-T} = L_t^{-1} L_t L_t^T L_t^{-T} = I.$$

Приложение В

К задаче управления

В.1 Задача ЛКГ-управления

Пусть система адекватно описывается n -мерным стохастическим разностным уравнением состояния

$$x(t_{i+1}) = \Phi(t_{i+1}, t_i)x(t_i) + B_d(t_i)u(t_i) + G_d(t_i)w_d(t_i) \quad (\text{В.1})$$

и m -мерным уравнением измерений

$$z(t_i) = H(t_i)x(t_i) + v(t_i), \quad (\text{В.2})$$

где r -мерное управление $u(t_i)$ приложено к входу системы; $\{w(t_0), w(t_1), \dots\}$ и $\{v(t_1), v(t_2), \dots\}$ представляют собой две независимые последовательности независимых нормально распределенных случайных векторов возмущений и погрешностей с нулевым средним значением, имеющие, соответственно, размерности q и m , обладающие ковариациями $Q_d(t_i)$ и $R(t_i)$ и независимые от случайного нормально распределенного начального состояния $x(t_0)$ со средним значением \bar{x}_0 и ковариацией P_0 .

Цель заключается в определении оптимального физически осуществимого закона управления u^* , оптимального в смысле минимума ожидаемой квадратической функции стоимости

$$J_0 = \mathbf{E} \left\{ \sum_{i=0}^N \frac{1}{2} [x^T(t_i)V_x(t_i)x(t_i) + u^T(t_i)V_u(t_i)u(t_i)] \right\} + \frac{1}{2} \mathbf{E} \{x^T(t_{N+1})V_f x(t_{N+1})\}, \quad (\text{В.3})$$

где t_0 — начальный момент времени, t_{N+1} — конечный (финальный) момент времени. Симметрические матрицы $V_x(t_i)$ и $V_u(t_i)$, а также $V_f(t_i)$ задают вес

(удельную значимость) потерь из-за отклонений от нуля следующих величин: текущих состояний и управлений на интервале управления $[t_0, t_{N+1}]$, а также финального состояния $x(t_{N+1})$.

ЗАМЕЧАНИЕ В.1. В силу критерия (В.3), данная задача называется задачей *ЛКГ-управления*, что читается как *линейная квадратично-гауссова задача управления* [117, 120].

ЗАМЕЧАНИЕ В.2. В минимизации критерия (В.3) могут участвовать лишь те векторы состояния, которые поддаются изменению, и только те векторы управления, которые могут их изменить. Так, начальное состояние $x(t_0)$ существует до начала процесса управления, и его нельзя изменить никаким управлением. Этот факт можно отразить в (В.3) выбором матрицы $V_x(t_0) = 0$, то есть $x(t_0)$ в критерии не учитывать. Поэтому, наряду с (В.3), произвольно выбирают (то есть только на основании данных рассуждений) запись критерия в виде

$$J_1 = \mathbf{E} \left\{ \sum_{i=0}^N \frac{1}{2} [x^T(t_{i+1})V_x(t_{i+1})x(t_{i+1}) + u^T(t_i)V_u(t_i)u(t_i)] \right\}, \quad (\text{В.4})$$

в котором $V_x(t_{N+1}) \triangleq V_f$ и, следовательно, $J_0 = J_1 + \mathbf{E} \{x^T(t_0)V_x(t_0)x(t_0)\}$. Отсюда видно, что J_0 обеспечивает более общий подход к постановке проблемы управления, поскольку не требует тех предварительных рассуждений, которые предшествуют выбору критерия (В.4) вместо (В.3). Действительно, управление $u(t_0)$, прикладываемое в момент t_0 , при оптимальном образе действия должно зависеть от исходного состояния $x(t_0)$, непосредственно предшествующего этому управлению. Вместе с тем, применяют оба варианта, (В.3) и (В.4).

В.2 Решение задачи управления

Введем, исходя из критерия (В.4), *функцию Беллмана*

$$C_{N+1-i}^*(t_i) \triangleq \min_{U(t_i, t_N)} E \left\{ \sum_{j=i}^N \frac{1}{2} \left[x^T(t_{j+1})V_x(t_{j+1})x(t_{j+1}) + u^T(t_j)V_u(t_j)u(t_j) \right] \right\}, \quad i = 0, 1, \dots, N,$$

которая представляет собой минимальную среднюю стоимость управления $U(t_i, t_N) \triangleq \{u(t_i), u(t_{i+1}), \dots, u(t_N)\}$ на оставшихся $(N + 1 - i)$ шагах от

текущего момента t_i . Для $0 \leq i < s \leq N$, согласно *принципу оптимальности*, имеем *функциональное уравнение Беллмана*

$$C_{N+1-i}^*(t_i) = \min_{U(t_i, t_{s-1})} E \left\{ \sum_{j=i}^{s-1} \frac{1}{2} \left[x^T(t_{j+1}) V_x(t_{j+1}) x(t_{j+1}) + \right. \right. \\ \left. \left. + u^T(t_j) V_u(t_j) u(t_j) \right] + C_{N+1-s}^*(t_s) \right\},$$

которое для пошаговой оптимизации, при $s = i + 1$, запишется в виде

$$C_{N+1-i}^*(t_i) = \min_{u(t_i)} E \left\{ \frac{1}{2} \left[x^T(t_{i+1}) V_x(t_{i+1}) x(t_{i+1}) + \right. \right. \\ \left. \left. + u^T(t_i) V_u(t_i) u(t_i) \right] + C_{N-i}^*(t_{i+1}) \right\}, \quad (\text{B.5})$$

где $i = N, N-1, \dots, 0$ и $C_0^*(t_{N+1}) = 0$.

ТЕОРЕМА В.1. Оптимальный закон ЛГК-управления для задачи стохастического управления с критерием (В.4) разделяется на две части (часть *I* и часть *II*), соединенные последовательно (*II* вслед за *I*) и синтезируемые независимо друг от друга:

I. Оптимальный линейный фильтр (Калмана), ФК

А. Для $i = 0, 1, \dots, N$ ФК вычисляет *экстраполяционные оценки* $\hat{x}(t_{i+1}^-)$ состояния $x(t_{i+1})$, получаемые при экстраполяции отфильтрованных оценок $\hat{x}(t_i^+)$ от момента t_i к моменту t_{i+1} в виде

$$\hat{x}(t_{i+1}^-) = \Phi(t_{i+1}, t_i) \hat{x}(t_i^+) + B_d(t_i) u^*(t_i)$$

с начальным значением $\hat{x}(t_0^+) = \mathbf{E} \{x(t_0)\} = \bar{x}_0$, и также *их ковариации*

$$P(t_{i+1}^-) = \Phi(t_{i+1}, t_i) P(t_i^+) \Phi^T(t_{i+1}, t_i) + G_d(t_i) Q_d(t_i) G_d^T(t_i)$$

с начальным значением $P(t_0^+) = \mathbf{E} \{[x(t_0) - \bar{x}_0][x(t_0) - \bar{x}_0]^T\} = P_0$.

В. Для $i = 1, 2, \dots, N$ ФК вычисляет *отфильтрованные оценки* $\hat{x}(t_i^+)$, обновленные по измерению $z(t_i) = z_i$ с ковариацией $R(t_i) > 0$ ошибок измерений в момент t_i , в виде

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + K_f(t_i)[z_i - H(t_i)\hat{x}(t_i^-)]$$

с коэффициентом усиления фильтра

$$K_f(t_i) = P(t_i^-) H^T(t_i) [H(t_i) P(t_i^-) H^T(t_i) + R(t_i)]^{-1}$$

и также ковариации отфильтрованных оценок

$$P(t_i^+) = P(t_i^-) - K_f(t_i)H(t_i)P(t_i^-).$$

II. Оптимальный линейный регулятор, ОЛР

ОЛР обеспечивает минимальную ожидаемую стоимость завершения процесса управления на оставшихся $(N + 1 - i)$ шагах из момента t_i , равную

$$C_{N+1-i}^*(t_i) = \frac{1}{2} \mathbf{E} \{x^T(t_i)M(t_i)x(t_i)\} + \alpha(t_i), \quad i = N, N-1, \dots, 0 \quad (\text{B.6})$$

с помощью оптимального управляющего воздействия

$$u^*(t_i) = -G_r(t_i)\hat{x}(t_i^+), \quad i = 0, 1, \dots, N. \quad (\text{B.7})$$

Управляющая функция этого стохастического регулятора

$$u^*[t_i, (\cdot)] = -G_r(t_i)(\cdot) \quad (\text{B.8})$$

идентична управляющей функции детерминистского линейного регулятора, причем для матрицы $G_r(t_i)$ в (B.8), для матрицы $M(t_i)$ и для скалярной величины $\alpha(t_i)$ в (B.6) справедлив следующий алгоритм.

Алгоритм последовательных вычислений регулятора (B.9)

- (1) $\Pi(t_{N+1}) = V_f \triangleq V_x(t_{N+1}), \quad \alpha(t_{N+1}) = 0,$
- (2) $A(t_i) = B_d^T(t_i)\Pi(t_{i+1})B_d(t_i) + V_u(t_i),$
- (3) $\Lambda(t_i) = \Phi^T(t_{i+1}, t_i)\Pi(t_{i+1})B_d(t_i)A^{-1}(t_i)B_d^T(t_i)\Pi(t_{i+1})\Phi(t_{i+1}, t_i),$
- (4) $M(t_i) = \Phi^T(t_{i+1}, t_i)\Pi(t_{i+1})\Phi(t_{i+1}, t_i) - \Lambda(t_i),$
- (5) $K_r(t_i) = A^{-1}(t_i)B_d^T(t_i)\Pi(t_{i+1}),$
- (6) $G_r(t_i) = K_r(t_i)\Phi(t_{i+1}, t_i),$
- (7) $\beta(t_i) = \frac{1}{2} \mathbf{tr} \{ \Lambda(t_i)P(t_i^+) + G_d^T(t_i)\Pi(t_{i+1})G_d(t_i)Q_d(t_i) \},$
- (8) $\alpha(t_i) = \alpha(t_{i+1}) + \beta(t_i),$
- (9) $\Pi(t_i) = V_x(t_i) + M(t_i),$

причем в алгоритме (В.9) для $i = N$ действуют пп. (1)–(9), а для следующих итераций $i = N - 1, \dots, 1, 0$ действуют пп. (2)–(9), хотя $\Pi(t_0)$, найденное по п. (9) при $i = 0$, далее не используется (конец вычислений).

В.3 Двойственность задач фильтрации и управления

Матрица $\Pi(t_i)$ в алгоритме (В.9) удовлетворяет *обратному алгебраическому уравнению Риккати*

$$\Pi(t_i) = V_x(t_i) + \Phi^T(t_{i+1}, t_i) \left\{ \begin{aligned} &\Pi(t_{i+1}) - \Pi(t_{i+1})B_d(t_i)[V_u(t_i) + \\ &+ B_d^T(t_i)\Pi(t_{i+1})B_d(t_i)]^{-1}B_d^T(t_i)\Pi(t_{i+1})\} \Phi(t_{i+1}, t_i), \end{aligned} \right\} \quad (В.10)$$

$$i = N, N - 1, \dots, 1, 0$$

с терминальным условием $\Pi(t_{N+1}) = V_f \triangleq V_x(t_{N+1})$ в начальный момент $i = N$ счета в обращенном времени. Оно является двойственным *прямоу алгебраическому уравнению Риккати*

$$P(t_{j+1}^-) = G_d(t_j)Q_d(t_j)G_d^T(t_j) + \Phi(t_{j+1}, t_j) \left\{ \begin{aligned} &P(t_j^-) - P(t_j^-)H^T(t_j) \times \\ &\times [R(t_j) + H(t_j)P(t_j^-)H^T(t_j)]^{-1}H(t_j)P(t_j^-) \} \Phi^T(t_{j+1}, t_j), \end{aligned} \right\} \quad (В.11)$$

$$j = 0, 1, \dots, N$$

с начальным условием $\{\cdot\} = P_0$ при $j = 0$ для задачи фильтрации.¹

Из сопоставления (В.10) и (В.11) устанавливаются *соотношения двойственности* между матрицами, которые описывают задачу оптимальной линейной фильтрации (и соответствующую часть I оптимального закона управления) и задачу оптимального линейного регулятора (и соответствующую часть II этого закона), — см. выше алгоритм (В.9), стр. 346. Эти соотношения двойственности показаны в табл. В.1, стр. 348.

В.4 Вычислительные алгоритмы задачи управления

Математический алгоритм управления (В.9) достаточно сложен в реализации. Кроме этого, он имеет принципиальную особенность: вычисления должны вестись в обращенном времени — от финального (терминального) момента $i = N$ в уравнении Риккати (В.10) к начальному моменту управления при $i = 0$. Эта особенность является принципиальной не потому, что

¹ $\{\cdot\}$ означает выражение внутри фигурных скобок в (В.11), ср. с формулой (14.9), стр. 318.

Таблица В.1. Соотношения двойственности для двух задач: оптимального ЛКГ-управления и оптимального ЛКГ-оценивания

Задача оптимального ЛКГ ^а -управления						
$\Pi(t_i)$	$V_x(t_i) \geq 0$	$\Phi^T(t_{i+1}, t_i)$	$B_d(t_i)$	$V_u(t_i) > 0$	$i = N, \dots, 0$	терминальное условие в (В.10) $\Pi(t_{N+1}) = V_f$
Задача оптимального ЛКГ ^а -оценивания						
$P(t_{j+1}^-)$	$Q(t_j) \geq 0$ ^б	$\Phi(t_{j+1}, t_j)$	$H^T(t_j)$	$R(t_j) > 0$	$j = N - i$	начальное условие в (В.11) $\{\cdot\}^c = P_0$

^а ЛКГ—Линейные модели, Квадратический критерий качества, Гауссовы помехи^б $Q(t_j) \triangleq G_d(t_j)Q_d(t_j)G_d^T(t_j)$ ^с $\{\cdot\}$ означает выражение внутри фигурных скобок в (В.11), ср. с формулой (14.9)

оптимальный закон управления найден методом динамического программирования Беллмана. Вне зависимости от метода вывода этого закона вычисления должны вестись именно в обращенном времени — от финального момента к начальному моменту времени. Это означает, что алгоритм (В.9) должен быть применен до начала процесса управления; его результаты — все значения матрицы $G_r(t_i)$ для формулы (В.7) — должны быть заранее вычислены и сохранены для воспроизведения управляющей функции (В.8) в формуле (В.7) в реальном процессе управления.

Взаимная двойственность, присущая двум задачам — оптимального ЛКГ-управления и оптимального ЛКГ-оценивания, — позволяет искать средства эффективной численной реализации алгоритма (В.9) по аналогии [70] с теми вычислительными методами оценивания, которые детально рассмотрены в данной книге.

Список иллюстраций

3.1	Строчно ориентированная схема $\bar{L}U$ -разложения	54
3.2	Столбцово ориентированная схема $\bar{L}U$ -разложения	54
3.3	Сложение n чисел методом сдвигания для $n = 8$ [8]	57
3.4	ij (слева) и ji (справа) формы матрично-векторного умножения [8]	59
3.5	Операция сложения в компьютере типа «регистр–регистр» [8]	60
3.6	Столбцово ориентированная схема $\bar{L}U$ -разложения с отложенными модификациями (jki -алгоритм, см. с. 64) [8]	61
3.7	Способ доступа к данным для kij -формы (слева) и для kji -формы (справа) $\bar{L}U$ -разложения. Обозначения: \bar{L} , U — вычисление закончено, обращений больше нет; \star — главный элемент (ГЭ); \oslash — деление на ГЭ (нормировка) [8]	65
3.8	Способ доступа к данным для jki -формы и для jik -формы (слева) и для ikj -формы и для ijk -формы (справа) $\bar{L}U$ -разложения. Обозначения: \bar{L} , U — вычисление закончено, обращения больше не производятся; \star — главный элемент (ГЭ); \oslash — деление на ГЭ (нормировка); \emptyset — обращений не было [8]	65
3.9	Алгоритмы скалярных произведений (слева) и столбцовый для обратной подстановки (справа)	69
4.1	Алгоритмы окаймления известной части $\bar{L}U$ -разложения: столбцовый (слева) и алгоритм скалярных произведений (справа)	74
4.2	Доступ к данным в алгоритмах окаймления известной части разложения. \bar{L} , U — вычисление закончено, но обращения происходят. A — обращений не было. Вычисляются: j -й столбец матрицы U и j -я строка матрицы \bar{L}	75
4.3	Доступ к данным в алгоритмах окаймления неизвестной части разложения. \bar{L}_{11} , U_{11} — вычисление закончено, обращений больше нет. L_{31} , U_{13} — вычисление закончено, но обращения происходят. A_{33} — обращений не было. Вычисляются: j -й столбец l_{3j} матрицы \bar{L} и j -я строка u_{j3}^T матрицы U	76
4.4	Алгоритмы Донгарры–Айзенштата окаймления неизвестной части $\bar{L}U$ -разложения: алгоритм линейных комбинаций (слева) и алгоритм скалярных произведений (справа)	77
6.1	Алгоритмы окаймления известной части LL^T -разложения: строчный (слева); алгоритм скалярных произведений (справа)	100
6.2	Алгоритмы окаймления неизвестной части LL^T -разложения: алгоритм линейных комбинаций (слева); алгоритм скалярных произведений (справа)	100

7.1	Алгебраически эквивалентные задачи, решаемые методом наименьших квадратов значений невязки v или среднего квадрата погрешности e : (а) — оптимальное моделирование неизвестной системы по экспериментальным условиям A и данным z ; (б) — оптимальное оценивание неизвестного вектора по наблюдениям Ax в присутствии случайных помех v с характеристиками $\mathbf{E}\{v\} = 0$ и $\mathbf{E}\{vv^T\} = I$	108
7.2	Геометрия преобразования Хаусхолдера. Задача 1 (прямая): даны векторы u и y , найти вектор y_r , отраженный от гиперплоскости U_\perp	112
7.3	Геометрия преобразования Хаусхолдера. Задача 2 (обратная): даны векторы y и y_r , найти вектор u , задающий отражающую гиперплоскость U_\perp ; здесь $y_r = se_1 = [s \mid 0 \cdots 0]^T$. Докажите, что здесь показан вектор $\frac{1}{2}u$, а не u (см. Замечание 7.2)	114
7.4	Представление возможных случаев применения теоремы 7.1 к матрице $A(m, n)$; (а) недоопределенная система: $k = m - 1 \leq n$; (б) определенная система: $k = n - 1$, $m = n$; (с) переопределенная система: $k = n < m$; (д) $k < n < m$	115
7.5	<i>Вверху:</i> Сохранение преобразования T и вычисление вектора $y = Tz$, $\forall y \in \mathbb{R}^m$. <i>Внизу:</i> Вычисление матрицы A^{-1} после сохранения преобразования T	121
7.6	Геометрия вращений	122
7.7	Вычисление матрицы $P_{1,j}$	123
7.8	Преобразование Гивенса: (а) столбцово ориентированная схема вычисления матрицы PA , где $P = P^{(j)}$ при $j = \min(m - 1, n)$ (нижняя матрица слева); (б) вычисление координаты r вектора $(a, b)^T$, повернутого до совмещения с первой осью, а также косинуса и синуса угла поворота и рабочего признака ζ ; (в) строчно ориентированная схема вычисления матрицы PA (верхняя матрица (г) восстановление косинуса и синуса угла поворота из признака ζ ; (д) получение вектора y теми преобразованиями $P_{j,i}$ произвольного вектора $z \in \mathbb{R}^m$, которые сохранены в рабочих признаках $\zeta_{j,i}$ и восстанавливаются из них; (е) вследствие п. (б) векторы 1, 2, 3 и 4 поворачиваются к положительному направлению первой координатной оси, а векторы 5, 6, 7 и 8 — к отрицательному направлению этой оси.	125
10.1	Теорема об ортогональной проекции	206
10.2	Матрица A и ее псевдообратная A^+ [13]	213
10.3	Проектирование вектора на линейное подпространство	229
11.1	Линейная задача наименьших квадратов	239
11.2	Включение априорных данных в линейную задачу НК	242
11.3	Рекурсия МНК в стандартной ковариационной форме (схема Калмана). Этап 1 — обновление модели по наблюдениям. Этап 2 — экстраполяция модели между наблюдениями. Априорная модель дает предсказание \hat{z} для отклика z объекта. Разность ν имеет смысл обновляющего процесса. Весовая матрица K , умноженная на ν , обеспечивает обновление априорной модели по наблюдению z , т. е. переход к апостериорной модели	247
11.4	Представление экспериментальных данных z в виде результата измерения неизвестного вектора x с помощью матрицы A в присутствии случайных ошибок v	252
11.5	Статистическая задача получения оценки \hat{x} по критерию МАВ и ее решение (полная статистическая интерпретация МНК-решения). Между моментами получения текущих данных апостериорные данные занимают место априорных данных, и процесс решения повторяется	258
13.1	Сечение поверхности критерия качества $J(x) = (z - Ax)^T(z - Ax)$ (для упражнения 13.10) плоскостью уровня	304

Список таблиц

1.1	Влияние неумажительных пропусков на оценку	18
2.1	Свойства специальных элементарных матриц	38
2.2	Позтапное перемножение $L_4^{-1}(L_3^{-1}(L_2^{-1}L_1^{-1}))$	44
2.3	Позтапное перемножение $\bar{U}_2^{-1}(\bar{U}_3^{-1}\bar{U}_4^{-1})$	46
3.1	Вычисления по алгоритму jki -формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом	66
3.2	Вычисления по алгоритму jik -формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом	66
3.3	Вычисления по алгоритму ikj -формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом	67
3.4	Вычисления по алгоритму ijk -формы для примера 3.3. Позиции ГЭ (без их реального выбора) показаны выделенным шрифтом	68
3.5	Варианты задания на лабораторный проект № 2	72
4.1	Вычисления по алгоритмам на рис. 4.1 для примера 3.3. Позиции элемента-делителя столбца \bar{L} показаны выделенным шрифтом	75
4.2	Вычисления по алгоритмам на рис. 4.4 для примера 3.3. Позиции элемента-делителя столбца \bar{L} показаны выделенным шрифтом	78
4.3	Варианты задания на лабораторный проект № 3	81
5.1	Варианты задания на лабораторный проект № 4	89
6.1	Варианты задания на лабораторный проект № 5	106
7.1	Эффективное обращение верхней треугольной матрицы $U := R^{-1}$	120
7.2	Варианты задания на лабораторный проект № 6	139
8.1	Варианты задания на лабораторный проект № 7	159
12.1	Варианты задания на лабораторный проект № 8	269
13.1	Скаляризованный фильтр Калмана в задаче регрессионного моделирования с мультиколлинеарностью регрессоров	303
14.1	Номер и содержание проекта № 10	334
14.2	Варианты исходных данных для проекта № 10	334
В.1	Соотношения двойственности для двух задач: оптимального ЛКГ-управления и оптимального ЛКГ-оценивания	348

Библиографический список

Основная литература

1. Алберт, А. Регрессия, псевдоинверсия и рекуррентное оценивание / А. Алберт. — М.: Наука, 1977.
2. Бахвалов, Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. — М.: Наука, 1987.
3. Вержбицкий, В. М. Основы численных методов: учебное пособие для вузов / В. М. Вержбицкий. — М.: Высш. шк., 2002, — 2-е изд., перераб. — М.: Высш. шк., 2005.
4. Воеводин, В. В. Вычислительные основы линейной алгебры / В. В. Воеводин. — М.: Наука, 1977.
5. Воеводин, В. В. Численные методы алгебры. Теория и алгоритмы / В. В. Воеводин. — М.: Наука, 1966.
6. Горбаченко, В. И. Вычислительная линейная алгебра с примерами на MATLAB / В. И. Горбаченко. — СПб.: БХВ, 2011. — Аннотация:
<http://www.exponenta.ru/educat/news/gorbachenko/1.asp>. Содержание:
http://www.bhv.ru/books/full_contents.php?id=189162. Visited 25.12.2011.
7. Методические материалы для проведения итоговой государственной аттестации выпускников вузов по направлению «Прикладная математика и информатика»: учебно-методический сборник / Тверь: Твер. Гос. ун-т, 2003.
8. Ортега, Дж. Введение в параллельные и векторные методы решения линейных систем / Дж. Ортега. — М.: Мир, 1991.
9. Ортега, Дж. Введение в численные методы решения дифференциальных уравнений / Дж. Ортега, У. Пул. — М.: Наука, 1986.
10. Писсанецки, С. Технология разреженных матриц / С. Писсанецки. — М.: Мир, 1988.
11. Райс, Дж. Матричные вычисления и математическое обеспечение / Дж. Райс. — М.: Мир, 1984.
12. Самарский, А. А. Численные методы / А. А. Самарский, А. В. Гулин. — М.: Наука, 1989.
13. Стренг, Г. Линейная алгебра и ее применения / Г. Стренг. — М.: Мир, 1980.
14. Фаддеев, Л. К. Вычислительные методы линейной алгебры / Л. К. Фаддеев, В. Н. Фаддеева. — М.: Физматгиз, 1963.

15. Bierman, G. J. Factorization methods for discrete sequential estimation / G. J. Bierman. — New York, San Francisco, London: Academic Press, 1977.
16. Grewal, M. S. Kalman Filtering: Theory and Practice Using MATLAB / M. S. Grewal, A. P. Andrews. — Second Edition. — N-Y: John Wiley and Sons Inc., 2001. ISBNs: 0-471-39254-5 (Hardback) and 0-471-26638-8 (Electronic).
17. Kailath, T. Linear Estimation / T. Kailath, A. H. Sayed, B. Hassibi. — Prentice Hall, NJ, 1999.
18. Rao, C. R. Linear Models: Least Squares and Alternatives, Second Edition / C. Radhakrishna Rao, Helge Toutenburg. — New York: Springer-Verlag, Inc., 1999.

Литература для дальнейшего чтения

19. Айвазян, С. А. Основы эконометрики / С. А. Айвазян. — М.: ЮНИТИ, 2001.
20. Айвазян, С. А. Теория вероятностей и математическая статистика / С. А. Айвазян, В. С. Мхитарян. — 2-е изд. — М.: ЮНИТИ-ДАНА, 2001.
21. Алгоритмы и программы восстановления зависимостей // Под ред. В. Н. Вапника. — М.: Наука, 1984.
22. Бард, Я. Нелинейное оценивание параметров / Я. Бард. — М.: Статистика, 1979.
23. Беллман, Р. Введение в теорию матриц / Р. Беллман. — М.: Наука, 1969.
24. Боровков, А. А. Теория вероятностей / А. А. Боровков. — М.: Наука, 1976.
25. Бугров, Я. С. Элементы линейной алгебры и аналитической геометрии / Я. С. Бугров, С. М. Никольский. — М.: Наука, 1984.
26. Валеев, С. Г. Регрессионное моделирование при обработке данных / С. Г. Валеев. — Казань: ФЭН, 2001.
27. Вапник, В. Н. Восстановление зависимостей по экспериментальным данным / В. Н. Вапник. — М.: Наука, 1979.
28. Воеводин, В. В. Матрицы и вычисления / В. В. Воеводин, Ю. А. Кузнецов. — М.: Наука, 1984.
29. Вулих, Б. З. Краткий курс теории функций вещественной переменной (введение в теорию интеграла) / Б. З. Вулих. — М.: Наука, 1973.
30. Вулих, Б. З. Введение в функциональный анализ / Б. З. Вулих. — М.: Наука, 1967.
31. Годунов, С. К. Решение систем линейных уравнений / С. К. Годунов. — Новосибирск: Наука, 1980.
32. Демиденко, Е. З. Линейная и нелинейная регрессии / Е. З. Демиденко. — М.: Финансы и статистика, 1981.
33. Денисов, В. И. Активная параметрическая идентификация стохастических линейных систем / В. И. Денисов, В. М. Чубич, О. С. Черникова, Д. И. Бобылева. — Новосибирск: НГТУ, 2009.

34. Дрейпер, Н. Прикладной регрессионный анализ / Н. Дрейпер, Г. Смит. — М.: Статистика, 1973. — 2-е изд. — М.: Финансы и Статистика, Кн. 1, 1986; Кн. 2, 1987.
35. Дэннис, Дж. Численные методы безусловной оптимизации и решения нелинейных уравнений / Дж. Дэннис, Р. Шнабель. — М.: Мир, 1988.
36. Жданов, А. А. Прямые рекуррентные алгоритмы решения линейных задач метода наименьших квадратов / А. А. Жданов // *ЖВМиМФ*. — 1994. — Т. 34, № 6. — С. 811–814.
37. Жданов, А. А. Прямой последовательный метод решения систем линейных алгебраических уравнений / А. А. Жданов // *Докл. РАН*. — 1997. — Т. 356, № 4. — С. 442–444.
38. Жданов, А. А., Взвешенный метод полных наименьших квадратов / А. А. Жданов, О. П. Шевченко // *Обозрение прикладной и промышленной математики*. — 2001. — Т. 8, № 1. — С. 169–170.
39. Икрамов, Х. Д. Численное решение линейных задач метода наименьших квадратов / Х. Д. Икрамов // *Математический анализ. Итоги науки и техники*, Т. 23. — М.: ВИНТИ, 1985.
40. Иллюстрированный самоучитель по MATLAB / URL: <http://www.matlabing.com/>. Visited 25.12.2011.
41. Захаров, К. В. Динамическая настройка алгоритма обнаружения маневра корабля / К. В. Захаров // *Автоматизация процессов управления*. — 2011. — № 4(26). — С. 23–30.
42. Калиткин, Н. Н. Численные методы / Н. Н. Калиткин. — М.: Наука, 1978.
43. Каханер, Д. Численные методы и программное обеспечение / Д. Каханер, К. Моулер, С. Нэш. — М.: Мир, 2001.
44. Кирьянов, Д. В. MathCad 13 в подлиннике / Д. В. Кирьянов. — СПб.: БХВ, 2003.
45. Кирьянов, Д. В. Вычислительная математика: Электронный учебник / Д. В. Кирьянов. — URL: <http://keldysh.ru/comma/>. Visited 25.12.2011.
46. Коваленко, И. Н. Теория вероятностей и математическая статистика / И. Н. Коваленко, А. А. Филиппова. — М.: Высшая школа, 1973.
47. Колмогоров, А. А. К обоснованию метода наименьших квадратов / А. А. Колмогоров // *Успехи математических наук*. — 1946. — Т. 1, № 1. — С. 57–71.
48. Костомаров, Д. П. Вводные лекции по численным методам: учебное пособие / Д. П. Костомаров, А. П. Фаворский. — М.: Логос, 2004.
49. Крянев, А. В. Математические методы обработки неопределенных данных / А. В. Крянев, Г. В. Лукин. — М.: ФИЗМАТЛИТ, 2003.
50. Крянев, А. В. Статистическая форма регуляризованного метода наименьших квадратов А. Н. Тихонова / А. В. Крянев // *Докл. АН СССР*. — 1986. — Т. 291, № 4. — С. 780–785.
51. Куликова, М. В. Методы вычисления логарифмической функции правдоподобия и ее градиента в алгоритмах калмановской фильтрации: дисс. к.ф.-м.н. / М. В. Куликова. — Ульяновск: УлГУ, 2005.

52. Лапчик, М. П. Численные методы: учебное пособие для студентов вузов / М. П. Лапчик, М. И. Рагулина, Е. К. Хеннер. — 2-е изд., стер. — М.: Издательский центр «Академия», 2005.
53. Линник, Ю. В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений / Ю. В. Линник. — М.: Физматгиз, 1962.
54. Лоусон, Ч. Численное решение задач метода наименьших квадратов / Ч. Лоусон, Р. Хенсон. — М.: Наука, 1986.
55. Лозв, М. Теория вероятностей / М. Лозв. — М.: ИЛ, 1962.
56. Мазмишвили, А. И. Теория ошибок и метод наименьших квадратов / А. И. Мазмишвили. — М.: Недра, 1978.
57. Марчук, Г. И. Методы вычислительной математики / Г. И. Марчук. — М.: Наука, 1989.
58. MATLAB — Образовательный математический сайт Exponenta.ru / URL: <http://exponenta.ru/soft/matlab/matlab.asp>. Visited 25.12.2011.
59. MATLAB — Список литературы на Образовательном математическом сайте Exponenta.ru / URL: http://exponenta.ru/soft/matlab/matlab_book.asp. Visited 25.12.2011.
60. Маттис, А. Оптимальное управление морских подвижных объектов / А. Маттис // *Автоматизация процессов управления*. — 2011. — № 1(23). — С. 88–92.
61. Медич, Дж. Статистически оптимальные линейные оценки и управление / Дж. Медич. — М.: Энергия, 1973.
62. Мэйдоналд, Дж. Вычислительные алгоритмы в прикладной статистике / Дж. Мэйдоналд. — М.: Финансы и статистика, 1988.
63. Невё, Ж. Математические основы теории вероятностей / Ж. Невё. — М.: Мир, 1969.
64. Огарков, М. А. Методы статистического оценивания параметров случайных процессов / М. А. Огарков. — М.: Энергоатомиздат, 1980.
65. Пугачев, В. С. Теория вероятностей и математическая статистика / В. С. Пугачев. — М.: Физматлит, 2002.
66. Рао, С. Р. Линейные статистические методы и их применение / С. Р. Рао. — М.: Наука, 1968.
67. Регуляризация / URL: <http://keldysh.ru/comma/html/inverse/regular.html>. Visited 25.12.2011.
68. Самохвалов, К. М. Статистическое моделирование региональных геопотенциальных полей: дисс. к.т.н. / К. М. Самохвалов. — Ульяновск: УлГТУ, 2004.
69. Себер, Д. Линейный регрессионный анализ / Д. Себер. — М.: Мир, 1980.
70. Семушин, И. В. Алгоритмы решения обратного уравнения Риккати для дискретных задач управления / И. В. Семушин, Ю. В. Цыганова, Н. Д. Старостина // *Автоматизация процессов управления*. — 2012. — № 3(29) [в разработке].

71. Семушин, И. В. Параметрическая идентификация модели погрешностей инерциальных навигационных систем / И. В. Семушин, Ю. В. Цыганова // *Автоматизация процессов управления*. — 2011. — № 4(26). — С. 15–22.
72. Семушин, И. В. Адаптивный квадратно–корневой ковариационный алгоритм фильтрации / И. В. Семушин, Ю. В. Цыганова // *Автоматизация процессов управления*. — 2011. — № 1(23). — С. 83–87.
73. Семушин, И. В. Адаптивные системы фильтрации, управления и обнаружения / И. В. Семушин (Гл. 1 и Гл. 2), Ю. В. Цыганова (Гл. 3), М. В. Куликова (Гл. 4), О. А. Фатьянова (Гл. 5), А. Е. Кондратьев (Гл. 5). — Ульяновск: УлГУ, 2011.
74. Семушин, И. В. Численные методы алгебры: учебное пособие для вузов / И. В. Семушин. — Ульяновск: УлГТУ, 2006.
75. Семушин, И. В. Детерминистские модели динамических систем: учебное пособие / И. В. Семушин, Ю. В. Цыганова. — Ульяновск: УлГТУ, 2006.
76. Семушин, И. В. Практикум по методам оптимизации. Компьютерный курс: учебное пособие для вузов / И. В. Семушин. — 3-е изд, перераб. и доп. — Ульяновск: УлГТУ, 2005.
77. Семушин, И. В. Сборник лабораторных работ и контрольных тестовых заданий по вычислительной линейной алгебре: учебное пособие / И. В. Семушин, Г. Ю. Куликов. — Ульяновск: УлГТУ, 2000.
78. Семушин, И. В. Эффективные алгоритмы обновления оценок по измерениям / И. В. Семушин // *Судостроительная промышленность*, сер. Вычислительная техника. — 1992. — № 28. — С. 55–59.
79. Семушин, И. В. Экстраполяция по времени *LD*-ковариационных факторов для фильтра Калмана // *Судостроительная промышленность*, сер. Вычислительная техника. — 1992. — № 28. — С. 27–30.
80. Семушин, И. В. Разработка методов, программ и практических рекомендаций для обработки экспериментальных данных на ЭВМ общего и специального назначения: научно-технический отчет № 14-129.8. Номер гос. регистрации СССР 80042284, код «Сигма» // И. В. Семушин. — Ульяновск: УлПИ, 1983.
81. Тихонов, А. Н. Методы решения некорректных задач / А. Н. Тихонов, В. Я. Арсенин. — 2-е изд. — М.: Наука, 1979.
82. Уидроу, Б. Адаптивная обработка сигналов / Б. Уидроу, С. Стирнз. — М.: Радио и Связь, 1989.
83. Феллер, В. Введение в теорию вероятностей и ее приложения / В. Феллер, Т. 1, 2. — М.: Мир, 1967.
84. Форсайт, Дж. Численное решение систем линейных алгебраических уравнений / Дж. Форсайт, К. Молер. — М.: Мир, 1969.
85. Халмош, П. Р. Конечномерные векторные пространства / П. Р. Халмош. — М.: Физматгиз, 1963.

86. Чернодаров, А. В. Применение модифицированной ортогонализации Грама–Шмидта для построения вычислительно устойчивых алгоритмов адаптивной фильтрации / А. В. Чернодаров, И. А. Белозеров, И. А. Москалевич // Научно-методические материалы по авиационному оборудованию, вып. 7. Под ред. М. К. Духовного. — Рига: РВВАИУ им. Я. Алксниса, 1985. — С. 62–68.
87. Четыркин, Е. М. Вероятность и статистика / Е. М. Четыркин, Е. Л. Калихман. — М.: Финансы и статистика, 1982.
88. Щиголев, Б. М. Математическая обработка наблюдений / Б. М. Щиголев. — М.: Наука, 1969.
89. Bar-Itzhack, I. V. Efficient Square Root Algorithm for Measurement Update in Kalman Filtering / I. V. Bar-Itzhack, Y. Medan // *Journal of Guidance, Control, and Dynamics*. — 1983. — Vol. 6, No. 3. — P. 129–134. [Русский перевод: Бар-Ицхак, И. И. Эффективный алгоритм коррекции измерений в фильтре Калмана / И. И. Бар-Ицхак, И. Мэден // *Аэрокосмическая техника*. — 1984. — Т. 2, № 1. — С. 141–147.]
90. Bar-Shalom, Y. Estimation with Application to Tracking and Navigation / Y. Bar-Shalom, X. R. Li, T. Kurubaran. — New York: John Wiley & Sons, 2001.
91. Brockmann, E. Combination of Solutions for Geodetic and Geodynamic Applications of the Global Positioning System (GPS) / E. Brockmann // *Geodatisch-geophysikalische Arbeiten in der Schweiz*, Vol. 55. — Schweizerische Geodatische Kommission, 1997.
92. Caines, P. E. Linear stochastic systems / P. E. Caines. — New York, Chichester, Brisbane, Toronto, Singapore: John Wiley & Sons, 1988.
93. Carraro, C. Square Root Iterative Filter: Theory and Applications to Econometric Models / C. Carraro, D. Sartore // *Annales d'Economie et de Statistique*. — 1987. — No. 6/7. — P. 435–459.
94. Chung, K. L. A Course in Probability Theory / K. L. Chung. — New York: Harcourt, Brace and World, 1968.
95. Daowang, F. Square-Root Second-Order Extended Kalman Filter and Its Application in Target Motion Analysis / F. Daowang, L. Teng, H. Z. Tao // *IET Radar, Sonar & Navigation*. — 2008. — No. 4, Iss. 3. — P. 329–335. doi 10.1049/iet-rsn.2008.0070.
96. Duff, I. S. Direct Methods for Sparse Matrices / I. S. Duff, A. M. Erisman, J. K. Reid. — Oxford: Clarendon Press, 1986.
97. Farebrother, R. W. Linear Least Squares / R. W. Farebrother. — New York and Basel: Marcel Dekker, Inc., 1988.
98. Fitzgerald, R. J. Divergence of the Kalman Filter / R. J. Fitzgerald // *IEEE Trans. on Automatic Control*. — 1971. — Vol. AC-16, No. 6. — P. 736–747.
99. Fröberg, C. E. Introduction to Numerical Analysis / C. E. Fröberg. — Reading, Massachusetts · Menlo Park, California · London · Don Mills, Ontario: Addison-Wesley, 1969.
100. Golub G. N. An Analysis of the Total Least Squares Problem / G. N. Golub, C. F. Van Loan // *SIAM J. Numer. Anal.*. — 1980. — No. 17. — P. 883–893.

101. Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems / R. E. Kalman // *ASME Journal of Basic Engineering*. — 1960. — Vol. 82. — P. 34–45.
102. Hackbusch, W. Multi-grid methods and applications / W. Hackbusch. — New York: Springer Verlag, 1985.
103. Kaminski, P. G. Square Root Filtering and Smoothing for Discrete Processes: Ph.D. thesis / P. G. Kaminski. — Stanford University, 1971.
104. Kaminski, P. G. Discrete Square Root Filtering: A Survey of Current Techniques / P. G. Kaminski, A. E. Bryson, S. F. Schmidt // *IEEE Trans. on Automatic Control*. — 1971. — Vol. AC-16. — P. 727–736. [Русский перевод: *Зарубежная радиоэлектроника*. — 1973. — № 6. — С. 37–53.]
105. Kapinski, M. Measure, Integral and Probability / M. Kapinski, P. E. Kopp. — 2nd ed. — London: Springer, 2004.
106. Kariya, T. Generalized Least Squares / Takeaki Kariya, Hiroshi Kurata. — Chichester: John Wiley & Sons, 2004.
107. Kulikova, M. V. On Scalarized Calculation of the Likelihood Function in Array Square-Root Filtering Algorithms / M. V. Kulikova // *Automation and Remote Control*. — 2009. — Vol. 70, No. 5. — P. 855–871.
108. Kulikova, M. V. Likelihood Gradient Evaluation Using Square-Root Covariance Filters / M. V. Kulikova // *IEEE Trans. on Automatic Control*. — March 2009. — Vol. 54, No. 3. — P. 646–651.
109. Lancaster, P. Algebraic Riccati Equations / P. Lancaster, L. Rodman. — New York: Oxford University Press, Inc., 1995.
110. Lange, A. A. An Algorithmic Approach for Improving and Controlling the Quality of Upper-Air Data / A. A. Lange // *WMO Instruments and Observing Methods Report*. — 1989. — No. 35. — World Meteorological Organization, Geneva, Switzerland.
111. Lange, A. A. Optimal Kalman Filtering for Ultra-Reliable Tracking, ESA CD-ROM WPP-237 / A. A. Lange // *Atmospheric Remote Sensing Using Satellite Navigation Systems*. — Proceedings, Special Symposium of the URSI Joint Working Group FG, 13–15 October 2003, Matera, Italy.
112. Lawson, C. L. Solving Least Squares Problems / C. L. Lawson, R. J. Hanson. — New Jersey: Prentice-Hall, Engle-wood Cliffs, 1974.
113. Lewis, F. L. Optimal Estimation with an Introduction to Stochastic Control Theory / F. L. Lewis. — New York: John Wiley & Sons, 1986.
114. Li, X. R. Survey of Maneuvering Target Tracking. Part I: Dynamic Models / X. R. Li, V. P. Jilkov // *IEEE Trans. on Aerospace and Electronic Systems*. — October 2003. — Vol. 39, No. 4. — P. 1333–1364.
115. Maybeck, P. S. Stochastic Models, Estimation, and Control, Volume 1 / P. S. Maybeck. — New York, San Francisco, London: Academic Press, Inc., 1979.
116. Maybeck, P. S. Stochastic Models, Estimation, and Control, Volume 2 / P. S. Maybeck. — New York, San Francisco, London: Academic Press, Inc., 1982.

117. Maybeck, P. S. Stochastic Models, Estimation, and Control, Volume 3 / P. S. Maybeck. — New York, San Francisco, London: Academic Press, Inc., 1982.
118. McCormick, S. F. (Ed.) Multigrid methods / S. F. McCormick. — New York: Marcel Dekker, 1988.
119. Morf M., Kailath T. Square-Root Algorithms for Least-Squares Estimation / M. Morf, T. Kailath // *IEEE Trans. on Automatic Control*. — 1975. — Vol. AC-20, No. 4. — P. 487–497.
120. Mosca, E. Optimal, Predictive, and Adaptive Control / E. Mosca. — New Jersey: 1995.
121. Murgu, A. Neural Networks for Planning and Control in Communication Networks: PhD thesis / A. Murgu. — University of Jyväskylä, Finland, 1995.
122. Noble, B. Applied Linear Algebra / B. Noble, J. W. Daniel. — Englewood Cliffs, New Jersey: Prentice-Hall, 1977.
123. Park, P., Kailath, T. New Square-Root Algorithms for Kalman Filtering / P. Park, T. Kailath // *IEEE Trans. on Automatic Control*. — 1995. — Vol. AC-40, No. 5. — P. 895–899.
124. Piovoso, M., Laplante, P. A. Kalman Filter Recipes for Real-Time Image Processing / M. Piovoso, P. A. Laplante // *Real-Time Imaging—Special Issue on Software Engineering Archive*. — December 2003. — Vol. 9, Iss. 6.
125. Potter, J. E., Stern, R. G. Statistical Filtering of Space Navigation Measurements / J. E. Potter, R. G. Stern // *Proceedings of 1963 AIAA Guidance and Control Conference*. — 1963. — New York: AIAA.
126. Saad, Y. Iterative Solution of Linear Systems in the 20th Century / Y. Saad, H. Van der Vorst // *J. Comput. Appl. Math.*. — 2000. — No. 123. — P. 1–33.
127. Saad, Y. Iterative Methods for Sparse Linear Systems / Y. Saad. — 2nd ed. — SIAM, 2003.
128. Sage, A. P. System Identification / A. P. Sage, J. L. Melsa. — New York: Academic Press, 1971. [Translated into Russian, ed. by N. S. Raibman. — M.: Nauka, 1974.]
129. Schmidt, S. F. Computational Techniques in Kalman Filtering / S. F. Schmidt // In: *Theory and Applications of Kalman Filtering*, NATO Advisory Group for Aerospace Research and Development, AGARDograph 139, Feb. 1970.
130. Semushin, I. V. Personal website / I. V. Semushin — URL: <http://staff.ulsu.ru/semushin/>. Visited 25.12.2011.
131. Semushin, I. V. Adaptation in Stochastic Dynamic Systems — Survey and New Results II / I. V. Semushin // *Int. J. Communications, Network and System Sciences*. — 2011. — Vol. 4, No. 4. — P. 266–285.
132. Semushin, I. V. Score Evaluation Within the Extended Square-Root Information Filter / I. V. Semushin, M. V. Kulikova // *Lecture Notes in Computer Science*. — 2006. — Vol. 3991, Pt. 1. — P. 473–481.
133. Semushin, I. V. On the Evaluation of Log Likelihood Gradient for Gaussian Signals / I. V. Semushin, M. V. Kulikova // *International Journal of Applied Mathematics & Statistics*. — 2005. — Vol. 3, No. S05. — P. 1–14.

134. Semushin, I. V. Computational and Soft Skills Development Through the Project Based Learning / I. V. Semushin, Ju. V. Tsyganova, V. V. Ugarov // *Lecture Notes in Computer Science*. — 2003. — Vol. 2658, Pt. 2. — P. 1098–1106.
135. Semushin, I. V. Stable Estimate Renewal According to Measurements / I. V. Semushin, E. V. Dulov, L. V. Kalinin // *Pattern Recognition and Image Analysis*, Conference Proceedings. — St. Petersburg–Moscow: MAIK «Nauka / Interperiodica» Publishing. — 1996. — Vol. 6, No. 1. — P. 86.
136. Shewchuk, J. An Introduction to the Conjugate Gradient Method without the Agonizing Pain / J. Shewchuk. — Technical Report No. CMU-CS-94-125. — Carnegie Mellon University, 1994.
137. Stigler, S. M. An Attack on Gauss, Published by Legendre in 1820 / S. M. Stigler // *Historia Mathematica*. — 1977. — No. 4. — P. 31–35.
138. Tsyganova, Yu. V. Computing the Gradient of the Auxiliary Quality Functional in the Parametric Identification Problem for Stochastic Systems / Yu. V. Tsyganova // *Automation and Remote Control*. — 2011. — Vol. 72, No. 9. — P. 1925–1940. [Original Russian Text published in *Avtomatika i Telemekhanika*. — 2011. — No. 9. — P. 142–160.]
139. Van der Vorst, H. Iterative Krylov Methods for Large Linear Systems / H. Van der Vorst. — Cambridge: Cambridge University Press. — 2003.
140. Verkhovsky, B. Algorithm with Nonlinear Acceleration for a System of Linear Equations / B. Verkhovsky. — Research Report. — No. 76-WR-1. — Princeton University, 1976.
141. Veroy (now Verkhovsky), B. Convergence and Complexity of Aggregation–Disaggregation Algorithm / B. Veroy (now Verkhovsky). — Research Report. — No. CS-87-04. — New Jersey Institute of Technology, 1987.
142. Verkhovsky, B., Polyakov, Yu. Feedback Algorithm for the Single-facility Minisum Problem / B. Verkhovsky, Yu. Polyakov // *Annals of the European Academy of Sciences*. — 2003. — P. 127–136.
143. Watkins, D. S. The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods / D. S. Watkins. — SIAM, 2007.
144. Wesseling, P. An Introduction to Multigrid Methods / P. Wesseling. — Chichester: John Wiley & Sons, 1992.

Предметный указатель

А

Аги и Тернер (Agee and Turner) 278

алгоритм

- Бар-Ицхака сокращенный (reduced Bar-Itzhack algorithm) 289
- Бирмана (Bierman algorithm) 281
- Бирмана сокращенный (reduced Bierman algorithm) 287
- быстрый квадратно-корневой (fast square root filter algorithm) 319
- векторных сумм (vector sums algorithm) 67
- Джозефа скаляризованный (scalarized Joseph algorithm) 250
- Джозефа стабилизированный (Joseph stabilized algorithm) 275
- Донгарры–Айзенштата (Dongarra–Eisenstat algorithm) 76
- Калмана скаляризованный (scalarized Kalman algorithm) 249
- Калмана стабилизированный (stabilized Kalman algorithm) 250
- Калмана стандартный (standard Kalman algorithm) 247
- Карлсона обновления по измерениям (Carlson measurement update algorithm) 285
- $L\bar{U}$ компактной схемы Краута (Crout compact schemata, $L\bar{U}$ algorithm) 35, 36
- $\bar{L}U$ компактной схемы Краута (Crout compact schemata, $\bar{L}U$ algorithm) 36
- Крылова с предобуславливанием (Krylov algorithm with preconditioning) 156
- метода Жордана (Jordan method algorithm) 37
- модифицированного квадратно-корневого информационного фильтра (modified square root information filter algorithm) 323
- обратной подстановки (back substitution algorithm) 67

- оценивания ковариационный (covariation estimation algorithm) 261
- оценивания информационный (information estimation algorithm) 262
- Поттера (Potter algorithm) 251
- прямой подстановки (forward substitution algorithm) 69
- расширенного квадратно-корневого информационного фильтра (extended square root information filter algorithm) 322
- столбцовый (column algorithm) 67
- фильтра Поттера (Potter filter algorithm) 277
- Хаусхолдера столбцово ориентированный (column based Hausholder algorithm) 116
- Хаусхолдера строчно ориентированный (row based Hausholder algorithm) 116

Б

- бидиагонализация квадратной матрицы (square matrix bidiagonalization) 129, 130
- блочные алгоритмы с ортогонализацией (array orthogonalization algorithms) 316

В

- вариационные методы (variation methods) 150
- ведущий столбец
- в преобразовании Гивенса (leading column in Givens transform) 122
 - в преобразовании Хаусхолдера (leading column in Hausholder transform) 114
- векторные регистры (vector registers) 59
- Винер (Wiener) 272
- выбор ведущего (главного) элемента (pivoting) 33
- по активной подматрице (active submatrix based pivoting) 32
 - по столбцу (column based pivoting) 31, 54
 - по строке (row based pivoting) 31

- разреженной матрицы оптимальный (optimal sparse matrix pivoting) 85

Г

Галилей (Galilei) 272

Гаусс (Gauss) 109, 199, 272

геометрическая интерпретация проектирования (geometric interpretation for projection) 230

Грама–Шмидта (Gram–Schmidt)

- ортогонализация (Gram-Schmidt orthogonalization) 132
- — взвешенная (weighted Gram-Schmidt orthogonalization) 280
- — модифицированная (modified Gram-Schmidt orthogonalization) 132
- — — с выбором ведущего столбца (modified Gram-Schmidt orthogonalization with pivoting) 133

гребневое оценивание (ridge estimation) 302

Д

декорреляция и скаляризация (decorrelation and scalarization) 270

диагональное преобладание (diagonal prevailing) 91

З

задача

- линейных наименьших квадратов (linear least squares problem) 109
- ЛКГ-управления (LQG-control problem) 344
- обобщенная на собственные значения (generalized eigenvalue problem) 148
- QR -разложения матрицы (problem of matrix QR -decomposition) 133
- управления линейно-квадратическая гауссова (linear quadratic gaussian control problem) 344

заполнение локальное (local filling) 84

И

исключение

- по столбцам (column based elimination) 32
- по строкам (row based elimination) 33
- гауссово по строкам (Gauss row based elimination with row based pivoting) 33

— полное (full elimination) 37

итерационный метод (ИМ) (iterative method) 141

— Зейделя (iterative Seidel method) 143

— многошаговый (multi step iterative methods) 145

— нестационарный (non-stationary IM) 142, 145

— неявный (indirect IM) 141

— одношаговый (one-step iterative methods) 145

— — стационарный в терминах погрешности (one-step stationary IM in terms of errors) 146

— стационарный (stationary IM) 142, 145

— явный (direct method) 141, 142, 150

— Якоби (iterative Jacobi method) 143

информационная форма последовательного МНК (information form of sequential LS method) 245

источник данных (data source) 272

итерационный параметр (iteration parameter) 145

итерационная формула (iteration formula) 142

К

квадратичная форма (quadratic form) 90

квадратный корень матрицы (matrix square root) 91

Калман (Kalman) 272

каноническая форма одношагового ИМ (canonical form of one-step iterative method) 145

ковариация (covariation) 241

— отфильтрованных оценок (filtered estimates covariance) 346

— экстраполяционных оценок (extrapolated estimates covariance) 345

компактная схема (compact schemata) 34

— Краута (Crout compact schemata) 34

— — строчно ориентированная (row based Crout compact schemata) 34

конвейеризация (conveyerization) 56

коэффициенты корреляции (correlation coefficients) 91

критерий

— качества квадратический (quadratic performance criterion) 109

— качества оценщика (estimator performance criterion) 317

- максимума апостериорной вероятности (maximum aposterioiri probability criterion) 257
- максимума правдоподобия (maximum likelihood criterion) 258
- наименьших квадратов (least squares criterion) 300
- — взвешенный (weighted least squares criterion) 262
- остановки (stopping criterion) 143
- Сильвестера (Silvester criterion) 91

Л

- Лежандр (Legendre) 109, 199
- линейная независимость системы (linear independence of a system) 200
- линейная оболочка (linear span) 205
- — векторов (linear vector span) 201
- линейное подпространство в \mathbb{R}^n (Linear subspace in \mathbb{R}^n) 200

М

- матрица
- идемпотентная (idempotent matrix) 112
- информационная (information matrix) 241, 262
- ковариационная (covariation matrix) 262
- лидирующая (leading matrix) 141, 145
- обратная (inverse matrix) 41
- ортогональная (orthogonal matrix) 107, 233
- перестановок (purturbation matrix) 30
- — элементарная (elementary purturbation matrix) 30
- переходная погрешности (error transition matrix) 147
- плохо обусловленная (ill-conditioned matrix) 47, 149
- положительно определенная (positive definite matrix) 90
- проекционная (projection matrix) 207, 225
- псевдообратная (pseudo-inverse matrix) 109, 211, 212, 239
- симметрическая (symmetric matrix) 112
- унитарная (unitary matrix) 233
- Хаусхолдера (Hausholder matrix) 112
- элементарная (elementary matrix) 37
- — специальная (special elementary matrices) 37

- эрмитова (Hermitian matrix) 233
- метод
- верхней релаксации (Succesive Over-Relaxation, SOR method) 146
- Гаусса, полный шаг (full step of Gauss method) 28
- Гаусса, прямой ход (forward move of Gauss method) 29
- Гаусса, обратный ход (back move of Gauss method) 30
- главных компонент (principal components method) 305
- делинеаризации (delinearization method) 156
- минимальных невязок (minimum residual method) 151
- — поправок (minimum correction method) 152
- многосеточный (multi-grid method) 156
- наименьших квадратов (МНК) (least squares method) 109, 137, 236
- простой итерации (simple iteration method) 145
- регуляризации Тихонова (Tikhonov's regularisation method) 302
- Ричардсона (Richardson method) 146
- скорейшего спуска (quickest descent method) 153
- сопряженных градиентов (conjugate gradients method) 154
- Юнга (Young method) 146
- минимальная ожидаемая стоимость завершения процесса управления (minimal expected cost for completing the control process) 346
- миноры, главные (main minors) 28
- модификации
- немедленные (immediate modifications) 63
- отложенные (delayed modifications) 64
- мультиколлинеарность (multicollinearity) 300

Н

- направляющий вектор (direction vector) 111
- невязка (residual) 109, 150
- норма вектора (vector norm) 201
- обобщенная (generalized vector norm) 148
- норма матрицы типа «бесконечность» («infinity» matrix norm) 50
- нормализованная погрешность (normalized error) 241

нормализованные экспериментальные данные (normalized experimental data) 240
 нормализованные псевдоизмерения (normalized pseudo measurements) 296
 нормальная система (normal system) 301
 нормальное псевдорешение (normal pseudo-solution) 110, 137, 210, 239, 261, 301
 нормальные уравнения (normal equations) 109, 137, 209, 238, 261
 нормировка (norming) 27
 нуль-пространство матрицы (matrix zero space) 206
 — левое (left matrix zero space) 207

О

обновление
 — матрицы (matrix update) 54
 — одноранговое (one-rank update) 278
 — по времени (temporal update) 252
 — системы (system update) 28
 обратная подстановка (back substitution) 111
 обращение верхней треугольной матрицы (inverting upper triangular matrix) 118
 окаймление (bordering) 73
 — известной части LU -разложения (bordering the known part of LU decomposition) 74
 — неизвестной части LU -разложения (bordering the unknown part of LU decomposition) 76
 оптимальный итерационный параметр (optimal iteration parameter) 149
 оптимальный линейный регулятор (optimal linear regulator, OLR) 346
 ортогонализация (orthogonalization) 271
 ортогональное дополнение (orthogonal complement) 201
 ортогональность векторов (vector orthogonality) 201
 ортогональные векторы (orthogonal vectors) 205
 отраженный вектор (reflected vector) 112
 оценщик (estimator) 259
 — байесовский (Bayesian estimator) 259
 — максимума апостериорной вероятности (maximum a posteriori probability estimator) 263
 — максимума правдоподобия (maximum likelihood estimator) 262

— оптимальный минимум среднеквадратический (optimum mean square estimator) 317
 — фишеровский (Fisher estimator) 259
 оценка
 — оптимальная (optimum estimate) 317
 — отфильтрованная (filtered estimate) 317, 345
 — сглаженная (smoothed estimate) 317
 — экстраполяционная (predicted estimate) 317, 345

П

переопределенная система (overdetermined system) 137
 перестановка неявная (indirect interchanging) 54
 погрешность
 — методическая (method error) 141
 — округления (roundoff error) 142
 — трансформированная (transform error) 142
 подпространство (subspace) 205
 — собственное (proper subspace) 205
 подстановка
 — прямая (forward substitution) 29, 53
 — обратная (back substitution) 29, 53
 полная статистическая интерпретация МНК (full statistical LS method interpretation) 252
 поправка ИМ (correction of IM) 151
 предиктивная форма фильтра Калмана (predicted form of Kalman filter) 318
 предупреждение
 — о термине $L\bar{U}^{-1}$ -«разложение» ($L\bar{U}^{-1}$ «decomposition» term precaution) 40
 — о термине $\bar{L}^{-1}U$ -«разложение» ($\bar{L}^{-1}U$ «decomposition» term precaution) 41
 преобразование
 — Хаусхолдера (Householder transform) 111
 — Гивенса (Givens transform) 120
 — — строчно ориентированное (row based Givens transform) 124
 — — столбцово ориентированное (column based Givens transform) 124
 принцип оптимальности (optimality principle) 345
 проектно ориентированное изучение (project based learning) 338
 проектор (projector) 211
 проекция вектора (vector projection) 111, 204

— на $\mathfrak{L}(y)$ (vector projection on $\mathfrak{L}(y)$) 208
 пространство
 — евклидово (Euclidean space) 201
 — Крылова (Krylov space) 156
 — столбцов матрицы (matrix column space) 206
 — строк матрицы (matrix space of rows) 206
 процедура ортогонализации системы векторов (vector system orthogonalization procedure) 131
 псевдоизмерения (pseudo measurements) 294
 псевдонаблюдения (pseudo-observations) 342

Р

разложение
 — LU (LU -decomposition) 27
 — $L\bar{U}$ ($L\bar{U}$ -decomposition) 29
 — — гауссовым исключением по строкам (Gauss $L\bar{U}$ decomposition with row based elimination) 33
 — — жордановым исключением, $L\bar{U}^{-1}$ -«разложение» (« $L\bar{U}^{-1}$ decomposition» $A = L\bar{U}$ by Jordan method) 40
 — — с выбором главного элемента ($L\bar{U}$ decomposition with pivoting) 32
 — $\bar{L}U$ ($\bar{L}U$ -decomposition) 29
 — — с выбором главного элемента ($\bar{L}U$ decomposition with pivoting) 32
 — сингулярное матрицы (matrix singular decomposition) 215
 — Холецкого (Cholesky decomposition) 92
 — — нижнее треугольное (lower triangular Cholesky decomposition) 92
 — — — без квадратных корней (square root free lower triangular Cholesky decomposition) 93
 — — верхнее треугольное (upper triangular Cholesky decomposition) 93
 — — — без квадратных корней (square root free upper triangular Cholesky decomposition) 94
 размерность пространства (dimension of a space) 200
 ранг матрицы (matrix rank) 206
 расстояние (distance) 201
 расходимость (divergence) 270
 расширенная система (extended system) 242
 расщепление системы (system splitting) 243
 регрессионное моделирование (regression modeling) 273

регрессия (regression) 236
 решение
 — задачи оптимального оценивания (solution to the problem of optimal estimation) 273
 — одновременное (simutaneous solution) 264
 — по МНК (МНК-решение) (least squares solution) 209, 238, 301

С

свойство
 — идемпотентности (idempotency feature) 31
 — несмещенности (unbiasedness property) 241
 — ортогональности (orthogonality feature) 31
 сингулярные числа матрицы (singular numbers of a matrix) 215
 скаляризация (scalarization) 270
 скалярное произведение (scalar product) 67, 201
 скалярный параметр (scalar parameter) 141
 собственное значение (eigenvalue) 233
 собственный вектор (eigenvector) 233
 соотношения двойственности (duality relations) 347
 спектральный радиус матрицы (spectral radius of matrix) 147
 степень параллелизма (parallelization power) 56
 — средняя (mean parallelization power) 57
 стохастический регулятор (stochastic regulator) 346
 стратегии выбора ведущего (главного) элемента (pivoting strategies) 31
 сходимость итерационного метода (convergence of iterative method) 141, 147

Т

таблица множителей (table of multipliers) 39
 теорема
 — Гаусса–Маркова (Gauss–Markov theorem) 263
 — линейной алгебры основная (main theorem of Linear Algebra) 206
 — об ортогональной проекции (theorem of orthogonal projection) 205
 — об ортогональном разложении пространства (theorem of space orthogonal decomposition) 203
 — о разложении Фурье (Fourie decomposition theorem) 208
 — о сингулярном разложении (singular decomposition theorem) 215

— о спектральном разложении эрмитовой матрицы (theorem of Hermitian matrix spectral decomposition) 234
 теория гильбертовых пространств (Hilbert spaces theory) 204
 требование к реализации (implementation requirement) 32
 триада (triada) 54
 триангуляризация матрицы
 — преобразованиями Хаусхолдера (matrix triangularization by Hausholder transform) 113, 114
 — преобразованиями Гивенса (matrix triangularization by Givens transform) 121, 122
 тридиагонализация симметрической матрицы (symmetric matrix tridiagonalization) 128, 129

У

упакованная форма матрицы (packed matrix forms) 82
 уравнение
 — дискретное алгебраическое Риккати (Discrete Algebraic Riccati Equation) 275
 — обратное алгебраическое Риккати (inverse algebraic Riccati equation) 347
 — прямое алгебраическое Риккати (direct algebraic Riccati equation) 347
 — функциональное Беллмана (Bellman functional equation) 345
 условие полного столбцового ранга (full column rank condition) 109

Ф

факторизация (factorization) 270
 фильтр
 — Бар-Ицхака сокращенный (Bar-Itzhack–Medan reduced filter) 290
 — Калмана (Kalman filter) 273
 — — ковариационный квадратно-корневой (square root covariance Kalman filter) 319
 — — обыкновенный (the conventional Kalman filter) 270
 — — расширенный (Extended Kalman Filter, EKF) 296

— — со скалярным обновлением по измерениям (Kalman filter with scalar measurement update) 274
 — — стандартный ковариационный (standard Kalman covariance filter, SCF) 317, 339, 340
 — оптимальный линейный (Калмана) (optimal linear (Kalman) filter, KF) 345
 — Поттера квадратно-корневой (Potter Square Root Filter) 270
 — расширенный (extended filter) 272
 — — второго порядка (second order extended filter) 300
 — — ковариационный квадратно-корневой (extended square root covariance filter) 320
 — — первого порядка (first order extended filter) 297
 — скаляризованный информационный квадратно-корневой (СККИФ) (scalarized square root information filter) 328
 — скаляризованный ковариационный квадратно-корневой (СКККФ) (scalarized square root covariance filter) 327
 — скаляризованный комбинированный квадратно-корневой (СКоККФ) (scalarized combined square root filter) 329
 — скаляризованный модифицированный квадратно-корневой информационный (СМККИФ) (scalarized modified square root covariance filter) 329
 функция Беллмана (Bellman function) 344

Х

характеристическое уравнение (characteristic equation) 233
 хранение матриц одномерное (linear (one dimension) matrix storing) 101

Э

элиминативная форма обратной матрицы (eliminative form of inverse matrix) 41
 экономия памяти (saving memory) 29
 элемент допустимый (feasible element) 84

Учебное издание

Семушин Иннокентий Васильевич

**ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ
И ОЦЕНИВАНИЯ**

Учебное пособие

Редактор М. А. Теленкова

ЛР № 020640 от 22.10.97

Оригинал-макет изготовлен автором в системе L^AT_EX2_ε

Подписано в печать 20.12.2011. Формат 60×84/16. Усл. печ. л. 21,62.

Гарнитура Computer Modern. Тираж 200 экз. Заказ № 85

ЭИ № 158.

Ульяновский государственный технический университет

432027, г. Ульяновск, ул. Северный Венец, д. 32

Типография УлГТУ, 432027, г. Ульяновск, ул. Северный Венец, д. 32