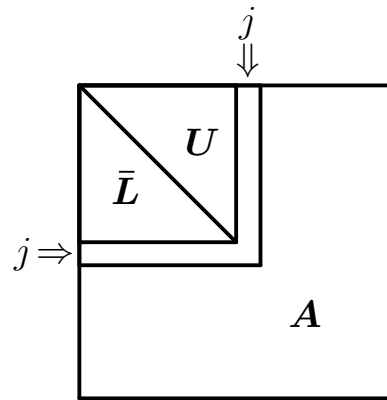


И. В. Семушин

ЧИСЛЕННЫЕ МЕТОДЫ АЛГЕБРЫ



Ульяновск
2006

Федеральное агентство по образованию
Государственное образовательное учреждение высшего
профессионального образования
Ульяновский государственный технический университет

И. В. Семушин

ЧИСЛЕННЫЕ МЕТОДЫ АЛГЕБРЫ

Ульяновск
2006

УДК 519.61
ББК 73я7
С30

Рецензенты: д-р физ.-мат. наук, профессор С. П. Мищенко
канд. техн. наук, доцент В. В. Селиванов

Утверждено редакционно-издательским советом университета
в качестве учебного пособия

Семушин И. В.

С30 Численные методы алгебры : учебное пособие для вузов. — Ульяновск : УлГТУ, 2006. — 178 с.
ISBN 5-89146

Первая часть текста (разд. 2–7) содержит необходимые теоретические сведения и задания на 6 лабораторных проектов по основным темам линейной алгебры. Во второй части (разд. 8–9) приведены задачи, контрольные задания и рабочая программа курса.

Для студентов, изучающих численные методы по программе подготовки в области математики, экономики и управления, естественных или технических наук по специальностям: 010101 – «Математика», 010501 – «Прикладная математика и информатика», 010503 – «Математическое обеспечение и администрирование информационных систем», 210400 – «Телекоммуникации», 230200 – «Информационные системы», 230301 – «Моделирование и исследование операций в организационно-технических системах», 080801 – «Прикладная информатика (по областям)», 090105 – «Комплексное обеспечение информационной безопасности автоматизированных систем» и другим.

УДК 519.61
ББК 73я7

ISBN 5-89146

© Оформление. УлГТУ, 2006
© И. В. Семушин, 2006

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

—Аристотель

Оглавление

Предисловие	7
1 ВВЕДЕНИЕ	9
1.1 Учебные цели студента	9
1.2 Оценка работы студента	10
1.3 Кодекс студента	15
1.4 Краткое описание курса	17
2 СТАНДАРТНЫЕ АЛГОРИТМЫ LU-РАЗЛОЖЕНИЯ	19
2.1 Алгоритмы метода Гаусса	19
2.2 Выбор ведущего элемента	21
2.3 Компактные схемы	23
2.4 Алгоритмы метода Жордана	26
2.5 Вычисление обратной матрицы	28
2.6 Плохо обусловленные матрицы	34
2.7 Задание на лабораторный проект № 1	35
2.8 Варианты задания	38
3 СОВРЕМЕННЫЕ АЛГОРИТМЫ LU-РАЗЛОЖЕНИЯ	40
3.1 Гауссово исключение и ijk -алгоритмы	40
3.2 Распараллеливание вычислений	42
3.3 Параллельное умножение матрицы на вектор	45
3.4 Параллельное LU -разложение	46
3.5 LU -разложение и его ijk -формы	49
3.6 Треугольные системы	56
3.7 Задание на лабораторный проект № 2	57
3.8 Варианты задания	59
4 АЛГОРИТМЫ ОКАЙМЛЕНИЯ В LU-РАЗЛОЖЕНИИ	61
4.1 Метод окаймления	61

4.2	Окаймление известной части разложения	61
4.3	Окаймление неизвестной части разложения	64
4.4	Задание на лабораторный проект № 3	66
4.5	Варианты задания	69
5	РАЗРЕЖЕННЫЕ ФОРМЫ LU-РАЗЛОЖЕНИЯ	70
5.1	Упакованные формы хранения матриц	70
5.2	Выбор ведущего элемента	72
5.3	Задание на лабораторный проект № 4	75
5.4	Варианты задания	76
6	РАЗЛОЖЕНИЯ ХОЛЕССКОГО	78
6.1	Положительно определенные матрицы	78
6.2	Квадратные корни из P и алгоритмы Холесского	79
6.3	Программная реализация алгоритмов Холесского	82
6.4	Разложение Холесского: ijk -формы	84
6.5	Разложение Холесского: алгоритмы окаймления	86
6.6	Задание на лабораторный проект № 5	89
6.7	Варианты задания	92
7	ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ	94
7.1	Ортогональные матрицы и приложения	94
7.2	Линейная задача наименьших квадратов	96
7.3	Ортогональные матрицы в задаче о наименьших квадратах	97
7.4	Преобразование Хаусхолдера	98
7.5	Шаг триангуляризации матрицы преобразованием Хаусхолдера	103
7.6	Решение треугольной системы $Rx = z$ и обращение матриц R и A	104
7.7	Преобразование Гивенса	107
7.8	Варианты заполнения матрицы R	113
7.9	Правосторонние ортогональные преобразования и их применение	114
7.10	Двусторонние ортогональные преобразования и их применение	115
7.11	Ортогонализация Грама-Шмидта	118
7.12	Алгоритмы ортогонализации Грама-Шмидта	120
7.13	Решение систем после ортогонализации Грама-Шмидта	123

7.14	Обращение матриц после ортогонализации	
	Грама-Шмидта	123
7.15	Задание на лабораторный проект № 6	124
7.16	Варианты задания	125
8	ФОНД ЗАДАЧ	127
8.1	Типовые задачи	127
8.2	Решения и рекомендации к типовым задачам	131
8.3	Варианты контрольных заданий	134
8.4	Задачи для контрольных заданий и экзамена	141
9	ПРОГРАММА КУРСА	164
9.1	Общая информация	164
9.2	Рабочая программа	165
	Заключение	175
	Библиографический список	178

Предисловие

Курс «Численные методы» в университетах преследует следующие 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 значительную часть времени отводят на «Численные методы алгебры». Эту часть общего курса называют также «Численные методы – I». Когда акцент в этом курсе делают на вычислительные алгоритмы, используют термин «Вычислительная линейная алгебра». Согласно требованиям Государственного образовательного стандарта [1], к этой части численных методов относят следующие темы:

- 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 нелинейных уравнений;
- тема 6 – методы решения проблемы собственных значений матриц.

В качестве дополнительных тем в часть I иногда включают линейное программирование и/или решение переопределенных систем. В данное учебное пособие включены первые три темы из приведенного списка тем. В лабораторных проектах затрагивается решение переопределенных систем, а в задачи включены вопросы по итерационным методам решения линейных систем уравнений. Линейное программирование изложено в отдельной книге [2].

Значение «Численных методов алгебры» во многих областях науки и техники трудно переоценить, — оно растет очень быстро. В связи с этим важно, чтобы студенты, готовящиеся стать специалистами в области математического моделирования, численных методов и комплексов программ, обладали истинно 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*, 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.

Ульяновск,
декабрь 2006



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 современной вычислительной техники.

Таким образом, безусловно каждому из вас потребуются хорошая компьютерная подготовка, чтобы выжить на рынке труда и успешно функциониро-

вать среди грамотных компьютерных пользователей. Было бы образовательным 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 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 = attendance (посещаемость),

H = homework (домашняя работа) и

E = exams (экзамены)

выражается целым числом не выше 100 баллов.

- Эта итоговая оценка затем отображается на стандартную шкалу оценок:

86 - 100 = «отлично»,

71 - 85 = «хорошо»,

56 - 70 = «удовлетворительно»,

0 - 55 = «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 «хорошо».

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. 85, 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 (A)

- 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 ^a	Балл A	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

^a 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.
- По курсу «Численные методы» 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.
- Итак, каждому студенту мы предлагаем выполнить в течение семестра 3 лабораторные работы, чтобы покрыть все три темы из списка на стр. 7, 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 заработает 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 заработает 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.

- 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} и P_{KP-2} письменных контрольных работ ($KP-1$) и ($KP-2$) в течение семестра и результата $P_{УЭ}$ устного экзамена ($УЭ$) во время экзаменационной сессии. Это означает, что

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

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 не менее двух контрольных работ 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, всегда 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

На стр. 18 вы найдете краткое — занимающее один лист — описание данного курса.

ЧИСЛЕННЫЕ МЕТОДЫ АЛГЕБРЫ		
Кредиты: 3 ^a Семестр: 1 ^b Обязательный: ДА		
Формат:	Лекции	34 ч
	Семинары	0 ч
	Лабораторные работы	17 ч
	Самостоятельная работа	49 ч
Преподаватель: проф. И. В. Семушин		
Содержание:		
Этот курс излагает основы современных и классических численных методов решения алгебраических задач: линейных систем уравнений, нелинейных уравнений, аппроксимации функций и отыскания собственных значений матриц. Его программа охватывает следующие темы: методы исключения неизвестных, алгоритмы разложения Холецкого, ортогональные преобразования, итерационные методы, а также обзор алгоритмов метода наименьших квадратов и методов решения проблемы собственных значений.		
Ожидаемые результаты изучения: продемонстрировать —		
знание и понимание:	структуры погрешностей решения вычислительных задач; корректности задач; прямых и итерационных методов решения линейных систем; 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 (Fortran 77/90, 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 две 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. <i>или:</i> 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.

2

СТАНДАРТНЫЕ АЛГОРИТМЫ LU -РАЗЛОЖЕНИЯ

2.1 Алгоритмы метода Гаусса

Обычный метод Гаусса, осуществляющий LU -разложение матрицы A , заключается в последовательном исключении переменных из уравнений системы

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

На первом шаге для исключения первой переменной x_1 в первом уравнении

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

достаточно потребовать выполнения условия $a_{11} \neq 0$. Тогда, разделив обе части (2.2) на a_{11} , получим уравнение, в котором коэффициент при x_1 равен 1. После этого, умножая полученное уравнение на a_{i1} и вычитая результат из i -го уравнения системы (2.1), $i = 2, \dots, n$, последовательно исключаем переменную x_1 из всех уравнений системы (кроме первого). На втором шаге метода Гаусса описанный алгоритм повторяем для переменной x_2 , т. е. берем систему (2.1), преобразованную в результате первого шага, и нормируем в ней второе уравнение

$$a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = f_2^{(1)},$$

считая, что $a_{22}^{(1)} \neq 0$. После этого исключаем переменную x_2 из оставшихся $n - 2$ уравнений системы (2.1). Таким образом, за n шагов получим эквивалентную систему линейных алгебраических уравнений с верхней треугольной матрицей, на диагонали которой стоят единицы¹

$$\bar{U}x = y. \quad (2.3)$$

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

Система (2.3) легко решается обратной подстановкой

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

При решении системы (2.1) на ЭВМ достаточно хранить коэффициенты матрицы A и элементы вектора f . Тогда каждый шаг метода Гаусса состоит в пересчете коэффициентов активной подматрицы матрицы A и соответствующих элементов вектора f . В результате прямого хода метода Гаусса на месте матрицы A образуются две треугольные матрицы: нижняя треугольная матрица L с ненулевыми элементами на главной диагонали и верхняя треугольная матрица U с единицами на главной диагонали, т. е. \bar{U} .

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 .

Алгоритм, дающий $L\bar{U}$ -разложение матрицы A , кратко можно записать 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, дающий $\bar{L}U$ -разложение 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.

Описанные выше алгоритмы в том виде, в каком они приведены, на практике используются очень редко, т. е. только в том случае, если можно га-

рантировать, 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. главу 4), поэтому в общем случае используется метод Гаусса с выбором главного (ведущего) элемента.

2.2 Выбор ведущего элемента

Существуют *три стратегии выбора главного (ведущего) элемента*.

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 переставленной 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 будут иметь следующий вид.

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.

Приведенный алгоритм называют *исключением по столбцам*, так как он исключает переменную x_k из всей поддиагональной части k -го столбца.

ЗАМЕЧАНИЕ 2.2. Во всех таких алгоритмах должно быть выполнено дополнительное *требование к реализации*: все действия должны выполняться в одном и том же массиве чисел, т. е. исходная матрица A должна замещаться числами, составляющими результирующее разложение. 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 трех стратегий, описанных выше.

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.

Под гауссовым *исключением по строкам* с выбором главного элемента по строке понимается такая разновидность метода Гаусса, в которой на каждом шаге исключения изменяется одна строка — первая строка активной подматрицы.

Первый шаг этого метода заключается в выборе главного элемента в первой строке матрицы A и в нормировке этой строки. На следующем шаге из

второй строки вычитается первая, умноженная на a_{21} . Затем во второй строке отыскивается главный элемент, и строка нормируется. На третьем шаге из третьей строки матрицы A вычитаются две первых. Сначала вычитается первая строка, умноженная на a_{31} , затем — вторая строка, умноженная на a_{32} . Потом в измененной третьей строке отыскивается главный элемент и строка нормируется. Повторяя описанный алгоритм n раз, получим $L\bar{U}$ -разложение матрицы A .

ЗАМЕЧАНИЕ 2.4. В описаниях алгоритмов упоминаются элементы матрицы A , например, только что упоминались элементы a_{21} , a_{31} , a_{32} . Естественно, речь идет о текущих, а не об исходных значениях элементов, занимающих указанные позиции матрицы A .

ЗАМЕЧАНИЕ 2.5. При UL -разложении матрицы A все действия выполняются аналогично, но в противоположном порядке нумерации строк и столбцов: снизу-вверх и справа-налево.

Таким образом, наряду с гауссовым исключением неизвестных по столбцам существует возможность проводить это исключение по строкам. Такая возможность 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 .

2.3 Компактные схемы

Следующей разновидностью метода Гаусса являются *компактные схемы*. Первая называется методом Краута, а вторая — компактной схемой «строка за строкой». В методе 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 метода 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

$$a_{ij} = \sum_{p=1}^n l_{ip}u_{pj}. \quad (2.4)$$

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

$$a_{ik} = l_{ik} + \sum_{p=1}^{k-1} l_{ip}u_{pk}, \quad i \geq k. \quad (2.5)$$

Из (2.5) следует

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

Таким образом, k -й столбец матрицы L становится известен. Теперь из (2.4) для k -й строки матрицы A имеем

$$a_{kj} = l_{kk}u_{kj} + \sum_{p=1}^{k-1} l_{kp}u_{pj}, \quad j > k. \quad (2.7)$$

Из (2.7) находим

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

Итак, (2.8) 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 (2.6) и (2.8) 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 (2.6) и того, что первым столбцом матрицы \bar{U} является первый координатный вектор e_1 . Здесь предполагается, что если нижний предел суммирования меньше верхнего, то значение суммы равно нулю. После этого в первом столбце выбирается главный элемент. Затем по формулам

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

вычисляется первая строка матрицы \bar{U} . Повторяя указанную последовательность действий n раз, с помощью формул (2.6) и (2.8) получаем $L\bar{U}$ -разложение матрицы A .

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

Для $k = 1$ до n

По формуле (2.6) вычисляем k -й столбец матрицы L .

Выбираем среди элементов k -го столбца главный элемент.

По формуле (2.8) 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 (2.6) и (2.8), а также последовательность вычисления столбцов матрицы \bar{L} и строк матрицы U . Таким образом, на k -м шаге сначала по формуле

$$u_{kj} = a_{kj} - \sum_{p=1}^{k-1} l_{kp}u_{pj}, \quad j \geq k, \quad (2.9)$$

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

$$l_{ik} = \left(a_{ik} - \sum_{p=1}^{k-1} l_{ip}u_{pk} \right) / u_{kk}, \quad i \geq k. \quad (2.10)$$

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

Для $k = 1$ до n

По формуле (2.9) вычисляем k -ю строку матрицы U .

Выбираем среди элементов k -й строки главный элемент.

По формуле (2.10) вычисляем k -й столбец матрицы \bar{L} .

Компактная схема «строка за строкой», дающая $L\bar{U}$ -разложение матрицы A , использует те же самые формулы (2.6) и (2.8). Меняется только последовательность вычисления элементов матрицы L .

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 (2.6) вычисляем ненулевые элементы k -й строки

матрицы L . По формулам (2.8) 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 (2.6) вычисляем элементы k -й строки матрицы L .

По формуле (2.8) 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 *методы Жордана*. Эти методы используют те же самые формулы, что и обычный метод Гаусса, но в отличие от него на k -м шаге метода Жордана пересчитываются все строки матрицы A , а не только строки активной подматрицы. Это соответствует исключению i -й переменной из всех уравнений кроме i -го. Таким образом, методы Жордана дают решение системы линейных алгебраических уравнений за один проход.

ЗАМЕЧАНИЕ 2.6. Термин $L\bar{U}^{-1}$ -разложение, который мы используем ниже для краткости в применении к методу Жордана, означает, что на самом деле отыскивается $L\bar{U}$ -разложение матрицы A , но алгоритм по этому методу 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 получаются с противоположными знаками.

Чтобы получить $L\bar{U}^{-1}$ -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

$$a_{ij} = -a_{ij}$$

Еще раз обратим внимание на замечание 2.6: в отличие от всех рассмотренных ранее случаев LU -разложения $L\bar{U}^{-1}$ -разложение не является разложением в обычном смысле, т. е. $L\bar{U}^{-1} \neq A$. Здесь под $L\bar{U}^{-1}$ -разложением понимается то, что на месте матрицы A образуются две треугольные матрицы: нижняя треугольная матрица L и верхняя треугольная матрица \bar{U}^{-1} с противоположными знаками внедиагональных элементов, причем $L\bar{U} = A$.

Таким образом, мы рассмотрели основные типы алгоритмов гауссова исключения, дающие LU -разложение матрицы A . Аналогичные алгоритмы для UL -разложения легко получаются из рассмотренных выше алгоритмов,

если проводить исключение, начиная с последней переменной и до первой. На матричном языке это означает модификацию строк и столбцов матрицы A , начиная с последних и до первых. Под активной подматрицей $A^{(k)}$ в этом случае понимаются первые k строк и столбцов матрицы A . Кроме того, $\bar{L}^{-1}U$ -разложение $A = \bar{L}U$ Жордана подразумевает смену знака на противоположный у всех элементов, находящихся ниже главной диагонали.

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*, 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 возьмем следующий пример.

ПРИМЕР 2.1. Пусть для данной матрицы A каким-либо способом найдено разложение $A = L\bar{U}$:

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

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

$$\begin{array}{l} \text{исходный массив} \Rightarrow \text{результатирующий массив} \\ \begin{bmatrix} 2 & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & \boxed{2} & -2 & 3 \\ 1 & 2 & \boxed{2} & -1 \\ 3 & 2 & \boxed{3} & -2 \\ 2 & 1 & 2 & \boxed{4} \end{bmatrix}. \end{array} \quad (2.11)$$

Следовательно, перед началом 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} должны вычисляться также по специальным процедурам, для которых исходные данные \bar{U} и L берутся из массива, названного в выражении (2.11) *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 (2.11). 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 процедурные действия соответствуют указанному выражению, представленному в следующем виде:

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/4 \end{bmatrix} \times \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1/3 & \\ & & -2/3 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & & & \\ & 1/2 & & \\ & -2/2 & 1 & \\ & -1/2 & & 1 \end{bmatrix} \times \begin{bmatrix} 1/2 & & & \\ -1/2 & 1 & & \\ -3/2 & & 1 & \\ -2/2 & & & 1 \end{bmatrix}.$$

В действительности же это означает, что сначала — на этапе **1** — результирующий массив из (2.11) пересчитывают по указанным правилам для вычисления L^{-1} , приводя его к следующему стартовому виду:

$$\left[\begin{array}{cccc|cccc} 2 & & & & 2 & -2 & 3 & \\ 1 & & & & 2 & & 2 & -1 \\ 3 & & & & 2 & & 3 & -2 \\ 2 & & & & 1 & & 2 & 4 \end{array} \right] \xRightarrow{\text{1}} \left[\begin{array}{cccc|cccc} 1/2 & & & & 2 & -2 & 3 & \\ -1/2 & & & & 1/2 & & 2 & -1 \\ -3/2 & & & & -2/2 & & 1/3 & -2 \\ -2/2 & & & & -1/2 & & -2/3 & 1/4 \end{array} \right]. \quad (2.12)$$

Чтобы понять, как в этом массиве должна работать процедура вычисления матрицы L^{-1} , рассмотрим произведение матриц перед выражением (2.12) 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. 2.1.

Из табл. 2.1 видно, что после получения (2.12), т. е. на этапе **2**, пересчитывают только элементы a_{21}, a_{31} и a_{41} . В данном случае имеем

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

Таблица 2.1. Поэтапное перемножение $L_4^{-1}(L_3^{-1}(L_2^{-1}L_1^{-1}))$

$$\begin{aligned}
& \begin{matrix} L_4^{-1} \\ \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/4 \end{array} \right] \end{matrix} \times \begin{matrix} L_3^{-1} \\ \left[\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1/3 \\ & & & -2/3 \end{array} \right] \end{matrix} \times \begin{matrix} L_2^{-1} \\ \left[\begin{array}{ccc} 1 & & \\ & 1/2 & \\ & & 1 \\ & & & -1/2 \end{array} \right] \end{matrix} \times \begin{matrix} L_1^{-1} \\ \left[\begin{array}{ccc} 1/2 & & \\ -1/2 & 1 & \\ -3/2 & & 1 \\ -2/2 & & & 1 \end{array} \right] \end{matrix} \\
& \begin{matrix} L_3^{-1} \\ \left[\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1/3 \\ & & & -2/3 \end{array} \right] \end{matrix} \times \begin{matrix} (L_2^{-1}L_1^{-1}) \\ \left[\begin{array}{ccc} 1/2 & & \\ -1/4 & 1/2 & \\ -1 & -1 & 1 \\ -3/4 & -1/2 & & 1 \end{array} \right] \end{matrix} \quad \Leftarrow \textcircled{2} \\
& \begin{matrix} L_4^{-1} \\ \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1/4 \end{array} \right] \end{matrix} \times \begin{matrix} (L_3^{-1}(L_2^{-1}L_1^{-1})) \\ \left[\begin{array}{cccc} 1/2 & & & \\ -1/4 & 1/2 & & \\ -1/3 & -1/3 & 1/3 & \\ -1/12 & 1/6 & -2/3 & 1 \end{array} \right] \end{matrix} \quad \Leftarrow \textcircled{3} \\
& \begin{matrix} L_4^{-1}(L_3^{-1}(L_2^{-1}L_1^{-1})) \\ \left[\begin{array}{cccc} 1/2 & & & \\ -1/4 & 1/2 & & \\ -1/3 & -1/3 & 1/3 & \\ -1/48 & 1/24 & -1/6 & 1/4 \end{array} \right] \end{matrix} \quad \Leftarrow \textcircled{4}
\end{aligned}$$

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

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

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

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

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.

$$\bar{U} \quad \bar{U}_4 \quad \bar{U}_3 \quad \bar{U}_2$$

$$\begin{bmatrix} 1 & 2 & -2 & 3 \\ & 1 & 2 & -1 \\ & & 1 & -2 \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & 3 \\ & 1 & & -1 \\ & & 1 & -2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & -2 \\ & 1 & & 2 \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix},$$

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

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

$$\bar{U}_2^{-1} \quad \bar{U}_3^{-1} \quad \bar{U}_4^{-1}$$

$$\begin{bmatrix} 1 & -2 & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \times \begin{bmatrix} 1 & & 2 & \\ & 1 & -2 & \\ & & 1 & \\ & & & 1 \end{bmatrix} \times \begin{bmatrix} 1 & & -3 & \\ & 1 & 1 & \\ & & 1 & 2 \\ & & & 1 \end{bmatrix}. \quad (2.13)$$

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 (2.12) — результирующий массив из (2.11) пересчитывают по указанным правилам для вычисления \bar{L}^{-1} и \bar{U}^{-1} , приводя его к следующему стартовому виду:

$$\begin{bmatrix} 2 & \begin{array}{|c} 2 & -2 & 3 \end{array} \\ 1 & \begin{array}{|c} 2 & 2 & -1 \end{array} \\ 3 & \begin{array}{|c} 2 & 3 & -2 \end{array} \\ 2 & \begin{array}{|c} 1 & 2 & 4 \end{array} \end{bmatrix} \xrightarrow[\text{❶}]{\text{①}} \begin{bmatrix} 1/2 & \begin{array}{|c} -2 & 2 & -3 \end{array} \\ -1/2 & \begin{array}{|c} 1/2 & -2 & 1 \end{array} \\ -3/2 & \begin{array}{|c} -2/2 & 1/3 & 2 \end{array} \\ -2/2 & \begin{array}{|c} -1/2 & -2/3 & 1/4 \end{array} \end{bmatrix}. \quad (2.14)$$

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 (2.13) отразим в табл. 2.2.

Таблица 2.2. Поэтапное перемножение $\bar{U}_2^{-1}(\bar{U}_3^{-1}\bar{U}_4^{-1})$

$$\begin{aligned}
& \begin{matrix} \bar{U}_2^{-1} & & \bar{U}_3^{-1} & & \bar{U}_4^{-1} \\ \left[\begin{array}{ccc} 1 & -2 & \\ & 1 & \\ & & 1 \end{array} \right] & \times & \left[\begin{array}{ccc} 1 & & 2 \\ & 1 & -2 \\ & & 1 \end{array} \right] & \times & \left[\begin{array}{ccc} 1 & & -3 \\ & 1 & 1 \\ & & 2 \\ & & & 1 \end{array} \right] \end{matrix} \\
& \begin{matrix} \bar{U}_2^{-1} & & (\bar{U}_3^{-1}\bar{U}_4^{-1}) \\ \left[\begin{array}{ccc} 1 & -2 & \\ & 1 & \\ & & 1 \end{array} \right] & \times & \left[\begin{array}{ccc} 1 & 2 & 1 \\ & 1 & -2 & -3 \\ & & 1 & 2 \\ & & & 1 \end{array} \right] & \Longleftarrow & \textcircled{2} \end{matrix} \\
& \begin{matrix} \bar{U}_2^{-1}(\bar{U}_3^{-1}\bar{U}_4^{-1}) \\ \left[\begin{array}{ccc} 1 & -2 & 6 & 7 \\ & 1 & -2 & -3 \\ & & 1 & 2 \\ & & & 1 \end{array} \right] & \Longleftarrow & \textcircled{3} \end{matrix}
\end{aligned}$$

Из табл. 2.2 видно, что после получения верхнетреугольной части в (2.14) 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 $\textcircled{3}$ после (2.14), получаем

$$\left[\begin{array}{cccc} 1/2 & -2 & 2 & -3 \\ -1/2 & 1/2 & -2 & 1 \\ -3/2 & -2/2 & 1/3 & 2 \\ -2/2 & -1/2 & -2/3 & 1/4 \end{array} \right] \xrightarrow[\textcircled{3}]{\textcircled{2}} \left[\begin{array}{cccc} 1/2 & -2 & 2 & 1 \\ -1/4 & 1/2 & -2 & -3 \\ -1 & -1 & 1/3 & 2 \\ -3/4 & -1/2 & -2/3 & 1/4 \end{array} \right].$$

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

$$\left[\begin{array}{cccc} 1/2 & -2 & 2 & 1 \\ -1/4 & 1/2 & -2 & -3 \\ -1 & -1 & 1/3 & 2 \\ -3/4 & -1/2 & -2/3 & 1/4 \end{array} \right] \xrightarrow[\textcircled{3}]{\textcircled{3}} \left[\begin{array}{cccc} 1/2 & -2 & 6 & 7 \\ -1/4 & 1/2 & -2 & -3 \\ -1/3 & -1/3 & 1/3 & 2 \\ -1/12 & 1/6 & -2/3 & 1/4 \end{array} \right]$$

завершает операции над верхнетреугольной частью, а для нижнетреугольной части завершение $\textcircled{4}$ показано отдельно на с. 31.

2.6 Плохо обусловленные матрицы

Следующие матрицы используются 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 :

$$\begin{aligned} a_{ii} &= \alpha^{|n-2i|/2}, \\ a_{1j} &= a_{j1} = a_{11}/\alpha^j; \\ a_{nj} &= a_{jn} = a_{nn}/\alpha^j; \\ a_{ij} &= 0 \text{ для остальных значений } i \text{ и } j. \end{aligned}$$

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 заданный 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 заданному 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 исполнения 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

$$\|x\|_{\infty} = \max_i(|x_i|). \quad (2.15)$$

4. Повторить пункт 3 задания для плохо обусловленных матриц (см. разд. 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

$$\|A_T^{-1} - A_{\text{пр}}^{-1}\| \leq \|I - AA_{\text{пр}}^{-1}\| \cdot \|A\|^{-1}. \quad (2.16)$$

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

$$\|A\|_{\infty} = \max_i \left(\sum_{j=1}^n |a_{ij}| \right), \quad (2.17)$$

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 2.7. 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 2.8. 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

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

3

СОВРЕМЕННЫЕ 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 главе 5.

Как изложено в разд. 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 .

$$\begin{array}{l}
 \text{Для } k = 1 \text{ до } n - 1 \\
 \quad \text{Для } i = k + 1 \text{ до } n \\
 \quad \quad l_{ik} = a_{ik}/a_{kk} \\
 \quad \quad \text{Для } j = k + 1 \text{ до } n \\
 \quad \quad \quad a_{ij} = a_{ij} - l_{ik}a_{kj}
 \end{array}$$

Рис. 3.1. Строчно ориентированная схема $\bar{L}U$ -разложения

$$\begin{array}{l}
 \text{Для } k = 1 \text{ до } n - 1 \\
 \quad \text{Для } s = k + 1 \text{ до } n \\
 \quad \quad l_{sk} = a_{sk}/a_{kk} \\
 \quad \quad \text{Для } j = k + 1 \text{ до } n \\
 \quad \quad \quad \text{Для } i = k + 1 \text{ до } n \\
 \quad \quad \quad \quad a_{ij} = a_{ij} - l_{ik}a_{kj}
 \end{array}$$

Рис. 3.2. Столбцово ориентированная схема $\bar{L}U$ -разложения

ОПРЕДЕЛЕНИЕ 3.1. Тριάдой называют операцию вида $\mathbf{a} + \alpha\mathbf{b}$, где \mathbf{a} и \mathbf{b} суть векторы, а α — скаляр.¹

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. *пересчет*) 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 конкретного 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, описанные в разд. 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. разд. 3.2 и 3.3.

³ Подробнее о стратегиях выбора главного элемента см. разд. 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). Однако 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. Типичная ситуация этого рода — обращение матрицы с помощью решения матричного уравнения $AX = E$, где E — 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. разд. 2.2).

3.2 Распараллеливание вычислений

В этом разделе мы излагаем некоторые простые 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. Таким образом, 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 таком конвейере в данный 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 для решения одной задачи нескольких процессоров, работающих сообща. 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 выполнить 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.

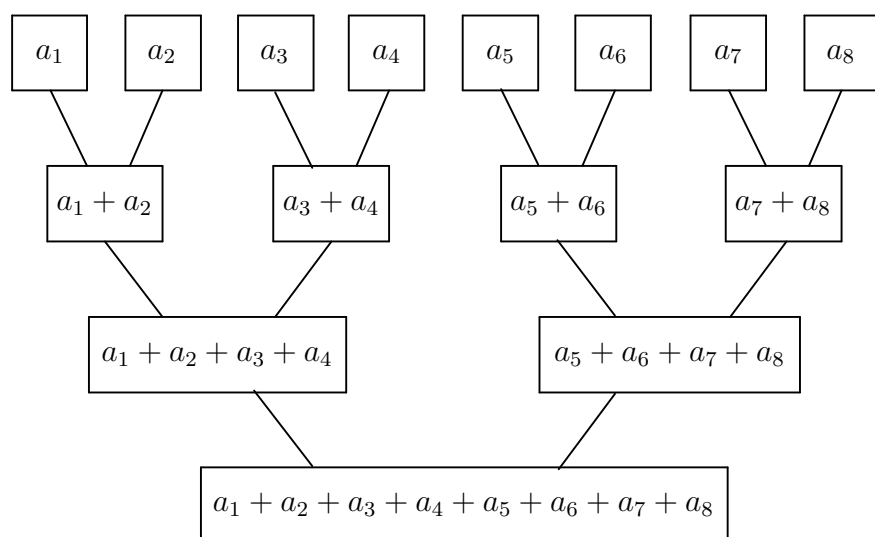


Рис. 3.3. Сложение n чисел методом сдваивания для $n = 8$

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

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

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

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

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 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

Как выше (в разд. 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 двух 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. (Однако 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.

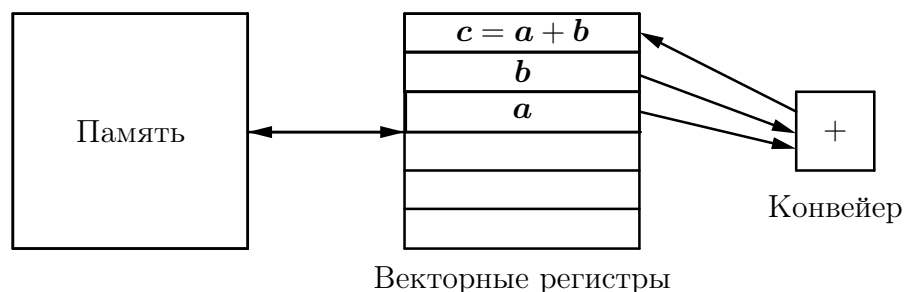


Рис. 3.5. Операция сложения в компьютере типа «регистр–регистр»

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 .

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

Записать в память модифицированный второй столбец. (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. Мы можем модифицировать алгоритм, чтобы устранить задержку, вызванную записью в память (3.7). Идея состоит в том, чтобы выполнить всю необходимую обработку для j -го столбца, прежде чем перейти к $(j + 1)$ -му столбцу. Таким образом, 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. 51)

Опишем несколько первых операций j -го шага вычислений, которые показывают 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; тогда результат, требуемый от алгоритма на рис. 3.6, либо достигается переходом к программированию на языке ассемблера, либо аппроксимируется путем развертывания 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 разделов 3.2, 3.3 и 3.4 приведен для того, чтобы объяснить те приложения, для которых создаются алгоритмы с отложенными модификациями: алгоритм на рис. 3.6 и другие, показанные ниже в разд. 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 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 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

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

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, 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 производится по строкам. Исключение — по столбцам. Модификации — немедленные. ГЭ — любая из трех стратегий.

$$\begin{array}{l}
 \text{Для } k = 1 \text{ до } n - 1 \\
 \text{Для } i = k + 1 \text{ до } n \\
 \quad l_{ik} = a_{ik}/a_{kk} \\
 \text{Для } j = k + 1 \text{ до } n \\
 \quad a_{ij} = a_{ij} - l_{ik}a_{kj}
 \end{array}$$

2) kji -алгоритм, рис. 3.2.

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

$$\begin{aligned} &\text{Для } k = 1 \text{ до } n - 1 \\ &\quad \text{Для } s = k + 1 \text{ до } n \\ &\quad \quad l_{sk} = a_{sk}/a_{kk} \\ &\quad \text{Для } j = k + 1 \text{ до } n \\ &\quad \quad \text{Для } i = k + 1 \text{ до } n \\ &\quad \quad \quad a_{ij} = a_{ij} - l_{ik}a_{kj} \end{aligned}$$

**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 производится по столбцам. Исключение — по столбцам. Модификации — отложенные. ГЭ — по $(j - 1)$ -му столбцу.

$$\begin{aligned} &\text{Для } j = 2 \text{ до } n \\ &\quad \text{Для } s = j \text{ до } n \\ &\quad \quad l_{s,j-1} = a_{s,j-1}/a_{j-1,j-1} \\ &\quad \text{Для } k = 1 \text{ до } j - 1 \\ &\quad \quad \text{Для } i = k + 1 \text{ до } n \\ &\quad \quad \quad a_{ij} = a_{ij} - l_{ik}a_{kj} \end{aligned}$$

4) jik -алгоритм.

Доступ к элементам матрицы A производится 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.

$$\begin{aligned} &\text{Для } j = 2 \text{ до } n \\ &\quad \text{Для } s = j \text{ до } n \\ &\quad \quad l_{s,j-1} = a_{s,j-1}/a_{j-1,j-1} \\ &\quad \text{Для } i = 2 \text{ до } j \\ &\quad \quad \text{Для } k = 1 \text{ до } i - 1 \\ &\quad \quad \quad a_{ij} = a_{ij} - l_{ik}a_{kj}. \\ &\quad \text{Для } i = j + 1 \text{ до } n \\ &\quad \quad \text{Для } k = 1 \text{ до } j - 1 \\ &\quad \quad \quad a_{ij} = a_{ij} - l_{ik}a_{kj} \end{aligned}$$

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

5) ikj -алгоритм.

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

$$\begin{aligned} &\text{Для } i = 2 \text{ до } n \\ &\quad \text{Для } k = 1 \text{ до } i - 1 \\ &\quad \quad l_{i,k} = a_{i,k}/a_{k,k} \\ &\quad \quad \text{Для } j = k + 1 \text{ до } n \\ &\quad \quad \quad a_{ij} = a_{ij} - l_{ik}a_{kj} \end{aligned}$$

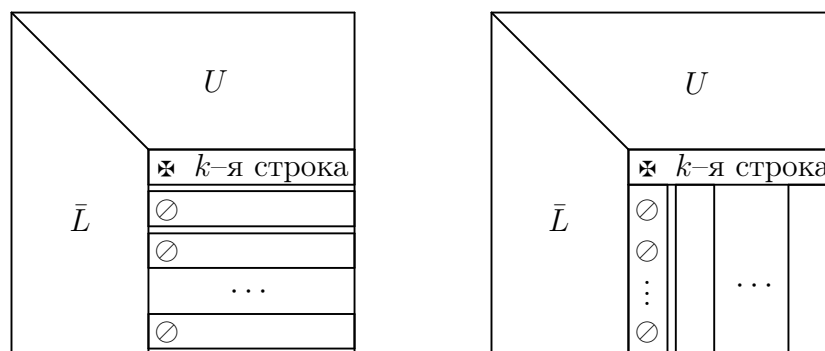


Рис. 3.7. Способ доступа к данным для kij -формы (слева) и для kji -формы (справа) $\bar{L}U$ -разложения. Обозначения: \bar{L}, U — вычисление закончено, обращений больше нет; \star — главный элемент (ГЭ); \circ — деление на ГЭ (нормировка)

6) ijk -алгоритм.

Доступ к элементам матрицы A производится по строкам. Исключение — по строкам. Модификации — отложенные. Первый цикл по j — 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 $i - 1$
 $a_{ij} = a_{ij} - l_{ik}a_{kj}$

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 выше 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 kij -формы и kji -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.

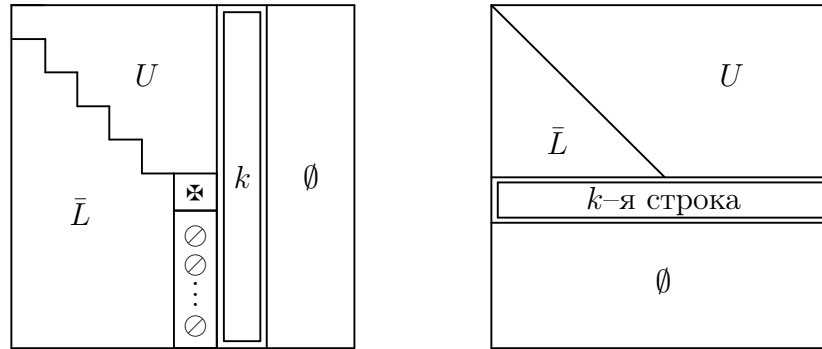


Рис. 3.8. Способ доступа к данным для jki -формы и для jik -формы (слева) и для ikj -формы и для ijk -формы (справа) $\bar{L}U$ -разложения. Обозначения: \bar{L} , U — вычисление закончено, обращения больше не производятся; \star — главный элемент ($\Gamma\Theta$); \circ — деление на $\Gamma\Theta$ (нормировка);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

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

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

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

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

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

A	$i = 2$	$i = 3$	$i = 4$
$\begin{bmatrix} 2 & 4 & -4 & 6 \\ 1 & 4 & 2 & 1 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & 2 & 4 & -2 \\ 3 & 8 & 1 & 1 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & 2 & 4 & -2 \\ 3/2 & 2 & 7 & -8 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -6 \\ 2/2 & 1 & 4 & -1 \end{bmatrix}$
\Uparrow исходная матрица	\Uparrow $(i - 1)$ нормировок и вычитаний в i -й строке	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & -6 \\ 2 & 5 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & 3 & -6 \\ 2/2 & 1/2 & 2 & 0 \end{bmatrix}$
		\Uparrow $(i - 1)$ нормировок и вычитаний в i -й строке	$\begin{bmatrix} \mathbf{2} & 4 & -4 & 6 \\ 1/2 & \mathbf{2} & 4 & -2 \\ 3/2 & 2/2 & \mathbf{3} & -6 \\ 2/2 & 1/2 & 2/3 & 4 \end{bmatrix}$

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

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

3.6 Треугольные системы

По окончании этапа приведения в гауссовом исключении мы должны 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).

<p>Для $j = n$ до 1 с шагом -1 Для $j = i + 1$ до n $c_i = c_i - u_{ij}x_j$ $x_i = c_i/u_{ii}$</p>	<p>Для $j = n$ до 1 с шагом -1 $x_j = c_j/u_{jj}$ Для $i = j - 1$ до 1 с шагом -1 $c_i = c_i - x_j u_{ij}$</p>
---	---

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

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

И алгоритм скалярных произведений, 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 заданный 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 заданному 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. разд. 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

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

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 $A_{\text{пр}}^{-1}$ — 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 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. 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.

3.8 Варианты задания

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

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

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*. Математически ее можно представить следующим образом. Разобьем матрицу A на блоки и, соответственно, разобьем на блоки сомножители \bar{L} и U искомого разложения $\bar{L}U = A$:

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

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 способа реализации окаймления в LU -разложении.

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 Для $k = 1$ до $j - 2$ Для $i = k + 1$ до $j - 1$ $a_{ij} = a_{ij} - l_{ik} a_{kj}$ Для $k = 1$ до $j - 1$ $l_{jk} = a_{jk} / a_{kk}$ Для $i = k + 1$ до j $a_{ji} = a_{ji} - l_{jk} a_{ki}$</p>	<p>Для $j = 2$ до n Для $i = 2$ до $j - 1$ Для $k = 1$ до $i - 1$ $a_{ij} = a_{ij} - l_{ik} a_{kj}$ Для $i = 2$ до j $l_{j,i-1} = a_{j,i-1} / a_{i-1,i-1}$ Для $k = 1$ до $i - 1$ $a_{ji} = a_{ji} - l_{jk} a_{ki}$</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 style="border-collapse: collapse; text-align: left; width: 100%;"> <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 style="border-collapse: collapse; text-align: left; width: 100%;"> <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 style="border-collapse: collapse; text-align: left; width: 100%;"> <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 style="border-collapse: collapse; text-align: left; width: 100%;"> <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.

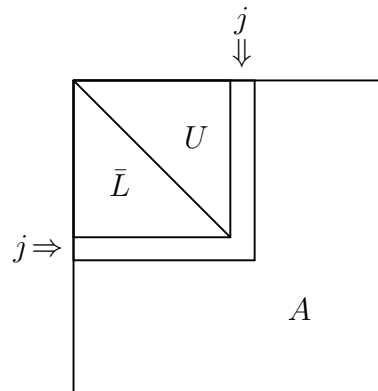


Рис. 4.2. Доступ к данным в алгоритмах окаймления известной части разложения. \bar{L} , U — вычисление закончено, но обращения происходят. A — обращений не было. Вычисляются: j -й столбец матрицы U и j -я строка матрицы \bar{L}

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.

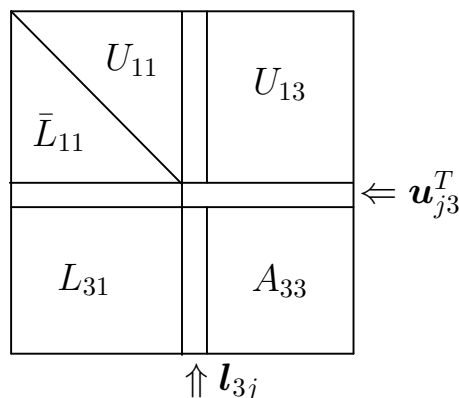


Рис. 4.3. Доступ к данным в алгоритмах окаймления неизвестной части разложения. \bar{L}_{11} , U_{11} — вычисление закончено, обращений больше нет. L_{31} , U_{13} — вычисление закончено, но обращения происходят. A_{33} — обращений не было. Вычисляются: j -й столбец $\bar{\mathbf{L}}$ и j -я строка \mathbf{u}_{j3}^T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

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

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 . Заметим, что при $j = i$ результатом модификации будет первая величина u_{jj} в (4.4). Во втором цикле по k, 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 . Обратим внимание на то, что, 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

$j = 1$	$j = 2$	$j = 3$	$j = 4$																																																																
<table style="border-collapse: collapse; width: 100%;"> <tr><td>2</td><td>4</td><td>-4</td><td>6</td></tr> <tr><td>1/2</td><td>4</td><td>2</td><td>1</td></tr> <tr><td>3/2</td><td>8</td><td>1</td><td>1</td></tr> <tr><td>2/2</td><td>5</td><td>0</td><td>5</td></tr> </table>	2	4	-4	6	1/2	4	2	1	3/2	8	1	1	2/2	5	0	5	<table style="border-collapse: collapse; width: 100%;"> <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>1</td><td>1</td></tr> <tr><td>2/2</td><td>1/2</td><td>0</td><td>5</td></tr> </table>	2	4	-4	6	1/2	2	4	-2	3/2	2/2	1	1	2/2	1/2	0	5	<table style="border-collapse: collapse; width: 100%;"> <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>5</td></tr> </table>	2	4	-4	6	1/2	2	4	-2	3/2	2/2	3	-6	2/2	1/2	2/3	5	<table style="border-collapse: collapse; width: 100%;"> <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																																																																
1/2	4	2	1																																																																
3/2	8	1	1																																																																
2/2	5	0	5																																																																
2	4	-4	6																																																																
1/2	2	4	-2																																																																
3/2	2/2	1	1																																																																
2/2	1/2	0	5																																																																
2	4	-4	6																																																																
1/2	2	4	-2																																																																
3/2	2/2	3	-6																																																																
2/2	1/2	2/3	5																																																																
2	4	-4	6																																																																
1/2	2	4	-2																																																																
3/2	2/2	3	-6																																																																
2/2	1/2	2/3	4																																																																

Как и в алгоритмах окаймления известной части разложения (разд. 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 заданный 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 заданному 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 исполнения 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. разд. 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 $A_{\text{пр}}^{-1}$ — 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 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

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

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.

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. Ноль 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.

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

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 выбора этого оптимального элемента используются следующие две стратегии.

Стратегия I. Локальное заполнение на $(k+1)$ -м шаге метода Гаусса будет минимальным, если в качестве главного 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 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 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} — единичная матрица 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. разд. 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 все 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 150-го порядка не более чем за 3 минуты (для персонального компьютера 386 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 из лабораторного проекта № 1, либо из лабораторного проекта № 2 (в соответствии со своим вариантом по лабораторному проекту № 1 или № 2). Матрица в этом случае 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 двух способов решения);
- ▷ зависимость времени решения от порядка матриц (для двух способов решения).

5.4 Варианты задания

В табл. 5.1 приведены 48 номеров вариантов задания на лабораторный проект № 4 с двумя стратегиями I и II выбора главного элемента, четырьмя схемами упаковки разреженной матрицы A и шестью вариантами метода исключения.

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 .

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 любой i -й строки и i -го столбца, также является невырожденной.

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

Условие 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 Холецкого

Если матрица P может быть представлена как

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

с квадратной матрицей S , то S называют *квадратным корнем* из P . 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 ($T T^T = I$ и $T^T T = I$) матрица, то $S T$ также удовлетворяет (6.1).

ЗАМЕЧАНИЕ 6.1. Хотя определение (6.1) часто используется, оно не является универсальным. Обобщения возможны в направлении отказа 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.

ОПРЕДЕЛЕНИЕ 6.2. 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*). Любая 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:

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

ЗАМЕЧАНИЕ 6.2. Приведенная выше 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*). Любая 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*). Любая симметрическая матрица $P > 0$ имеет разложение $P = UU^T$, 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*). Любая 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:

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

6.3 Программная реализация алгоритмов Холецкого

В справочных целях включаем реализацию трех из четырех приведенных выше алгоритмов на языке FORTRAN. 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 разд. 6.6.

Верхнетреугольное разложение Холесского:

$$\begin{array}{l}
 P(N, N), \quad P = UU^T, \quad P > 0, \quad U - \text{верхнетреугольная матрица.} \\
 \hline
 \text{DO } 5 \quad J = N, 2, -1 \qquad \qquad \qquad \text{© } \quad j = n, n - 1, \dots, 2 \\
 U(J, J) = \text{SQRT}(P(J, J)) \\
 \alpha = 1./U(J, J) \\
 \text{DO } 5 \quad K = 1, J - 1 \\
 U(K, J) = \alpha * P(K, J) \\
 \beta = U(K, J) \\
 \text{DO } 5 \quad I = 1, K \\
 5 \quad P(I, K) = P(I, K) - \beta * U(I, J) \\
 U(1, 1) = \text{SQRT}(P(1, 1)) \\
 \hline
 \end{array}$$

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 Холесского:

$$\begin{array}{l}
 P(N, N), \quad P = \bar{U}D\bar{U}^T, \quad P > 0, \quad \bar{U} - \text{верхнетреугольная матрица} \\
 \text{с единичной диагональю, } \quad D - \text{диагональная матрица.} \\
 \hline
 \text{DO } 5 \quad J = N, 2, -1 \qquad \qquad \qquad \text{© } \quad j = n, n - 1, \dots, 2 \\
 D(J) = P(J, J) \\
 \alpha = 1./D(J) \\
 \text{DO } 5 \quad K = 1, J - 1 \\
 \beta = P(K, J) \\
 \bar{U}(K, J) = \alpha * \beta \\
 \text{DO } 5 \quad I = 1, K \\
 5 \quad P(I, K) = P(I, K) - \beta * \bar{U}(I, J) \\
 D(1) = P(1, 1) \\
 \hline
 \end{array}$$

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 разд. 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. разд. 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

$$\begin{aligned}
 &(l_{11} = p_{11}^{1/2}) \\
 &\text{Для } k = 1 \text{ до } n - 1 \\
 &\quad \text{Для } i = k + 1 \text{ до } n \\
 &\quad \quad l_{ik} = p_{ik}/p_{kk} \\
 &\quad \quad (l_{ik} = p_{ik}/l_{kk}) \\
 &\quad \quad \text{Для } j = k + 1 \text{ до } i \\
 &\quad \quad \quad p_{ij} = p_{ij} - l_{ik}p_{jk} \\
 &\quad \quad \quad (p_{ij} = p_{ij} - l_{ik}l_{jk}) \\
 &\quad (l_{k+1,k+1} = p_{k+1,k+1}^{1/2})
 \end{aligned}$$

2) kji -алгоритм

$$\begin{aligned}
 &(l_{11} = p_{11}^{1/2}) \\
 &\text{Для } k = 1 \text{ до } n - 1 \\
 &\quad \text{Для } s = k + 1 \text{ до } n \\
 &\quad \quad l_{sk} = p_{sk}/p_{kk} \quad (p_{sk}/l_{kk}) \\
 &\quad \quad \text{Для } j = k + 1 \text{ до } n \\
 &\quad \quad \quad \text{Для } i = j \text{ до } n \\
 &\quad \quad \quad \quad p_{ij} = p_{ij} - l_{ik}p_{jk} \\
 &\quad \quad \quad \quad (p_{ij} = p_{ij} - l_{ik}l_{jk}) \\
 &\quad (l_{k+1,k+1} = p_{k+1,k+1}^{1/2})
 \end{aligned}$$

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

3) jki -алгоритм

$$\begin{aligned}
 &(l_{11} = p_{11}^{1/2}) \\
 &\text{Для } j = 2 \text{ до } n \\
 &\quad \text{Для } s = j \text{ до } n \\
 &\quad \quad l_{s,j-1} = p_{s,j-1}/p_{j-1,j-1} \\
 &\quad \quad (l_{s,j-1} = p_{s,j-1}/l_{j-1,j-1}) \\
 &\quad \quad \text{Для } k = 1 \text{ до } j - 1 \\
 &\quad \quad \quad \text{Для } i = j \text{ до } n \\
 &\quad \quad \quad \quad p_{ij} = p_{ij} - l_{ik}p_{jk} \\
 &\quad \quad \quad \quad (p_{ij} = p_{ij} - l_{ik}l_{jk}) \\
 &\quad (l_{j,j} = p_{j,j}^{1/2})
 \end{aligned}$$

4) jik -алгоритм

$$\begin{aligned}
 &(l_{11} = p_{11}^{1/2}) \\
 &\text{Для } j = 2 \text{ до } n \\
 &\quad \text{Для } s = j \text{ до } n \\
 &\quad \quad l_{s,j-1} = p_{s,j-1}/p_{j-1,j-1} \\
 &\quad \quad (l_{s,j-1} = p_{s,j-1}/l_{j-1,j-1}) \\
 &\quad \quad \text{Для } i = j \text{ до } n \\
 &\quad \quad \quad \text{Для } k = 1 \text{ до } j - 1 \\
 &\quad \quad \quad \quad p_{ij} = p_{ij} - l_{ik}p_{jk} \\
 &\quad \quad \quad \quad (p_{ij} = p_{ij} - l_{ik}l_{jk}) \\
 &\quad (l_{j,j} = p_{j,j}^{1/2})
 \end{aligned}$$

5) ikj -алгоритм

$$\begin{aligned}
 & (l_{11} = p_{11}^{1/2}) \\
 & \text{Для } i = 2 \text{ до } n \\
 & \quad \text{Для } k = 1 \text{ до } i - 1 \\
 & \quad \quad l_{i,k} = p_{i,k}/p_{k,k} \\
 & \quad \quad (l_{i,k} = p_{i,k}/l_{k,k}) \\
 & \quad \quad \text{Для } j = k + 1 \text{ до } i \\
 & \quad \quad \quad p_{ij} = p_{ij} - l_{ik}p_{jk} \\
 & \quad \quad \quad (p_{ij} = p_{ij} - l_{ik}l_{jk}) \\
 & \quad \quad (l_{i,i} = p_{i,i}^{1/2})
 \end{aligned}$$

6) ijk -алгоритм

$$\begin{aligned}
 & (l_{11} = p_{11}^{1/2}) \\
 & \text{Для } i = 2 \text{ до } n \\
 & \quad \text{Для } j = 2 \text{ до } i \\
 & \quad \quad l_{i,j-1} = p_{i,j-1}/p_{j-1,j-1} \\
 & \quad \quad (l_{i,j-1} = p_{i,j-1}/l_{j-1,j-1}) \\
 & \quad \quad \text{Для } k = 1 \text{ до } j - 1 \\
 & \quad \quad \quad p_{ij} = p_{ij} - l_{ik}p_{jk} \\
 & \quad \quad \quad (p_{ij} = p_{ij} - l_{ik}l_{jk}) \\
 & \quad \quad (l_{i,i} = p_{i,i}^{1/2})
 \end{aligned}$$

ЗАМЕЧАНИЕ 6.6. Приведенные алгоритмы $\bar{L}D\bar{L}^T$ и LL^T -разложений Холесского матрицы получены из соответствующих ijk -алгоритмов $\bar{L}U$ -разложения матрицы A (см. разд. 3.5). Для получения $\bar{U}D\bar{U}^T$ и UU^T -разложений Холесского 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 Холесского, 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 Холесского: алгоритмы окаймления

Как и для LU -разложения, для разложения Холесского 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 разд. 6.2.

Покажем, как выводятся такие матрично-векторные алгоритмы для одного из четырех вариантов разложения Холецкого (6.2), а именно, для 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 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) и 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) и алгоритм скалярных произведений (справа)

6.6 Задание на лабораторный проект № 5

Написать и отладить программу, реализующую заданный вариант алгоритма разложения Холецкого, 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 Холецкого;
- и) 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 Холецкого. 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.

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 (2.15) 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 лабораторного 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 закрытого интервала вида

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

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 $x^* = (1, 2, \dots, n)$, где 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 (2.15) из лабораторного проекта № 1.

ЗАМЕЧАНИЕ 6.7. 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 6.8. 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.

6.7 Варианты задания

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	36	39	40

^a — строчный алгоритм;

^b — алгоритм скалярных произведений;

^c — алгоритм линейных комбинаций.

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, если и только если $TT^T = I$.

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 $T^T T = I$.

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

Свойство Д. 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 Д играют существенную роль в квадратно-корневым алгоритмах решения прикладных задач оптимального моделирования и оптимального оценивания методом наименьших квадратов (рис. 7.1).

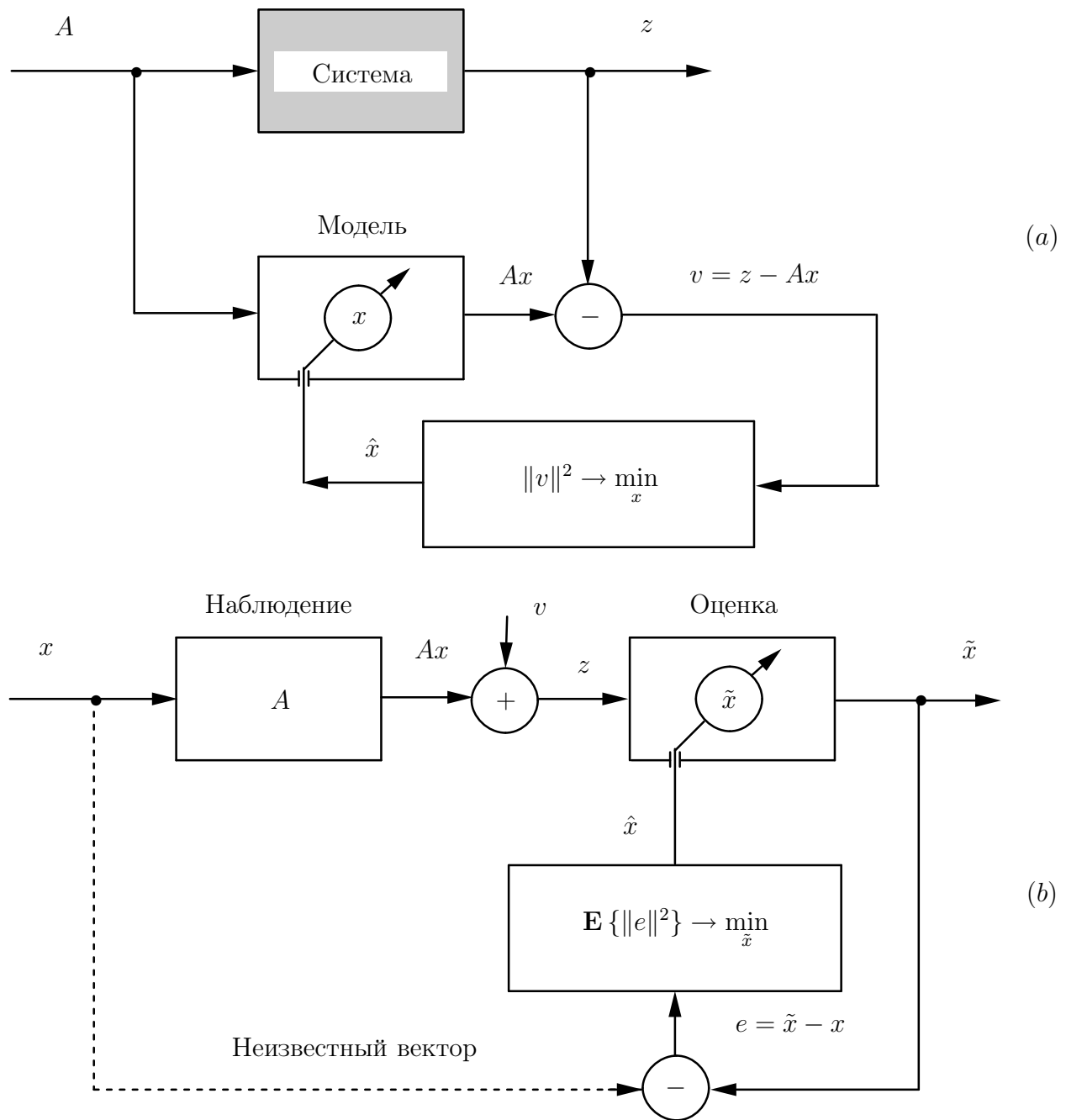


Рис. 7.1. Алгебраически эквивалентные задачи, решаемые методом наименьших квадратов значений невязки v или среднего квадрата погрешности e : (a) — оптимальное моделирование неизвестной системы по экспериментальным условиям A и данным z ; (b) — оптимальное оценивание неизвестного вектора по наблюдениям 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.

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. $\hat{x}_0 \in \mathcal{R}(A^T)$.

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.

7.3 Ортогональные матрицы в задаче о наименьших квадратах

В рассматриваемой задаче о наименьших квадратах

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

Пусть T , $T(m, m)$, есть матрица некоторого ортогонального преобразования. В силу свойства С (см. разд. 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 разд. 7.4 и 7.7 мы покажем, как можно выбрать T , чтобы преобразованная матрица имела вид

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

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

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

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

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 видеть, что искомым вектор \hat{x} , минимизирующий этот критерий, должен удовлетворять уравнению

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

которое легко решается обратной подстановкой (см. разд. 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). Но 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. 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 $(y \mid \hat{u})$ вектора y на прямую, задаваемую направлением u , равна $(y^T \hat{u})\hat{u}$. Следовательно,

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

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

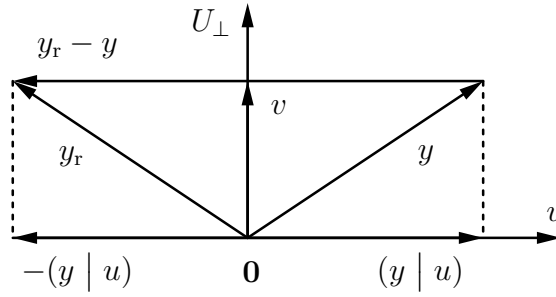


Рис. 7.2. Геометрия преобразования Хаусхолдера. Задача 1 (прямая): даны векторы u и y , найти вектор y_r , отраженный от гиперплоскости U_\perp

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 $T_u \triangleq (I - \beta u u^T)$, 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 важно 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 преобразованиями 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, разд. 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, где видно, что вектор y_r ортогонален гиперплоскости U_\perp и параллелен вектору u .

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 .

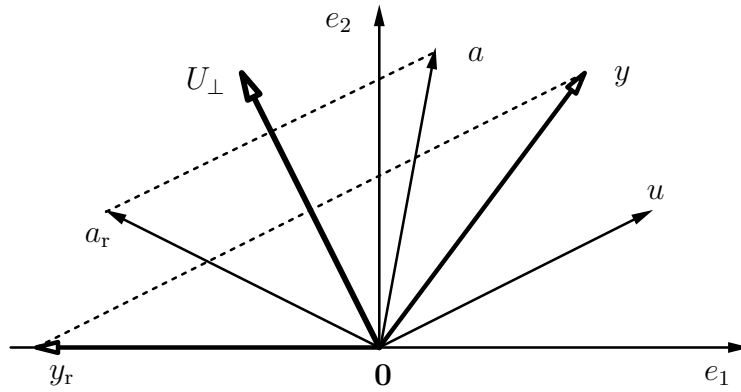


Рис. 7.3. Геометрия преобразования Хаусхолдера. Задача 2 (обратная): даны векторы y и y_r , найти вектор u , задающий отражающую гиперплоскость U_\perp ; здесь $y_r = se_1 = [s \mid 0 \cdots 0]^T$

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

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

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, 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

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

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

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

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 разд. 7.5.

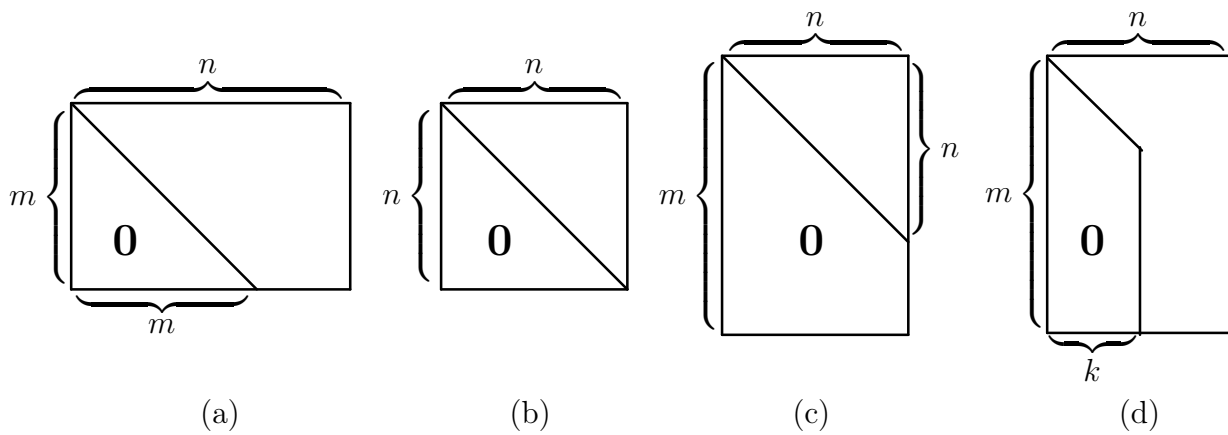


Рис. 7.4. Схематическое представление возможных случаев применения теоремы 7.1 к матрице $A(m, n)$; (a) недоопределенная система: $k = m - 1 \leq n$; (b) определенная система: $k = n - 1, m = n$; (c) переопределенная система: $k = n < m$; (d) $k < n < m$

7.5 Шаг триангуляризации матрицы преобразованием 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

$$T_u A = \begin{matrix} & \overbrace{}^1 & \overbrace{\phantom{\tilde{A}}}^{n-1} \\ \begin{matrix} 1 \\ \vdots \\ m-1 \end{matrix} \left\{ \begin{array}{c|c} s & \\ \mathbf{0} & \tilde{A} \end{array} \right. \end{matrix} . \quad (7.22)$$

Алгоритм. Для вычисления s следует применять свойство 6, т. е. выполнять (7.23) для всех $k = 1$ до $\min(m - 1, n)$. Затем для вычисления \tilde{A} следует применять свойство 5, т. е. последовательно в цикле по $j = 2, \dots, n$ для всех $i = k, \dots, m$ выполнять (7.24) с $\lambda \triangleq -\gamma$ (см. (7.15)). Здесь введена величина $\alpha \triangleq -\beta$ (см. (7.14) и 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*, 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. 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).

<p>Для $k = 1$ до $\min(m - 1, n)$</p> $s_k = -\operatorname{sgn} [A(k, k)] \left(\sum_{i=k}^m [A(i, k)]^2 \right)^{1/2},$ $u_k(1) = A(k, k) - s_k,$ $u_k(i) = A(k + i - 1, k), \quad i = 2, \dots, m - k + 1,$ $\alpha_k = 1/(s_k u_k(1)) \quad (\alpha_k < 0).$	} (7.25)
<p>Для $j = k + 1, \dots, n$</p> $\lambda_k(j - k) := \alpha_k \cdot \sum_{i=k}^m u_k(i - k + 1)A(i, j),$ <p>Для $i = k, k + 1, \dots, m$</p> <p>Для $j = k + 1, \dots, n$</p> $A(i, j) := A(i, j) + \lambda_k(j - k)u_k(i - k + 1).$	} (7.26)

7.6 Решение треугольной системы $Rx = z$ и обращение матриц R и A

Как отмечено в подразд. 7.3, мы часто заинтересованы в решении уравнения

$$Rx = z, \tag{7.27}$$

где $R = R(n, n)$ — верхнетреугольная 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 вычислить 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 обратную матрицу $U \triangleq R^{-1}$ рекуррентно, т. е. , 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. 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 верхнетреугольной матрицы: $U := R^{-1}$. Реализуются формулы (7.31), (7.32) и (7.33). Если нужно, U может замещать R .

$R(N, N), U(N, N), R$ и U — верхнетреугольные 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 $I = K, JM1$
 10 $SUM = SUM - U(K, I) * R(I, J)$
 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 и 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. 105, 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 верхнетреугольной матрицы: $U := R^{-1}$. Реализуются формулы (7.31), (7.32) и (7.33). Верхнетреугольные матрицы R и U хранятся как векторы размерности $N(N+1)/2$. Если нужно, U может замещать R .

$R(N, N), U(N, N), R$ и U — верхнетреугольные матрицы.	
$U(1) = 1./R(1)$	© $R(1) \equiv R(1, 1)$
$JJ = 1$	
DO 20 $J = 2, N$	
$\overline{JJ} = JJ$	© $\overline{JJ} = J(J-1)/2 = (J-1, J-1)$
$JJ = JJ + J$	© $JJ = J(J+1)/2 = (J, J)$
$U(JJ) = 1./R(JJ)$	
$JM1 = J - 1$	
$KK = 0$	
DO 20 $K = 1, JM1$	
$KK = KK + K$	© $KK = K(K+1)/2$
$KI = KK$	
$SUM = 0.$	
DO 10 $I = K, JM1$	
$SUM = SUM - U(KI) * R(\overline{JJ} + I)$	© $KI = (K, I), \overline{JJ} + 1 = (I, J)$
10 $KI = KI + I$	© $KI = (K, I + 1)$
20 $U(\overline{JJ} + K) = SUM * U(JJ)$	© $\overline{JJ} + K = (K, J)$

Как видно из (7.24), для отражения любого 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.

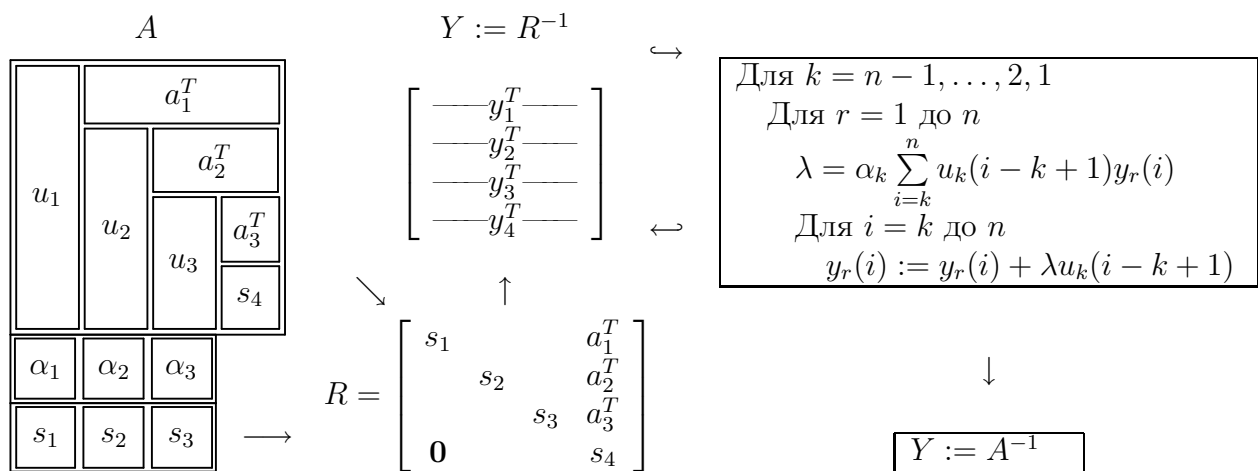
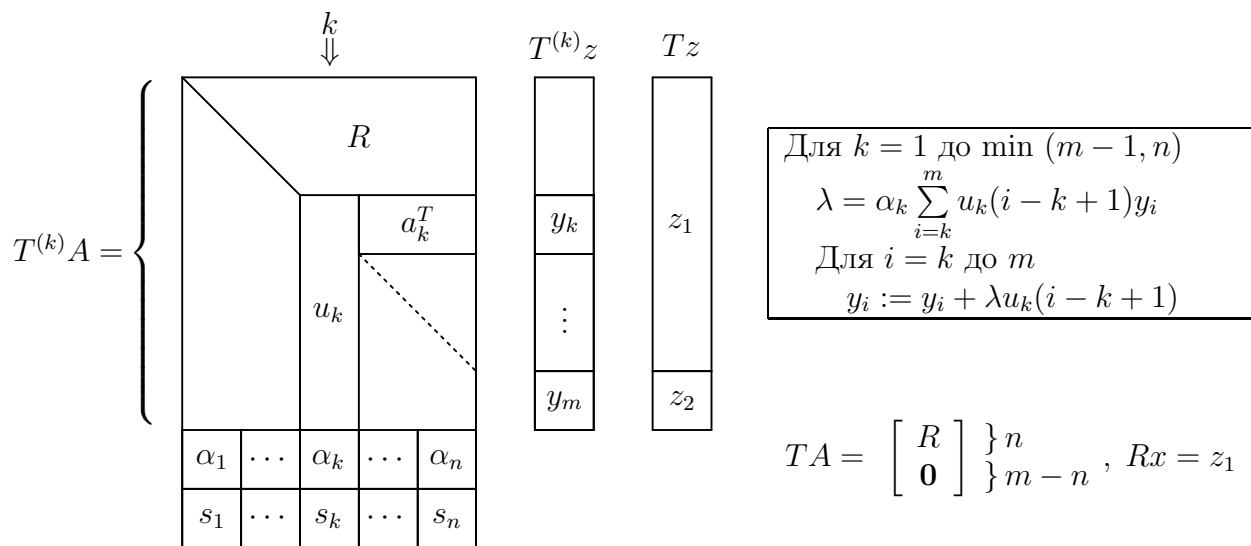


Рис. 7.5. Вверху: Сохранение преобразования T и вычисление вектора $y = Tz, \forall y \in \mathbb{R}^m$. Внизу: Вычисление матрицы A^{-1} после сохранения преобразования T

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 .

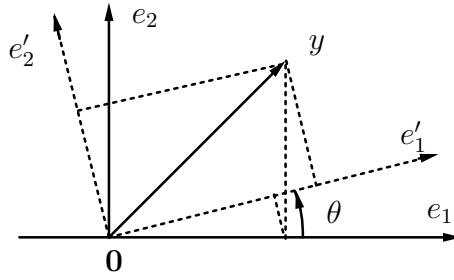


Рис. 7.6. Геометрия вращений

Триангуляризация матрицы преобразованиями Гивенса. Выбор θ такой, что вторая координата вектора y 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*, 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

$$P_1 A = \begin{matrix} & \overbrace{\quad}^1 & \overbrace{\quad}^{n-1} \\ \begin{matrix} 1 \\ \vdots \\ m-1 \end{matrix} \left\{ \begin{array}{c|c} r & \tilde{A} \\ \mathbf{0} & \end{array} \right. \end{matrix}, \quad (7.34)$$

$$P_1 = P_{1,m} \cdots P_{1,3} P_{1,2},$$

и матрицы $P_{1,j}$ определяются согласно 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 следующим образом.

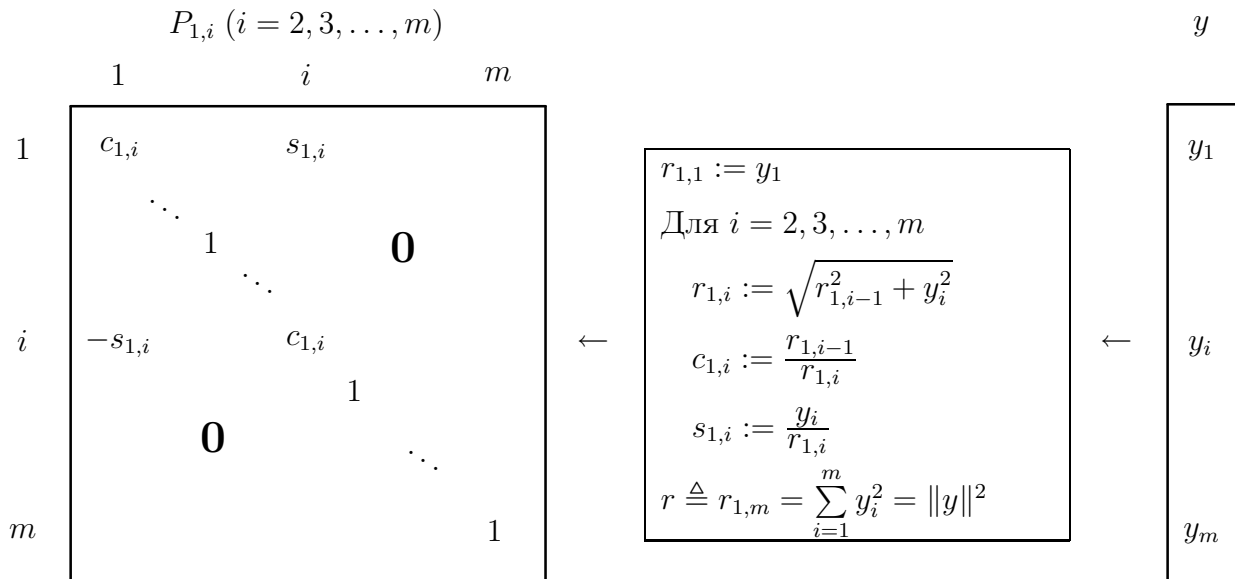


Рис. 7.7. Вычисление матрицы $P_{1,j}$

Для ведущего (левого) столбца y_j матрицы A_j эта P_j выбрана так, что

$$P_j A_j = \begin{matrix} 1 \\ \vdots \\ m-j \end{matrix} \left\{ \begin{matrix} \overbrace{1} \\ \vdots \\ \mathbf{0} \end{matrix} \left| \begin{matrix} \overbrace{a_j^T} \\ \vdots \\ A_{j+1} \end{matrix} \right. \right\}, \quad j = 1, \dots, k; \quad k \leq \min(m-1, n). \quad (7.35)$$

Тогда в процессе после k повторных применений леммы 7.2 имеем следующий промежуточный результат триангуляризации матрицы A :

$$P^{(k)} A = \begin{bmatrix} r_1 & & & a_1^T \\ & r_2 & & a_2^T \\ & & \dots & \vdots \\ & & & r_k & a_k^T \\ & \mathbf{0} & & & \hline & & & & A_{k+1} \end{bmatrix} \quad (7.36)$$

с отвечающей этому моменту процесса итоговой матрицей преобразований

$$P^{(k)} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & P_k \end{bmatrix} \cdots \begin{bmatrix} I_1 & 0 \\ 0 & P_2 \end{bmatrix} P_1, \quad P_j = P_{j,m-j+1} \cdots P_{j,3} P_{j,2}, \quad (7.37)$$

$$\begin{array}{c}
P_{j,t} \ (t = 2, 3, \dots, l), \ l \triangleq m - j + 1, \ j = 1, 2, \dots, k, \ i = j + t - 1. \\
\begin{array}{ccc}
1 & & t & & l \\
\begin{array}{|c|} \hline 1 \\ \hline \end{array} & \begin{array}{|c|} \hline c_{j,i} \\ \hline \end{array} & & \begin{array}{|c|} \hline s_{j,i} \\ \hline \end{array} & \\
& \ddots & & & \\
& & 1 & & \mathbf{0} \\
& & \ddots & & \\
t & \begin{array}{|c|} \hline -s_{j,i} \\ \hline \end{array} & & \begin{array}{|c|} \hline c_{j,i} \\ \hline \end{array} & \\
& & & 1 & \\
& & \mathbf{0} & & \ddots \\
l & & & & 1 \\
\end{array} & \leftarrow & \begin{array}{|c|} \hline r_{j,1} := y_{1,j} \\ \hline \text{Для } t = 2, \dots, l \\ \hline r_{j,t} := \sqrt{r_{j,t-1}^2 + y_{t,j}^2} \\ \hline c_{j,j+t-1} := \frac{r_{j,t-1}}{r_{j,t}} \\ \hline s_{j,j+t-1} := \frac{y_{t,j}}{r_{j,t}} \\ \hline r_j \triangleq r_{j,l} = \sum_{t=1}^l y_{t,j}^2 = \|y_j\|^2 \\ \hline \end{array} & \leftarrow & \begin{array}{|c|} \hline y_{1,j} \\ \hline y_{t,j} \\ \hline y_{l,j} \\ \hline \end{array} \\
\end{array}$$

Формула (7.37) имеет рекуррентный вид произведения

$$\left. \begin{array}{l}
P^{(j)} = \left[\begin{array}{cc} I_{j-1} & 0 \\ 0 & P_j \end{array} \right] P^{(j-1)}, \quad P^{(1)} = P_1 \\
P_j = P_{j,m-j+1} \cdots P_{j,3} P_{j,2}, \quad j = 2, \dots, N
\end{array} \right\}, \quad N = \min(m-1, n). \quad (7.38)$$

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

$$PA = \left[\begin{array}{c} R \\ \cdots \\ \mathbf{0} \end{array} \right], \quad R = \begin{array}{|c|} \hline \square \\ \hline \mathbf{0} \\ \hline \end{array} \triangleq R_{\text{не}}, \quad (7.39)$$

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 и (2) столбцово ориентированная схема (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.

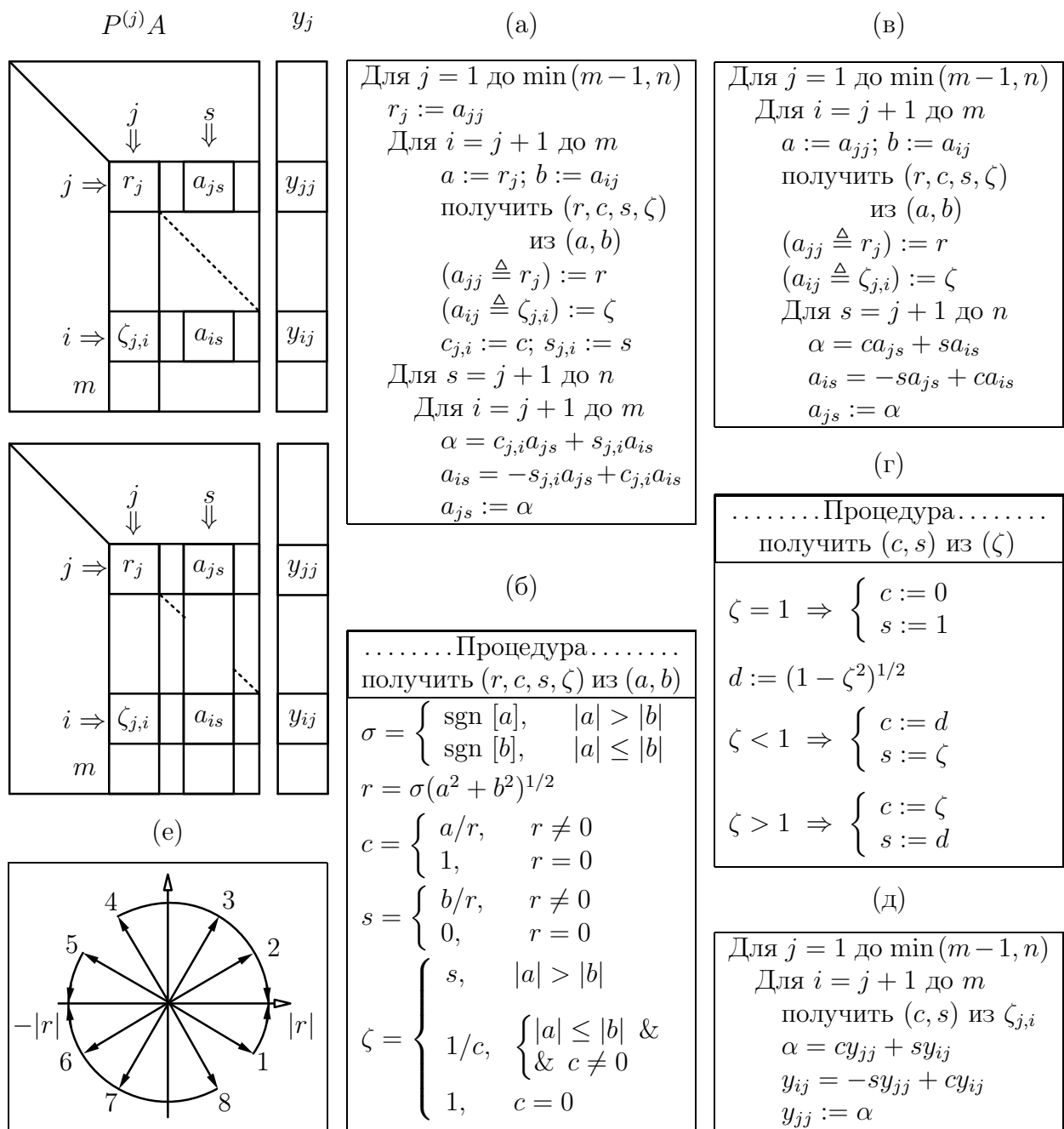


Рис. 7.8. Преобразование Гивенса: (а) столбцово ориентированная схема вычисления матрицы PA , где $P = P^{(j)}$ при $j = \min(m-1, n)$ (нижняя матрица слева); (б) вычисление координаты r вектора $(a, b)^T$, повернутого до совмещения с первой осью, а также косинуса и синуса угла поворота и рабочего признака ζ ; (в) строчно ориентированная схема вычисления матрицы PA (верхняя матрица слева); (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 поворачиваются 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 означает значения косинуса и синуса, однако их сохранение было бы неэффективным решением. Gentleman (1973) предложил способ, 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, любые ортогональные преобразования (выше рассмотрены T — преобразование Хаусхолдера и P — преобразование Гивенса, ниже будет рассмотрено Q — преобразование Грама-Шмидта) 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 , а именно: первый вариант показан в (7.39), следующие три имеют вид

$$PA = \begin{bmatrix} R \\ \cdots \\ \mathbf{0} \end{bmatrix}, \quad R = \begin{array}{|c|} \hline \text{треугольн.} \\ \hline \mathbf{0} \\ \hline \end{array} \triangleq R_{\text{nw}}, \quad (7.40)$$

где индекс nw подчеркивает, что в треугольной матрице R заполненной частью может быть только «северо-западная» (*north-by-west*) часть,

$$PA = \begin{bmatrix} \mathbf{0} \\ \cdots \\ R \end{bmatrix}, \quad R = \begin{array}{|c|} \hline \mathbf{0} \\ \hline \text{треугольн.} \\ \hline \end{array} \triangleq R_{\text{se}}, \quad (7.41)$$

где индекс $_{se}$ подчеркивает, что в треугольной матрице R заполненной частью может быть только «юго-восточная» (*south-by-east*) часть, и

$$PA = \begin{bmatrix} \mathbf{0} \\ \dots \\ R \end{bmatrix}, \quad R = \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} \triangleq R_{sw}, \quad (7.42)$$

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 и их применение

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. 7.9). 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, обращааемых в нуль.

После такого преобразования матрицы A решение системы $Ax = z$ сводится к решению эквивалентной системы с треугольной матрицей $Ry = z$. Затем искомым 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 и их применение

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. 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

$$\left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_2 \end{array} \right] A \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_2^T \end{array} \right] = \begin{array}{c} \overbrace{1} \\ \overbrace{1} \\ \underbrace{n-2} \\ \left[\begin{array}{c|c} a_1 & s_1 \quad \mathbf{0} \\ \hline s_1 & \tilde{A} \\ \mathbf{0} & \end{array} \right] \end{array}. \quad (7.43)$$

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

$$\left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_{j+1} \end{array} \right] A_j \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \hline \mathbf{0} & Q_{j+1}^T \end{array} \right] = \begin{array}{c} \underbrace{1} \quad \underbrace{1} \quad \underbrace{n-j-1} \\ \left\{ \begin{array}{l} 1 \\ 1 \\ \vdots \\ n-j-1 \end{array} \right. \left[\begin{array}{c|cc} a_j & s_j & \mathbf{0} \\ \hline \mathbf{0} & & A_{j+1} \end{array} \right] \end{array} \quad (7.44)$$

Тогда в процессе после k повторных применений леммы 7.3 имеем отвечающую этому моменту процесса итоговую матрицу преобразований

$$Q^{(k)} = \begin{bmatrix} I_k & 0 \\ 0 & Q_{k+1} \end{bmatrix} Q^{(k-1)}, \quad 1 \leq k \leq N = n - 2, \quad Q^{(0)} = I_n \quad (7.45)$$

и промежуточный результат тридиагонализации данной матрицы A в виде

$$Q^{(k)} A (Q^{(k)})^T = \left[\begin{array}{cccc} a_1 & s_1 & & \\ s_1 & a_2 & s_2 & \\ & s_2 & \dots & \dots \\ & & \dots & a_k & s_k \\ & & & s_k & \\ \mathbf{0} & & & & A_{k+1} \end{array} \right].$$

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

$$Q^{(1)}AS^{(1)} = \begin{matrix} & \overbrace{}^1 & \overbrace{}^1 & \overbrace{\phantom{\mathbf{0}}}^{n-2} \\ 1\left\{ \left[\begin{array}{c|cc} s_1 & a_1 & \mathbf{0} \\ \mathbf{0} & \tilde{A} & \end{array} \right] \right. \\ n-2\left\{ \end{matrix}, \quad \left\{ \begin{array}{l} Q^{(1)} = Q_1, \\ S^{(1)} = \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \mathbf{0} & S_2 \end{array} \right]. \end{array} \right. \quad (7.46)$$

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

$$Q_j A_j \left[\begin{array}{c|c} I_1 & \mathbf{0} \\ \mathbf{0} & S_{j+1} \end{array} \right] = \begin{matrix} & \overbrace{}^1 & \overbrace{}^1 & \overbrace{\phantom{\mathbf{0}}}^{n-j-1} \\ 1\left\{ \left[\begin{array}{c|cc} s_j & a_j & \mathbf{0} \\ \mathbf{0} & A_{j+1} & \end{array} \right] \right. \\ n-j\left\{ \end{matrix}. \quad (7.47)$$

Тогда в процессе после k повторных применений леммы 7.4 имеем отвечающие этому моменту процесса итоговые матрицы преобразований

$$\left. \begin{array}{l} Q^{(k)} = \left[\begin{array}{cc} I_{k-1} & 0 \\ 0 & Q_k \end{array} \right] Q^{(k-1)}, \quad k \leq n-2, \quad Q^{(0)} = I_n, \quad Q^{(1)} = Q_1, \\ S^{(k)} = S^{(k-1)} \left[\begin{array}{cc} I_k & 0 \\ 0 & S_{k+1} \end{array} \right], \quad k \leq n-2, \quad S^{(1)} = \left[\begin{array}{cc} I_1 & 0 \\ 0 & S_2 \end{array} \right] \end{array} \right\} \quad (7.48)$$

и промежуточный результат бидиагонализации данной матрицы A в виде

$$Q^{(k)}AS^{(k)} = \left[\begin{array}{c|cc} s_1 & a_1 & \\ \hline & s_2 & a_2 & \mathbf{0} \\ & \dots & \dots & \\ & s_k & a_k & \\ \hline \mathbf{0} & & & A_{k+1} \end{array} \right].$$

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

$$Q^{(n-1)}AS^{(n-2)} = \left[\begin{array}{cccccc} s_1 & a_1 & & & & \\ & s_2 & a_2 & & & \\ & & \dots & \dots & & \\ & & & s_{n-1}a_{n-1} & & \\ & & & & s_n & \\ \mathbf{0} & & & & & \mathbf{0} \end{array} \right] .$$

Основное применение указанных двусторонних ортогональных преобразований заключается в вычислении сингулярных значений произвольной матрицы $A = A(n, n)$, а также в решении проблемы собственных значений. Но 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.

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 этого множества выберем такое 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 $A = QR$.

7.12 Алгоритмы ортогонализации Грама-Шмидта

Рассмотрим задачу QR -разложения матрицы $A(m, n)$, $m \geq n$, 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

$$\begin{aligned} r_{11}q_1 &= a_1, \\ r_{12}q_1 + r_{22}q_2 &= a_2, \\ r_{13}q_1 + r_{23}q_2 + r_{33}q_3 &= a_3. \end{aligned}$$

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 7.7. 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 $k = 1, 2, \dots, n$. Две его более современные версии показаны на стр. 122.

Алгоритм ГШО (классическая схема)

$\begin{aligned} &\text{Для } k = 1 \text{ до } n \\ &r_{ik} = a_k^T q_i, \quad i = 1, 2, \dots, k-1, \\ &v = a_k - \sum_{i=1}^{k-1} r_{ik} q_i, \\ &r_{kk} = (v^T v)^{1/2}, \\ &q_k = v / r_{kk}. \end{aligned}$

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
 Найти № l , для которого $\|a_{q(l)}\| = \max_s \|a_{q(s)}\|,$
 Переставить номера: $q(k) \leftrightarrow q(l),$
 $r_{q(k),q(k)} = \|a_{q(k)}\| = (a_{q(k)}^T a_{q(k)})^{1/2},$
 $a_{q(k)} = a_{q(k)} / r_{q(k),q(k)},$
 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), 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 Грама-Шмидта

1. Пусть дана система линейных алгебраических уравнений с квадратной невырожденной матрицей $Ax = f$. Тогда после ортогонального приведения имеем $A = QR$. 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. 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 Грама-Шмидта

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 заданный 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 заданному 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

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

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 решить методом исключения переменных из лабораторной работы № 1 (в соответствии со своим вариантом). Таким образом, каждую систему линейных алгебраических уравнений, заданную сгенерированной матрицей A и образованной по ней правой частью f , 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

В табл. 7.2 каждому номеру варианта соответствует свой вариант заполнения треугольной матрицы R , своя разновидность ортогонального преобразования и своя схема организации вычислений.

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
 $\triangleq R_{ne}$	1	2	3	4	5	6	7
 $\triangleq R_{nw}$	8	9	10	11	12	13	14
 $\triangleq R_{se}$	15	16	17	18	19	20	21
 $\triangleq R_{sw}$	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.

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 ниже 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.

8.1 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, 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, разд. 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:

- 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:

- 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:

- 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
- $$\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}, \quad x \in \mathbb{R}^3.$$

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, разд. 6.7) 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:

- 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, разд. 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:

- 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 $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

8.2 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.

$$A^{-1} = \frac{1}{75} \begin{pmatrix} 39 & -48 & 30 \\ 12 & -9 & 15 \\ -2 & -11 & 10 \end{pmatrix}, \quad M_A = 12 \cdot 117/75 = 18.72.$$

8.3 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:

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:

- 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 $\|\cdot\|_\infty$.

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:

- 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:

- 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:

- 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:

- 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 $\|\cdot\|_\infty$.

Задание 3. Для матрицы

$$P = \begin{pmatrix} 15 & 3 & -12 & 3 \\ 3 & 9 & 4 & 1 \\ -12 & 4 & 13 & -2 \\ 3 & 1 & -2 & 1 \end{pmatrix}$$

выполнить:

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:

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:

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:

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 $\|\cdot\|_\infty$.

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:

- 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:

- 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:

- 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:

- 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 $\|\cdot\|_\infty$.

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:

- 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:

- 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 $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

8.4 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- На основе метода Зейделя сформулировать неявный метод скорейшего спуска в каноническом виде.
- Определить оптимальный параметр τ_1 для нулевого начального приближения, т.е. $x^0 = (0, 0, 0)^T$?
- 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:

- На основе метода Зейделя сформулировать метод минимальных поправок в каноническом виде.
- 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:

- 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:

- 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:

- 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:

- 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:

- 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

$$P x = b,$$

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:

- 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

$$A x = b,$$

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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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 $\|\cdot\|_\infty = \max_{i=1,2,3} \{|x_i|\}$, $x \in \mathbb{R}^3$.

9

ПРОГРАММА КУРСА

9.1 Общая информация

Цель – базовые умения и навыки в области разработки компьютерно ориентированных вычислительных алгоритмов решения задач, возникающих в процессе математического моделирования законов реального мира. Понимание основных идей численных методов, особенностей и условий их применения в реальных условиях.

Задачи – изучение основных методов «Вычислительной линейной алгебры», численного решения нелинейных уравнений и задач аппроксимации функций по экспериментальным данным.

Формат – 2/0/1 или 2/1/1 (т. е. в неделю часов на лекции / семинары / лабораторные работы), в зависимости от специальности, формы обучения и университета. Например, при семестре в 17 недель возможны варианты:

- специальность 01050165, очная форма обучения:
лекции (34 час), лабораторные работы (17 час), самостоятельная работа (49 час), экзамен
- специальность 08080165, очная форма обучения:
лекции (34 час), лабораторные работы (16 час), самостоятельная работа (50 час), зачет
- специальность 08080165, заочная форма обучения:
лекции (17 час), лабораторные работы (8,5 час), самостоятельная работа (74,5 час), зачет
- специальность 08080165, очно-заочная форма обучения:
лекции (34 час), практические занятия (10 час), самостоятельная работа (66 час), зачет

Пререквизиты – «Математический анализ», «Геометрия и алгебра», «Информатика», «Языки программирования и методы трансляции» и «Практикум на ЭВМ».

Кореквизиты¹ – «Практикум на ЭВМ» (17 час), зачет.

Постреквизиты – «Численные методы II», «Теория игр и исследование операций», специальные дисциплины и выпускная работа.

9.2 Рабочая программа

ЛЕКЦИИ

1. Введение (1 час). Задачи и методы вычислительной математики и их приложения в различных сферах деятельности. Математическое моделирование и вычислительный эксперимент. Численные методы как раздел современной математики. Роль компьютерно-ориентированных численных методов в исследовании сложных математических моделей.

2. Системы линейных алгебраических уравнений (27 час).

2.1. Прямые методы решения систем (12 час). LU -разложение и методы исключения Гаусса и Жордана. Стратегии выбора главного элемента. Обращение матриц. Компактные схемы (Краут). Положительно определенные матрицы. Разложения Холецкого и метод квадратного корня из матриц.

2.2. Плохая обусловленность и анализ ошибок (3 час). Нормы матриц и линейных преобразований. Обращение возмущенных матриц (лемма Банаха). Обусловленность линейных уравнений (число обусловленности и теорема об относительной ошибке). Прямой и обратный анализы ошибок. Примеры и последствия плохой обусловленности систем. Приемлемое решение неопределенной системы (теорема Оттля-Прагера).

¹ «Практикум на ЭВМ» в УлГУ поддерживает курс «Численные методы I» для специальностей «Прикладная математика и информатика» и «Математика», при этом для специальности «Математика» на него отводится 34 часа.

2.3. Методы ортогонального приведения (3 час). Ортогонализация Грама-Шмидта (обыкновенная и модифицированная). QR -разложение матриц и решение систем. Плоские вращения Гивенса. Элементарные отражения Хаусхолдера. Решение систем с применением ортогональных преобразований.

2.4. Итерационные методы решения систем. (7 час) Примеры и канонический вид итерационных методов решения систем линейных алгебраических уравнений. Исследование сходимости итерационных методов. Необходимое и достаточное условие сходимости стационарных итерационных методов. Оценка скорости сходимости стационарных итерационных методов. Итерационные методы вариационного типа. Апостериорная оценка погрешности итерационных методов.

2.5. Метод наименьших квадратов (3 час). Подгонка данных, построение моделей и задача МНК. Нормальные уравнения и нормальное псевдорешение. Элементарная статистическая интерпретация решения задачи МНК. Рекурсия в решении задачи МНК. Информационная форма решения. Эффективные вычислительные схемы МНК, использующие факторизацию или ортогональное приведение матриц.

3. Проблема собственных значений (3 час). Общая характеристика проблемы. Решение полной проблемы собственных значений при помощи QR -алгоритма.

4. Корни нелинейных уравнений (3 час). Нелинейные задачи с одной переменной. Метод простой итерации. Метод Ньютона. Сходимость метода Ньютона. Методы для случая, когда производные не заданы. Метод Ньютона для систем нелинейных уравнений. Локальная сходимость метода Ньютона. Теорема Канторовича и теорема о сжимающем отображении.

ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа № 1: Стандартные алгоритмы LU -разложения.

Лабораторная работа № 2: Современные алгоритмы LU -разложения.

Лабораторная работа № 3: Алгоритмы окаймления в LU -разложении.

Лабораторная работа № 4: Разреженные формы LU -разложения.

Лабораторная работа № 5: Разложения Холецкого.

Лабораторная работа № 6: Ортогональные преобразования.

САМОСТОЯТЕЛЬНАЯ РАБОТА

В зачет самостоятельной работы студента включены упражнения, представляющие собой доказательство вариантов теорем, доказанных на лекциях, решение задач, а также написание программ для лабораторных работ и чтение учебной литературы.

ВОПРОСЫ ЭКЗАМЕНА

1. Теорема о существовании и единственности LU -разложения. Связь разложения и метода Гаусса исключения неизвестных.
2. Теорема о существовании и единственности UL -разложения. Связь разложения и метода Гаусса исключения неизвестных.
3. Метод Гаусса: расчетные формулы и подсчет числа действий умножения/деления в процедуре факторизации матрицы.
4. Метод Гаусса: расчетные формулы и подсчет числа действий умножения/деления в процедурах прямой и обратной подстановки.
5. Элементарные треугольные матрицы. Теорема об алгоритме LU -разложения с замещением исходной матрицы матрицами L и U .
6. Элементарные треугольные матрицы. Теорема об алгоритме UL -разложения с замещением исходной матрицы матрицами U и L .
7. Метод Гаусса с выбором главного элемента (ГЭ): стратегии и программная реализация. Выбор ГЭ по строке и решение систем.
8. Теорема о методе Гаусса (об LU -разложении) с выбором главного элемента по столбцу активной подматрицы.

9. Теорема о методе Гаусса (об LU -разложении) с выбором главного элемента по строке активной подматрицы.
10. Вычисление определителя и обращение матрицы (два способа) с учетом выбора главного элемента.
11. Метод Гаусса-Жордана: теорема об алгоритме LU -разложения с получением U^{-1} . Подсчет числа действий умножения/деления.
12. Метод Гаусса-Жордана: теорема об алгоритме UL -разложения с получением L^{-1} . Подсчет числа действий умножения/деления.
13. Компактные схемы: вариант LU -разложения. Алгоритм и пример.
14. Компактные схемы: вариант UL -разложения. Алгоритм и пример.
15. Алгоритмы LU -разложения с исключением по столбцам и по строкам. Примеры.
16. Алгоритмы UL -разложения с исключением по столбцам и по строкам. Примеры.
17. Положительно-определенные матрицы и разложения Холецкого. Вывод алгоритмов Холецкого из алгоритмов LU -разложения.
18. LL^T -разложение положительно-определенных матриц: вывод по методу квадратичных форм.
19. LDL^T -разложение положительно-определенных матриц: вывод по методу квадратичных форм.
20. UU^T -разложение положительно-определенных матриц: вывод по методу квадратичных форм.
21. UDU^T -разложение положительно-определенных матриц: вывод по методу квадратичных форм.
22. Нормы вектора и матрицы. Норма с индексом бесконечность. Оценка для собственных значений через норму матрицы.
23. Число обусловленности системы линейных алгебраических уравнений. Свойства стандартного числа обусловленности.
24. Обращение возмущенных матриц (лемма Банаха).

25. Полная оценка относительной погрешности решения линейных систем.
26. Прямой и обратный анализы ошибок. Приемлемое решение неопределенной системы (теорема Оттля-Прагера).
27. Элементарные отражения Хаусхолдера: прямая и обратная задачи.
28. Ортогональные преобразования Хаусхолдера: приведение матрицы к верхней треугольной форме.
29. Элементарные плоские вращения Гивенса. Приведение матрицы к верхней треугольной форме вращениями Гивенса.
30. Решение систем и обращение матрицы после приведения матрицы к верхней треугольной форме ортогональными преобразованиями (Хаусхолдера или Гивенса).
31. Итерационные методы. Классические методы Якоби и Зейделя.
32. Каноническая форма и разновидности итерационных методов.
33. Определение сходимости итерационных методов, матричное неравенство $C > 0$ и нижняя грань для (Cx, x) .
34. Теорема о сходимости стационарного одношагового метода с симметрической положительно-определенной матрицей системы.
35. Следствие о сходимости метода Якоби для задач со строгим диагональным преобладанием матрицы системы.
36. Следствие о сходимости метода верхней релаксации для задач с симметрической положительно-определенной матрицей системы.
37. Следствие о сходимости метода простой итерации для задач с симметрической положительно-определенной матрицей системы.
38. Необходимое и достаточное условие сходимости стационарных одношаговых итерационных методов. Необходимость.
39. Достаточное условие сходимости стационарных итерационных методов: случай полной системы собственных векторов матрицы S , – переходной матрицы погрешности.

40. Достаточное условие сходимости стационарных итерационных методов: случай неполной системы собственных векторов матрицы S , – переходной матрицы погрешности.
41. Апостериорная оценка погрешности итерационных методов.
42. Задача линейных наименьших квадратов. Нормальные уравнения и нормальное псевдорешение.
43. Элементарная статистическая интерпретация решения задачи линейных наименьших квадратов.
44. Рекурсия в решении задачи линейных наименьших квадратов. Информационная форма решения.
45. Рекурсия в решении задачи линейных наименьших квадратов. Ковариационная форма решения.
46. Степенной метод решения проблемы собственных значений.
47. Метод Якоби решения проблемы собственных значений.
48. Метод Гивенса решения проблемы собственных значений.
49. Метод Хаусхолдера решения проблемы собственных значений.
50. QR -метод Френсиса решения проблемы собственных значений.
51. Метод простой итерации решения одного уравнения с одним неизвестным.
52. Метод Ньютона решения одного уравнения с одним неизвестным.
53. Сходимость метода Ньютона решения одного уравнения с одним неизвестным.
54. Конечно-разностный метод Ньютона и метод секущих решения одного уравнения с одним неизвестным.
55. Метод Ньютона решения систем нелинейных уравнений.
56. Локальная сходимость метода Ньютона решения систем нелинейных уравнений.
57. Теорема Канторовича и теорема о сжимающем отображении.

ЛИТЕРАТУРА

Основная

1. Воеводин В. В. Численные методы алгебры. Теория и алгоритмы. — М.: Наука, 1966.
2. Самарский А. А., Гулин А. В. Численные методы. — М.: Наука, 1989.

Дополнительная

К разделу 2.1:

1. Калиткин Н. Н. Численные методы. — М.: Наука, 1978.
2. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления. — М.: Наука, 1984.
3. Годунов С. К. Решение систем линейных уравнений. — Новосибирск: Наука, 1980.
4. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем: Пер. с англ. — М.: Мир, 1991.
5. Писсанецки С. Технология разреженных матриц: Пер. с англ. — М.: Мир, 1988.
6. Тьюарсон Р. Разреженные матрицы: Пер. с англ. — М.: Мир, 1977.
7. Фаддеев Л. К., Фаддеева В. Н. Вычислительные методы линейной алгебры. — М.: Физматгиз, 1963.
8. Stoer J., Bulirsch R. Introduction to Numerical Analysis: Translated from German. — 2nd ed. New York: Springer-Verlag, 1993. [Оригинальное издание: J. Stoer, R. Bulirsch. Einführung in die Numerische Mathematik, I, II. — Berlin, Heidelberg: Springer-Verlag, 1972, 1976.]

К разделу 2.2:

1. Воеводин В. В. Вычислительные основы линейной алгебры. — М.: Наука, 1977.
2. Ортега Дж., Пул У. Введение в численные методы решения дифференциальных уравнений. — М.: Наука, 1986.
3. Райс Дж. Матричные вычисления и математическое обеспечение: Пер. с англ. — М.: Мир, 1984.
4. Херцбергер Ю., Алефельд Г. Введение в интервальные вычисления: Пер. с англ. — М.: Мир, 1987.
5. Fröberg Carl-Erik. Introduction to Numerical Analysis. — 2nd ed. Reading Massachusetts Menlo Park (California) London Don Mills (Ontario): Addison-Wesley Publishing Company, 1969.
6. Golub G. H., Ortega J. M. Scientific Computing and Differential Equations. An Introduction to Numerical Methods. — San Diego New York Boston London Sydney Tokyo Toronto: Academic Press, 1992. [*Есть русский перевод*]
7. Noble B., Daniel J. W. Applied Linear Algebra. — 2nd ed. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1977.
8. Stoer J., Bulirsch R. Introduction to Numerical Analysis: Translated from German. — 2nd ed. New York: Springer-Verlag, 1993. [Оригинальное издание: J. Stoer, R. Bulirsch. Einführung in die Numerische Mathematik, I, II. — Berlin, Heidelberg: Springer-Verlag, 1972, 1976.]

К разделу 2.3:

1. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления. — М.: Наука, 1984.
2. Калиткин Н. Н. Численные методы. — М.: Наука, 1978.
3. Райс Дж. Матричные вычисления и математическое обеспечение: Пер. с англ. — М.: Мир, 1984.
4. Размыслов Ю. П., Ищенко С. Я. Практикум по вычислительным методам алгебры. — М.: Изд-во МГУ, 1989.

5. Семушин И. В., Куликов Г. Ю. Сборник лабораторных работ и контрольных, тестовых заданий по курсу «Вычислительная линейная алгебра». — Ульяновск: УлГТУ, 2000.
6. Stoer J., Bulirsch R. Introduction to Numerical Analysis: Translated from German. — 2nd ed. New York: Springer-Verlag, 1993. [Оригинальное издание: J. Stoer, R. Bulirsch. Einführung in die Numerische Mathematik, I, II. — Berlin, Heidelberg: Springer-Verlag, 1972, 1976.]

К разделу 2.4:

1. Самарский А. А., Гулин А. В. Численные методы. — М.: Наука, 1989.

К разделу 2.5:

1. Беклемишев Д. В. Дополнительные главы линейной алгебры. — М.: Наука, 1983.
2. Bierman G. J. Factorization Methods for Discrete Sequential Estimation. — N.-Y.: Academic Press, 1977.
3. Stoer J., Bulirsch R. Introduction to Numerical Analysis: Translated from German. — 2nd ed. New York: Springer-Verlag, 1993. [Оригинальное издание: J. Stoer, R. Bulirsch. Einführung in die Numerische Mathematik, I, II. — Berlin, Heidelberg: Springer-Verlag, 1972, 1976.]

К разделу 3:

1. Бахвалов Н. С. Численные методы. — М.: Наука, 1975.
2. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. — М.: Наука, 1987.
3. Уилкинсон Дж. Х. Алгебраическая проблема собственных значений: Пер. с англ. — М.: Наука, 1970.
4. Stoer J., Bulirsch R. Introduction to Numerical Analysis: Translated from German. — 2nd ed. New York: Springer-Verlag, 1993. [Оригинальное издание: J. Stoer, R. Bulirsch. Einführung in die Numerische Mathematik, I, II. — Berlin, Heidelberg: Springer-Verlag, 1972, 1976.]

К разделу 4:

1. Дэннис Дж., Шнабель Р. Численные методы безусловной оптимизации и решения нелинейных уравнений: Пер. с англ. — М.: Мир, 1988.
2. Самарский А. А., Гулин А. В. Численные методы. — М.: Наука, 1989.
3. Stoer J., Bulirsch R. Introduction to Numerical Analysis: Translated from German. — 2nd ed. New York: Springer-Verlag, 1993. [Оригинальное издание: J. Stoer, R. Bulirsch. Einführung in die Numerische Mathematik, I, II. — Berlin, Heidelberg: Springer-Verlag, 1972, 1976.]

Составитель: профессор, д.т.н. И. В. Семушин

Рецензент: доцент, к.ф.-м.н. А. В. Жарков.

ЗАКЛЮЧЕНИЕ

В данное учебное пособие включены учебные тексты по следующим темам курса «Численные методы»:

▸ тема 1 – методы исключения в решении систем:

- текст и лабораторная работа № 1, стр. 19–39. В этот раздел включены:
 - 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;
- текст и лабораторная работа № 2, стр. 40–59. В этот раздел включены:
 - 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;
- текст и лабораторная работа № 3, стр. 61–69. В этот раздел включены:
 - метод окаймления,

- окаймление известной части LU -разложения,
- окаймление неизвестной части LU -разложения,
- задание на лабораторный проект № 3,
- 16 вариантов задания;
- текст и лабораторная работа № 4, стр. 70–76. В этот раздел включены:
 - упакованные формы хранения матриц,
 - выбор ведущего элемента,
 - задание на лабораторный проект № 4,
 - 48 вариантов задания;
- ▷ тема 2 – разложения Холецкого положительно определенных матриц:
 - текст и лабораторная работа № 5, стр. 78–92. В этот раздел включены:
 - 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:
 - текст и лабораторная работа № 6, стр. 94–125. В этот раздел включены:
 - 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.

В качестве дополнительного материала включены:

- фонд задач, стр. 127–163 и
- программа курса, стр. 164–174.

Введение (разд. 1) описывает следующие важные для студента разделы, которые следует прочитать в первую очередь:

- цели учебной работы студента, стр. 9–10,
- оценка учебной работы студента, стр. 10–16,
- кодекс студента, стр. 15–17 и
- краткое описание курса, стр. 17–18.

Библиографический список

1. Методические материалы для проведения итоговой государственной аттестации выпускников вузов по направлению «Прикладная математика и информатика»: Учеб.-метод. сб. / Тверь, Твер. Гос. ун-т. — 2003.
2. Семушин, И. В. Практикум по методам оптимизации. Компьютерный курс: учеб. пособие для вузов / И. В. Семушин. — 3-е изд, перераб. и доп. — Ульяновск : УлГТУ, 2005.
3. Воеводин, В. В. Численные методы алгебры. Теория и алгоритмы / В. В. Воеводин. — М.: Наука, 1966.
4. Самарский, А. А. Численные методы / А. А. Самарский, А. В. Гулин. — М.: Наука, 1989.
5. Ортега, Дж. Введение в параллельные и векторные методы решения линейных систем / Дж. Ортега. — М.: Мир, 1991.
6. Писсанецки, С. Технология разреженных матриц / С. Писсанецки. — М.: Мир, 1988.
7. Фаддеев, Л. К. Вычислительные методы линейной алгебры / Л. К. Фаддеев, В. Н. Фаддеева. — М.: Физматгиз, 1963.
8. Воеводин, В. В. Вычислительные основы линейной алгебры / В. В. Воеводин. — М.: Наука, 1977.
9. Ортега, Дж. Введение в численные методы решения дифференциальных уравнений / Дж. Ортега, У. Пул. — М.: Наука, 1986.
10. Райс, Дж. Матричные вычисления и математическое обеспечение / Дж. Райс. — М.: Мир, 1984.

Учебное издание
СЕМУШИН Иннокентий Васильевич
Численные методы алгебры
Учебное пособие

Редактор О. С. Бычкова

Оригинал-макет изготовлен в системе $\text{\LaTeX}2\epsilon$.

Подписано в печать 23.12.2006.

Формат 60×84/16. Бумага офсетная. Усл. печ. л. 10,4.

Уч.-изд. л. 10,1. Гарнитура Computer Modern.

Тираж 200 экз. Заказ 1732.

Ульяновский государственный технический университет
432027, Ульяновск, ул. Северный венец, 32