Сибирское отделение Российской Академии наук
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ им.Г.И. Будкера

Э.А. Бибердорф, Н.И. Попова

РЕШЕНИЕ ЛИНЕЙНЫХ СИСТЕМ С ГАРАНТИРОВАННОЙ ОЦЕНКОЙ ТОЧНОСТИ РЕЗУЛЬТАТОВ

(часть первая)

ИЯФ 99-49

http://www.inp.nsk.su/publications

НОВОСИБИРСК
1999

Сибирское отделение Российской Академии наук
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ им.Г.И. Будкера

Э.А. Бибердорф, Н.И. Попова

РЕШЕНИЕ ЛИНЕЙНЫХ СИСТЕМ С ГАРАНТИРОВАННОЙ ОЦЕНКОЙ ТОЧНОСТИ РЕЗУЛЬТАТОВ

(часть первая)

ИЯФ 99-49

НОВОСИБИРСК
1999

Решение линейных систем с гарантированной оценкой точности результатов (часть первая)

Н.И. Попова

Институт ядерной физики им. Г.И. Будкера, 630090, Новосибирск

Э.А. Бибердорф

Институт Математики СО РАН, Россия

В предлагаемой работе подробно описывается созданный нами на языке ФОРТРАН–90 пакет программ, реализующих новый тип алгоритмов решения систем линейных уравнений. При его написании были широко использованы результаты и рекомендации законченной теории таких алгоритмов, разработанной под руководством акад. С.К. Годунова в Институте Математики СО РАН. Предлагаемый подход, в отличие от известных и распространённых, позволяет решать системы не только с квадратными, но и с произвольными прямоугольными матрицами. При этом используются исключительно ортогональные преобразования, выделенные тем, что они не ухудшают обусловленности исследуемой системы. Из наиболее важных черт нового метода особо отметим две: 1) учитываются и суммируются неизбежно возникающие при счёте машинные погрешности, что позволяет *наряду с результатом привести оценку его точности*, 2) при возникновении аварийной ситуации выполняется диагностика и *в сообщении об аварии указывается также ее возможная причина.*

Solution of the linear systems with the guaranteed estimate of the results accuracy (part 1)

E.A. Biberdorf, N.I. Popova

Abstract

Developed by us a FOTRAN-90 based program package is described in details. The package realize new types of algorithms for linear equations systems solveing. We widely used results and recommendations of completed theory of such algorithms that was created under leadership of academician S. K. Godunov in Institute of Mathematics SB RAS. The suggested way is different from common ones. It allows to solve systems not only with squared but also with rectangular matrices. At that only orthogonal transformations are used because they do not deteriorate condition of the investigated system. From the most important features of the new method we'll remark out two: 1) all the computer rounding errors that inevitably appear during calculations are taken into account and summed and along with the result an estimate of it's accuracy is carried out; 2) in emergency situation a diagnostick program starts and shows a description on possible cause of fault.

1 Введение

Поводом для проведения данной работы послужили серьезные затруднения, с которыми пришлось столкнуться при применении метода наименьших квадратов к обработке результатов численных экспериментов.

Некоторую функцию $y(z)$, приближенные значения которой в ряде точек $z_1, z_2, \ldots, z_N$ известны, надо было аппроксимировать полиномом $g(z) = \sum_{j=1}^{M} c_j z^{p(j)}$ и подобрать коэффициенты c_j так, чтобы значения этого полинома в точках $z_1, z_2, \ldots, z_N$ оказались минимально удалены от значений $y(z_i)$. Возникающую при этом линейную по отношению к неизвестным c_j задачу нетрудно представить в форме

$$Zc + r = y\,, \tag{1}$$

где матрица Z размера $N \times M$ состоит из элементов $Z_{ij} = z_i^{p(j)}$. Интерполяцию по методу наименьших квадратов можно считать успешной, если найдены векторы c и r, причем норма $\|r\|$ минимальна (по поводу обозначений см. раздел 2).

Классически первым шагом решения систем подобных (1) всегда является сведение их к так называемой *нормальной* системе уравнений [4],[9]

$$Ac = f \tag{2}$$

с квадратной $M \times M$ матрицей A. К такой системе теоретически применим весь арсенал методов как точных (различные модификации метода исключения Гаусса), так и итерационных (см. [7]).

На практике же пришлось столкнуться с рядом осложнений. Сначала при решении системы методом Гаусса вычислительная программа внезапно прекращала свою работу и сообщала о делении на нуль. Это могло свидетельствовать как о том, что исходная матрица в машинном представлении оказалась вырожденной, так и о том, что она "портилась"и

вырождалась в процессе связанных с методом Гаусса преобразований за счёт машинных округлений.

Следующим был испробован способ, предусматривающий предварительное разложение матрицы A на произведение BC нижней и верхней треугольных матриц и их последовательное обращение по методу Гаусса (см. §19 в книге [7] и статью [4]). Матрицу A удалось обратить, но это не была победа: среди диагональных элементов обратной матрицы A^{-1} находились большие отрицательные числа, что противоречило теории (в обсуждаемом случае диагональные элементы A^{-1} являются обратными *квадратами* весов неизвестных [4]). Избавиться от отрицательных элементов удалось, используя возможности пакета написанных на ФОРТРАНе программ [8], позволяющих производить вычисления с произвольной точностью. При этом для обращения матрицы размера 9×9 пришлось работать с вещественными числами, десятичное представление которых содержало 50 знаков после запятой.

Но и эта ситуация не могла считаться удовлетворительной, поскольку естественно возник и оставался открытым кардинальной важности вопрос: *сколько же знаков полученного решения являются верными*?

Напомним, что в методе наименьших квадратов недостаточно найти значения неизвестных c_j, необходимо также вычислить и указать их дисперсии σ_j, которые возникают из–за "шума"входных данных и определяются соотношением [4],[9]

$$\sigma_j = \|r\| \sqrt{\frac{\alpha_{jj}}{N-M}},$$

где за α_{jj} обозначен j-ый диагональный элемент матрицы A^{-1}. Но, как было показано выше, значения элементов обратной матрицы могут быть найдены с ошибкой, информация о величине которой полностью отсутствует. Сам собой напрашивается закономерный вывод: не имея гарантированной оценки точности вычислений невозможно достоверно оценить и влияние статистических погрешностей.

Мы проиллюстрировали проблематику на примерах конкретных алгоритмов, но необходимо подчеркнуть, что подобные недостатки присущи фактически всем используемым сегодня классическим способам решения систем уравнений. В случаях, когда эти системы никак не связаны с методом наименьших квадратов статистический аспект, конечно, полностью исчезает, но вопрос о точности результата остается в полной мере. Также неудовлетворительно, на наш взгляд, почти во всех доступных библиотеках по линейной алгебре организована реакция на аварийные

ситуации. Чаще всего пользователь получает в этом случае краткое сообщение типа "divided by zero"или еще более загадочное "invalid floating point operation"и ему предоставляется полная свобода в поисках причины неполадки.

Поэтому имеется острая необходимость в создании вычислительных алгоритмов нового поколения, которые отвечали бы следующим общим требованиям:

1. Если счёт успешно завершен, то вместе с результатом должна быть приведена оценка его точности (число верных знаков). Напомним в этой связи, что в физике, например, указание приближенной величины без одновременного обозначения интервала возможной ошибки не признается полноценной информацией.

2. Если происходит аварийная остановка (деление на нуль, переполнение), то должна производиться диагностика причины (слишком большая матрица коэффициентов, слишком большая или маленькая правая часть, близость матрицы к вырожденной и т.п.). В будущем, возможно, наши программы смогут давать и советы типа "следует увеличить точность вычислений до N знаков после запятой".

Отметим, что теоретическая база для успешного решения этой важной задачи сегодня имеется (см. ниже).

Практически во всех руководствах по вычислительной линейной алгебре, подчеркивается, что точность решения линейной системы зависит от такой численной характеристики матрицы коэффициентов A как обусловленность $\mu(A)$, которая всегда больше единицы или равна ей. Если матрица плохо обусловлена (число обусловленности велико), то решение системы может очень резко меняться при малых изменениях элементов матрицы и правой части. Точно также чутко реагирует решение плохо обусловленной системы на вычислительные погрешности. К сожалению, далеко не все разработчики пакетов программ математического обеспечения уделяют обусловленности системы должное внимание. В этой связи с положительной стороны следует отметить известный пакет MATLAB. Его программы решения системы линейных уравнений и обращения матрицы одновременно вычисляют число обусловленности и, если оно оказалось велико, огорчают пользователя предупреждением, что результат вычислений *может быть недостоверным*. При этом неизвестной остаётся не только погрешность результата, но и погрешность вычисления самого числа обусловленности.

Заметим, что имеют место следующие соотношения

$$\mu(A^*A) = \mu(A)^2 \quad \text{и} \quad \mu(AB) \le \mu(A)\mu(B), \tag{3}$$

откуда следует весьма важная и противоречащая классическому подходу практическая рекомендация: при решении задачи методом наименьших квадратов следует *отказаться от приведения системы к нормальному виду* (2), а пытаться решать исходную систему (1). А из неравенства (3) видно, что любое преобразование, применённое к решаемой системе может ухудшить её обусловленность. Поэтому в монографии [6] - одной из первых и наиболее основательных работ, посвященных детальному анализу влияния машинных округлений на точность результата вычислений - особо подчеркивается, что основой для точного алгоритма решения линейных алгебраических систем должны быть *ортогональные преобразования*. Только они, в отличие от всех других не меняют (а следовательно не ухудшают) обусловленности системы.

Под руководством С.К.Годунова в Институте Математики СО РАН много лет велась работа по созданию алгоритмов, отвечающих сформулированным выше требованиям. Один из первых опытов по их программной реализации – комплекс программ по линейной алгебре (КОПЛА) на языке АЛГОЛ – был предназначен для машины БЭСМ-6 и позволял производить расчёты с произвольными квадратными матрицами размера порядка 120×120. За подробным описанием этой программной библиотеки можно обратиться к [3]. Результаты теоретических разработок и вычислительных экспериментов суммированны и подробно изложены в монографии [2]. Позже в [5] А.Н. Малышевым исправлены некоторые неточности, допущенные в [2], и предложен ряд усовершенствований. В книге проведён по возможности лаконичный анализ погрешностей, что делает её более доступной для первого чтения. Там же приведены алгоритмы на языке ФОРТРАН-77 – библиотека ЛИНА – недостатком которой, на наш взгляд, является как раз отсутствие оценок погрешностей.

В настоящей работе мы попытались максимально использовать возможности представленной в [2] технологии учёта вычислительных ошибок. Теоретическая часть посвящена краткому изложению известных фактов из линейной алгебры, основанным на них алгоритмам и способам контроля за точностью вычислений. Опущенные подробности можно найти в предложенной литературе. Описание процедур на языке ФОРТРАН-90 помещено в Приложение.

Мы предполагаем и далее продолжать эту деятельность. Реализованные в данной работе алгоритмы существенно ориентированы на стандартные модели представления вещественных чисел с одинарной, двойной или четверной точностью. Иногда это может оказаться серьезным ограничением – один такой случай уже рассматривался выше. В практике ИЯФ'а известна ситуация, когда успешное решение серьезной физической задачи потребовало использования в десятичном представлении числа 300 знаков после запятой [10].

Ясно, что вычислительные алгоритмы нового поколения должны стать доступны тем, кто работает с произвольными точностями и это мы считаем своей первой задачей. Отметим, что здесь, возможно, потребуется разработка специального механизма учёта вычислительных ошибок.

Вторая задача, которую мы также ставим перед собой, формулируется так: создать вычислительные программы, для которых оценка погрешности была бы не выходным, а *входным* параметром и задавалась бы пользователем. Для решения обеих этих задач мы планируем широко использовать возможности пакета [8] для счета с произвольной точностью. Планируется также применить итерационные методы для уточнения получаемых в процессе вычислений результатов.

2 Краткие сведения из линейной алгебры

Назначение этого раздела – договориться об обозначениях. Все факты и определения, упомянутые здесь, давно используются в вычислительной линейной алгебре, но неспециалистам некоторые из них могут быть неизвестны. Все утверждения мы приводим без доказательств. Для тех, кого они заинтересуют, мы даём ссылки на литературу.

2.1 Основные понятия

Строчку $x = (x_1, x_2, \ldots, x_M)$ или столбец

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix}$$

мы будем называть *векторами* длины M из пространства $\mathbf{R_M}$. *Скалярное произведение* для них определим формулой

$$(x, y) = x_1 y_1 + x_2 y_2 + \ldots + x_M y_M \,.$$

Длиной или *нормой* вектора x называется число

$$\|x\| = \sqrt{(x, x)} = (\sum_{j=1}^{N} |x_i|^2)^{1/2} \,.$$

Отметим важные неравенства, которые играют исключительную роль в оценках точности вычисления скалярных произведений и умножения матрицы на вектор.

Неравенство треугольника:

$$\big|\|x\| - \|y\|\big| \le \|x + y\| \le \|x\| + \|y\|.$$

Неравенство Коши-Буняковского:

$$\big|(x, y)\big| \le \|x\|\,\|y\|.$$

Прямоугольная таблица

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{pmatrix},$$

состоящая из N строк и M столбцов, называется *матрицей* с размерами $N \times M$.

Квадратная матрица, элементы которой суть $a_{ij} = \delta_{ij}$, где δ_{ij} – символ Кронеккера, называется *единичной* и обозначается I.

Транспонированием матрицы называется замена строк этой матрицы её столбцами с сохранением их номеров. Матрица, полученная таким образом из матрицы A, называется *транспонированной* по отношению к матрице A и обозначается A^T. Если размеры A равны $N \times M$, то размеры транспонированной матрицы A^T равны $M \times N$. Повторное транспонирование приводит к исходной матрице.

Пусть A – матрица с комплексными элементами. $\bar{A}$ – матрица, комплексносопряженная к A, т.е. получаемая из матрицы A заменой её элементов на комплексносопряженные. Матрица $A^* = \bar{A}^T$ называется *сопряженной* к A.

Если $A^* = A$ (что может иметь место только для квадратных матриц), то матрица A - *самосопряженная (эрмитова)*. Если при этом матрица A – вещественная, то она называется *симметрической*.

Пусть A – квадратная матрица размера $M \times M$, тогда матрица B называется *обратной* к матрице A, если $BA = I$. Если такая B существует, то матрица A называется неособенной. Обратную матрицу обозначают A^{-1}. Заметим, что $A^{-1}A = AA^{-1}$. Задача обращения матрицы тесно связана с задачей решения неоднородной системы уравнений. Если для матрицы A известна обратная матрица, то умножая обе части уравнения $Ax = f$ слева на A^{-1}, получаем решение системы $x = A^{-1}f$.

Определение 1 *Если $A^{-1} = A^*$, то матрица A называется ортогональной.*

Свойства: Ортогональные матрицы не меняют нормы вектора: $\|x\| = \|Ax\|$. Это свойство ортогональных матриц эквивалентно их определению. Произведение двух ортогональных матриц есть ортогональная матрица.

Пример 1. Примером ортогональной матрицы может служить *преобразование отражения* – матрица

$$P = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1N} \\ P_{21} & P_{22} & \dots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \dots & P_{NN} \end{pmatrix},$$

элементы которой вычисляются по формулам

$$P_{ij} = \delta_{ij} - \frac{2r_i r_j}{\|r\|^2}, \qquad (4)$$

где $r = (r_1, r_2, \dots, r_N)$ - некоторый ненулевой вектор. Преобразование $Px = y$, действующее в пространстве вектор-столбцов $\mathbf{R_N}$ можно выразить так

$$y = x - \frac{2(r, x)}{\|r\|^2} r\,. \qquad (5)$$

Пусть вектор p коллинеарен вектору r и специальным образом нормирован: $p = \sqrt{2}r/\|r\|$, $\|p\| = \sqrt{2}$. Тогда формула (5) несколько упрощается:

$$y = Px = x - (p, x)p.$$

Геометически преобразование P интерпретируется как отражение вектора x относительно гиперплоскости, определённой уравнением $(p, x) = 0$, проходящей через начало координат и ортогональной вектору p^T. Очевидные свойства матрицы P: $P^* = P = P^{-1}$. Преобразование отражения понадобится нам для построения алгоритма приведения прямоугольных матриц к двухдиагональному виду. При этом мы будем использовать теорему

Теорема 1 *Для любых двух векторов x, y из $\mathbf{R_N}$ таких, что $\|x\| = \|y\|$, существует матрица отражения P такая, что $Px = y$.*

Очевидно, что элементы матрицы P определяются вектором r, $r^T = y - x$ по формуле (4).

Пример 2. Другим примером ортогональной матрицы является матрица Q размера $N \times N$, которая при действии на вектор x с координатами x_j, $1 \le j \le N$, меняет две его координаты местами: $y = Qx$, $y_j = x_j$ при $j \ne j_1, j_2$; $y_{j_1} = x_{j_2}$ и $y_{j_2} = x_{j_1}$. В результате умножения некоторой матрицы на Q слева у исходной матрицы меняются местами строки с номерами j_1 и j_2, а при умножении на Q справа, местами меняются столбцы j_1 и j_2.

2.2 Численные характеристики матриц

Определение 2 *Число $\lambda = \lambda_j(A)$ называется собственным значением квадратной матрицы A, если существует ненулевой вектор-столбец x такой, что*

$$Ax = \lambda x\,.$$

Свойства: Квадратная матрица размера $N \times N$ имеет ровно N собственных значений возможно кратных. Если у матрицы T есть обратная T^{-1}, то собственные значения A и TAT^{-1} совпадают. Если матрица Q ортогональна, то $\lambda_j(A) = \lambda_j(QAQ^*)$, так как по определению ортогональной матрицы $Q^* = Q^{-1}$. Квадратная матрица является особенной, т.е. не имеет обратной, если и только если среди собственных чисел есть нулевые. Собственные числа самопряженной матрицы вещественны. Собственные числа ортогональных матриц по модулю равны единице.

Любая прямоугольная $N \times M$-матрица A допускает представление в виде $A = UKV^*$, где U и V – ортогональные квадратные $N \times N$– и $M \times M$–матрицы, а матрица K состоит из диагональной квадратной клетки Σ размером $N_0 \times N_0$ ($N_0 = \min\{N, M\}$) с неотрицательными элементами на главной диагонали и при $N \ne M$ из дополнительных нулевых строк или столбцов

$$K = \left(\Sigma \vdots\ 0 \right) \quad \text{при } M < N\,,$$

$$K = \begin{pmatrix} \Sigma \\ \dots \\ 0 \end{pmatrix} \quad \text{при } M > N\,,$$

$$K = \Sigma \quad \text{при } N = M \quad (\sigma_{N_0} \ge \sigma_{N_0-1} \ge \dots \ge \sigma_1 > 0)\,,$$

где

$$\Sigma = \begin{pmatrix} \sigma_{N_0} & 0 & \dots & 0 \\ 0 & \sigma_{N_0-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_1 \end{pmatrix}.$$

Определение 3 *Числа $\sigma_j = \sigma_j(A)$ называются сингулярными числами матрицы A, а разложение $A = UKV^*$ - сингулярным разложением матрицы A.*

Свойства: Сингулярные числа определяются по матрице A однозначно. Если $B = QA$ или $B = AQ$, где Q - ортогональная матрица, то из определения видно, что $\sigma_j(A) = \sigma_j(B)$. Сингулярные числа самосопряженных матриц совпадают с абсолютными значениями их собственных значений, т.е. если $A = A^*$, то $\sigma_j(A) = |\lambda_j(A)|$. Существует связь между сингулярными числами $\sigma_j(A)$, $j = 1, \dots, N_0$, $N_0 = \min\{N, M\}$ матрицы A размера $N \times M$ и собственными значениями составной матрицы размера $(N + M) \times (N + M)$

$$\mathcal{A} = \begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix}.$$

Пронумеруем собственные числа $\lambda_j(\mathcal{A})$, $j = 1, \dots, N + M$ по возрастанию (матрица $\mathcal{A}$ – самосопряженная, следовательно, как сказано выше, все её собственные значения вещественны). Тогда

$$\begin{array}{lll} \lambda_1(\mathcal{A}) = -\sigma_{N_0}(A) & \lambda_{N_0+1}(\mathcal{A}) = 0 & \lambda_{N+M-N_0}(\mathcal{A}) = \sigma_1 \\ \lambda_2(\mathcal{A}) = -\sigma_{N_0+1} & \lambda_{N_0+2}(\mathcal{A}) = 0 & \lambda_{N+M-N_0+2}(\mathcal{A}) = \sigma_2 \\ \dots & \dots & \dots \\ \lambda_{N_0}(\mathcal{A}) = -\sigma_1 & \lambda_{N+M-N_0}(\mathcal{A}) = 0 & \lambda_{N+M}(\mathcal{A}) = \sigma_{N_0} \end{array}.$$

Все сингулярные числа ортогональной матрицы равны единице: $Q^* = Q^{-1}$, $\sigma_j(Q) = 1$.

Определение 4 *Нормой матрицы A называется число*

$$\|A\| = \sup_{x \ne 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|\,.$$

Свойства: Оказывается, норма матрицы совпадает с её максимальным сингулярным числом $\|A\| = \sigma_{max}(A)$.

Введём числовые характеристики матриц, которые легко вычисляются непосредственно через её элементы. Положим

$$\mathcal{M}(A) = \max\left\{ \max_i \sum_{j=1}^{M} |a_{ij}| \,,\ \max_j \sum_{i=1}^{N} |a_{ij}| \right\}, \tag{6}$$

так что величина $\mathcal{M}(A)$ равна максимуму по строкам и по столбцам сумм модулей элементов соответствующих строк и столбцов. Для двухдиагональной матрицы D

$$D = \begin{pmatrix} d_1 & b_1 & & & \\ & d_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & d_{M-1} & b_{M-1} \\ & & & & d_M \end{pmatrix}$$

формула (6) принимает вид

$$\mathcal{M}(D) = \max\left\{ \begin{array}{l} \max_{1\le i\le N-1} \left(|d_i| + |b_{i+1}|\right), \\ \max_{2\le i\le N} \left(|d_i| + |b_i|\right) . \end{array} \right. \tag{7}$$

Норму матрицы $\|A\|$ и величину $\mathcal{M}(A)$ связывают неравенства

$$\|A\| \le \mathcal{M}(A) \le \max\{\sqrt{M}, \sqrt{N}\}\|A\|. \tag{8}$$

Числовую характеристику $\mathcal{F}(A) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} |a_{ij}|^2}$ будем называть фробениусовой нормой A. Известно, что

$$\|A\| \le \mathcal{F}(A) \le \min\{\sqrt{M}, \sqrt{N}\}\|A\|.$$

Числовая характеристика $\mathcal{F}(A)$ так же, как и операторная норма $\|A\|$, является ортогональным инвариантом матрицы A, однако $\mathcal{F}(A)$ просто вычисляется по явным формулам, что в некоторых случаях делает удобным вычисление $\mathcal{F}(A)$ и последующую оценку $\|A\|$ через неё с помощью этого неравенства.

При оценке точности решения системы линейных уравнений исключительную роль играет число обусловленности матрицы:

Определение 5 *Числом обусловленности квадратной матрицы называется величина*

$$\mu(A) = \sup_{x\ne 0\ \xi\ne 0} \left\{ \frac{\|Ax\|}{\|A\xi\|} \frac{\|\xi\|}{\|x\|} \right\}.$$

Заметим, что при $\sigma_1(A) \ne 0$

$$\mu(A) = \frac{\sup_{x\ne 0}(\|Ax\|/\|x\|)}{\inf_{\xi\ne 0}(\|A\xi\|/\|\xi\|)} = \frac{\sigma_N(A)}{\sigma_1(A)} < \infty .$$

Если $\sigma_1(A) = 0$, то, очевидно, $\mu(A) = \infty$. Таким образом $\mu(A)$ является безразмерной характеристикой матрицы, обращающейся в бесконечность для вырожденных матриц. Из определения вытекает, что $\mu(A) \ge 1$. $\mu(A)$ называют и *числом обусловленности системы* $Ax = f$. Чем меньше $\mu(A)$, тем система считается лучше обусловленной. Плохо обусловленная система – это система с очень большим $\mu(A)$. Следующая теорема устанавливает зависимость между точностью задания элементов матрицы A и правой части f, точностью решения системы x и числом обусловленности:

Теорема 2 *Пусть A и B – квадратные матрицы размера $N \times N$, f и g – векторы длины N, причем*

$$\frac{\|B\|}{\|A\|}\mu(A) < 1,$$

тогда для решений x и y систем

$$Ax = f, \qquad (A+B)y = f+g$$

справедливо следующее неравенство

$$\frac{\|y-x\|}{\|x\|} \le \left(\frac{\|B\|}{\|A\|} + \frac{\|g\|}{\|f\|}\right) \frac{\mu(A)}{1 - \frac{\|B\|}{\|A\|}\mu(A)}.$$

2.3 Последовательности Штурма трёхдиагональных матриц

Определим последовательности Штурма, которые понадобятся в дальнейшем при вычислениях собственных значений симметричных трёхдиагональных матриц (и сингулярных чисел двухдиагональных матриц).

Пусть заданы последовательности вещественных чисел

$$\begin{array}{l} b_1 \ne 0,\ b_2 \ne 0, \ldots, \\ c_2 \ne 0,\ c_3 \ne 0, \ldots, \\ d_1,\ d_2,\ d_3, \ldots, \end{array}$$

причём $b_j c_j > 0$ для $j \geq 2$. Определим дробно–рациональные функции $\mathcal{P}_0(\lambda)$, $\mathcal{P}_1(\lambda)\ldots$ по следующим формулам:

$$\mathcal{P}_0(\lambda) = 0, \qquad \mathcal{P}_k(\lambda) = \frac{|c_{k+1}|}{d_k - \lambda - |b_k|\mathcal{P}_{k-1}(\lambda)} \quad (k = 1, 2, \ldots). \tag{9}$$

Функции $\mathcal{P}_k(\lambda)$ всюду, за исключением своих полюсов, принимают конечные значения. Будем считать, что значения $\mathcal{P}_k(\lambda)$ в полюсах равны $+\infty$. Видно, что полюс $\mathcal{P}_{k-1}(\lambda)$ одновременно является нулём $\mathcal{P}_k(\lambda)$.

Определение 6 *Последовательность* $\mathcal{P}_k(\lambda) \quad (k = 1, 2, \ldots)$ *называется последовательностью Штурма.*

Последовательность (9) для $k = 1, 2, \ldots, N$ можно связать с главными минорами $\mathcal{D}_k(\lambda)$ k-го порядка трёхдиагональной матрицы

$$A = \begin{pmatrix} d_1 - \lambda & c_2 & 0 & \ldots & 0 \\ b_2 & d_2 - \lambda & c_3 & \ldots & \vdots \\ 0 & b_3 & d_3 - \lambda & c_4 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & b_{M-1} & d_{M-1} - \lambda & c_M \\ 0 & \ldots & 0 & b_M & d_M - \lambda \end{pmatrix}.$$

Полагая $\mathcal{D}_0(\lambda) = 1, \quad \mathcal{D}_1(\lambda) = d_1 - \lambda$, получим равенства, связывающие три последовательных минора

$$\mathcal{D}_k(\lambda) = (d_k - \lambda)\mathcal{D}_{k-1}(\lambda) - b_k c_k \mathcal{D}_{k-2}(\lambda) \quad (2 \leq k \leq N),$$

которые легко преобразовать к виду

$$|d_k| \left(\frac{|c_k|\mathcal{D}_{k-2}(\lambda)}{\mathcal{D}_{k-1}(\lambda)} \right) - (d_k - \lambda) + |c_{k+1}| \left(\frac{\mathcal{D}_k(\lambda)}{|c_{k+1}|\mathcal{D}_{k-1}(\lambda)} \right) = 0$$

или

$$\frac{|c_{k+1}|\mathcal{D}_{k-1}(\lambda)}{\mathcal{D}_k(\lambda)} = \frac{|c_{k+1}|}{d_k - \lambda - |b_k||c_k|\mathcal{D}_{k-2}(\lambda)/\mathcal{D}_{k-1}(\lambda)}.$$

Последнее соотношение совпадает с (9) при $\mathcal{P}_k(\lambda) = |c_{k+1}|\mathcal{D}_{k-1}(\lambda)/\mathcal{D}_k(\lambda)$ $(1 \leq k \leq N-1)$, где $\mathcal{D}_k(\lambda)$ – главные миноры трёхдиагональной матрицы, зависящие от параметра λ. Таким образом дробно–рациональные функции $\mathcal{P}_k(\lambda)$ для $k = 1, 2, \ldots, N-1$ совпадают с отношениями $\frac{|c_{k+1}|\mathcal{D}_{k-1}(\lambda)}{\mathcal{D}_k(\lambda)}$.

Элемент c_{N+1} в матрице отсутствует, поэтому будем считать, что $c_{N+1} = 1$ и

$$\mathcal{P}_N(\lambda) = \frac{\mathcal{D}_{N-1}(\lambda)}{\mathcal{D}_N(\lambda)} = \frac{1}{d_N - \lambda - |b_N|\mathcal{P}_{N-1}(\lambda)}.$$

Суть следующей леммы в том, что несмотря на то, что матрица A не является симметричной, можно гарантировать, что все корни её главных миноров, в том числе и все собственные числа, вещественны и различны:

Лемма 1 *Все нули полиномов* $\mathcal{D}_i(\lambda)$ $(1 \leq i \leq N)$ *вещественны и различны, причем между каждыми двумя соседними нулями* $\mathcal{D}_{j+1}(\lambda)$ *имеется в точности один нуль* $\mathcal{D}_j(\lambda)$ $(1 \leq j \leq N-1)$.

Отсюда следует, что корни и полюса $\mathcal{P}_i(\lambda)$ вещественны и различны. В дальнейшем мы будем использовать следующую теорему Штурма:

Теорема 3 (Штурм) *Для любого вещественного* λ_0 *число корней* λ *уравнения* $\mathcal{D}_N(\lambda) = 0$ *(то есть собственных значений матрицы* A*) таких, что* $\lambda < \lambda_0$*, совпадает с числом неположительных значений в последовательности*

$$\mathcal{P}_1(\lambda_0),\ \mathcal{P}_2(\lambda_0),\ \ldots,\ \mathcal{P}_N(\lambda_0).$$

Подробное доказательство теоремы приведено в книге [2].

3 Описание алгоритмов

3.1 Решение линейной алгебраической системы

Рассмотрим линейную систему уравнений

$$Ax + r = f, \tag{10}$$

где A – числовая матрица размера $N \times M$ и f – числовой вектор длины N являются заданными величинами. Предположим, что нам удалось подобрать такие ортогональные матрицы P и Q, размера $N \times N$ и $M \times M$ соответственно, что матрица PAQ имеет очень простой вид

$$PAQ = \begin{pmatrix} D \\ \mathbf{0} \end{pmatrix}, \tag{11}$$

где D – квадратная двухдиагональная матрица $M \times M$

$$D = \begin{pmatrix} d_1 & b_1 & & & \\ & d_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & d_{M-1} & b_{M-1} \\ & & & & d_M \end{pmatrix},$$

а за 0 обозначена нулевая матрица размера $(N-M)\times M$. Способ реализации такого разложения будет описан в следующем разделе. Второе наше предположение состоит в том, что все диагональные элементы матрицы D отличны от нуля: $|d_i| > 0$.

Итак, пусть оба наших предположения выполнены. Домножим уравнение (10) на матрицу P слева. Получим

$$PAx + Pr = Pf.$$

Воспользуемся тем, что матрица Q ортогональна, то есть $QQ^* = I$, где I – единичная матрица размера $M \times M$. Тогда

$$PAx + Pr = PAIx + Pr = PAQQ^*x + Pr = Pf.$$

Обозначим $w = Q^*x$, $\rho = Pr$, $h = Pf$, то есть $x = Qw$, $r = P^*\rho$, $f = P^*h$. Причем векторы ρ и h представим в виде

$$\rho = \begin{pmatrix} \rho^{(1)} \\ \rho^{(2)} \end{pmatrix} \qquad h = \begin{pmatrix} h^{(1)} \\ h^{(2)} \end{pmatrix}.$$

Здесь векторы $\rho^{(1)}$ и $h^{(1)}$ длины M, а векторы $\rho^{(2)}$ и $h^{(2)}$ длины $N - M$. С этими обозначениями система (10) распадается на две

$$Dw + \rho^{(1)} = h^{(1)}, \qquad \rho^{(2)} = h^{(2)}. \tag{12}$$

Заметим, что $\|\rho\| = \|Pr\| = \|r\|$, так как умножение вектора на ортогональную матрицу не меняет его нормы, а с другой стороны, $\|\rho\| = \sqrt{\|\rho^{(1)}\|^2 + \|\rho^{(2)}\|^2}$. Компонента $\rho^{(2)}$ вектора ρ из уравнений (12) находится однозначно. Поэтому, чтобы минимизировать $\|\rho\|$, нужно правильно распорядиться компонентой $\rho^{(1)}$. Но уравнение для w и $\rho^{(1)}$ является нормальной системой вида (2) с невырожденной матрицей. То есть $\min\|\rho^{(1)}\| = 0$, если w является решением системы $Dw = h^{(1)}$, которое легко вычисляется по рекурентным формулам:

$$\begin{aligned} w_M &= \tfrac{h^{(1)}_M}{d_M} \\ w_i &= \tfrac{1}{d_i}\left(h^{(1)}_i - b_i w_{i+1}\right), \qquad 1 \le i < M. \end{aligned} \tag{13}$$

Таким образом для решения системы (10) нужно проделать следующую последовательность шагов:

Шаг 1. Найти ортогональные матрицы P, Q и D такие, что выполнено равенство (11).

Шаг 2. Вычислить $h = \begin{pmatrix} h^{(1)} \\ h^{(2)} \end{pmatrix} = Pf$.

Шаг 3. Найти вектор w по формулам (13).

Шаг 4. Вернуться в исходный базис

$$x = Qw, \quad r = P^* \begin{pmatrix} 0 \\ h^{(2)} \end{pmatrix}.$$

3.2 Приведение матрицы к двухдиагональному виду

В этом разделе мы ставим перед собой задачу построить ортогональные матрицы P и Q, осуществляющие приведение произвольной матрицы A размера $N \times M, N > M$ к виду (11). Возьмем первый столбец матрицы $A^{(0)} = A$:

$$\begin{pmatrix} a^{(0)}_{11} \\ a^{(0)}_{21} \\ \vdots \\ a^{(0)}_{N1} \end{pmatrix}.$$

Его норма как вектора размера N равна $\sqrt{\sum_{i=1}^{N}\left(a^{(0)}_{i1}\right)^2}$. Положим $d_1 = -\mathrm{sgn}(a^{(0)}_{11})\sqrt{\sum_{i=1}^{N}\left(a^{(0)}_{i1}\right)^2}$.

Замечание. Специальный выбор знака d_1 важен для последующего вывода оценки точности вычислений.

Построим новый вектор

$$\begin{pmatrix} a^{(1)}_{11} \\ a^{(1)}_{21} \\ \vdots \\ a^{(1)}_{N1} \end{pmatrix} = \begin{pmatrix} d_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Очевидно, что норма нового вектора-столбца равна норме исходного. Значит существует нормированный вектор $p^{(1)}$, $\|p^{(1)}\| = \sqrt{2}$ размера N и ортогональная матрица отражения $P^{(1)}$ размера $N \times N$

$$r_i^{(1)} = a_{i1}^{(1)} - a_{i1}^{(0)}$$
$$p^{(1)} = \sqrt{2}r^{(1)}/\|r^{(1)}\|$$
$$P_{ij}^{(1)} = \delta_{ij} - p_i^{(1)} p_j^{(1)}.$$

такие, что $P^{(1)}A^{(0)} = A^{(1)}$, где

$$A^{(1)} = \begin{pmatrix} d_1 & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \dots & * \end{pmatrix}.$$

Звездочками мы обозначаем, вообще говоря, ненулевые элементы матрицы.

Следующим нашим шагом будет зануление элементов первой строки матрицы $A^{(1)}$, начиная с третьего. Пусть $b_1 = -\mathrm{sgn}(a_{12}^{(1)})\sqrt{\sum_{i=2}^{M} a_{1i}^{(1)}}$ и $(a_{11}^{(2)}, a_{12}^{(2)}, \dots, a_{1M}^{(2)}) = (d_1, b_1, 0, \dots, 0)$ – вектор размера M. Положим

$$s_i^{(1)} = a_{1i}^{(2)} - a_{1i}^{(1)}$$
$$q^{(1)} = \sqrt{2}s/\|s\|$$
$$Q_{ij}^{(1)} = \delta_{ij} - q_i^{(1)} q_j^{(1)}.$$

Заметим, что матрица $Q^{(1)}$ имеет вид

$$Q^{(1)} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \dots & * \end{pmatrix}.$$

Следовательно умножение любого вектора размера M на матрицу Q не меняет его первой компоненты. Поэтому умножение на $Q^{(1)}$ слева любой матрицы соответствующего размера не изменит первой строки исходной матрицы, домножение справа не изменит первого столбца, то есть первые столбцы у матриц $A^{(1)}$ и $A^{(2)} = A^{(1)}Q^{(1)}$ совпадают. При этом у матрицы $A^{(2)}$ в первой строчке будет стоять вектор

$$(d_1, b_1, 0, \dots, 0).$$

Тем самым установлено, что

$$A^{(2)} = \begin{pmatrix} d_1 & b_1 & 0 & \dots & 0 \\ 0 & * & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & * & * & \dots & * \end{pmatrix}.$$

Очевидно, что этот процесс можно продолжать по индукции. И для того, чтобы с помощью ортогональных преобразований превратить исходную матрицу в двухдиагональную, нужно следовать следующему алгоритму:

Шаг 0. Присвоение начальных данных

$$A^{(0)} = A, \qquad P_0 = I_N, \qquad Q_0 = I_M$$

Шаг 2i-1. $(1 \le i \le M)$ Зануление элементов i-го столбца матрицы $A^{(2i-2)}$, начиная с $(i+1)$-го элемента

Вычисление i-го диагонального элемента

$$d_i = -\mathrm{sgn}(a_{i,i}^{(2i-2)})\sqrt{\sum_{j=i}^{N}\left(a_{ji}^{(2i-2)}\right)^2}.$$

Присвоение новых значений элементов i-го столбца

$$a_{ji}^{(2i-1)} = a_{ji}^{(2i-2)},\ j < i; \quad a_{ii}^{(2i-1)} = d_i; \quad a_{ji}^{(2i-1)} = 0, \quad j > i.$$

Вычисление нормали к гиперплоскости отражения

$$r_j^{(i)} = a_{ji}^{(2i-1)} - a_{ji}^{(2i-2)}.$$

Нормировка нормали

$$p^{(i)} = \sqrt{2}r^{(i)}/\|r^{(i)}\|.$$

Вычисление матрицы отражения и ее действие на матрицу $A^{(2i-2)}$

$$P_{ij}^{(i)} = \delta_{ij} - p_i^{(i)} p_j^{(i)}$$
$$A^{(2i-1)} = P^{(i)}A^{(2i-1)}, \qquad P_i = P^{(i)}P_{i-1}.$$

Шаг 2i. $(1 \le i \le M-1)$ Зануление элементов i-ой строки матрицы $A^{(2i-1)}$, начиная с $(i+2)$-го элемента

Вычисление i-го наддиагонального элемента

$$b_i = -\mathrm{sgn}(a_{i,i+1}^{(2i)}) \sqrt{\sum_{j=i}^{N} \left(a_{ji}^{(2i-1)}\right)^2}$$

Присвоение новых значений элементов $i-1$-ой строки

$$a_{ij}^{(2i)} = a_{ij}^{(2i-1)}, \quad j \le i; \qquad a_{ii+1}^{(2i)} = b_i; \qquad a_{ij}^{(2i)} = 0, \quad j > i+1$$

Вычисление нормали к гиперплоскости отражения

$$s_j^{(i)} = a_{ij}^{(2i)} - a_{ij}^{(2i-1)}$$

Нормировка нормали

$$q^{(i)} = \sqrt{2} s^{(i)} / \|s^{(i)}\|$$

Вычисление матрицы отражения и ее действие на матрицу $A^{(2i-1)}$

$$Q_{ij}^{(i)} = \delta_{ij} - q_i^{(i)} q_j^{(i)};$$
$$A^{(2i)} = A^{(2i-1)} Q^{(i)}, \qquad Q_i = Q_{i-1} Q^{(i)}.$$

В результате: $P = P_M = P^{(M)} P^{(M-1)} \dots P^{(1)}$,

$Q = Q_{M-1} = Q^{(1)} Q^{(2)} \dots Q^{(M-1)}$, $PAQ = \begin{pmatrix} D \\ \mathbf{O} \end{pmatrix}$, где

$$D = \begin{pmatrix} d_1 & b_1 & & & \\ & d_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & d_{M-1} & b_{M-1} \\ & & & & d_M \end{pmatrix}.$$

3.3 Вычисление обусловленности матрицы

Пусть A – матрица размера $N \times M$, $N > M$. Как известно, число обусловленности μ и сингулярные числа $\sigma_1, \dots, \sigma_M$ матрицы не меняются при применении к ней ортогональных преобразований справа и слева:

$\mu(A) = \mu(PAQ)$. Выберем P и Q так, чтобы матрица PAQ была двухдиагональна:

$$PAQ = \begin{pmatrix} D \\ 0 \end{pmatrix},$$

то есть D, как и прежде, – квадратная размера $M \times M$ двухдиагональная матрица. Тогда $\mu(A) = \mu(D)$. Это позволяет свести вычисление числа обусловленности произвольной матрицы к нахождению минимального σ_1 и максимального σ_M сингулярных чисел квадратной двухдиагональной матрицы. Построим составную матрицу

$$\mathcal{D} = \begin{pmatrix} 0 & D \\ D^* & 0 \end{pmatrix}.$$

Она симметрическая, следовательно ее собственные числа $\lambda_1, \lambda_2, \dots, \lambda_{2M}$ вещественны. Будем считать, что собственные числа $\mathcal{D}$ занумерованы по возрастанию. Тогда связь между ними и сингулярными числами D выражается формулами

$$\begin{aligned} \lambda_1(\mathcal{D}) &= -\sigma_M(D), & \lambda_{M+1}(\mathcal{D}) &= \sigma_1(D), \\ \lambda_2(\mathcal{D}) &= -\sigma_{M-1}(D), & \lambda_{M+2}(\mathcal{D}) &= \sigma_2(D), \\ &\dots & &\dots \\ \lambda_M(\mathcal{D}) &= -\sigma_1(D), & \lambda_{2M}(\mathcal{D}) &= \sigma_M(D). \end{aligned}$$

Таким образом, вычисляя собственные числа $\lambda_{M+1}(\mathcal{D})$ и $\lambda_{2M}(\mathcal{D})$, мы тем самым находим σ_1 и σ_M. Заметим также, что ортогональными преобразованиями – перестановкой строк и столбцов – матрица $\mathcal{D}$ приводится к трехдиагональному виду :

$$Q\mathcal{D}Q^* = S = \begin{pmatrix} 0 & s_1 & 0 & \dots & 0 \\ s_1 & 0 & s_2 & \ddots & \vdots \\ 0 & s_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & s_{M-1} \\ 0 & \dots & 0 & s_{M-1} & 0 \end{pmatrix}. \tag{14}$$

где

$$s_{2i-1} = d_i, \qquad s_{2i} = b_i, \qquad 1 \le i \le M. \tag{15}$$

Из определения матрицы S следует, что ее собственные числа совпадают с собственными числами $\mathcal{D}$ (стр. 10). Тем самым задача о нахождении числа обусловленности произвольной матрицы A свелась к вычислению

собственных чисел трехдиагональной симметрической матрицы с нулевой диагональю.

Дополним последовательность (15) числами $s_0 = 1$, $s_{2M} = 1$. Для матрицы S и числа λ определим последовательность Штурма

$$\begin{aligned} &\mathcal{P}_0(\lambda) \equiv 0, \\ &\mathcal{P}_{i+1}(\lambda) = -\frac{|s_{i+1}|}{\lambda + |s_i| \mathcal{P}_i(\lambda)} \qquad 0 \le i < 2M. \end{aligned} \tag{16}$$

Алгоритм вычисления собственных чисел S базируется на теореме 3, сформулированной в разделе 2.3. Теперь очевидно, что если нам известен интервал $[\lambda_-, \lambda_+]$, содержащий собственное число λ_j матрицы S, то, поделив этот интервал пополам и вычислив для серединной точки $\lambda = \frac{1}{2}(\lambda_- + \lambda_+)$ последовательность Штурма $\mathcal{P}_i(\lambda)$, с помощью теоремы 3 можно выяснить в каком из подинтервалов $[\lambda_-, \lambda]$ или $[\lambda, \lambda_+]$ находится число λ_j. Этот подинтервал можно в свою очередь поделить пополам и применить к его середине теорему Штурма, и так далее. То есть методом бисекций теоретически мы можем найти приближенное значение λ_j с произвольной точностью. При практических вычислениях эта точность ограничена из-за неизбежных погрешностей.

Осталось обсудить, как выбирать начальный интервал $[\lambda_-, \lambda_+]$. Заметим, что собственные числа λ_{M+1} и λ_{2M}, которые нас интересуют, неотрицательны, поэтому в качестве нижней границы начального интервала естественно взять $\lambda_- = 0$. С другой стороны $\lambda_{M+1} \le \lambda_{2M} = \sigma_M = \|D\|$. А норму двухдиагональной матрицы D легко оценить через модули ее элементов d_i, b_i : $\|D\| \le \mathcal{M}(D) = \max\{\max\{|d_i| + |b_i|\}, \max\{|d_{i+1}| + |b_i|\}\}$. То есть число $\mathcal{M}(D)$ может служить верхней границей λ_+ начального интервала.

Исходя из всех приведенных соображений, алгоритм для приближенного вычисления собственного числа λ_j $(j = M+1, 2M)$ матрицы S мы формализуем следующим образом:

Шаг 1. Выбрать параметр ε допустимой погрешности. Присвоить исходные значения границам интервала

$$\lambda_- = 0, \qquad \lambda_+ = \mathcal{M}(D).$$

Шаг 2. Проверка малости интервала:
если $\lambda_+ - \lambda_- \le \varepsilon$, то перейти к **Шагу 4**;
если $\lambda_+ - \lambda_- > \varepsilon$, то перейти к **Шагу 3**.

Шаг 3. Присвоить $\lambda = \frac{1}{2}(\lambda_+ + \lambda_-)$. Вычислить последовательность Штурма по формулам (16). Пусть k – число неотрицательных элементов в последовательности $\mathcal{P}_1(\lambda), \mathcal{P}_2(\lambda), \ldots, \mathcal{P}_{2M}(\lambda)$.
Если $j \le k$, то присвоить $\lambda_+ = \lambda$.
Если $k < j$, то присвоить $\lambda_- = \lambda$.
Перейти к **Шагу 2**.

Шаг 4. Присвоить $\widehat{\lambda}_j = \frac{1}{2}(\lambda_+ + \lambda_-)$

Результатом исполнения алгоритма является $\widehat{\lambda}$ – приближенное значение λ_j, причем $|\widehat{\lambda}_j - \lambda_j| \le \varepsilon$.

4 Особенности компьютерных вычислений

4.1 Машинная арифметика, простейшие арифметические операции

При реализации на ЭВМ какого-либо численного алгоритма возникают вычислительные погрешности. Чтобы уметь их предсказывать, моделировать и оценивать, необходимо в первую очередь понимать природу ошибок, возникающих при выполнении компьютером элементарных арифметических действий. Причина неточного выполнения арифметических операций заключается в том, что вещественное число z, будь то операнд или результат операции, при размещении в памяти ЭВМ как правило заменяется на некоторое число $z_{\text{маш}}$, близкое к z. Действительно, если зафиксировать определенное целое положительное число γ, то любое вещественное число z можно представить в следующем виде:

$$z = \pm \gamma^{p(z)} m_\gamma(z)\,, \tag{17}$$

где $p(z)$ есть целое число и называется γ-ичным порядком, $m_\gamma(z) = \frac{a_1}{\gamma} + \frac{a_2}{\gamma^2} + \ldots$ – возможно бесконечная сумма правильных дробей (то есть a_j – натуральное число, $0 < a_1 < \gamma$, $0 \le a_j < \gamma$ при $2 \le j$), которая называется γ-ичной мантиссой. В то же время для размещения вещественных чисел в памяти компьютера задается γ (обычно 2, 8, или 16) и отводится конечное фиксированное число бит M_p под целое число $p_\gamma(z)$ и M_m бит под набор чисел $a_1, a_2, \ldots$ При этом очевидно, что существуют вещественные числа настолько большие, что для размещения их порядка M_p бит недостаточно. Коэффициенты a_j напротив все ограничены, но их может быть бесконечно много. То есть число, мантисса которого выражаются длинной или тем более бесконечной γ-ичной дробью, также не

может быть представлено в памяти компьютера. Иными словами, чтобы число z могло быть размещено в машинной памяти (будем называть такие числа *машинными*), его порядок должен быть ограничен

$$p_- \le p(z) \le p_+ ,$$

а мантисса должна раскладываться в конечную γ-ичную дробь

$$m_\gamma(z) = \frac{a_1}{\gamma} + \frac{a_2}{\gamma^2} + \ldots + \frac{a_k}{\gamma^k}.$$

Константы p_+ , p_- и k — абсолютные, определяются только величинами M_p и M_m и от числа z не зависят. В дальнейшем для оценок погрешностей вместо машинных характеристик γ, p_-, p_+, k нам будет удобно пользоваться следующими величинами

$$\varepsilon_0 = \gamma^{p_-}\frac{1}{\gamma}, \quad \varepsilon_\infty = \gamma^{p_+}(1 - \frac{1}{\gamma^k}), \quad \varepsilon_1 = \gamma^1\frac{1}{\gamma^k} \quad .$$

Из их определения следует, что ε_0 – минимальное положительное машинное число; ε_∞ – максимальное машинное число; ε_1 – положительное число, минимальное из всех машинных чисел $z_{\text{маш}}$ таких, что

$$(1 + z_{\text{маш}})_{\text{маш}} > 1 .$$

Очевидно, что множество машинных чисел конечно. Следовательно для любого машинного числа z кроме крайних двух определены предыдущий и последующий элементы этого множества. Будем их обозначать z_- и z_+.

Чтобы разместить в машинной памяти произвольное вещественное число z, его необходимо заменить на ближайшее к нему машинное $z_{\text{маш}}$. Оказывается, если $|z| < \varepsilon_0$, то

$$|z_{\text{маш}} - z| < \varepsilon_0;$$

а если $\varepsilon_0 \le |z|$, то

$$|z_{\text{маш}} - z| \le \varepsilon_1|z| .$$

Введем специальные обозначения для бинарных арифметических операций над машинными числами a, b:

$$a \oplus b = (a + b)_{\text{маш}}, \qquad a \ominus b = (a - b)_{\text{маш}},$$
$$a \otimes b = (a \times b)_{\text{маш}}, \qquad a \oslash b = (a/b)_{\text{маш}}.$$

При проведении оценок погрешностей, возникающих во время выполнения этих действий, будем пользоваться следующим утверждением

Лемма 2 *Пусть $\upsilon \in \{+, -, \times, \div\}$ – обозначение для одной из бинарных операций, a, b – машинные числа. Тогда, если $|a \upsilon b| < \varepsilon_0$, то*

$$|(a \upsilon b)_{\text{маш}} - (a \upsilon b)| \le \varepsilon_0 ;$$

если $\varepsilon_0 \le |a \upsilon b|$, то

$$|(a \upsilon b)_{\text{маш}} - (a \upsilon b)| \le \varepsilon_1|a \upsilon b| .$$

То есть погрешность машинной бинарной операции можно смоделировать:

$$(a \upsilon b)_{\text{маш}} = (a \upsilon b)(1 + \alpha) + \beta , \qquad (18)$$

где $|\alpha| \le \varepsilon_1$, $|\beta| \le \varepsilon_0$. На этом представлении основан так называемый метод обратного анализа погрешностей – метод, позволяющий интерпретировать результат численного выполнения каждой арифметической операции, а значит и алгоритма в целом, как результат точных вычислений с относительно возмущёнными исходными данными.

Среди арифметических операций особое место занимает операция вычисления квадратного корня. Результат, который дают специальные подпрограммы, входящие в состав стандартного математического обеспечения, может оказаться недостаточно точным для проведения гарантированных оценок вычислительных погрешностей. Для того, чтобы машинное значение квадратного корня $b = (\sqrt{a})_{\text{маш}}$ было установлено с относительной точностью

$$|b - \sqrt{a}| \le \varepsilon_1|\sqrt{a}|,$$

достаточно выполнения следующего условия

$$(b \otimes b)_- \le a \le (b_+ \otimes b_+)_- .$$

Если это условие не выполнено для результата $\widetilde{b}$ стандартной подрограммы, то его можно уточнить, проведя несколько итераций

$$b_0 = \widetilde{b}; \quad b_{i+1} = \frac{1}{2}\left(b_i + \frac{a}{b_i}\right).$$

Оценивая погрешности, мы будем считать, что квадратный корень вычисляется с относительной точностью.

Экстремальные ситуации. Если точный результат некоторой операции выходит за диапазон $[-\varepsilon_\infty , \varepsilon_\infty]$ представимых в ЭВМ чисел, то

возникает ситуация ПЕРЕПОЛНЕНИЯ и дальнейшие вычисления становятся невозможными. Если точный результат по абсолютной величине меньше ε_0, то из леммы (2) следует, что он не может быть вычислен с относительной погрешностью. Такая ситуация называется ИСЧЕЗНОВЕНИЕМ ПОРЯДКА и может повлечь за собой рост вычислительной погрешности, не поддающийся оценке.

4.2 Вычислительные погрешности в машинных операциях над матрицами и векторами

В этом разделе мы изучим вопросы, связанные с матричными и векторными вычислениями на ЭВМ. Они касаются осуществимости вычислений и их точности.

Уже при вводе вектора размера M, каждая компонента x_i вектора x заменяется на $(x_i)_{\text{маш}}$. Точность ввода вектора оценивается

$$\|x_{\text{маш}} - x\| \le \sqrt{\sum_{i=1}^{M} (\varepsilon_1 |x_i| + \varepsilon_0)^2} \le \varepsilon_1 \|x\| + \sqrt{M}\varepsilon_0 \ .$$

Если предположить, что вектор x не слишком мал : $\|x\| \ge \sqrt{M\varepsilon_0/\varepsilon_1}$, то оценке можно придать вид

$$\|x_{\text{маш}} - x\| \le 2\varepsilon_1 \|x\| \ .$$

Кроме того, для осуществимости ввода необходимо, чтобы все компоненты вектора x удовлетворяли неравенствам $|x_i| \le \varepsilon_\infty$. Ясно, что каждое из неравенств верно, если потребовать более сильного условия $\|x\| \le \varepsilon_\infty$. Как и для чисел обозначим $\oplus$ машинную операцию сложения векторов, $\ominus$ — машинное вычитание, $\otimes$ — машинное умножение вектора на скаляр и умножение матрицы на матрицу, $(*,*)_{\text{маш}}$ — машинное скалярное произведение и приведем оценки погрешностей, возникающих при этих операциях.

$$\begin{aligned} \|(x \oplus y) - (x + y)\| &\le \varepsilon_1 \|x + y\| + \varepsilon_0 \sqrt{M} \\ \|(\alpha \otimes x) - (\alpha x)\| &\le \varepsilon_1 |\alpha| \, \|x\| + \varepsilon_0 \sqrt{M}. \end{aligned} \qquad (19)$$

Заметим, что для того, чтобы выполнение сложения и умножения на компьютере было возможным, необходимо, чтобы каждая компонента результата была из промежутка $(-\varepsilon_\infty, \varepsilon_\infty)$. Мы будем требовать обычно даже больше: $\|x+y\| < \varepsilon_\infty$, $\|\alpha x\| < \varepsilon_\infty$. Если норма результата операции

не слишком мала

$$\frac{\sqrt{M}\varepsilon_0}{\varepsilon_1} \le \|x + y\|, \qquad \frac{\sqrt{M}\varepsilon_0}{\varepsilon_1} \le |\alpha| \, \|x\| \, ,$$

то оценки (19) упрощаются:

$$\begin{aligned} \|(x \oplus y) - (x + y)\| &\le 2\varepsilon_1 \|x + y\|, \\ \|(\alpha \otimes x) - (\alpha x)\| &\le 2\varepsilon_1 |\alpha| \, \|x\|. \end{aligned}$$

Если наоборот

$$\frac{\sqrt{M}\varepsilon_0}{\varepsilon_1} > \|x + y\|, \qquad \frac{\sqrt{M}\varepsilon_0}{\varepsilon_1} > |\alpha| \, \|x\|,$$

то оценки (19) можно огрубить иначе:

$$\begin{aligned} \|(x \oplus y) - (x + y)\| &\le 2\varepsilon_0 \sqrt{M}, \\ \|(\alpha \otimes x) - (\alpha x)\| &\le 2\varepsilon_0 \sqrt{M}. \end{aligned}$$

Замечание. Особое место занимает умножение на степень γ. Пусть $\alpha = \gamma^p$, причем $p \ge 0$, тогда, если не возникает ПЕРЕПОЛНЕНИЯ, машинное умножение на α производится без ошибок. Если $p < 0$, то возникающая погрешность не превосходит $\varepsilon_0 \sqrt{M}$.

Оценка погрешности для скалярного произведения имеет вид

$$\|(x, y)_{\text{маш}} - (x, y)\| \le \frac{\varepsilon_1 M}{1 - \varepsilon_1 M/2} \|x\| \, \|y\| + \frac{\varepsilon_0 M}{1 - \varepsilon_1 M/2}.$$

Если предположить,что машинные векторы x и y удовлетворяют оценкам

$$\begin{aligned} \sqrt{2M\varepsilon_0 \ / \ \varepsilon_1} \le \|x\| \le \sqrt{\varepsilon_\infty \ / \ 2}, \\ \sqrt{2M\varepsilon_0 \ / \ \varepsilon_1} \le \|y\| \le \sqrt{\varepsilon_\infty \ / \ 2}, \end{aligned}$$

а размерность M этих векторов не превосходит $\sqrt{1/\varepsilon_1}$, то

$$|(x, y)_{\text{маш}} - (x, y)| < (M + 1)\varepsilon_1 \|x\| \, \|y\|.$$

Все введённые нами дополнительные ограничения на векторы и их размерности при вычислениях на современных ЭВМ как правило не являются слишком обременительными. Поэтому в дальнейшем мы используем

только упрощённые неравенства, дающие нам оценки относительных погрешностей.

В наших расчетах мы, конечно, используем умножение матрицы на вектор а также значение нормы вектора. Пусть A - матрица размера $N \times M$, $N_0 = \min\{N, M\}$ и $\mathcal{F}(A) = \sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij}^2$ – фробениусова норма матрицы A. Наложим на матрицу A и x условия

$$\sqrt{2M\varepsilon_0 \,/\, \varepsilon_1} \le \|x\| \le \sqrt{\varepsilon_\infty \,/\, 2},$$
$$\sqrt{2M\varepsilon_0 \,/\, \varepsilon_1} \le \|\alpha_i\| \le \sqrt{\varepsilon_\infty \,/\, 2},$$

где α_i – i-я строка матрицы A. Если использовать неравенство

$$\|A\| \le \mathcal{F}(A) \le \sqrt{N_0}\|A\|$$

то вычислительные погрешности умножения матрицы на вектор оцениваются следующим образом:

$$\|A \otimes x - Ax\| \le (M+1)\sqrt{N_0}\varepsilon_1 \|A\| \, \|x\| .$$

Если вычисление квадратного корня производится с относительной точностью, то для погрешности вычисления нормы вектора нетрудно вывести оценку

$$\Big| \|x\|_{\text{маш}} - \|x\| \Big| \le \varepsilon_1 \frac{M+4}{2} \|x\| = \delta_1 \|x\| . \tag{20}$$

Эти простые неравенства будут использованы для оценки точности более сложных машинных вычислений.

4.3 Вычислительные погрешности при построении оператора отражения

В разделе 3.2 при приведении матрицы к двухдиагональному виду большое значение имели ортогональные преобразования отражения. Поэтому оценки точности вычисления операторов отражения и их действия на векторы и матрицы играют одну из ключевых ролей в оценках точности численного решения линейных систем по алгоритму из 3.1.

Итак, пусть дан вектор x размера N и пусть перед нами стоит задача построить оператор отражения P такой, что $Px = y$, $\|x\| = \|y\|$, $y_k = x_k$ при $k = 1, i-1$ и $y_k = 0$ при $i < k \le N$. Очевидно, что компонента y_i должна быть вычислена по формуле

$$y_i = \pm \sqrt{\sum_{j=i}^{N} |x_j|^2}. \tag{21}$$

Тогда согласно (20)

$$|(y_i)_{\text{маш}} - y_i| \le \varepsilon_1 \frac{N+4}{2} |y_i|. \tag{22}$$

То есть

$$\|y_{\text{маш}} - y\| \le \delta_1 \|y\|. \tag{23}$$

Оценим погрешность, с которой определяется вектор $p = \sqrt{2}(y-x)/\|y-x\|$ — специальным образом нормированная нормаль к гиперплоскости, относительно которой осуществляется отражение. Вычисляя на компьютере мы найдём некоторый машинный вектор $p_{\text{маш}}$:

$$p_{\text{маш}} = (\sqrt{2})_{\text{маш}} \otimes (y_{\text{маш}} \ominus x) \oslash \|y_{\text{маш}} \ominus x\|_{\text{маш}}.$$

Разобьём задачу на две подзадачи: во-первых, оценим точность вычисления вектора $r = y - x$, а во-вторых, - точность его нормировки $p = \sqrt{2}r/\|r\|$. Заметим, что компоненты $(r_j)_{\text{маш}}$, $j \ne i$, могут быть вычислены точно. Ошибка возникает только в $(r_i)_{\text{маш}}$. Чтобы вектор r мог быть найден с относительной точностью, необходимо в формуле (21) правильно распорядиться знаком — выбрать $+$ или $-$. Если выбрать знак $(y_i)_{\text{маш}}$ противоположным знаку x_i, то есть $\operatorname{sgn}\Big((y_i)_{\text{маш}}\Big) = -\operatorname{sgn}(x_i)$, то вычитание $(y_i)_{\text{маш}} - x_i$ будет производиться с относительной точностью всегда, кроме случая $(y_i)_{\text{маш}} = 0$. Действительно, при условии $(y_i)_{\text{маш}} \ne 0$ будем иметь $|(y_i)_{\text{маш}} - x_i| > \varepsilon_0$, следовательно по лемме 2 погрешность результата будет относительной. Итак,

$$|(r_i)_{\text{маш}} - r_i| = \Big| \Big((y_i)_{\text{маш}} \ominus x_i\Big) - \Big(y_i - x_i\Big) \Big| =$$
$$\Big| \Big((y_i)_{\text{маш}} \ominus x_i\Big) \pm \Big((y_i)_{\text{маш}} - x_i\Big) - \Big(y_i - x_i\Big) \Big| \le$$
$$\Big| \Big((y_i)_{\text{маш}} \ominus x_i\Big) - \Big((y_i)_{\text{маш}} - x_i\Big) \Big| + |(y_i)_{\text{маш}} - y_i|.$$

Мы уже обсудили выше, что первое слагаемое при правильном выборе знака $(y_i)_{\text{маш}}$ можно оценить

$$\Big| \Big((y_i)_{\text{маш}} \ominus x_i\Big) - \Big((y_i)_{\text{маш}} - x_i\Big) \Big| \le \varepsilon_1 |(y_i)_{\text{маш}} - x_i|.$$

То есть

$$|(r_i)_{\text{маш}} - r_i| \le \varepsilon_1 |(y_i)_{\text{маш}} - x_i| + |(y_i)_{\text{маш}} - y_i|.$$

В то же время

$$|(y_i)_{\text{маш}} - x_i| \le |(y_i)_{\text{маш}} - y_i| + |y_i - x_i|.$$

Значит

$$|(r_i)_{\text{маш}} - r_i| \le \varepsilon_1|y_i - x_i| + (1+\varepsilon_1)|(y_i)_{\text{маш}} - y_i|.$$

Ко второму слагаемому применим оценку (22), получим

$$|(r_i)_{\text{маш}} - r_i| \le (1+\varepsilon_1)\delta_1|y_i| + \varepsilon_1|y_i - x_i|.$$

Так как мы специальным образом выбирали знак y_i, имеет место неравенство $|y_i| \le |y_i - x_i|$, поэтому в итоге

$$|(r_i)_{\text{маш}} - r_i| \le \Big((1+\varepsilon_1)\delta_1 + \varepsilon_1\Big)|y_i - x_i| = \delta_2|r_i|.$$

Заметим, что оценка для разности норм векторов r и $r_{\text{маш}}$ имеет вид:

$$\big|\|r_{\text{маш}}\| - \|r\|\big| \le \|r_{\text{маш}} - r\| = |(r_i)_{\text{маш}} - r_i| \le \delta_2|r_i| \le \delta_2\|r\|. \qquad (24)$$

Приступим к оценке точности нормировки. Но сначала отметим несколько очевидных неравенств. Например, нетрудно видеть, что точность вычисления $\sqrt{2}$ такова:

$$|(\sqrt{2})_{\text{маш}} - \sqrt{2}| \le \varepsilon_1\sqrt{2}.$$

Используя неравенство треугольника и оценки (20) и (24), получим

$$\Big|\|r_{\text{маш}}\|_{\text{маш}} - \|r\|\Big| \le \Big|\|r_{\text{маш}}\|_{\text{маш}} - \|r_{\text{маш}}\|\Big| + \Big|\|r_{\text{маш}}\| - \|r\|\Big| \le$$
$$\delta_1\|r_{\text{маш}}\| + \delta_2\|r\| \le (\delta_1 + \delta_2 + \delta_1\delta_2)\|r\| = \delta_3\|r\|.$$

Аналогично получается неравенство

$$\Big|1/\|r_{\text{маш}}\|_{\text{маш}} - 1/\|r\|\Big| \le \frac{\delta_1 + \delta_2 + \delta_1\delta_2}{(1-\delta_1)(1-\delta_2)}\frac{1}{\|r\|} \le \frac{\delta_3}{1-\delta_3}\frac{1}{\|r\|}.$$

Заметим еще, что при условии

$$\|x\| \le \frac{1}{\varepsilon_0\sqrt{2}}\frac{1+\varepsilon_1}{1-\delta_3}, \qquad (25)$$

которого легко можно добиться нормировкой, с относительной точностью производится деление

$$\left|\sqrt{2} \oslash \|r_{\text{маш}}\|_{\text{маш}} - \frac{(\sqrt{2})_{\text{маш}}}{\|r_{\text{маш}}\|_{\text{маш}}}\right| \le \frac{\varepsilon_1(1+\varepsilon_1)}{1-\delta_3}\frac{(\sqrt{2})_{\text{маш}}}{\|r_{\text{маш}}\|_{\text{маш}}},$$

т.к. модуль результата при этом достаточно большой:

$$\left|\frac{(\sqrt{2})_{\text{маш}}}{\|r_{\text{маш}}\|_{\text{маш}}}\right| > \varepsilon_0.$$

С условием (25) точность вычисления скалярного нормирующего множителя оценивается следующим образом:

$$\Big|\Big(\sqrt{2}/\|r_{\text{маш}}\|\Big)_{\text{маш}} - \sqrt{2}/\|r\|\Big| \le$$
$$\Big|\Big(\sqrt{2}/\|r_{\text{маш}}\|\Big)_{\text{маш}} - (\sqrt{2})_{\text{маш}}/\|r_{\text{маш}}\|_{\text{маш}}\Big| + \Big|(\sqrt{2})_{\text{маш}} - \sqrt{2}\Big| \Big/ \|r_{\text{маш}}\|_{\text{маш}} +$$
$$\sqrt{2}\Big|1/\|r_{\text{маш}}\|_{\text{маш}} - 1/\|r\|\Big| \le \Big(\tfrac{\varepsilon_1(1+\varepsilon_1)}{1-\delta_3} + \tfrac{\varepsilon_1}{1-\delta_3} + \tfrac{\delta_3}{1-\delta_3}\Big)\tfrac{\sqrt{2}}{\|r\|} \le$$
$$\tfrac{2\varepsilon_1+\varepsilon_1^2+\delta_3}{1-\delta_3}\tfrac{\sqrt{2}}{\|r\|} = \Big(\tfrac{(1+\delta_2)^2}{1-\delta_3} - 1\Big)\tfrac{\sqrt{2}}{\|r\|} = \delta_4\tfrac{\sqrt{2}}{\|r\|}.$$

Теперь у нас в распоряжении есть всё необходимое, чтобы оценить точность вычисления нормированного вектора p:

$$\|p_{\text{маш}} - p\| \le \left\|\left(\tfrac{\sqrt{2}}{\|r_{\text{маш}}\|}\right)_{\text{маш}} \otimes r_{\text{маш}} - \tfrac{\sqrt{2}}{\|r\|}r\right\| \le$$
$$\left\|\left(\tfrac{\sqrt{2}}{\|r_{\text{маш}}\|}\right)_{\text{маш}} \otimes r_{\text{маш}} - \left(\tfrac{\sqrt{2}}{\|r_{\text{маш}}\|}\right)_{\text{маш}} r_{\text{маш}}\right\| + \left|\left(\tfrac{\sqrt{2}}{\|r_{\text{маш}}\|}\right)_{\text{маш}} -\right.$$
$$\left.\tfrac{\sqrt{2}}{\|r\|}\right|\|r_{\text{маш}}\| + \tfrac{\sqrt{2}}{\|r\|}\|r_{\text{маш}} - r\| + \delta_2\sqrt{2} \le \qquad (26)$$
$$\varepsilon_1\left\|\left(\tfrac{\sqrt{2}}{\|r_{\text{маш}}\|}\right)_{\text{маш}}\right\|\|r_{\text{маш}}\| + \varepsilon_0\sqrt{N-i+1} + \delta_4\|r_{\text{маш}}\|\tfrac{\sqrt{2}}{\|r\|} \le$$
$$\sqrt{2}\Big(\varepsilon_1(1+\delta_2)(1+\delta_4) + \delta_4(1+\delta_2) + \delta_2\Big) + \varepsilon_0\sqrt{N-i+1} =$$
$$\delta_5\sqrt{2} + \varepsilon_0\sqrt{N-i+1}.$$

Таким образом норма вектора $p_{\text{маш}}$, вообще говоря, отлична от $\sqrt{2}$. Но как и любой ненулевой вектор размера N, вектор $p_{\text{маш}}$ определяет некоторое преобразование отражения $P_{\text{маш}}$ по формуле

$$b = P_{\text{маш}}a = a - \frac{2(p_{\text{маш}}, a)}{\|p_{\text{маш}}\|^2}p_{\text{маш}},$$

здесь a и b – векторы размерности N, и подразумевается, что все вычисления производятся точно. Наряду с преобразованием $P_{\text{маш}}$ рассмотрим преобразование $\widetilde{P}_{\text{маш}}$, действующее по формуле

$$\widetilde{b} = \widetilde{P}_{\text{маш}}a = a - (p_{\text{маш}}, a)p_{\text{маш}}. \qquad (27)$$

Из-за того, что вектор $p_{\text{маш}}$ ненормирован, преобразование $\widetilde{P}_{\text{маш}}$ не будет ортогональным. И тем не менее результат его действия близок к результату $P_{\text{маш}}$. Оценим

$$\|\widetilde{b}-b\| = \|(p_{\text{маш}},a)p_{\text{маш}} - \tfrac{2(p_{\text{маш}},a)}{(p_{\text{маш}},p_{\text{маш}})}p_{\text{маш}}\| \le \tfrac{|(p_{\text{маш}},a)|}{\|p_{\text{маш}}\|}\left|2-\|p_{\text{маш}}\|^2\right| \le \|a\|\left|2-\|p_{\text{маш}}\|^2\right|.$$

Учитывая неравенство (26), получим

$$\left|2-\|p_{\text{маш}}\|^2\right| \le (\delta_5\sqrt{2}+\varepsilon_0\sqrt{N})\left((1+\delta_5)\sqrt{2}+\varepsilon_0\sqrt{N}\right) = \delta_6.$$

Иными словами

$$\|p_{\text{маш}}\|^2 \le 2+\delta_6. \tag{28}$$

Следовательно

$$\left|\|\widetilde{b}\|-\|b\|\right| \le \|\widetilde{b}-b\| \le \delta_6\|a\|\,, \tag{29}$$

а значит

$$\|\widetilde{b}\| \le \|b\|+\delta_6\|a\| = (1+\delta_6)\|a\|. \tag{30}$$

По формуле (27) мы будем вычислять результат действия оператора $\widetilde{P}_{\text{маш}}$ на компьютере, и поэтому вместо $\widetilde{b}$ нами будет получено некоторое $\widetilde{b}_{\text{маш}}$. Необходимо оценить погрешность :

$$\|\widetilde{b}_{\text{маш}}-\widetilde{b}\| = \|(a\ominus(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}})-(a-(p_{\text{маш}},a)p_{\text{маш}})\|.$$

Для этого нам понадобится ряд вспомогательных оценок. Как уже упоминалось в пункте (4.2), точность вычисления скалярных произведений может быть оценена следующим образом:

$$|(p_{\text{маш}},a)_{\text{маш}}-(p_{\text{маш}},a)| \le \varepsilon_1(N+1)\|p_{\text{маш}}\|\,\|a\|\,, \tag{31}$$

при условии, если $\sqrt{2N\varepsilon_0/\varepsilon_1} \le \|p_{\text{маш}}\|$, $\sqrt{2N\varepsilon_0/\varepsilon_1} \le \|a\|$. То есть

$$|(p_{\text{маш}},a)_{\text{маш}}| \le (1+\varepsilon_1(N+1))\|p_{\text{маш}}\|\,\|a\|.$$

Точность умножения на вектор

$$\begin{aligned}
&\|(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}} - (p_{\text{маш}},a)p_{\text{маш}}\| \le\\
&\|(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}} - (p_{\text{маш}},a)_{\text{маш}}p_{\text{маш}}\|+\\
&\|p_{\text{маш}}\|\,|(p_{\text{маш}},a)_{\text{маш}}-(p_{\text{маш}},a)|\varepsilon_1|(p_{\text{маш}},a)_{\text{маш}}|\|p_{\text{маш}}\|+\varepsilon_0\sqrt{N}+\\
&\|p_{\text{маш}}\|\,|(p_{\text{маш}},a)_{\text{маш}}-(p_{\text{маш}},a)| \le\\
&\varepsilon_1(N+2+\varepsilon_1(N+1))\|p_{\text{маш}}\|^2\|a\|+\varepsilon_0\sqrt{N}.
\end{aligned} \tag{32}$$

Теперь очевидно, что

$$\begin{aligned}
&\|\widetilde{b}_{\text{маш}}-\widetilde{b}\| = \|(a\ominus(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}})-(a-(p_{\text{маш}},a)p_{\text{маш}})\| \le\\
&\|(a\ominus(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}})-(a-(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}})\|+\\
&\|(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}})-(p_{\text{маш}},a)_{\text{маш}}p_{\text{маш}})\|+\\
&\varepsilon_1\|a-(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}}\|+\\
&\varepsilon_0\sqrt{N}+\|(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}})-(p_{\text{маш}},a)_{\text{маш}}p_{\text{маш}})\| \le\\
&\varepsilon_1\|a-(p_{\text{маш}},a)_{\text{маш}}p_{\text{маш}}\|+\\
&\varepsilon_0\sqrt{N}+(\varepsilon_1+1)\|(p_{\text{маш}},a)_{\text{маш}}\otimes p_{\text{маш}})-(p_{\text{маш}},a)_{\text{маш}}p_{\text{маш}})\|.
\end{aligned}$$

Применив неравенства (30), (32) и (28), получим

$$\begin{aligned}
&\|\widetilde{b}_{\text{маш}}-\widetilde{b}\| \le \varepsilon_1\|\widetilde{b}_{\text{маш}}\|+\varepsilon_0\sqrt{N}+\varepsilon_1(N+2+\varepsilon_1(N+1))\|p_{\text{маш}}\|^2\|a\|+\\
&\varepsilon_0\sqrt{N} \le \varepsilon_1(1+\delta_6)\|a\|+\varepsilon_1(N+2+\varepsilon_1(N+1))(2+\delta_6)\|a\|+\\
&2\varepsilon_0\sqrt{N} \le \delta_7\|a\|+2\varepsilon_0\sqrt{N}.
\end{aligned} \tag{33}$$

Объединие неравенств (29) и (33) дает оценку точности

$$\begin{aligned}
&\|(\widetilde{P}_{\text{маш}}a)_{\text{маш}}-P_{\text{маш}}a\| = \|\widetilde{b}_{\text{маш}}-b\| \le (\delta_6+\delta_7)\|a\|+2\varepsilon_0\sqrt{N} =\\
&\Delta_p(N)\|a\|+o_p(N).
\end{aligned} \tag{34}$$

Выпишем правило, по которому вычисляются коэффициенты погрешности $\Delta_p(N)$ и $o_p(N)$:

$$\begin{aligned}
&\delta_1 = \varepsilon_1(N+1)/2\,, \qquad \delta_2 = (1+\varepsilon_1)\delta_1+\varepsilon_1\,,\\
&\delta_3 = \delta_1+\delta_2+\delta_1\delta_2\,, \qquad \delta_4 = (1+\delta_2)^2/(1-\delta_3)-1\,,\\
&\delta_5 = \varepsilon_1(1+\delta_2)(1+\delta_4)+\delta_4(1+\delta_2)+\delta_2\,,\\
&\delta_6 = (\delta_5\sqrt{2}+\varepsilon_0\sqrt{N})\left((1+\delta_5)\sqrt{2}+\varepsilon_0\sqrt{N}\right),\\
&\delta_7 = \varepsilon_1(1+\delta_6)+\varepsilon_1(N+2+\varepsilon_1(N+1))(2+\delta_6)\,,\\
&\Delta_p(N) = \delta_5+\delta_7\,,\\
&o_p(N) = 2\varepsilon_0\sqrt{N}.
\end{aligned} \tag{35}$$

В заключение оценим точность применения преобразования отражения к матрице A размера $N\times M$. В соответствии с правилами перемножения матриц применение преобразования слева к матрице равносильно его применению к каждому столбцу матрицы, в то время как применение его справа равносильно применению к каждой строке. Будем считать, что преобразуется максимально возможное число компонент (элементов столбцов матрицы). Тогда применяя преобразование отражения слева (преобразование действует в N–мерном пространстве), имеем оценку:

$$\begin{aligned}
&\|(\widetilde{P}_{\text{маш}}A)_{\text{маш}}-P_{\text{маш}}A\| \le \mathcal{F}\left((\widetilde{P}_{\text{маш}}A)_{\text{маш}}-P_{\text{маш}}A\right) \le\\
&\Delta_p(N)\mathcal{F}(A)+\sqrt{M}o_p(N) \le \Delta_p(N)\sqrt{N_0}\|A\|+\sqrt{M}o_p(N).
\end{aligned}$$

Для преобразований, действующих справа, имеет место несколько иная оценка:

$$\|(A\widetilde{P}_{\text{маш}})_{\text{маш}} - AP_{\text{маш}}\| \le \mathcal{F}\Big((A\widetilde{P}_{\text{маш}})_{\text{маш}} - AP_{\text{маш}}\Big) \le$$
$$\Delta_p(M)\mathcal{F}(A) + \sqrt{N}o_p(M) \le \Delta_p(M)\sqrt{N_0}\|A\| + \sqrt{N}o_p(M).$$

В этих формулах $N_0 = \min\{N, M\}$.

4.4 Вычислительные погрешности при приведении матрицы к двухдиагональному виду

Приведем оценки вычислительных погрешностей при упрощении вида матрицы. Алгоритм приведения $N \times M-$ матрицы A $(N > M)$ к двухдиагональной форме сводится к реализации формулы

$$K = P^{(M)} \dots P^{(2)} P^{(1)} A Q^{(1)} Q^{(2)} \dots Q^{(M-2)},$$

где

$$K = \begin{pmatrix} D \\ \dots \\ 0 \end{pmatrix},$$

$$D = \begin{pmatrix} d_1 & b_1 & & & \\ & d_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & d_{M-1} & b_{M-1} \\ & & & & d_M \end{pmatrix}.$$

$P^{(j)}$ и $Q^{(j)}$ – ортогональные преобразования отражений, построенные по соответствующим нормированным векторам $p^{(j)}$, $q^{(j)}$. Как было показано в предыдущем разделе, машинная реализация не позволяет нам вычислить $P^{(j)}$, $Q^{(j)}$ и D точно. Вместо них мы находим некие $\widetilde{P}^{(j)}_{\text{маш}}$, $\widetilde{Q}^{(j)}_{\text{маш}}$ и $D_{\text{маш}}$. Причем существуют примеры, когда $\|D_{\text{маш}} - D\| \sim \|A\|$ даже при высокой точности машинной арифметики. Пусть

$$P_{\text{маш}} = P^{(M)}_{\text{маш}} \dots P^{(2)}_{\text{маш}} P^{(1)}_{\text{маш}},$$
$$Q_{\text{маш}} = Q^{(1)}_{\text{маш}} Q^{(2)}_{\text{маш}} \dots Q^{(M-2)}_{\text{маш}}$$

ортогональные преобразования, которые определяются векторами $p^{(j)}_{\text{маш}}$, $q^{(j)}_{\text{маш}}$. Мы будем оценивать

$$\|K_{\text{маш}} - P_{\text{маш}} A Q_{\text{маш}}\|,$$

что вполне достаточно для наших целей.

Погрешности процесса двухдиагонализации на компьютере смоделируем следующим образом:

$$A^{(0)} = \rho A + H^{(0)},$$
$$A^{(1)} = P^{(1)}_{\text{маш}} A^{(0)} + H^{(1)},$$
$$A^{(2)} = A^{(1)} Q^{(1)}_{\text{маш}} + H^{(2)},$$
$$\dots\dots\dots\dots\dots\dots$$
$$A^{(2M-2)} = P^{(M)}_{\text{маш}} A^{(2M-3)} + H^{(2M-2)},$$
$$K_{\text{маш}} = (1/\rho) A^{(2M-2)} + H^{(2M-1)},$$

где $A^{(j)}$ – промежуточные результаты, $H^{(j)}$ – матрицы погрешностей $(j = 0, 1, 2, \dots, 2M-1)$, ρ – нормирующий множитель. Видно, что процесс двухдиагонализации включает в себя предварительную нормировку матрицы и разнормировку результата. Матрица $A^{(1)}$ имеет следующую структуру:

$$A^{(1)} = \begin{pmatrix} a^{(1)}_{11} & a^{(1)}_{12} & \dots & a^{(1)}_{1M} \\ 0 & a^{(1)}_{22} & \dots & a^{(1)}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a^{(1)}_{N2} & \dots & a^{(1)}_{NM} \end{pmatrix},$$

где $a^{(1)}_{11} = (d_1)_{\text{маш}}$. Используя оценки для оператора отражения, оценим фробениусову норму H_1, предварительно оценив норму каждого ее столбца $H^{(1)}_1, H^{(1)}_2, \dots, H^{(1)}_M$ в отдельности. Для первого столбца $H^{(1)}_1$ применимо неравенство (23) стр. 29:

$$\|H^{(1)}_1\| \le \delta_1 \|A^{(0)}_1\|.$$

Здесь $A^{(0)}_1$ – первый столбец матрицы $A^{(0)}$. Так как $\delta_1 < \Delta_p(N)$, то

$$\|H^{(1)}_1\| \le \Delta_p(N) \|A^{(0)}_1\|. \tag{36}$$

Для оценки нормы $H^{(1)}_j$ используем неравенство (34) стр. 33. Получим

$$\|H^{(1)}_j\| \le \Delta_p(N) \|A^{(0)}_j\| + o_p(N). \tag{37}$$

Просуммировав неравенства (36) и (37), легко получить следующую оценку

$$\mathcal{F}(H^{(1)}) \le \Delta_p(N) \mathcal{F}(A^{(0)}) + \sqrt{M-1}\, o_p(N). \tag{38}$$

В матрице $H^{(2)}$ элементы первого столбца будут нулями, поэтому в оценке должна участвовать фробениусова норма матрицы, составленной из последних $M-1$ столбцов $A^{(1)}$:

$$\mathcal{F}(H^{(2)}) \le \Delta_p(M-1)\mathcal{F}(A^{(1)}) + \sqrt{N-1}\, o_p(M-1)\,. \qquad (39)$$

Продолжая этот процесс, можно записать следующие оценки:

$$\begin{array}{ll}
\mathcal{F}(H^{(2i)}) \le \Delta_p(M-i)\mathcal{F}(A^{(2i-1)}) + \sqrt{N-i}\, o_p(M-i)\,, & 1 \le i \le M-2\,; \\
\mathcal{F}(H^{(2j+1)}) \le \Delta_p(N-j)\mathcal{F}(A^{(2j)}) + \sqrt{M-j}\, o_p(N-j)\,, & 0 \le j \le M-1\,; \\
\mathcal{F}(H^{(2M-2)}) \le \Delta_p(N-M)\mathcal{F}(A^{(2M-3)})\,. &
\end{array} \qquad (40)$$

Для упрощения этих выражений используются следующие очевидные факты: функции $\Delta_p(n)\,, o_p(n)$ монотонно зависят от своих аргументов, поэтому неравенства (38)–(40) только усилятся при замене $\Delta_p(n)\,, o_p(n)$ на максимально возможные значения $\Delta_p(N)\,, o_p(N)$:

$$\begin{array}{ll}
\mathcal{F}(H^{(2i)}) \le \Delta_p(N)\mathcal{F}(A^{(2i-1)}) + \sqrt{N}o_p(N)\,, & 1 \le i \le M-2\,; \\
\mathcal{F}(H^{(2j+1)}) \le \Delta_p(N)\mathcal{F}(A^{(2j)}) + \sqrt{N}o_p(N)\,, & 0 \le j \le M-1\,; \\
\mathcal{F}(H^{(2M-2)}) \le \Delta_p(N)\mathcal{F}(A^{(2M-3)})\,. &
\end{array} \qquad (41)$$

Верна следующая лемма

Лемма 3 *Если*

$$\frac{2\sqrt{N}(M-1)o_p(N)}{\Delta_p(N)} \le \mathcal{F}(A^{(0)})\,, \qquad (M-1)^2\Delta_p(N) < 1\,,$$

то

$$\mathcal{F}(A^{(2M-2)} - P_{\text{маш}}A^{(0)}Q_{\text{маш}}) \le (2M-1)\Delta_p(N)\mathcal{F}(A_0).$$

То есть

$$\|K_{\text{маш}} - P_{\text{маш}}AQ_{\text{маш}}\| \le \sqrt{M}(2M-1)\Delta_p(N)\|A_0\|.$$

Замечание 1. Первое неравенство в условии леммы по сути означает, что норма матрицы $A^{(0)}$ для проведения оценок должна быть не слишком мала. Для того, чтобы это условие выполнялось и введена начальная нормировка.

Замечание 2. Аналогичным образом можно моделировать погрешности применения последовательности ортогональных преобразований

$$P_{\text{маш}} = P^{(M)}_{\text{маш}} \dots P^{(2)}_{\text{маш}} P^{(1)}_{\text{маш}}$$

к вектору f

$$g = P_{\text{маш}}f,$$

считая вектор матрицей, состоящей из одного столбца:

$$\begin{array}{l}
f^{(0)} = \rho f + h^{(0)}\,, \\
f^{(1)} = P^{(1)}_{\text{маш}} f^{(0)} + h^{(1)}\,, \\
\dots\dots\dots\dots\dots\dots \\
f^{(M)} = P^{(M)}_{\text{маш}} f^{(M-1)} + h^{(M)}\,, \\
g_{\text{маш}} = (1/\rho) f^{(2M-2)} + h^{(M+1)}\,.
\end{array}$$

Причём

$$\|h^{(j)}\| \le \Delta_p \|f^{(j-1)}\| + o_p.$$

Тогда по лемме 3 имеем оценку

$$\|f^{(M)} - P^{(M)}_{\text{маш}} f^{(0)}\| \le M\Delta_p\|f\|$$

при условии, что норма $f^{(0)}$ не слишком мала:

$$\frac{2\sqrt{N}(M-1)o_p(N)}{\Delta_p(N)} \le \|f^{(0)}\|.$$

Погрешности нормировок сказываются следующим образом: так как первое умножение делается с целью увеличить норму исходной матрицы, то разумно в качестве ρ выбрать γ^p, где $p \ge 0$ – целое число. Тогда, как мы уже указывали в замечании на странице 27, $H_0 = 0$. На последнем шаге, когда происходит разнормировка, умножение производится соответственно на $\rho = \gamma^{-p}$ с погрешностью $\mathcal{F}(H_{2M-1}) \le \sqrt{NM}\varepsilon_0$.

Итоговые оценки принимают вид

$$\begin{array}{l}
\|K_{\text{маш}} - P_{\text{маш}}AQ_{\text{маш}}\| = \\
\|\frac{1}{\rho} \otimes A^{(2M-2)} - \frac{1}{\rho}A^{(2M-2)}\| + \frac{1}{\rho}\|A^{(2M-2)} - P_{\text{маш}}A^{(0)}Q_{\text{маш}}\| \le \\
\sqrt{NM}\varepsilon_0 + \sqrt{M}(2M-1)\Delta_p(N)\|A\|.
\end{array} \qquad (42)$$

$$\|g_{\text{маш}} - P_{\text{маш}}f\| \le \sqrt{N}\varepsilon_0 + M\Delta_p(N)\|f\|, \qquad (43)$$

где коэффициент $\Delta_p(N)$ определён на стр. 33.

4.5 Вычислительные погрешности при решении линейной системы с двухдиагональной матрицей

В этом разделе мы получим оценку погрешностей, возникающих при реализации алгоритма решения линейной системы уравнений

$$Dx = f$$

с двухдиагональной $M \times M$-матрицей коэффициентов D:

$$D = \begin{pmatrix} d_1 & b_1 & & & \\ & d_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & d_{M-1} & b_{M-1} \\ & & & & d_M \end{pmatrix}$$

и обсудим вопрос об осуществимости всех необходимых машинных операций.

Решение системы производится исключением неизвестных в обратном порядке, начиная с x_M:

$$\begin{aligned} &x_M = f_M/d_M\ , \\ &x_{k-1} = (f_{k-1} - b_k x_k)/d_{k-1} \quad (k = M, M-1, \ldots, 2)\ . \end{aligned}$$

Машинная реализация этих формул имеет вид:

$$\begin{aligned} &(x_M)_{\text{маш}} = f_M \oslash d_M\ , \\ &(x_{k-1})_{\text{маш}} = (f_{k-1} \ominus b_k \otimes (x_k)_{\text{маш}}) \oslash d_{k-1} \quad (k = M, M-1, \ldots, 2)\ . \end{aligned} \tag{44}$$

Используя равенство (18) стр. 25, их можно переписать, заменив машинные операции на обычные:

$$\begin{aligned} &(x_M)_{\text{маш}} = f_M(1+\varphi_M)/d_M + \xi_M\ , \\ &(x_{k-1})_{\text{маш}} = \frac{(1+\delta_{k-1})\Big(\chi_{k-1} + (1+\varphi_{k-1})\Big(f_{k-1} - \psi_{k-1} - b_k(1+\beta_k)(x_k)_{\text{маш}}\Big)\Big)}{d_{k-1}} + \xi_{k-1} \\ &\text{при} (M \geq k \geq 2)\ . \end{aligned}$$

Причем имеют место оценки

$$\begin{aligned} &|\varphi_j| \leq \varepsilon_1,\ \ |\beta_j| \leq \varepsilon_1,\ \ |\delta_j| \leq \varepsilon_1, \\ &|\xi_j| \leq \varepsilon_0,\ \ |\chi_j| \leq \varepsilon_0,\ \ |\psi_j| \leq \varepsilon_0. \end{aligned} \tag{45}$$

Если ввести обозначения:

$$\begin{aligned} &\overline{x}_k = (x_k)_{\text{маш}} \quad \text{при} \quad 1 \leq k \leq M\ ; \\ &\overline{d}_M = d_M\ , \\ &\overline{d}_k = d_k/(1+\delta_k)\ , \qquad \overline{b}_{k+1} = b_{k+1}(1+\beta_{k+1})(1+\varphi_k) \,\text{при} \quad 1 \leq k \leq M-1\ ; \\ &\overline{f}_M = f_M(1+\varphi_M) + d_M\xi_M\ , \\ &\overline{f}_k = f_k(1+\varphi_k) + \chi_k - \psi_{k-1}(1+\varphi_{k-1}) + d_{k-1}\xi_{k-1}/(1+\delta_{k-1}) \\ &\text{при} \quad 1 \leq k \leq M-1\ , \end{aligned}$$

то система (44) запишется в виде

$$\begin{aligned} &\overline{x}_M = \overline{f}_M/\overline{d}_M\ , \\ &\overline{x}_{k-1} = (\overline{f}_{k-1} - \overline{b}_k\overline{x}_k)/\overline{d}_{k-1}\ . \end{aligned}$$

Из неравенств (45) следует, что коэффициенты новой системы не слишком отличаются от коэффициентов исходной:

$$\frac{|\overline{d}_k - d_k|}{|d_k|} \leq \frac{2\varepsilon_1}{1-2\varepsilon_1}, \qquad \frac{|\overline{b}_k - b_k|}{|b_k|} \leq \frac{2\varepsilon_1}{1-2\varepsilon_1}. \tag{46}$$

То есть $\mathcal{F}(\overline{D} - D) \leq \frac{2\varepsilon_1}{1-2\varepsilon_1}\mathcal{F}(D)$, или

$$\|\overline{D} - D\| \leq \sqrt{M}\frac{2\varepsilon_1}{1-2\varepsilon_1}\|D\|. \tag{47}$$

А для правой части имеет место неравенство

$$\|\overline{f} - f\| \leq \varepsilon_1\|f\| + \sqrt{M}\varepsilon_0 + \sqrt{M}\varepsilon_0(1+\varepsilon_1) + \frac{\varepsilon_0\mathcal{F}(D)}{1-\varepsilon_1}. \tag{48}$$

В виду оценок (46) и (48) и чрезвычайной малости машинных констант ε_0 и ε_1 следует ожидать, что решения x и $\overline{x}$ в определённом смысле близки. Чтобы оценить их разницу необходимо применить теорему 2, которая гарантирует нам следующую оценку:

$$\|x - \overline{x}\| = \|x - x_{\text{маш}}\| \leq \Delta_D(M, D, f)\|x\|\ , \tag{49}$$

где

$$\Delta_D(M, D, f) = \Big(\varepsilon_0\sqrt{M}\tfrac{(2+\varepsilon_1)(1-\varepsilon_1)+\|D\|}{\|f\|(1-\varepsilon_1)(1-2\varepsilon_1)} + \varepsilon_1(\sqrt{M}+1-2\varepsilon_1)\Big) \times \frac{\mu(D)}{1-2\varepsilon_1(1+\sqrt{M}\,\mu(D))},$$

при условии

$$2\varepsilon_1(1+\sqrt{M}\mu(D)) < 1.$$

Видно, что для оценки погрешности необходимо знать число обусловленности исходной матрицы.

Кроме этого необходимо разрешить вопрос об осуществимости вычислений, то есть следует удостовериться, что все промежуточные результаты допускают представление в ЭВМ. Достаточным условием для этого является выполнение неравенств

$$\|f\| \le \min\left\{\sigma_1(D)\varepsilon_\infty\ ,\ \frac{\varepsilon_\infty}{2\mu(D)}\right\}.$$

4.6 Погрешности при вычислении сингулярных чисел матрицы

Как уже упоминалось, вычисление сингулярных чисел произвольной матрицы сводится к расчету собственных чисел симметрической трехдиагональной матрицы. В разделе 3.3 был описан соответствующий алгоритм, основанный на использовании теоремы Штурма. Здесь мы рассмотрим машинную реализацию этого алгоритма. Как обычно нас интересуют вопросы осуществимости промежуточных вычислений и относительная точность конечного результата.

Итак, пусть симметрическая трехдиагональная матрица S имеет вид

$$S = \begin{pmatrix} t_1-\lambda & s_2 & 0 & \dots & 0 \\ s_2 & t_2-\lambda & s_3 & \dots & \vdots \\ 0 & s_3 & t_3-\lambda & s_4 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & s_{M-1} & t_{M-1}-\lambda & s_M \\ 0 & \dots & 0 & s_M & t_M-\lambda \end{pmatrix}. \qquad (50)$$

Тогда машинные значения последовательности Штурма $[\mathcal{P}_j(\lambda)]_{\text{маш}}$ можно вычислить по формулам

$$\begin{aligned} &[\mathcal{P}_0(\lambda)]_{\text{маш}} \equiv 0, \\ &[\mathcal{P}_{i+1}(\lambda)]_{\text{маш}} = |s_{i+1}| \oslash (t_i \ominus_0 \lambda \ominus_0 |s_i| \otimes [\mathcal{P}_i(\lambda)_{\text{маш}}]) \qquad 0 \le i < N, \\ &[\mathcal{P}_N(\lambda)]_{\text{маш}} = 1 \oslash (t_N \ominus_0 \lambda \ominus_0 |s_N| \otimes [\mathcal{P}_i(\lambda)]_{\text{маш}}). \end{aligned} \qquad (51)$$

Операция $\ominus_0$ представляет собой специальным образом определенную операцию вычитания:

$$a \ominus_0 b = \begin{cases} a \ominus b & \text{при } a \ominus b \ne 0, \\ \varepsilon_1/\gamma \max\{|a|,|b|\} & \text{при } a \ominus b = 0. \end{cases}$$

Таким образом, если модули операндов не слишком малы, то результат операции $\ominus_0$ отличен от нуля.

Следующая лемма отвечает на вопрос об осуществимости промежуточных операций в формулах (51) для матриц, элементы которых обладают специальными свойствами.

Лемма 4 *Пусть параметры разрядной сетки ЭВМ удовлетворяют неравенству*

$$\varepsilon_1 \ge 2\gamma \max\left\{\sqrt[4]{\varepsilon_0}\ ,\ \sqrt[4]{\frac{2}{\varepsilon_\infty}}\right\},$$

элементы матрицы подчинены условиям

$$\frac{\varepsilon_1}{\gamma} \le |t_i| \le 1, \quad \frac{\varepsilon_1}{\gamma} \le |s_j| \le 1, \qquad (52)$$

а параметр λ изменяется в интервале $[-3,3]$. Тогда вычисления элементов последовательности Штурма по формулам (51) не приводят к ПЕРЕПОЛНЕНИЯМ.

Замечание. Интервал $[-3,3]$ для аргумента последовательности Штурма не является жестким ограничением. Последовательность Штурма применяется для расчета собственных значений матрицы, следовательно в качестве области изменения λ достаточно взять интервал $[-\|S\|, \|S\|]$, так как он содержит все собственные числа S. Но из (8) и (52) следует $\|S\| \le \mathcal{M}(S) \le 3$ для матриц, элементы которых удовлетворяют неравенствам (52).

Если исходная матрица S получена из двухдиагональной матрицы D по формуле (14), то она очевидно не удовлетворяет этим ограничениям, так как $t_i = 0$. В этом случае матрицу S необходимо подвергнуть дополнительным преобразованиям и возмущениям для того, чтобы можно было судить об осуществимости промежуточных операций при вычислении последовательностей Штурма:

1. Умножить матрицу на число ρ, подобранное так, чтобы максимальный из ненулевых элементов матрицы $(\rho S)_{\text{маш}}$ удовлетворял условию

$$\frac{1}{\gamma} \le \max_{i,j}\{|\rho t_i|\ ,\ |\rho s_j|\} < 1\ .$$

2. Заменить все нулевые элементы на главной и побочных диагоналях на число ε_1/γ.

После этого элементы преобразованной матрицы, обозначим её через $\widetilde{S}$, будут удовлетворять условиям леммы 4. Как следует из леммы, последовательность Штурма для $\widetilde{S}$ вычисляется без ПЕРЕПОЛНЕНИЙ, то есть все промежуточные результаты вычислений могут быть размещены в машинной памяти.

Перейдем к моделированию машинных погрешностей в вычислении последовательности Штурма $\left[\widetilde{\mathcal{P}}_j(\lambda)\right]_{\text{маш}}$ для нормированой и возмущенной матрицы $\widetilde{S}$:

$$\left[\widetilde{\mathcal{P}}_j\right]_{\text{маш}} = \frac{|s_{j+1}|(1+\varphi_{j+1})}{\Big(t_j(1+\alpha_j)-\lambda(1+\overline{\alpha}_j)\Big)(1+\chi_j)-|s_j|(1+\psi_j)\left[\widetilde{\mathcal{P}}_{j-1}\right]_{\text{маш}}(1+\overline{\chi}_j)(1+\overline{\chi}_j)\xi_j},$$

причём

$$|\xi_j| \le \varepsilon_0,\ |\varphi_j| \le \varepsilon_1,\ |\alpha_j| \le \varepsilon_1,\ |\overline{\alpha}_j| \le \varepsilon_1,$$
$$|\chi_j| \le \varepsilon_1,\ |\overline{\chi_j}| \le \varepsilon_1,\ |\psi_j| \le \varepsilon_1.$$

Заметим, что

$$\frac{1+\alpha_j}{1+\overline{\alpha_j}} = 1+\tilde{\alpha}_j,\ \frac{1+\chi_j}{1+\overline{\chi}_j} = 1+\tilde{\chi}_j,\quad \text{где } |\tilde{\alpha}_j| \le \frac{\varepsilon_1}{\gamma},\ |\tilde{\chi}_j| \le \frac{\varepsilon_1}{\gamma}.$$

Если обозначить

$$b_j = s_j\sqrt{\frac{1+\psi_j}{(1+\overline{\alpha}_j)(1+\tilde{\chi}_j)}\frac{1+\varphi_j}{1+\overline{\alpha}_{j-1}}},\quad d_j = t_j(1+\tilde{\alpha}_j),$$

то оказывается, что машинные значения последовательности Штурма $\left[\widetilde{\mathcal{P}}_j(\lambda)\right]_{\text{маш}}$ для матрицы $\widetilde{S}$ совпадают с точными значениями последовательности Штурма $\widetilde{\widetilde{\mathcal{P}}}_j(\lambda)$ для симметрической матрицы $\widetilde{\widetilde{S}}$

$$\widetilde{\widetilde{S}} = \begin{pmatrix} d_1-\lambda & b_2 & 0 & \dots & 0 \\ b_2 & d_2-\lambda & b_3 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_{M-1} & d_{M-1}-\lambda & b_M \\ 0 & \dots & 0 & b_M & d_M-\lambda \end{pmatrix},$$

которая мало отличается от $\widetilde{S}$. Легко проверить, что

$$\|\widetilde{\widetilde{S}} - \widetilde{S}\| \le \mathcal{M}(\widetilde{\widetilde{S}} - \widetilde{S}) \le \frac{\varepsilon_1(5\gamma+1)}{2\gamma-\varepsilon_1(5\gamma+1)}\mathcal{M}(\widetilde{S}) \le 4\varepsilon_1\mathcal{M}(\widetilde{S})$$

следовательно

$$|\lambda_j(\widetilde{\widetilde{S}}) - \lambda_j(\widetilde{S})| \le 4\varepsilon_1\mathcal{M}(\widetilde{S}) = \delta_{st}\mathcal{M}(\widetilde{S}) \le 3\delta_{st}.$$

Последние оценки показывают, что при проведении вычислений по алгоритму Штурма (стр.22) в применении к матрице $\widetilde{S}$, параметр ε допустимой погрешности не имеет смысла выбирать меньше, чем $3\delta_{st}\mathcal{M}(\widetilde{S})$. Если же мы выберем $\varepsilon = 3\delta_{st}\mathcal{M}(\widetilde{S})$, то в результате исполнения алгоритма будет найдено $\left[\lambda_j(\widetilde{S})\right]_{\text{маш}}$ такое, что

$$\left|\left[\lambda_j(\widetilde{S})\right]_{\text{маш}} - \lambda_j(\widetilde{S})\right| \le \frac{3}{2}\delta_{st}\mathcal{M}(\widetilde{S}).$$

По построению $\widetilde{S}$ число $\rho\lambda_j(S)$ почти равно $\lambda_j(\widetilde{S})$, поэтому следует ожидать близости величин $\lambda_j(S)$ и $\frac{1}{\rho}\left[\lambda_j(\widetilde{S})\right]_{\text{маш}}$. Оценим общую погрешность, которая складывается из ошибок при нормировке, при замене нулевых элементов матрицы на ненулевые, при вычислении последовательности Штурма и при конечной разнормировке.

$$|(\lambda_j(S))_{\text{маш}} - \lambda_j(S)| \le |[(\lambda_j(\widetilde{S}))_{\text{маш}}/\rho]_{\text{маш}} - (\lambda_j(\widetilde{S}))_{\text{маш}}/\rho| +$$
$$|(\lambda_j(\widetilde{S}))_{\text{маш}}/\rho - \lambda_j(\widetilde{S})/\rho| + |\lambda_j(\widetilde{S})/\rho - \lambda_j((\rho S)_{\text{маш}})/\rho| +$$
$$+\left|\frac{\lambda_j((\rho S)_{\text{маш}})}{\rho} - \frac{\lambda_j(\rho S)}{\rho}\right|.$$

Применяя стандартную в таких случаях технику оценок и выбрав в качестве ρ подходящую степень γ, в итоге получим неравенство

$$|(\lambda_j(S))_{\text{маш}} - \lambda_j(S)| \le 3\varepsilon_0 + (3\varepsilon_1+\delta_{st})\,\mathcal{M}(S) \le \Delta_{eig}\|S\|,$$

где

$$\Delta_{eig} = \sqrt{N}\varepsilon_1\left(4+\frac{5\gamma+1}{2\gamma-\varepsilon_1(5\gamma+1)}\right),\ \delta_{st} = 4\varepsilon_1.$$

4.7 Вычислительные погрешности при решении линейной системы произвольного вида

В этом разделе мы оценим погрешности при решении системы вида

$$Ax + r = f \tag{53}$$

по алгоритмам, описанным выше. Здесь мы предполагаем, что матрица A прямоугольная размера $N \times M$, $N \geq M$ полного ранга, то есть $\sigma_{min}(A) > 0$. Требование минимальности для невязки можно сформулировать как свойство ортогональности r всем столбцам матрицы A: $A^*r = 0$. Поэтому наряду с системой (53) рассмотрим расширенную систему

$$\mathcal{A}\begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{54}$$

где матрица $\mathcal{A}$ имеет блочный вид:

$$\mathcal{A} = \begin{pmatrix} I & A \\ A^* & 0 \end{pmatrix}.$$

Система (54) разрешима однозначно, следовательно следующее определение корректно:

Определение 7 *Параметром несовместности системы (53) назовем величину*

$$\theta(A, f) = \frac{\|r\|}{\|f - r\|}.$$

В дальнейших оценках мы будем опираться на следующую теорему

Теорема 4 *Пусть A и $\widetilde{A}$ – прямоугольные матрицы $N \times M$, $N \geq M$ полного ранга, f и $\widetilde{f}$ – векторы правых частей, $\theta = \theta(A, f)$, $\widetilde{\theta} = \widetilde{\theta}(\widetilde{A}, \widetilde{f})$, причем $\|A - \widetilde{A}\| \leq \alpha\|A\|, \|f - \widetilde{f}\| \leq \varphi\|f\|$, где α и φ достаточно малы:*

$$2\alpha\mu(A) < 1, \qquad \frac{2(\alpha + \varphi)\mu(A)}{1 - 2\alpha\mu(A)}\sqrt{2\theta^2 + 1} < 1.$$

Тогда верны следующие оценки

$$\frac{\|x - \widetilde{x}\|}{\|x\|} \leq \frac{2(\alpha + \varphi)\mu(A)\sqrt{2\theta^2 + 1}}{1 - 2\alpha\mu(A)}, \tag{55}$$

$$\widetilde{\theta} \leq \frac{\theta + \frac{2\sqrt{2}(\alpha+\varphi)\mu(A)\sqrt{2\theta^2+1}}{1-2\alpha\mu(A)}}{1 - \alpha\mu(A)\left(1 - \frac{2(\alpha+\varphi)\mu(A)\sqrt{2\theta^2+1}}{1-2\alpha\mu(A)}\right)}, \tag{56}$$

$$\widetilde{\theta} \geq \frac{\theta - \frac{2\sqrt{2}(\alpha+\varphi)\mu(A)\sqrt{2\theta^2+1}}{1-2\alpha\mu(A)}}{1 - \alpha\mu(A)\left(1 + \frac{2(\alpha+\varphi)\mu(A)\sqrt{2\theta^2+1}}{1-2\alpha\mu(A)}\right)}. \tag{57}$$

Замечание. Параметры несовместности для систем, связанных ортогональными преобразованиями совпадают.

Займемся моделированием погрешностей, возникающих при решении системы (53). В первую очередь система приводится к двухдиагональному виду. Как было показано ранее, вычислительные ошибки на этом этапе алгоритма сказываются следующим образом: система (53) превращается в систему

$$\widetilde{A}\widetilde{x} + \widetilde{r} = \widetilde{f} \tag{58}$$

с двухдиагональной матрицей

$$\widetilde{A} = \begin{pmatrix} \widetilde{D} \\ 0 \end{pmatrix},$$

где $\widetilde{A} = PAQ + \Omega^{(1)}$, $\widetilde{f} = Pf + \omega^{(1)}$. В данном случае P и Q – ортогональные преобразования, $\Omega^{(1)}$ и $\omega^{(1)}$ матрица и вектор погрешностей, которые согласно (42) и соответственно (43) оцениваются по формулам

$$\|\Omega^{(1)}\| = \sqrt{NM}\varepsilon_0 + \sqrt{M}(2M - 1)\Delta_p(N)\|A\|,$$
$$\|\omega^{(1)}\| = \sqrt{N}\varepsilon_0 + M\Delta_p(N)\|f\|,$$

которые можно несколько упростить при дополнительных, но не слишком жёстких требованиях к матрице A и правой части f. Так, если

$$\sqrt{N}\varepsilon_0 \leq (2M - 1)\Delta_p(N)\|A\|,$$
$$\sqrt{N}\varepsilon_0 \leq M\Delta_p(N)\|f\|,$$

то

$$\|\Omega^{(1)}\| = 2\sqrt{M}(2M - 1)\Delta_p(N)\|A\|,$$
$$\|\omega^{(1)}\| = 2M\Delta_p(N)\|f\|.$$

Отсюда легко получается

$$\|\Omega^{(1)}\| = 2\sqrt{M}(2M - 1)\Delta_p(N)(1 + 2\sqrt{M}(2M - 1)\Delta_p(N))\|\widetilde{A}\|,$$
$$\|\omega^{(1)}\| = 2M\Delta_p(N)(1 + 2M\Delta_p(N))\|\widetilde{f}\|. \tag{59}$$

Разбиваем векторы $\widetilde{f}$, $\widetilde{r}$ на две векторные компоненты длины M и $N-M$ соответственно:

$$\widetilde{f} = \begin{pmatrix} \widetilde{f}^{(1)} \\ \widetilde{f}^{(2)} \end{pmatrix}, \qquad \widetilde{r} = \begin{pmatrix} \widetilde{r}^{(1)} \\ \widetilde{r}^{(2)} \end{pmatrix},$$

тогда $\widetilde{r}^{(1)} = 0$, $\widetilde{r}^{(2)} = \widetilde{f}^{(2)}$, а $\widetilde{x}$ удовлетворяет системе $\widetilde{D}\widetilde{x} = \widetilde{f}^{(1)}$ с квадратной двухдиагональной матрицей $\widetilde{D}$. На этом шаге производится и оценка $\mu(\widetilde{D})$, и если эта величина оказалась слишком велика (то есть матрица $\widetilde{D}$ с точки зрения компьютера вырождена) и неравенство

$$\frac{\|\Omega^{(1)}\|}{\|D\|}\mu(\widetilde{D}) < 1,$$

(см. теорему 2) невыполнено, то от дальнейших вычислений следуют отказаться с указанием причины. Для системы (58) $\widetilde{\theta} = \theta(\widetilde{A}, \widetilde{f})$ по определению вычисляется следующим образом:

$$\widetilde{\theta} = \frac{\|\widetilde{f}^{(2)}\|}{\|\widetilde{f}^{(1)}\|}.$$

На основании теоремы 4 и неравенств (59) мы можем оценить погрешность

$$\|\widetilde{x} - Q^*x\| \le \widetilde{\Delta}\|\widetilde{x}\|,$$

где

$$\widetilde{\Delta} = \frac{2\mu(\widetilde{D})\sqrt{2\widetilde{\theta}^2+1}}{1-2\sqrt{M}(2M-1)\Delta_p(N)(1+2\sqrt{M}(2M-1)\Delta_p(N))\mu(\widetilde{D})} \times$$
$$\left(2\sqrt{M}(2M-1)\Delta_p(N)(1+2\sqrt{M}(2M-1)\Delta_p(N)) + 2M\Delta_p(N)(1+2M\Delta_p(N))\right)$$

Обратим внимание, что

$$\|\widetilde{x}\| \le \frac{\|Q^*x\|}{1-\widetilde{\Delta}} = \frac{\|x\|}{1-\widetilde{\Delta}},$$

и следовательно

$$\|\widetilde{x} - Q^*x\| \le \frac{\widetilde{\Delta}}{1-\widetilde{\Delta}}\|x\|.$$

На следующем шаге алгоритма происходит решение системы $\widetilde{D}\widetilde{x} = \widetilde{f}^{(1)}$. В результате вычислений будет найден некоторый вектор $\widetilde{\widetilde{x}}$, являющийся точным решением возмущенной системы $\widetilde{\widetilde{D}}\widetilde{\widetilde{x}} = \widetilde{\widetilde{f}}^{(1)}$, где

$\widetilde{\widetilde{D}} = \widetilde{\widetilde{D}} + \Omega^{(2)}$, $\widetilde{\widetilde{f}}^{(1)} = \widetilde{\widetilde{f}}^{(1)} + \omega^{(2)}$. Матрица $\Omega^{(2)}$ и вектор $\omega^{(2)}$ моделируют погрешности при решении системы и удовлетворяют оценкам

$$\|\Omega^{(2)}\| \le \sqrt{M}\frac{2\varepsilon_1}{1-2\varepsilon_1}\|\widetilde{D}\|,$$
$$\|\omega^{(2)}\| \le 2\varepsilon_1\|\widetilde{f}^{(1)}\|$$

согласно формулам (47) и (48). Поэтому по теореме 2

$$\|\widetilde{x} - \widetilde{\widetilde{x}}\| \le \widetilde{\widetilde{\Delta}}\|\widetilde{x}\|,$$

где мы обозначили

$$\widetilde{\widetilde{\Delta}} = \varepsilon_1\left(\frac{2\sqrt{M}}{1-2\varepsilon_1}+2\right)\frac{\mu(\widetilde{D})}{1-2\varepsilon_1\sqrt{M}\mu(\widetilde{D})/(1-2\varepsilon_1)}.$$

Очевидное следствие из последнего неравенства

$$\|\widetilde{\widetilde{x}}\| \le (1+\widetilde{\widetilde{\Delta}})\|\widetilde{x}\|.$$

Используя неравенство для $\|\widetilde{x}\|$, полученное выше, находим

$$\|\widetilde{\widetilde{x}}\| \le \frac{1+\widetilde{\Delta}}{1-\widetilde{\widetilde{\Delta}}}\|x\|.$$

Из неравенства треугольника легко получается

$$\|\widetilde{\widetilde{x}} - Q^*x\| \le \|\widetilde{\widetilde{x}} - \widetilde{x}\| + \|\widetilde{x} - Q^*x\| \le \frac{\widetilde{\widetilde{\Delta}} + \widetilde{\Delta}}{1-\widetilde{\Delta}}\|x\|.$$

Наконец на последнем шаге происходит применение преобразования Q к вектору $\widetilde{\widetilde{x}}$ и преобразования P^* к вектору невязки $\widetilde{r}$. При этом будут получены векторы $x_{\text{маш}} = Q\widetilde{\widetilde{x}} + \xi$ и $r_{\text{маш}} = P^*\widetilde{r} + \eta$. Согласно оценке (43) на странице 37 получим

$$\|\xi\| \le \sqrt{M}\varepsilon_0 + M\Delta_p(N)\|\widetilde{\widetilde{x}}\|.$$

В итоге оценка погрешности решения линейной системы произвольного вида, равная разности между машинным $x_{\text{маш}}$ и точным x значениями, выглядит так:

$$\|x_{\text{маш}} - x\| \le \|Q\widetilde{\widetilde{x}} + \xi - x\| \le \|\widetilde{\widetilde{x}} - Q^*x\| + \|\xi\| \le$$
$$\frac{\widetilde{\Delta}(1+M\Delta_p(N)) + \widetilde{\widetilde{\Delta}} + M\Delta_p(N)}{1-\widetilde{\Delta}}\|x\| + \sqrt{M}\varepsilon_0.$$

5 Заключение

Полученые выше теоретические выводы были использованы нами при разработке пакета программ для решения некоторых задач линейной алгебры. Подробное описание содержимого пакета на языке ФОРТРАН-90 можно найти в Приложении, а здесь мы дадим краткий обзор его возможностей.

Основными являются две процедуры. Первая – процедура *LinSystemSolution* для решения системы $Ax + r = f$ с произвольной прямоугольной $N \times M$-матрицей A, $(N \geq M)$. Её параметрами являются матрица A, правая часть f, решение x, невязка r и оценка точности решения. Вторая – процедура *InverseMatrix* для построения обратной к квадратной матрице, параметрами которой, кроме исходной и обратной матриц, также является оценка ошибки.

Помимо этих двух, пакет содержит ряд вспомогательных служебных процедур, некоторые из которых могут заинтересовать подготовленного пользователя. Так, подпрограмма *TwoDiagonalisation* с помощью ортогональных преобразований приводит произвольную $N \times M$-матрицу A, $(N \geq M)$ к двухдиагональному виду. Для полученой таким образом двухдиагональной матрицы функция *TwoDiagCond* выдает ее число обусловленности, а процедура *SingValue* – сингулярное число с заданным порядковым номером.

В самое ближайшее время мы планируем дополнить пакет рядом новых процедур, среди которых: 1) вычисление собственных чисел и векторов симметрической матрицы; 2) решение недоопределенной системы, то есть системы $Ax = f$ с прямоугольной матрицей размера $N \times M$ при $N < M$; 3) представление матрицы в виде произведения ортогональной и верхней треугольной матриц (т.н. QR-разложение). Все они также будут выдавать оценку точности вычислений.

Мы сознательно не включили в текст каких-либо доказательств правильности работы наших программ. Дело в том, что создатели теории используемых нами алгоритмов собрали коллекцию ярких иллюстративных примеров работы механизмов оценки точности [2]. Именно эти примеры мы и использовали в качестве тестов. Результаты тестирования полностью удовлетворяют теории и совпадают с приведенными в указанных монографиях.

Авторы искренне благодарят А.М.Блохина за инициативу по организации работ, направленных на создание вычислительных алгоритмов нового поколения, Л.Ф.Хайло за консультации и советы, Б.В.Чирикова и В.В.Вечеславова за всестороннюю поддержку и помощь.

A Описание программ

Программный пакет состоит из четырёх модулей: *SingMod*, *TDiagMod*, *LSysMod* и *ArithMod*. Первые три из них являются основными и содержат подпрограммы вычисления сингулярных чисел матрицы, числа обусловленности, решение системы линейных уравнений, обращения матрицы. Последний модуль является служебным. В нём собраны такие процедуры, как определение машинных констант, вычисление уточнённого значения квадратного корня, генерация нулевой и единичной матриц, вычисление нормы вектора и другие.

A.1 Модуль *SingMod*

В этом модуле основной является функция *TwoDiagCond*, значение которой есть число обусловленности двухдиагональной матрицы. Так как число обусловленности равно отношению максимального и минимального сингулярных чисел, то очень важной является также процедура *SingValue*, вычисляющая их значения.

A.1.1 Функция NormEstimation

Назначение:

Вычисляет оценку сверху евклидовой нормы двухдиагональной матрицы. Возвращает вещественное положительное число с двойной точностью, которым оценивается норма двухдиагональной матрицы.

Обращение к функции:

NormEstimation(MainDiag,SecondaryDiag)

Параметры:

MainDiag – входной параметр, главная диагональ.

SecondaryDiag – входной параметр, побочная диагональ.

Точки остановки и сообщения: отсутствуют.
Используемые внешние подпрограммы:
MachConst (Модуль ArithMod)

A.1.2 Подпрограмма SingValue

Назначение:

Вычисляет J-е сингулярное число двухдиагональной матрицы. Сингулярные числа считаются расположенными по возрастанию.

Обращение к подпрограмме:

Call SingValue(Sigma,MainDiag,SecondaryDiag,J,error)

Параметры:

Sigma – выходной параметр, действительное число с двойной точностью, содержит вычисленное значение J-го сингулярного числа.

MainDiag – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержит элементы главной диагонали двухдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $N-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали двухдиагональной матрицы.

J – входной параметр, целое положительное число меньше N, номер расчитываемого сингулярного числа.

error – выходной параметр (является необязательным), вещественное число с двойной точностью, содержит оценку ошибки при расчёте *Sigma*: $||Sigma - \sigma_j|| \leq error$.

Точки остановки и сообщения: отсутствуют.
Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*NormEstimation (Модуль SingMod)*

A.1.3 Функция TwoDiagCond

Назначение:

Вычисляет число обусловленности двухдиагональной матрицы. Возвращает вещественное число с двойной точностью $\mu = \frac{\sigma_{max}}{\sigma_{min}}$, где σ_{max} , σ_{min} – максимальное и минимальное сингулярные числа исходной матрицы.

Обращение к функции:

TwoDiagCond(MainDiag,SecondaryDiag)

Параметры:

MainDiag – входной параметр, главная диагональ.

SecondaryDiag – входной параметр, побочная диагональ.

Точки остановки и сообщения: отсутствуют
Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*SingValue (Модуль SingMod)*

A.2 Модуль *TDiagMod*

Подпрограммы этого модуля работают с двухдиагональными матрицами. Процедура *TwoDiagSystemSolution* решает систему с двухдиагональной матрицей коэффициентов, процедура *InvTwoDiagMatrix* производит обращение двухдиагональной матрицы.

A.2.1 Подпрограмма TwoDiagSystemSolution

Назначение:

По заданным диагоналям двухдиагональной матрицы и заданной правой части вычисляет решение линейной системы, невязку с минимальной нормой, оценку ошибки.

Обращение к подпрограмме:

Call TwoDiagSystemSolution(Sol,Md,Sd,RHS,r,error)

Параметры:

Sol – выходной параметр, одномерный массив длины M вещественных чисел с двойной точностью. После исполнения программы содержит компоненты решения системы.

Md – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью. Содержит значения элементов главной диагонали матрицы решаемой системы.

Sd – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью. Содержит значения элементов побочной диагонали матрицы решаемой системы.

RHS – входной параметр, одномерный массив длины $N \geq M$ вещественных чисел с двойной точностью. Содержит значения компонент вектора правой части системы.

r – выходной необязательный параметр, одномерный массив длины N, Содержит значения компонент вектора невязки системы.

error – выходной необязательный параметр, положительное вещественное чисело с двойной точностью. Содержит оценку нормы ошибки:

$$\frac{\|Sol - x\|}{\|x\|} \leq error,$$

где x – точное решение системы.

Точки остановки и сообщения:
Если размерности массивов Sol, Md, Sd, RHS, r неудовлетворяют описанию, то программа прекращает свою работу с сообщением:

Bad data dimension in "TwoDiagSystemSolution"

Если норма правой части $\|RHS\|$ превосходит минимум из $\sigma_1\varepsilon_\infty$ и $\frac{\varepsilon_\infty}{2\mu}$, где σ_1 – минимальное сингулярное число матрицы системы, μ – обусловленность матрицы системы, ε_∞ – максимальное положительное машинное число, то вычисления могут быть прерваны из-за невозможности, размещения результатов промежуточных вычислений в памяти машины. Поэтому, если $\|RHS\| \geq \min\left\{\sigma_1\varepsilon_\infty\ ,\ \frac{\varepsilon_\infty}{2\mu}\right\}$, то выдаётся предупреждение:

Warning: Calculation can be interrupt, Right Hand Side is too big

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*SingValue (Модуль SingMod)*
3.*VectorNorm (Модуль ArithMod)*

A.2.2 Подпрограмма InvTwoDiagMatrix

Назначение:

Находит обратную матрицу к исходной двухдиагональной.

Обращение к подпрограмме:

Call InvTwoDiagMatrix(Md,Sd,Inverse,error)

Параметры:

Md – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью. Содержит значения элементов главной диагонали матрицы .

Sd – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью. Содержит значения элементов побочной диагонали матрицы .

Inverse – выходной параметр, двумерный массив размера $M \times M$ вещественных чисел с двойной точностью. Содержит значения элементов обратной матрицы.

error – выходной необязательный параметр, положительное вещественное чисело с двойной точностью. Содержит оценку нормы ошибки:

$$\frac{\|Inverse - A^{-1}\|}{\|A^{-1}\|} \leq error,$$

где A^{-1} – точное значение обратной матрицы.

Точки остановки и сообщения:
Если размеры массивов-параметров не соответствуют описанию, программа прекращает работу. При этом выдаётся сообщение:

Bad Data Dimension in "InvTwoDiagMatrix"

Используемые внешние подпрограммы:
1.*Reflection (Модуль LSysMod)*
2.*VectorReflection (Модуль LSysMod)*
3.*MachConst (Модуль ArithMod)*
4.*Eye (Модуль ArithMod)*
5.*Sgn (Модуль ArithMod)*

A.3 Модуль *LSysMod*

Основные программы этого модуля – *LinSystemSolution* (решение линейной системы) и *InverseMatrix* (обращение квадратной матрицы). Основой

использованных в них алгоритмов является приведение матрицы к двухдиагональному виду при помощи ортогональных преобразований. Эти действия осуществляются подпрограммами *TwoDiagonalisation* (приведение к двухдиагональному виду), *Reflection* (вычисление преобразования отражения), *VectorReflection* (применение преобразования отражения к произвольному вектору).

A.3.1 Подпрограмма TwoDiagonalisation

Назначение:

Приводит произвольную матрицу A размера $N \times M$, $N \geq M$ к двухдиагональному виду при помощи ортогональных преобразований: $D = PAQ$, A – исходная матрица, D – двухдиагональная матрица того же размера.

Обращение к подпрограмме:

Call TwoDiagonalisation(Matrix,MainDiag,SecondaryDiag,MirLeft, MirRight,error)

Параметры:

Matrix – входной и выходной параметр, двумерный массив размера $N \times M$, $N \geq M$, вещественных чисел с двойной точностью. На входе содержит элементы исходной матрицы, на выходе – вычисленные элементы двухдиагональной матрицы. P и Q – квадратные ортогональные матрицы соответствующих размеров.

MainDiag – выходной необязательный параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержащий вычисленные элементы итоговой двухдиагональной матрицы.

SecondaryDiag – выходной необязательный параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержащий вычисленные элементы побочной диагонали итоговой двухдиагональной матрицы.

MirLeft – выходной необязательный параметр, двумерный массив вещественных чисел с двойной точностью размера $N \times M$, содержащий в качестве столбцов последовательно вычисленные векторы нормалей к плоскостям, относительно которых производились отражения, применяемые слева.

MirRight – выходной необязательный параметр, двумерный массив вещественных чисел с двойной точностью размера $M \times (M-1)$, содержащий в качестве столбцов последовательно вычисленные векторы нормалей к плоскостям, относительно которых производились отражения, применяемые справа.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, оценивающее норму ошибки: $\|D_{\text{маш}} - P_{\text{маш}} A Q_{\text{маш}}\| \leq error$.

Точки остановки и сообщения:
Если размеры массивов-параметров не соответствуют описанию, программа прекращает работу. При этом выдаётся сообщение:

Bad Data Dimension in "TwoDiagonalization"

Используемые внешние подпрограммы:
1. *Reflection (Модуль LSysMod)*
2. *VectorReflection (Модуль LSysMod)*
3. *MachConst (Модуль ArithMod)*
4. *Eye (Модуль ArithMod)*
5. *Sgn (Модуль ArithMod)*

A.3.2 Подпрограмма Reflection

Назначение:

Если не задан параметр I, программа вычисляет вектор нормали к плоскости, относительно которой должен быть отражён вектор $Vector1$, чтобы быть коллинеарным вектору $Vector2$, и матрицу отражения. Если задан параметр I, то $Vector2$ принимает следующий вид: первые $I-1$ компоненты совпадают с соответствующими компонентами вектора $Vector1$, все компоненты начиная с $I+1$-ой равны нулю, а I-я компонента находится из условия равенства норм $Vector1$ и $Vector2$.

Обращение к подпрограмме:

Call Reflection(Vector1,Vector2,ReflVector,ReflMatrix,I)

Параметры:

Vector1 – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержащий компоненты исходного вектора.

Vector2 – входной-выходной необязательный параметр, одномерный массив длины N, вещественных чисел с двойной точностью, содержащий компоненты вектора, получаемого из исходного в результате действия оператора отражения.

ReflVector – выходной необязательный параметр, одномерный массив длины N, вещественных чисел с двойной точностью, содержащий на выходе вычисленные значения компонент вектора нормали гиперплоскости, относительно которой осуществляется отражение.

ReflMatrix – выходной необязательный параметр, двумерный массив размера $N \times N$, вещественных чисел с двойной точностью, содержащий на выходе вычисленные компоненты матрицы оператора отражения.

I – входной необязательный параметр, обязателен при отсутствии параметра Vector2. Целое число $1 \leq I \leq N$. Указывает на то, что искомое преобразование отражения, действуя на вектор Vector1, сохраняет первые $I-1$ компоненты, зануляет компоненты начиная с $I+1$-ой до N-ой, не изменяя при этом нормы.

Точки остановки и сообщения:
1. Если параметры программы не соответствуют описанию, то работа подпрограммы прекращается. При этом выдаётся сообщение:

Bad Data for "Reflection"

2. Если отсутствует параметр I, а элементы массива $Vector2$ слишком малы, то есть

$$\sqrt{\sum_{j=1}^{N} |Vector2(j)|^2} < \varepsilon_0 \ ,$$

где ε_0– минимальное положительное вещественное число с двойной точностью, то работа прекращается с выдачей соощения:

Bad Data for "Reflection"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*Eye (Модуль ArithMod)*
3.*Sgn (Модуль ArithMod)*
4.*Zeros (Модуль ArithMod)*

A.3.3 Подпрограмма VectorReflection

Назначение:

Отражение вектора относительно гиперплоскости с заданным вектором нормали.

Обращение к подпрограмме:

Call VectorReflection(ReflVector, Vector)

Параметры:

ReflVector – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержащий компоненты вектора нормали к некоторой плоскости.

Vector – входной-выходной, одномерный массив длины N, вещественных чисел с двойной точностью. На входе – начальный вектор, на выходе – координаты вектора, получаемого из исходного после его отражения относительно плоскости с нормалью $ReflVector$.

Точки остановки и сообщения:
Если размерности массивов $ReflVector$ и $Vector$ не совпадают, то программа прекращает работу с сообщением:

Bad Data Dimension in "VectorReflection"

Используемые внешние подпрограммы:
MachConst (Модуль ArithMod)

A.3.4 Подпрограмма LinSystemSolution

Назначение:

Подпрограмма решает линейную систему алгебраических уравнений с матрицей $Matrix$ размера $N \times M$, причём $N \geq M$, и правой частью RHS. В качестве решения выдается вектор Sol, а также вектор невязки r и оценка ошибки $error$:

$$error \geq \frac{\|Sol - x\|}{\|x\|} \ ,$$

где x – истинное решение системы.

Обращение к подпрограмме:

Call LinSystemSolution(Sol,Matrix,RHS,r,error)

Параметры:

Sol – выходной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержащий компоненты вектора решения системы.

Matrix – входной параметр, двумерный массив $N \times M$ вещественных чисел с двойной точностью, матрица коэффициентов линейной системы уравнений.

RHS – входной параметр, одномерный массив длины N, вещественных чисел с двойной точностью, содержащий значения компонент вектора правой части системы.

r – выходной необязательный параметр, одномерный массив длины N, Содержит значения компонент вектора невязки системы.

error – выходной необязательный параметр, положительное вещественное чисело с двойной точностью. Содержит оценку нормы ошибки: $\|Sol - x\|/\|x\| \leq error$, где x – точное решение системы.

Точки остановки и сообщения:
Если размеры параметров не соответствуют описанию, то программа прекращает работу. При этом выдаётся сообщение:

Bad Data Dimension in "LinSystemSolution"

Используемые внешние подпрограммы:
1. *TwoDiagonalisation (Модуль LSysMod)*
2. *TwoDiagSystemSolution (Модуль TDiagMod)*

A.3.5 Подпрограмма InverseMatrix

Назначение:

Подпрограмма по исходной матрице строит обратную к ней.

Обращение к подпрограмме:

Call InverseMatrix(Matrix,InvMatrix,error)

Параметры:

Matrix – входной параметр, массив $N \times N$, содержит элементы исходной матрицы.

InvMatrix – выходной параметр, двумерный массив $N \times N$. На выходе содержит вычисленные значения обратной матрицы.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью. Содержит оценку нормы ошибки:

$$\frac{\|InvMatrix - Matrix^{-1}\|}{\|Matrix^{-1}\|} \leq error$$

Точки остановки и сообщения:
Если размеры параметров не соответствуют описанию, то программа прекращает работу. При этом выдаётся сообщение:

Bad Data size in "InverseMatrix"

Используемые внешние подпрограммы:
1. *TwoDiagonalisation (Модуль LSysMod)*
2. *InvTwoDiagMatrix (Модуль TDiagMod)*

A.4 Модуль *ArithMod*

A.4.1 Подпрограмма MachConst

Назначение: определяет машиные константы $\gamma, \varepsilon_0, \varepsilon_1, \varepsilon_\infty$.
Обращение к подпрограмме: *Call MachConst*

A.4.2 Функция Sgn

Назначение: определяет знак целого или вещественного числа.
Обращение к подпрограмме: *Sgn(x)*
Параметр: входной параметр – целое или вещественное число с двойной точностью.
Результат: число того же типа, что и аргумент, равное $\mathrm{sign}(x)$.

A.4.3 Подпрограмма Zeros

Назначение: зануляет все элементы матрицы.
Обращение к подпрограмме: *Call Zeros(A)*

Параметр: *A* – входной-выходной параметр, может быть одномерным или двумерным массивом вещественных чисел с двойной точностью, на выходе все элементы массива равны нулю.
Используемые внешние подпрограммы: *MachConst (Модуль ArithMod)*

A.4.4 Подпрограмма Eye

Назначение: входную квадратную матрицу заменяет на единичную.
Обращение к подпрограмме: *Call Eye(A)*
Параметр: *A* – входной-выходной параметр, двумерный массив вещественных чисел с двойной точностью размера $N \times N$. На выходе содержит единичную матрицу.
Используемые внешние подпрограммы: *MachConst (Модуль ArithMod)*

A.4.5 Функция VectorNorm

Назначение: Вычисляет норму вектора.
Обращение к подпрограмме: *VectorNorm(Vector)*
Параметр: *Vector* – входной параметр, одномерный массив вещественных чисел с двойной точностью длины N.
Результат: вещественное число с двойной точностью, вычисленная норма вектора.

Литература

[1] Ф.Р. Гантмахер, *Теория матриц*, Наука, Москва (1967).

[2] С.К. Годунов, А.Г. Антонов, О.П. Кирилюк,В.И. Костин, *Гарантированная точность решения систем линейных уравнений в евклидовых пространствах*, Наука, Новосибирск (1992).

[3] З.А. Ляпидевская, *Комплекс процедур по линейной алгебре*, Препринт 259, Новосибирск, Вычислительный центр СОАН СССР (1980).

[4] С.Г. Маковер, *Решение системы нормальных уравнений при помощи матриц*, Астрономический журнал, XXXIII, 3 (1956).

[5] А.Н. Малышев, *Введение в вычислительную линейную алгебру*, Наука, Новосибирск (1991).

[6] Дж.Х. Уилкинсон, *Алгебраическая проблема собственных значений*, Наука, Москва (1970).

[7] Д.К. Фаддеев, В.Н. Фаддеева, *Вычислительные методы линейной алгебры*, Физматгиз, Москва (1960).

[8] D.H. Bailey, *ACM Translation on Math. Software* 19, 288 (1993).

[9] Б.М. Щиголев, *Математическая обработка наблюдений*, ГИФМЛ, Москва (1960).

[10] В.В.Вечеславов, Б.В.Чириков ЖЭТФ **114**, 1516 (1998)

Э.А. Бибердорф, Н.И. Попова

Решение линейных систем с гарантированной оценкой точности результатов (часть первая

E.A. Biberdorf, N.I. Popova

Solution of the linear systems with the guaranteed estimate of the results accuracy (part 1)

ИЯФ 99-49

Ответственный за выпуск А.М. Кудрявцев
Работа поступила 31.05. 1999 г.

Сдано в набор 15.06. 1999 г.
Подписано в печать 15.06. 1999 г.
Формат бумаги 60×90 1/16 Объем 2.5 печ.л., 2.0 уч.-изд.л.
Тираж 100 экз. Бесплатно. Заказ N° 49

Обработано на IBM PC и отпечатано на
ротапринте ИЯФ им. Г.И. Будкера СО РАН
Новосибирск, 630090, пр. академика Лаврентьева, 11.