В.И. РЕПНИКОВ

Белгосуниверситет

Кафедра вычислительной математики

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ

Курс лекций

Минск 2012

ВВЕДЕНИЕ

Мы приступаем к изучению нового цикла дисциплин, который в разное время назывался «Методы вычислений» или «Численные методы», а в настоящее время для студентов специальности «Прикладная математика» состоит из трех частей: «Вычислительные методы алгебры», «Методы численного анализа» и «Численные методы математической физики». Цель курса – вплотную подвести слушателей к проблемам, связанным с изучением количественных характеристик того или иного объекта или явления в ходе вычислительного эксперимента, познакомить с традиционными подходами к решению тех или иных классов задач, возникающих в процессе повседневной деятельности человека, к конкретным алгоритмам, особенностями их применения и характеристиками, подтолкнуть к мысли о том, что лучшие алгоритмы еще не созданы и, быть может, посеять сомнения во всемогуществе современных компьютеров (точнее, их технических возможностей, несмотря на впечатляющий рост в последние десятилетия). Другими словами, вам предстоит научиться применять те знания, которые вы приобрели в различных областях математики, к решению проблемы, которую иногда называют «доведение до числа».

Как уже говорилось выше, первая часть курса носит название «Вычислительные методы алгебры» (точнее – вычислительные методы ***линейной*** алгебры). К ним относятся:

1. Численные методы решения систем линейных алгебраических уравнений и связанные с ними методы обращения матриц и вычисления определителей;
2. Численные методы нахождения собственных значений и собственных векторов матриц.

При этом, конечно же, хотелось бы подчеркнуть, что ни один из разделов численных методов не может существовать изолированно от других. Прояснять и расширять эти связи – также одна из задач курса.

Ниже, прежде чем перейти к рассмотрению первых конкретных алгоритмов линейной алгебры, мы рассмотрим азбуку численных методов, о существовании и выводах которой, к сожалению, большинство из пользователей, встречающихся с необходимостью применения численных методов, как правило, не задумывается.

Прежде всего, попытаемся (см., например, [6]) осмыслить место теории численных методов в системе других областей знаний и рассказать о проблемах, связанных с ее применением.

Математика как наука возникла в связи с необходимостью решения практических задач: измерения на местности, навигации и т.д. Поэтому она всегда была ***численной*** математикой, ее целью являлось получение решения в виде числа.

Численное решение прикладных задач всегда интересовало математиков. Крупнейшие представители прошлого сочетали в своих исследованиях изучение явлений природы, получение их математического описания (современный термин – построение математической модели), и его исследование. При этом анализ усложненных моделей, более тонко учитывающих особенности изучаемого явления, требовал, как правило, создания специальных, обычно приближенных численных методов.

Настоящее время характерно резким расширением приложений математики, что во многом связано с развитием средств вычислительной техники. В результате появления компьютеров скорость выполнения арифметических операций возросла от 0.1 операции в секунду при ручном счете до 10^{16} операций на современных многопроцессорных вычислительных системах из TOP-500. Примерно в такой же степени выросли и другие характеристики (например, если размер оперативной памяти для компьютеров первого поколения составлял примерно 1 К, то сейчас уже никого не удивишь десятками Гигабайт для вполне себе заурядной персоналки).

Все это способствовало распространению мнения о всемогуществе компьютера и впечатления о том, что математики избавились почти от всех хлопот, связанных с численным решением задач, а следовательно, и разработка новых методов для их решения уже не столь существенна. В действительности же дело обстоит совершенно иначе, поскольку потребности эволюции, как правило, ставят перед наукой задачи, находящиеся на грани ее возможностей, а просмотр методов решения сложных задач показывает, что, как правило, эффект, достигаемый за счет совершенствования численных методов, по порядку сравним с эффектом, достигаемым за счет повышения производительности компьютера.

Подчеркнем, что численные методы – это методы, работающие, как правило, с приближенными данными. При этом представляется безусловно важным приобретение навыков в работе с приближенными величинами, о чем речь пойдет ниже.

§ 1. Источники и классификация погрешности

Погрешность решения задачи обусловливается следующими причинами:

1. Математическое описание задачи является неточным, в частности, неточно заданы входные данные описания;
2. Применяемый для решения метод часто не является точным: получение точного решения возникающей математической задачи требует неограниченного или неприемлемо большого числа арифметических операций, поэтому вместо точного решения задачи приходится прибегать к приближенному;
3. При вводе данных в компьютер, при выполнении арифметических операций и при выводе данных производится округление (что связано с ограниченностью разрядной сетки).

Погрешности, соответствующие названным причинам, называют: 1) ***неустранимой погрешностью***; 2) ***погрешностью метода***; 3) ***вычислительной погрешностью***.

Иногда неустранимую погрешность подразделяют на две части: а) неустранимой погрешностью называют лишь погрешность, являющуюся следствием неточности задания числовых данных, входящих в математическое описание задачи; б) погрешность, являющуюся следствием несоответствия математического описания задачи реальности, называют ***погрешностью математической модели***.

Таким образом, если I – точное значение отыскиваемого параметра, $\widetilde{I}$ – значение этого параметра, соответствующее принятому математическому описанию, I_h – решение задачи, получаемое при реализации численного метода в предположении отсутствия округлений, $\widetilde{I}_h$ – приближенное решение задачи, получаемое при реальных вычислениях, то

$$\begin{aligned} &\rho_1 = \widetilde{I} - I \text{ – неустранимая погрешность,} \\ &\rho_2 = I_h - \widetilde{I} \text{ – погрешность метода,} \\ &\rho_3 = \widetilde{I}_h - I_h \text{ – вычислительная погрешность.} \end{aligned} \tag{1.1}$$

Полная погрешность $\rho_0 = \widetilde{I}_h - I$, равная разности между реально получаемым и точным решениями задачи, очевидно, удовлетворяет равенству

$$\rho_0 = \rho_1 + \rho_2 + \rho_3 . \tag{1.2}$$

Во многих случаях под термином ***погрешность*** того или иного вида удобно понимать не рассмотренные выше разности между приближениями, а некоторые меры близости между ними, например, в скалярном случае полагают

$$\rho_0 = \left|\widetilde{I}_h - I\right|,\ \rho_1 = \left|\widetilde{I} - I\right|,\ \rho_2 = \left|I_h - \widetilde{I}\right|,\ \rho_3 = \left|\widetilde{I}_h - I_h\right|.$$

При таких обозначениях вместо (1.2) получим соотношение

$$\rho_0 \le \rho_1 + \rho_2 + \rho_3 . \tag{1.3}$$

Тип используемых мер близости зависит от того, каким функциональным пространствам принадлежат рассмотренные выше решения.

Заметим, что исследование вроде бы «бесполезной» неустранимой погрешности может помочь сформулировать разумные требования к точности результата численного решения (и в частности, например, разработчику программных комплексов – грамотно ставить «защиту от дурака»).

§ 2. Абсолютная и относительная погрешности. Форма записи данных

Если a – точное значение некоторой величины, а a^* – известное приближение к нему, то ***абсолютной погрешностью*** приближенного значения a^* обычно называют некоторую величину $\Delta(a^*)$, про которую известно, что

$$|a - a^*| \le \Delta(a^*).$$

Относительной погрешностью приближенного значения называют некоторую величину $\delta(a^*)$, про которую известно, что

$$\left|\frac{a - a^*}{a^*}\right| \le \delta(a^*).$$

Значащими цифрами числа называют все цифры в его записи, начиная с первой ненулевой слева.

Пример. У чисел $a^* = 0.\underline{03045}$, $a^* = 0.\underline{03045000}$ значащими цифрами являются подчеркнутые цифры. Число значащих цифр в первом случае равно 4, во втором – 7.

Значащую цифру называют ***верной***, если абсолютная погрешность числа не превосходит половины единицы разряда, соответствующего этой цифре.

Пример. $a^* = 0.\underline{03045}$, $\Delta(a^*) = 0.000003$, $a^* = 0.\underline{03045}000$, $\Delta(a^*) = 0.0000007$, подчеркнутые цифры являются верными.

Если все значащие цифры верные, то говорят, что число записано ***со всеми верными цифрами***.

Пример. При $a^* = 0.\underline{03045}$, $\Delta(a^*) = 0.000003$ число a^* записано со всеми верными цифрами.

Иногда употребляется термин ***число верных цифр после запятой***. В последнем примере это число равно 5.

Довольно часто информация о некоторой величине задается пределами ее изменения:

$$a_1 \le a \le a_2,$$

причем пределы записываются с одинаковым числом знаков после запятой. Абсолютную или относительную погрешность обычно записывают в виде числа, содержащего одну или две значащие цифры. Информация о том, что a^* является приближенным значением числа a с абсолютной погрешностью $\Delta(a^*)$, иногда записывают в виде

$$a = a^* \pm \Delta(a^*),$$

причем a^* и $\Delta(a^*)$ принято записывать с одинаковым числом знаков после запятой.

Точно так же информацию о том, что a^* является приближенным значением числа a с относительной погрешностью $\delta(a^*)$, записывают в виде

$$a = a^*(1 \pm \delta(a^*)).$$

§ 3. Запись чисел в компьютере

Современные компьютеры оперируют с числами, записанными в одной из двух приведенных ниже форм.

Первая форма записи – ***числа с фиксированной запятой***. Она характеризуется тремя параметрами:

β – основание системы счисления;

t – количество цифр, отведенных для представления числа в машинном слове;

f – количество цифр (разрядов), отведенных для представления дробной части числа.

Обозначим систему чисел с фиксированной точкой $P(\beta, t, f)$. В качестве примера рассмотрим систему $P(10,4,1)$. В ней содержится 19999 чисел, причем все соседние числа ***равноудалены*** друг от друга: –999.9, – 999.8, ..., 999.8, 999.9 . Любое вещественное число из интервала $(-1000; 1000)$ может быть представлено в этой системе элементом $fix(x) \in P(10,4,1)$ с абсолютной погрешностью, не превосходящей 0.05. Например, если $x = 864.54$, то его представление $fix(x)$ в системе $P(10,4,1)$ равно 865.5, а абсолютная погрешность такого представления равна 0.04. Соответственно, относительная погрешность этого представления равна $\frac{0.04}{865.5} \approx 0.0005$ (или 0.005%). С другой стороны, если $x = 0.86554$, то $fix(x) = 000.9$ и относительная и относительная ошибка составит уже около 4%. Это означает, что хотя числа системы $P(10,4,1)$ и образуют равномерную сетку на интервале $(-1000;1000)$, но ***относительная плотность*** этой сетки не яв-

ляется равномерной. Данный недостаток присущ всем системам представления чисел с фиксированной точкой.

Большинство компьютеров используют системы $F(\beta,t,L,U)$ представления чисел с ***плавающей точкой***, которые характеризуются четырьмя параметрами:

β – основание системы счисления, используемой компьютером;

t – количество цифр (разрядов), отведенных для представления мантиссы числа;

L, U – пределы изменения значений показателей степеней чисел (порядка).

Любое ненулевое число $x \in F$ имеет вид

$$x = \pm d_0.d_1d_2...d_{t-1} \times \beta^l,\ 0 \le d_i < \beta\ .$$

Если $d_0 \neq 0$, то такое число с плавающей точкой называется нормализованным. Числа с плавающей точкой, принадлежащие любой системе F, образуют неравномерную сетку, относительная плотность которой является равномерной. Наряду с указанными выше четырьмя параметрами β,t,L,U, однозначно определяющих систему F, на практике широко используются еще три параметра σ, λ и ε, которые выражаются через четыре основных параметра и имеют смысл соответственно наименьшего положительного числа с плавающей точкой данной системы, наибольшего положительного числа и ***машинного эпсилон*** (последнее, по сути, это расстояние между числом 1 и следующим за ним числом).

Отметим, что арифметические операции (сложение и умножение) над числами из F при правильном округлении коммутативны, однако для них не выполняются законы ассоциативности и дистрибутивности.

§ 4. О вычислительной погрешности

Ограничение на порядки чисел, представимых в компьютере иногда приводит к прекращению вычислений. В других же случаях относительно небольшая разрядность чисел, представимых в данной системе, приводит к недопустимому искажению результатов вычислительной погрешностью. Такие алгоритмы, где вследствие ограниченности порядка или малости t возникают подобные эффекты, называют «неустойчивыми». Построение «устойчивых» алгоритмов составляет существенную часть теории численных методов.

Рассмотрим несколько примеров.

1. Пусть отыскивается наименьший корень уравнения $y^2 - 140y + 1 = 0$. Для определенности условимся о следующих правилах округления. Вычисления производятся в десятичной системе счисления, причем в мантиссе числа после округлений удерживается 4 разряда. Имеем:

$$y = 70 - \sqrt{4899},\ \sqrt{4899} = 69.992...;$$

после округления получаем

$$\sqrt{4899} \approx 69.99,\ y \approx 70 - 69.99 = 0.0\underline{1}.$$

То же самое значение y можно, «избавившись от иррациональности в числителе», представить в виде

$$y = \frac{1}{70 + \sqrt{4899}}\ .$$

Последовательно производя вычисления, после окончательного округления получим: $y \approx 0.00\underline{7143}$ (все подчеркнутые цифры - верные). Таким образом, в условиях той же арифметики получен гораздо лучший результат.

2. Пусть требуется вычислить интеграл

$$I_n = \int_0^1 x^n e^{x-1} dx,\ n = 1,2,...$$

Путем интегрирования по частям для этих целей легко получить рекуррентную формулу

$$I_n = 1 - nI_{n-1},\ n = 2,3,...,\ I_1 = \frac{1}{e}.$$

Вычисления, проведенные в десятичной арифметике с $t = 6$, показывают (проделайте!), что уже I_9 становится отрицательным, чего, очевидно, быть не должно. Причиной такого накопления ошибок является то, что первоначальная ошибка в I_1, возникшая при представлении числа e и приблизительно равная 4.412×10^{-7}, при вычислении I_n умножается на $|n!|$. В частности, при $n = 9$ ошибка в I_9 примерно составит $4/412 \times 10^{-7} \times 9! \approx 0.1601$ и оказывается почти равной истинному значению I_9. Таким образом, выбранный алгоритм оказался неустойчивым.

Если же его переписать в виде

$$I_{n-1} = \frac{1 - I_n}{n}, \quad n = ...,3,2,$$

то легко видеть, что на n-м шаге ошибка не увеличивается, а уменьшается в n раз, т.е. алгоритм оказывается устойчивым. Недостаток его состоит, очевидно, в том, что нам неизвестно начальное приближение. Однако как бы грубо мы ни выбирали начальное приближение при $n >> 1$, начальная и промежуточные ошибки округления будут быстро уменьшаться на каждом шаге. Так как $I_n \le \frac{1}{n+1}$, то, положив, например, $I_{20} \approx 0$, получим: начальная ошибка не превосходит $\frac{1}{21}$. При вычислении I_{19} она умножится на $\frac{1}{20}$, т.е. станет равной примерно 0.0024. Когда мы дойдем до I_{15}, ошибка станет меньше 4×10^{-8}, т.е. станет меньше единственной ошибки округления. Поэтому, начиная с I_{15} значения I_n верны во всех шести знаках с точностью до возможной ошибки округления в последнем знаке.

§ 5. Прямая и обратная задачи теории погрешностей

Довольно часто возникает следующая задача. Искомая величина y является функцией от параметров $a_1, a_2, ..., a_n$: $y = f(a_1, a_2, ..., a_n)$. Известна область G в пространстве переменных $a_1, a_2, ..., a_n$, которой принадлежат эти параметры. Требуется получить приближение к y и оценить его погрешность.

Если y^* – приближенное значение величины y, то ***предельной абсолютной погрешностью*** $A(y^*)$ называют наилучшую при имеющейся информации оценку абсолютной погрешности величины y^*, т.е.

$$A(y^*) = \sup_{(a_1, a_2, ..., a_n) \in G} |f(a_1, a_2, ..., a_n) - y^*| \qquad (5.1).$$

Соответственно, ***предельной относительной погрешностью*** называют величину $\frac{A(y^*)}{|y^*|}$.

Рассмотрим наиболее распространенный случай, когда область G – прямоугольник:

$$|a_j - a_j^*| \le \Delta(a_j^*), \quad j = 1,...,n,$$

и за приближенное значение принимается величина

$$y^* = f(a_1^*, a_2^*, ..., a_n^*).$$

Если f – непрерывно дифференцируемая функция своих аргументов, то, согласно формуле Лагранжа,

$$f(a_1, a_2, ..., a_n) - y^* = \sum_{j=1}^{n} \frac{\partial f(a_1^* + \theta(a_1 - a_1^*), ..., a_n^* + \theta(a_n - a_n^*))}{\partial a_j} (a_j - a_j^*), \ 0 \le \theta \le 1. \qquad (5.2)$$

Отсюда следует оценка погрешности

$$|f(a_1, a_2, ..., a_n) - y^*| \le A_0(y^*) = \sum_{j=1}^{n} B_j \Delta(a_j^*), \qquad (5.3)$$

где

$$B_j = \sup_G \left| \frac{\partial f(a_1, a_2, ..., a_n)}{\partial a_j} \right|.$$

Положим

$$\rho = \sqrt{\sum_{j=1}^{n} \left(\Delta\left(a_j^*\right)\right)^2} \ .$$

Тогда, если частные производные функции f непрерывны, то

$$B_j = \left| \frac{\partial f\left(a_1^*, a_2^*, ..., a_n^*\right)}{\partial a_j} \right| + o(1) .$$

Следовательно, $A_0\left(y^*\right) = A^0\left(y^*\right) + o(\rho)$, где

$$A^0\left(y^*\right) = \sum_{j=1}^{n} \left| \frac{\partial f\left(a_1^*, a_2^*, ..., a_n^*\right)}{\partial a_j} \right| \Delta\left(a_j^*\right).$$

При практической работе вместо оценки (5.3) обычно пользуются более простой, вообще говоря неверной, «оценкой»

$$\left| f(a_1, a_2, ..., a_n) - y^* \right| \le A^0\left(y^*\right), \tag{5.4}$$

называемой ***линейной оценкой погрешности***.

Часто также приходится решать обратную задачу: с какой точностью надо задать значения аргументов $a_1^*, a_2^*, ...a_n^*$ функции $y = f(a_1, a_2, ..., a_n)$, чтобы погрешность приближенного значения $y^* = f\left(a_1^*, a_2^*, ..., a_n^*\right)$ не превосходила заданной величины ε?

Пусть относительно истинных и приближенных значений параметров справедливы предположения, сформулированные выше. Тогда в силу (5.3) имеем оценку погрешности

$$\left| f(a_1, a_2, ..., a_n) - y^* \right| \le \sum_{j=1}^{n} B_j \Delta\left(a_j^*\right). \tag{5.5}$$

Поэтому любая совокупность $\left(\Delta\left(a_1^*\right), ..., \Delta\left(a_n^*\right)\right)$ абсолютных погрешностей, удовлетворяющих неравенству

$$\sum_{j=1}^{n} B_j \Delta\left(a_j^*\right) \le \varepsilon ,$$

обеспечивает требуемую точность.

Если функция f зависит только от одного аргумента $(n = 1)$, то для достижения заданной точности достаточно взять $\Delta\left(a_1^*\right) \le \dfrac{\varepsilon}{B_1}$.

В случае $n > 1$ достаточно часто рекомендуют использовать следующие предположения:
1) принцип «равных влияний» (каждое слагаемое вносит равный вклад в правую часть оценки (5.5)). В этом случае соответствующие величины могут быть найдены по формулам

$$\Delta\left(a_j^*\right) \le \frac{\varepsilon}{nB_j}, \ \ j = 1, ...n ;$$

2) принцип «равных погрешностей». В этом случае предполагается, что все абсолютные погрешности аргументов равны между собой. Поэтому

$$\Delta\left(a_j^*\right) \le \frac{\varepsilon}{\sum_{k=1}^{n} B_k}, \ \ j = 1, ..., n .$$

В простейших случаях можно пользоваться сформулированными рецептами. Однако в более сложных случаях целесообразно подойти к вопросу более аккуратно. Например, можно ввести в рассмотрение функцию стоимости $F\left(\Delta\left(a_1^*\right), ..., \Delta\left(a_n^*\right)\right)$ затрат на задание точки $\left(a_1^*, ...a_n^*\right)$ с заданными абсолютными погрешностями координат $\Delta\left(a_1^*\right), ..., \Delta\left(a_n^*\right)$ и найти ее минимум в области $\sum_{j=1}^{n} B_j \Delta\left(a_j^*\right) \le \varepsilon, \ \Delta\left(a_j^*\right) \ge 0, \ \ j = 1, ..., n$.

РАЗДЕЛ I

Решение систем линейных алгебраических уравнений

Итак, более подробно поговорим о решении первой из задач курса «Вычислительные методы алгебры», сформулированных выше.

При формальном подходе к делу решение этой задачи не встречает трудностей: решение системы линейных алгебраических уравнений можно найти, например, пользуясь формулами Крамера (если, конечно, матрица системы – квадратная, а именно такие системы мы и будем рассматривать, как правило, если не оговорено противное, в дальнейшем). Но какова цена такого решения? При использовании правил Крамера необходимо вычислить $n+1$ определитель. Для этого, в свою очередь, нужно выполнить (если раскрывать определитель непосредственно по его определению) $n!(n-1)$ умножений. Следовательно, всего необходимо $n!\left(n^2-1\right)$ умножений. Сюда следует добавить еще n делений непосредственно для нахождения решений. Таким образом, всего для нахождения решения системы размерности $n\times n$ требуется $q_n = n!\left(n^2-1\right)+n$ операций умножения и деления. Уже при не очень больших n это число становится катастрофически большим. Например, при $n=30$ $q_n \sim 10^{35}$. Такой объем вычислений самым современным компьютерам не под силу. (максимальная производительность их достигает 10^{16} *flops* и, следовательно, время решения нашей задачи составит 10^{19} *сек* $\sim 2.5\cdot 10^{11}$ *лет*). При этом, помимо всего прочего, на результатах могут очень сильно сказаться ошибки округления. Все это вместе взятое говорит о том, что в вычислительной математике в изложенном виде правило Крамера использоваться не должно (как, впрочем, и к любым другим «очевидным» методам следует подходить с изрядной долей осторожности).

Методы решения систем линейных алгебраических уравнений можно разделить на три группы:

1) точные (или прямые);
2) итерационные;
3) вероятностные.

Метод решения систем линейных алгебраических уравнений относят к классу точных, если он позволяет получить решение любой системы в ***точном*** виде после конечного числа операций, если эти операции выполнять без округлений (т.е. фактически речь идет об отсутствии погрешности метода).

В случае итерационных методов точное решение можно получить лишь в пределе некоторого бесконечного процесса. Особое место здесь занимают вероятностные методы (типа методов Монте-Карло).

Классы задач, для решения которых обычно применяются методы этих групп можно условно назвать соответственно классами задач с малым, средним и большим числом неизвестных. Изменение объема и структуры памяти компьютеров, увеличение их быстродействия и развитие численных методов приводят к смещению границ применения методов в сторону систем более высоких порядков. В настоящее время точные методы обычно применяются при решении систем до порядка 10^4-10^5, итерационные – до порядка 10^7-10^9. Большее же число неизвестных заставляет обращаться к методам типа Монте-Карло.

ГЛАВА I

Прямые методы решения систем линейных алгебраических уравнений

Пусть задана система линейных алгебраических уравнений

$$Ax = f\,, \tag{1}$$

где A – $n \times n$ -матрица коэффициентов системы,

$x = (x_1, x_2, ..., x_n)^T$ – вектор-столбец неизвестных,

$f = (f_1, f_2, ..., f_n)^T$ – заданный вектор-столбец правых частей (свободных членов).

Общая идеология всех прямых методов решения линейных систем состоит в преобразовании этой системы к некоторому более простому виду

$$A_k x = f_k\,. \tag{2}$$

Это преобразование осуществляется обычно путем последовательного умножения слева исходной системы (1) на некоторые невырожденные матрицы $L_1, L_2, ..., L_k$, т.е.

$$A_k = L_k L_{k-1} ... L_1 A, \;\; f_k = L_k L_{k-1} ... L_1 f\,. \tag{3}$$

Цель подобных преобразований – добиться того, чтобы система (2) легко решалась, например, легко находилась обратная матрица. Тогда

$$x = A_k^{-1} f_k\,.$$

Такая ситуация очевидным образом реализуется, скажем, в случае, если матрица A_k является треугольной. Если же A_k удается сделать диагональной, то система (2) решается еще проще.

Достаточно часто преобразованиями вида (3) матрицу A приводят к двух- или трехдиагональному виду, а иногда добиваются того, чтобы матрица A_k была унитарной, т.е. $A_k A_k^* = E$, где E – единичная матрица соответствующей размерности. Тогда, очевидно, $A_k^{-1} = A_k^*$.

На основании формул (3) легко указать вид матрицы A через A_k и $L_1, L_2, ..., L_k$:

$$A = L_1^{-1} ... L_k^{-1} A_k\,.$$

Данное соотношение может быть использовано для решения задачи о вычислении определителя матрицы A и нахождения обратной к ней. Это может быть реализовано с использованием следующих формул:

$$\begin{cases} A^{-1} = A_k^{-1} L_k ... L_1\,, \\ \det A = |A| = \dfrac{|A_k|}{|L_1| \cdot |L_2| \cdot \ldots \cdot |L_k|}\,. \end{cases}$$

Перейдем сейчас к рассмотрению конкретных примеров такого сорта алгоритмов.

§ 1. Метод Гаусса

Запишем исходную систему (1) в развернутом виде (покоординатно):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = f_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = f_2, \\ \cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = f_n. \end{cases} \tag{1.1}$$

В методе Гаусса обычно на первом шаге выбирают некоторое уравнение и некоторое неизвестное в этом уравнении, считая, что соответствующий коэффициент при выбранном неизвестном отличен от нуля. Не нарушая общности изложения, предположим, что выбрано первое уравнение и первое (т.е. x_1) неизвестное в нем ($a_{11} \neq 0$). Разделив на a_{11} первое (выбранное) уравнение, преобразуем его к виду

$$x_1 + b_{12}x_2 + \cdots + b_{1n}x_n = g_1, \tag{1.2}$$

где

$$b_{1j} = \frac{a_{1j}}{a_{11}}, \quad j = 2,3,...,n, \quad g_1 = \frac{f_1}{a_{11}}.$$

После этого исключим неизвестное x_1 из всех остальных уравнений системы (1.1). Для этого умножим уравнение (1.2) на величину a_{i1} и вычтем из i-го уравнения (1.1). В итоге получим систему

$$\begin{cases} x_1 + b_{12}x_2 \quad + \cdots + b_{1n}x_n = g_1, \\ \quad a_{22,1}x_2 + \cdots + a_{2n,1}x_n = f_{2,1}, \quad a_{ij,1} = a_{ij} - a_{i1}b_{1j}, \\ \quad a_{32,1}x_2 + \cdots + a_{3n,1}x_n = f_{3,1}, \quad i,j = \overline{2,n}, \\ \qquad\qquad \cdots\cdots \qquad\qquad f_{i,1} = f_i - a_{i1}g_1, \\ \quad a_{n2,1}x_2 + \cdots + a_{nn,1}x_n = f_{n,1}, \quad i = \overline{2,n}. \end{cases} \tag{1.3}$$

В последних $(n-1)$ уравнениях системы (1.3) не содержится неизвестное x_1 и их можно решать отдельно. К этой подсистеме размерности $(n-1)\times(n-1)$ также можно применить описанное преобразование:

1) выбрать уравнение и неизвестное в нем с отличным от нуля коэффициентом (далее этот коэффициент будем называть ***ведущим элементом***);
2) разделить выбранное уравнение на ведущий элемент;
3) исключить выбранное неизвестное из оставшихся уравнений.

Этот процесс можно повторять до тех пор, пока либо не будут исчерпаны все уравнения системы, либо больше не будет отличных от нуля коэффициентов.

Пусть нам удалось выполнить m шагов $(m \le n)$, причем всякий раз выбирали в качестве ведущего элемента элемент, стоящий в верхнем левом углу текущей подматрицы. Возможны два варианта:

1. $m = n$. Тогда после окончания описанного процесса исходная система будем преобразована к виду

$$\begin{cases} x_1 + b_{12}x_2 + b_{13}x_3 + \cdots + b_{1n-1}x_{n-1} + b_{1n}x_n = g_1, \\ \qquad x_2 + b_{23}x_3 + \cdots + b_{2n-1}x_{n-1} + b_{2n}x_n = g_2, \\ \qquad\qquad \cdots\cdots \\ \qquad\qquad\qquad\qquad x_{n-1} + b_{n-1n}x_n = g_{n-1}, \\ \qquad\qquad\qquad\qquad\qquad x_n = g_n. \end{cases} \tag{1.4}$$

Матрица данной системы имеет треугольный вид и, следовательно, решение системы легко может быть найдено. Для этого из последнего уравнения получаем значение неизвестного x_n, подставляя его в предпоследнее уравнение, находим x_{n-1}, подставляя оба этих неизвестных в третье с конца уравнение, находим x_{n-2} и т.д.:

$$\begin{cases} x_n = g_n, \\ x_k = g_k - b_{kk+1}x_{k+1} - \cdots - b_{kn}x_n, \quad k = n-1,...,1. \end{cases} \tag{1.5}$$

Весь описанный процесс при отмеченных условиях (ведущим всегда выбирался элемент из верхнего левого угла текущей подматрицы) носит название ***схемы единственного деления***. При этом преобразование системы (1.1) к треугольному виду (1.4) называют ***прямым ходом*** метода Гаусса (формулы типа (1.2), (1.3)), а нахождение неизвестных по формулам (1.5) – ***обратным***.

2. $m < n$. Тогда вместо (1.4) система (1.1) в результате преобразований примет вид

$$\begin{cases} x_1 + b_{12}x_2 + b_{13}x_3 + \cdots + b_{1n}x_n = g_1, \\ \qquad x_2 + b_{23}x_3 + \cdots + b_{2n}x_n = g_2, \\ \qquad\qquad \cdots\cdots \\ \qquad x_m + b_{mm+1}x_{m+1} + \cdots + b_{mn}x_n = g_m, \\ \qquad\qquad\qquad\qquad 0 = f_{m+1m}, \\ \qquad\qquad\qquad \cdots\cdots \\ \qquad\qquad\qquad\qquad 0 = f_{nm}. \end{cases} \tag{1.6}$$

Если среди чисел $f_{m+1m},...,f_{nm}$ встретится хотя бы одно, отличное от нуля, то наша система несовместна, а следовательно, несовместна и исходная система (1.1), поскольку мы проводили лишь элементарные преобразования.

Если же все эти числа равны нулю, то наша система приводится к первым m уравнениям. Если теперь задать неизвестные $x_{m+1},...,x_n$ произвольно, то оставшиеся m неизвестных могут быть найдены единственным образом из системы с треугольной матрицей.

Таким образом, подведем краткий итог: в схеме единственного деления (на k-м шаге) ведущий элемент – $a_{kk,k-1}$. Поэтому окончательные формулы для выполнения прямого хода будут выглядеть следующим образом:

$$\begin{cases} a_{kj,0} = a_{kj}, \quad k, j = \overline{1,n}; \\ b_{kj} = \dfrac{a_{kj,k-1}}{a_{kk,k-1}}, \quad j = k+1,...,n; \ k = 1,...,n-1; \\ a_{ij,k} = a_{ij,k-1} - a_{ik,k-1} \cdot b_{kj}, \quad i, j = k+1,...,n; \ k = 1,...,n-1; \end{cases} \tag{1.7}$$

$$\begin{cases} f_{k,0} = f_k, \quad g_k = \dfrac{f_{k,k-1}}{a_{kk,k-1}}, \quad k = 1,...,n; \\ f_{i,k} = f_{i,k-1} - a_{ik,k-1} \cdot g_k, \quad i = k+1,...,n; \quad k = 1,...,n-1. \end{cases} \tag{1.8}$$

Обратный ход осуществляется по формулам (1.5).

Элементарный подсчет показывает (проведите (!)), что число умножений и делений, необходимых для нахождения решения системы из n уравнений по схеме единственного деления составляет величину порядка $\frac{1}{3}n^3$.

Установим теперь связь между изложенным вариантом метода Гаусса и формулами (2), (3) общей схемы.

Легко видеть, что первый этап k-го шага прямого хода (деление k-го уравнения на ведущий элемент) равносильно умножению системы уравнений слева на диагональную матрицу

$$C_k = \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ & \ddots & & & & & \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & a_{kk,k-1}^{-1} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ & & & & & \ddots & \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{1.9}$$

а второй этап – соответственно умножению слева на матрицу

$$C'_k = \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ & \ddots & & & & & \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -a_{k+1k,k-1} & 1 & \cdots & 0 \\ & & & \cdots & & \ddots & \\ 0 & \cdots & 0 & -a_{nk,k-1} & 0 & \cdots & 1 \end{bmatrix}. \tag{1.10}$$

Таким образом, исходная система (1) в результате этих преобразований запишется в виде (2), где $L_i = C'_i C_i$, $i = \overline{1,n}$, а C_i, C'_i определяются формулами (1.9), (1.10), т.е. описанная схема алгоритма фактически эквивалентна разложению матрицы исходной системы линейных алгебраических уравнений в произведение двух треугольных матриц (нижней (Low) и верхней (Up)) и дальнейшему решению двух систем с треугольными матрицами (отсюда – деление на прямой и обратный ход метода). Другими словами, если матрица A представима в виде

$$A = LU, \tag{1.11}$$

где L – нижняя треугольная, а U – верхняя треугольная, то решение исходной линейной системы (1) сводится к решению двух систем с треугольными матрицами

$$\begin{aligned} Ly &= f, \\ Ux &= y. \end{aligned} \tag{1.12}$$

Дадим обоснование возможности разложения (1.11). Обозначим через Δ_j угловой минор порядка j матрицы A, т.е.

$$\Delta_1 = a_{11},\ \Delta_2 = \begin{vmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{vmatrix}, ..., \Delta_n = \det A .$$

Тогда имеет место

***Теорема* 1** (об LU-разложении). Пусть все угловые миноры матрицы A отличны от нуля, т.е. $\Delta_j \neq 0,\ j = \overline{1,n}$. Тогда матрицу A можно представить, причем единственным образом, в виде (1.11), где L – нижняя треугольная матрица с ненулевыми диагональными элементами, а U – верхняя треугольная матрица с единичной диагональю.

Доказательство проведем методом математической индукции.

Вначале рассмотрим случай $n = 2$. Будем искать разложение матрицы A в виде

$$A = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{12} & l_{22} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} \\ 0 & 1 \end{bmatrix},$$

где l_{11}, l_{21}, l_{22}, u_{12} – неизвестные пока числа. Выполнив перемножение матриц и приравняв элементы произведения к соответствующим элементам матрицы A, для нахождения этих чисел придем к системе уравнений

$$\begin{cases} l_{11} = a_{11}, \\ l_{11}u_{12} = a_{12}, \\ l_{21} = a_{21}, \\ l_{21}u_{12} + l_{22} = a_{22}, \end{cases}$$

Которая имеет единственное решение:

$$l_{11} = a_{11},\ u_{12} = \frac{a_{12}}{l_{11}},\ l_{21} = a_{21},\ l_{22} = a_{22} - a_{21} \cdot \frac{a_{12}}{a_{11}} = \frac{\Delta_2}{\Delta_1}.$$

По предположению $\Delta_1 \neq 0$ и $\Delta_2 \neq 0$. Поэтому диагональные элементы матрицы L (т.е. l_{11} и l_{22}) отличны от нуля. Таким образом, в данном случае утверждение теоремы имеет место.

Пусть теперь утверждение теоремы справедливо для матриц порядка $k-1$. Докажем его справедливость для матриц порядка k. Представим матрицу A_k порядка k в виде

$$A_k = \begin{bmatrix} a_{11} & \cdots & a_{1k-1} & a_{1k} \\ & \cdots & & \\ a_{k-11} & \cdots & a_{k-1k-1} & a_{k-1k} \\ a_{k1} & \cdots & a_{kk-1} & a_{kk} \end{bmatrix} = \begin{bmatrix} A_{k-1} & a_{k-1} \\ b_{k-1} & a_{kk} \end{bmatrix},$$

где

$$A_{k-1}=\begin{bmatrix} a_{11} & \cdots & a_{1k-1} \\ & \cdots & \\ a_{k-11} & \cdots & a_{k-1k-1} \end{bmatrix},\ a_{k-1}=(a_{1k},...,a_{k-1k})^T,\ b_{k-1}=(a_{k1},...,a_{kk-1}).$$

Согласно предположению для матрицы порядка $k-1$ A_{k-1} существует разложение (1.11), т.е.

$$A_{k-1}=L_{k-1}U_{k-1}.$$

Разложение для матрицы A_k будем искать в виде

$$A_k=\begin{bmatrix} L_{k-1} & 0 \\ l_{k-1} & l_{kk} \end{bmatrix}\cdot\begin{bmatrix} U_{k-1} & u_{k-1} \\ 0 & 1 \end{bmatrix}, \qquad (1.13)$$

где $l_{k-1}=(l_{k1},...,l_{kk-1})$, $u_{k-1}=(u_{1k},...,u_{k-1k})^T$, l_{kk} – неизвестные пока величины.

Вновь выполняя перемножение (блочных (!)) матриц, получим:

$$\begin{cases} L_{k-1}U_{k-1}=A_{k-1}, \\ L_{k-1}u_{k-1}=a_{k-1}, \\ l_{k-1}U_{k-1}=b_{k-1}, \\ l_{k-1}u_{k-1}+l_{kk}=a_{kk}. \end{cases}$$

Первое из равенств системы выполняется тождественно в силу индуктивного предположения, а из оставшихся, учитывая существование матриц L_{k-1}^{-1} и U_{k-1}^{-1}, найдем:

$$u_{k-1}=L_{k-1}^{-1}a_{k-1},\ l_{k-1}=b_{k-1}U_{k-1}^{-1},\ l_{kk}=a_{kk}-l_{k-1}u_{k-1}.$$

Таким образом, LU -разложение матрицы A_k существует.

Докажем, что $l_{kk}\neq 0$. Из (1.13) следует (по правилу вычисления определителя произведения квадратных матриц), что

$$\det A_k=\det L_{k-1}\cdot l_{kk}\cdot\det U_{k-1}\cdot 1=\det L_{k-1}\cdot l_{kk}.$$

В силу условия теоремы $\det A_k=\Delta_k\neq 0$, поэтому $l_{kk}\neq 0$.

Осталось установить единственность разложения. Это мы сделаем методом от противного. Пусть существует два разложения:

$$A=L_1U_1=L_2U_2.$$

Отсюда, учитывая невырожденность сомножителей, последовательно находим:

$$L_2=L_1U_1U_2^{-1}\ \text{ и }\ L_1^{-1}L_2=U_1U_2^{-1}.$$

Матрица в левой части последнего равенства является нижней треугольной, а в правой – верхней треугольной (как произведения двух нижних и двух верхних треугольных соответственно). Поэтому равенство возможно лишь в том случае, когда оба этих произ-

ведения являются диагональными матрицами. Но на диагонали произведения $U_1U_2^{-1}$ стоят единицы. Следовательно, $U_1U_2^{-1} = L_1^{-1}L_2 = E$, откуда следуют равенства $U_1 = U_2$ и $L_1 = L_2$.

$\boxtimes$

Замечание. На диагонали матрицы L будут стоять ведущие элементы прямого хода, т.е. $l_{kk} = a_{kk,k-1}$ (показать!).

1.1. Метод Гаусса с выбором главного элемента

Возможность проведения процесса исключения в схеме единственного деления, изложенной выше, как это следует из доказанной теоремы, гарантируется условиями $\Delta_j \neq 0,\ j = \overline{1,n}$.

Однако при проведении расчетов заранее неизвестно, все ли угловые миноры отличны от нуля. При этом может оказаться, что система (1.1) имеет единственное решение, несмотря на то, что какой-либо из Δ_j равен нулю. Кроме того, фиксация ведущего элемента в случае его относительной малости может привести в процессе вычислений к сильному накоплению погрешностей. Избежать этих ситуаций (или по крайней мере сильно смягчить некоторые их последствия) позволяет ***метод Гаусса с выбором главного элемента***. Основная идея метода заключается в том, что в качестве ведущего элемента на каждом этапе исключения выбирается наибольший по модулю (главный) элемент. На практике обычно используются следующие варианты метода Гаусса с выбором главного элемента:

а) Метод Гаусса с выбором главного элемента по строке. Он эквивалентен применению схемы единственного деления к системе, в которой на каждом этапе проводится перенумерация переменных;

б) Метод Гаусса с выбором главного элемента по столбцу. Он эквивалентен применению схемы единственного деления к системе, в которой на каждом этапе проводится перенумерация уравнений;

в) Метод Гаусса с выбором главного элемента по всей подматрице. Он эквивалентен применению схемы единственного деления к системе, в которой на каждом этапе проводится соответствующая перенумерация и переменных, и уравнений.

Заметим, что с вычислительной точки зрения наиболее экономичен вариант б), так как а) и в) требуют для своей реализации дополнительного вектора, в котором хранится информация о выполненных переименованиях неизвестных, а кроме того, в варианте в) требуется поиск максимального по модулю элемента в массиве размерности $(n-k)\times(n-k),\ k = \overline{0,n-1}$, что также приводит к большим вычислительным затратам.

Получим сейчас матричные аналоги сформулированных алгоритмов, т.е. покажем, что они укладываются в общую схему (2) – (3).

Определение* 1. *Матрицей перестановок P называется квадратная матрица, в каждой строке и в каждом столбце которой только один элемент отличен от нуля и этот элемент равен единице.

Определение* 2. *Элементарной матрицей перестановок P_{km} назовем матрицу, полученную из единичной перестановкой k-й и m-й строк.

Примерами таких матриц являются, скажем, матрицы

$$P_{12} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad P_{24} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Отметим сейчас некоторые достаточно очевидные свойства определенных элементарных матриц перестановок:

1. Произведение любого числа элементарных матриц перестановок является матрицей перестановок (не обязательно элементарной);
2. Матрица $P_{km}A$ отличается от матрицы A перестановкой k-й и m-й строк;
3. Матрица AP_{km} отличается от матрицы A перестановкой k-го и m-го столбцов.

Упражнения.

1. Доказать сформулированные выше свойства элементарных матриц перестановок;
2. Вычислить определитель матрицы P_{km}.

С учетом введенных определений рассмотрим далее более подробно вариант б) метода Гаусса с выбором главного элемента. Легко видеть, что данный вариант алгоритма можно записать в виде

$$L_n L_{n-1} P_{n-1} L_{n-2} P_{n-2} \dots L_2 P_2 L_1 P_1 Ax = L_n L_{n-1} P_{n-1} L_{n-2} P_{n-2} \dots L_2 P_2 L_1 P_1 f\,, \qquad (1.14)$$

где P_k – элементарная матрица перестановок, т.е. $P_k = P_{km}$, $k < m \le n$, а L_k – элементарная треугольная матрица, рассмотренная нами ранее ($L_k = C_k' C_k$, см. (1.9), (1.10)).

Используя соотношения (здесь учтено еще одно свойство элементарных матриц перестановок: $P_{km} \cdot P_{km} = E$ (доказать!))

$$P_k L_{k-i}^{(i-1)} = P_k L_{k-i}^{(i-1)} P_k P_k = L_{k-i}^{(i)} P_k\,,\ k = n-1,...,i+1;\ i = 1,...,n-2;\ \ L_k^{(0)} = L_k\,,$$

где $L_{k-i}^{(i)} = P_k L_{k-i}^{(i-1)} P_k$ (таким образом, матрица $L_{k-i}^{(i)}$ – нижняя треугольная, имеющая обратную, так как де факто матрица $L_{k-i}^{(i)}$ отличается от матрицы $L_{k-i}^{(i-1)}$ перестановкой всего двух элементов в $(k-i)$-м столбце, причем ниже диагонали), из (1.14) получим:

$$L_n L_{n-1} L_{n-2}^{(1)} L_{n-3}^{(2)} \dots L_2^{(n-3)} L_1^{(n-2)} PAx = L_n L_{n-1} L_{n-2}^{(1)} L_{n-3}^{(2)} \dots L_2^{(n-3)} L_1^{(n-2)} Pf\,. \qquad (1.15)$$

Здесь $P = P_{n-1} P_{n-2} \dots P_1$ – результирующая матрица перестановок.

Таким образом, метод Гаусса с выбором главного элемента по столбцу можно записать в виде схемы единственного деления, примененной к системе

$$PAx = Pf\,,$$

где P, как указано выше, – некоторая результирующая матрица перестановок.

***Теорема* 2.** Если $\det A \ne 0$, то существует матрица перестановок P такая, что матрица PA имеет отличные от нуля угловые миноры.

Доказательство.

Вновь используем метод математической индукции.

Рассмотрим случай $n = 2$. Тогда

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

Если $a_{11} \ne 0$, то в этом случае теорема верна при $P = E$. Если же $a_{11} = 0$, то, поскольку $\det A \ne 0$, то $a_{21} \ne 0$. Тогда у матрицы

$$P_{12} A = \begin{bmatrix} a_{21} & a_{22} \\ a_{11} & a_{12} \end{bmatrix}$$

все угловые миноры отличны от нуля.

Пусть утверждение выполняется для любых квадратных матриц A_{k-1} порядка $k-1$. Покажем, что оно в таком случае имеет место и для матриц порядка k. Как и при доказательстве теоремы 1 представим матрицу A_k в виде

$$A_k = \begin{bmatrix} A_{k-1} & a_{k-1} \\ b_{k-1} & a_{kk} \end{bmatrix}.$$

Достаточно рассмотреть два случая:

1) $\det A_{k-1} \neq 0$.

Тогда по предположению индукции существует матрица перестановок P_{k-1} такая, что произведение $P_{k-1}A_{k-1}$ имеет отличные от нуля угловые миноры. Следовательно, для матрицы перестановок

$$P = \begin{bmatrix} P_{k-1} & 0 \\ 0 & 1 \end{bmatrix}$$

имеем:

$$PA_k = \begin{bmatrix} P_{k-1}A_{k-1} & P_{k-1}a_{k-1} \\ b_{k-1} & a_{kk} \end{bmatrix},$$

причем

$$\det(PA_k) = \pm \det A_k \neq 0.$$

Тем самым показано, что все угловые миноры матрицы PA_k отличны от нуля.

2) $\det A_{k-1} = 0$.

Так как $\det A_k \neq 0$, то найдется хотя бы один отличный от нуля минор порядка $k-1$ матрицы A_k, полученный вычеркиванием последнего столбца и какой-либо строки. Пусть, например,

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k-1} \\ \cdots & & \cdots & \cdots \\ a_{l-11} & a_{l-12} & \cdots & a_{l-1k-1} \\ a_{l+11} & a_{l+12} & \cdots & a_{l+1k-1} \\ \cdots & & \cdots & \cdots \\ a_{k1} & a_{k2} & \cdots & a_{kk-1} \end{vmatrix} \neq 0, \qquad (1.16)$$

где $l \neq k$.

Переставляя в матрице A_k строки с номерами l и k, получим матрицу $P_{lk}A$, у которой угловой минор порядка $k-1$ имеет вид

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k-1} \\ & \cdots & \cdots & \\ a_{l-11} & a_{l-12} & \cdots & a_{l-1k-1} \\ a_{k1} & a_{k2} & \cdots & a_{kk-1} \\ a_{l+11} & a_{l+12} & \cdots & a_{l+1k-1} \\ & \cdots & \cdots & \\ a_{k-11} & a_{k-12} & \cdots & a_{k-1k-1} \end{vmatrix}$$

и отличен от нуля, так как отличается от минора (1.16) только перестановкой строк. Следовательно, приходим к предыдущему случаю.

Таким образом, теорема полностью доказана. ⊠

Следствие. Если $\det A \neq 0$, то существует матрица перестановок P такая, что справедливо разложение

$$PA = LU, \tag{1.17}$$

где L – нижняя треугольная матрица с отличными от нуля диагональными элементами, а U – верхняя треугольная матрица с единичной главной диагональю.

Упражнение. Исследовать подобным образом другие варианты метода Гаусса с выбором главного элемента.

1.2. Вычисление определителей и обращение матриц

Выполнение преобразований матрицы A в процессе решения системы (1) позволяет также решить и некоторые другие задачи, связанные с ней. Это, например, – задача вычисления определителя и задача обращения матрицы. При этом для вычисления определителя практически не требуется дополнительных вычислительных затрат. Действительно, пусть имеет место разложение (1.17). Тогда

$$\det(PA) = \det(LU) = \det L \cdot \det U = \det L = l_{11} \cdot l_{22} \cdot \ldots \cdot l_{nn},$$

т.е. произведение диагональных элементов матрицы L равно определителю матрицы PA. А поскольку матрицы PA и A отличаются только перестановкой строк, определитель матрицы PA может отличатся от определителя матрицы A только знаком. Поэтому

$$\det A = (-1)^m a_{11,0} \cdot a_{22,1} \cdot \ldots a_{nn,n-1},$$

где m – количество перестановок, осуществляемых в процессе исключения, $a_{kk,k-1}$, $k = \overline{1, n-1}$ – ведущие элементы метода Гаусса.

Аналогичным образом выглядят и формулы для вычисления определителя в других модификациях метода Гаусса с выбором главного элемента.

Задача нахождения матрицы, обратной матрице A, эквивалентна задаче решения матричного уравнения

$$AX = E, \tag{1.18}$$

где X – искомая матрица. Обозначив ее столбцы через $x^{(j)}$, $j = 1,...,n$, а столбцы единичной матрицы – через $\delta^{(j)}$, $j = 1,...,n$, систему (1.18) перепишем в виде

$$Ax^{(j)} = \delta^{(j)}, \quad j = 1,...,n, \tag{1.19}$$

$x^{(j)} = \left(x_{1j},...,x_{nj}\right)^T$, $\delta^{(j)} = \left(\delta_{1j},...,\delta_{nj}\right)^T$, δ_{ij} – символ Кронекера,

т.е. каждый вектор $x^{(j)}$ является решением системы вида (1.19), а для нахождения всей обратной матрицы необходимо решить n систем с одной и той же матрицей A в качестве матрицы системы.

Поэтому для решения поставленной задачи достаточно один раз совершить прямой ход метода Гаусса, т.е. получить разложение $A = LU$. Обратный ход тогда может быть осуществлен путем решения систем

$$\begin{cases} Ly^{(j)} = \delta^{(j)}, \\ Ux^{(j)} = y^{(j)}, \quad j = 1,...,n. \end{cases} \tag{1.20}$$

С другой стороны, безусловно, разумными будут попытки объединить процесс решения всех систем (1.20). Для этого достаточно выполнять преобразования, приводящие матрицу A к треугольному виду, над ***всеми*** правыми частями одновременно.

При этом для решения задачи обращения матрицы несколько выгоднее оказывается еще одна модификация метода Гаусса, которая в литературе носит название ***метода Гаусса – Жордана*** (или метода Жордана). Ее отличие от разобранного выше метода Гаусса состоит в том, что в результате выполнения прямого хода алгоритма матрица преобразуется не к треугольному, а к ***диагональному*** виду. Технически для этого необходимо при выполнении k-го шага процесса исключения неизвестных исключать неизвестное x_k не только из уравнений с номерами $k+1,...,n$ (как в штатном варианте), но и из уравнений с номерами $1,...,k-1$. Таким образом, необходимость в выполнении обратного хода в методе Гаусса-Жордана отпадает.

Схематически метод Гаусса-Жордана применительно к решению системы (1) (в варианте схемы единственного деления) может быть изображен в виде

$$(A \mid f) \sim \left(E \mid A^{-1} f\right),$$

а для нахождения обратной матрицы –

$$(A \mid E) \sim \left(E \mid A^{-1}\right).$$

Аналогично могут быть реализованы варианты алгоритма с выбором главного элемента.

§ 2. Метод квадратного корня

Заметим вначале, что приведенное в § 1 доказательство теоремы об LU-факторизации является конструктивным, так как по сути дела представляет собой алгоритм для построения разложения квадратной матрицы A в произведение двух треугольных.

Упражнение. Получить расчетные формулы для построения LU-разложения.

Достаточно часто этим алгоритмом пользуются вместо штатного алгоритма метода Гаусса (соответственно он называется методом LU-факторизации) (например, в случае необходимости решения нескольких систем с одной и той же матрицей, скажем, в задаче обращения матрицы).

Особенно выгодным становится использование описанной процедуры в случае, когда матрица A системы (1) является эрмитовой, т.е. выполняется равенство $A = A^*$ или $a_{ij} = \overline{a}_{ji}$. Тогда матрица A (мы по-прежнему считаем ее удовлетворяющей условиям Теоремы 1) может быть представлена в виде

$$A = S^* D S, \tag{2.1}$$

где S – верхняя треугольная матрица с положительными элементами на главной диагонали, S^* – матрица, комплексно-сопряженная с матрицей S, D – диагональная матрица с элементами ± 1 по диагонали.

Действительно, так как A удовлетворяет условиям Теоремы 1 об LU-факторизации, то имеет место разложение

$$A = LU,$$

где L – нижняя треугольная матрица, имеющая обратную, а U – верхняя треугольная с единичной диагональю.

Представим матрицу L в виде произведения

$$L = MK,$$

где M – нижняя треугольная матрица с единичной главной диагональю, а K – диагональная матрица, главная диагональ которой совпадает с главной диагональю матрицы L, т.е.

$$K = \operatorname{diag}(l_{11}, l_{22}, ..., l_{nn}). \tag{2.2}$$

Так как $l_{ii} \neq 0$, $i = \overline{1,n}$, то указанное представление существует. Следовательно, справедливо равенство

$$A = MKU, \tag{2.3}$$

где M и U – треугольные матрицы с единичной главной диагональю, и K – диагональная матрица, имеющая обратную.

Тогда из условия $A = A^*$ получаем:

$$MKU = U^*K^*M^*.$$

Поэтому

$$U\left(M^*\right)^{-1} = K^{-1}M^{-1}U^*K^*. \tag{2.4}$$

Матрица, находящаяся в левой части равенства (2.4), является верхней треугольной, а в правой – нижней треугольной. Поэтому из (2.4) следует, что обе матрицы ($U\left(M^*\right)^{-1}$ и $K^{-1}M^{-1}U^*K^*$) являются диагональными.

Далее, так как матрица $U\left(M^*\right)^{-1}$ имеет единичную главную диагональ, то она является единичной, т.е. $U\left(M^*\right)^{-1} = E$, откуда следует, что $U = M^*$.

Тогда равенство (2.3) примет вид

$$A = MKM^*. \tag{2.5}$$

Наконец, представим матрицу K, определяемую формулой (2.2), в виде

$$K = |K|^{\frac{1}{2}} D |K|^{\frac{1}{2}},$$

где

$$|K|^{\frac{1}{2}} \overset{def}{=} \operatorname{diag}\left(\sqrt{|l_{11}|}, ..., \sqrt{|l_{nn}|}\right),$$
$$D = \operatorname{diag}(\operatorname{sign} l_{11}, ..., \operatorname{sign} l_{nn}).$$

Заметим, что подобное представление возможно, поскольку определитель эрмитовой матрицы вещественен и $l_{ii} = \dfrac{\Delta_i}{\Delta_{i-1}}$, $i = \overline{2,n}$, $l_{11} = \Delta_1$ (доказать !).

Поэтому из (2.5) получим разложение (2.1), в котором $S = |K|^{\frac{1}{2}} M^*$ – верхняя треугольная матрица с положительными элементами по главной диагонали.

Получим теперь расчетные формулы метода решения систем линейных алгебраических уравнений (1) с эрмитовой матрицей, который носит название ***метода квадратного корня*** и в основе которого лежит разложение (2.1).

Итак, пусть $A = S^* D S$, где

$$S = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ 0 & s_{22} & \dots & s_{2n} \\ & \dots & \dots & \\ 0 & 0 & \dots & s_{nn} \end{pmatrix}, \quad D = \begin{pmatrix} d_{11} & 0 & \dots & 0 \\ 0 & d_{22} & \dots & 0 \\ & \dots & \dots & \\ 0 & 0 & \dots & d_{nn} \end{pmatrix},$$

и, следовательно,

$$S^* = \begin{pmatrix} s_{11} & 0 & \dots & 0 \\ \bar{s}_{12} & s_{22} & \dots & 0 \\ & \dots & \dots & \\ \bar{s}_{1n} & \bar{s}_{2n} & \dots & s_{nn} \end{pmatrix}.$$

Заметим, что, согласно доказанному выше, матрица S имеет ***положительные*** элементы по главной диагонали. Поэтому знак сопряжения над диагональными элементами матрицы S^* мы не ставим.

Последовательно раскрывая произведение $S^* D S$, получаем:

$$(DS)_{ij} = \sum_{k=1}^{n} d_{ik} s_{kj} = d_{ii} s_{ij},$$

$$\left(S^* DS\right)_{ij} = \sum_{k=1}^{n} \bar{s}_{ki} (DS)_{kj} = \sum_{k=1}^{n} \bar{s}_{ki} d_{kk} s_{kj} = \sum_{k=1}^{i-1} \bar{s}_{ki} d_{kk} s_{kj} + s_{ii} d_{ii} s_{ij} + \sum_{k=i+1}^{n} \bar{s}_{ki} d_{kk} s_{kj}.$$

Учитывая, что $\bar{s}_{ki} = 0$ при $k > i$, перепишем последнее равенство в виде

$$\left(S^* DS\right)_{ij} = s_{ii} d_{ii} s_{ij} + \sum_{k=1}^{i-1} \bar{s}_{ki} d_{kk} s_{kj}.$$

Отсюда и из (2.1) имеем систему уравнений:

$$s_{ii} s_{ij} d_{ii} + \sum_{k=1}^{i-1} \bar{s}_{ki} d_{kk} s_{kj} = a_{ij}, \quad i = 1,2,\dots,j; \quad j = \overline{1,n}. \tag{2.6}$$

Полагая в (2.6) $i = j$, получаем:

$$s_{ii}^2 d_{ii} = a_{ii} - \sum_{k=1}^{i-1} |s_{ki}|^2 d_{kk},$$

откуда

$$d_{ii} = \operatorname{sign}\left(a_{ii} - \sum_{k=1}^{i-1} |s_{ki}|^2 d_{kk} \right), \tag{2.7}$$

$$s_{ii} = \sqrt{\left| a_{ii} - \sum_{k=1}^{i-1} |s_{ki}|^2 d_{kk} \right|}, \tag{2.8}$$

Далее из той же формулы (2.6) при $i < j$ последовательно находим:

$$s_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} \bar{s}_{ki} d_{kk} s_{kj}}{s_{ii} d_{ii}}, \quad j = i+1, ..., n. \tag{2.9}$$

Таким образом, формулы для вычисления элементов матриц S и D имеют вид (2.7) – (2.9), причем индекс i в них последовательно изменяется от 1 до n.

После факторизации матрицы A процесс решения системы (1), как это уже не раз отмечалось выше, распадается на решение двух систем с треугольными матрицами

$$\begin{cases} S^* y = f, \\ DSx = y \end{cases}$$

и может быть найдено по формулам

$$\begin{cases} y_i = \dfrac{f_i - \sum_{k=1}^{i-1} \bar{s}_{ki} y_k}{s_{ii}}, \quad i = \overline{1,n}, \\ x_j = \dfrac{y_j - \sum_{k=j+1}^{n} d_{kk} s_{kj} x_k}{d_{jj} s_{jj}}, \quad j = \overline{n,1}. \end{cases} \tag{2.10}$$

Алгоритм (2.7) – (2.10) называют алгоритмом ***метода квадратного корня***.

***Замечание* 1.** В случае вещественной симметричной положительно определенной матрицы A разложение (2.1) принимает более простой вид:

$$A = U^T U, \tag{2.11}$$

где U – верхняя треугольная матрица с положительными элементами на диагонали (докажите !). Это разложение называют ***разложением Холесского*** (англ. Cholesky). В этом случае формулы (2.7) – (2.9) несколько упрощаются.

***Замечание* 2.** По количеству умножений и делений, как несложно подсчитать (выполните соответствующие подсчеты !), метод квадратного корня при больших n примерно в 2 раза экономичнее метода Гаусса, что легко объяснить тем, что данный алгоритм, в отличие от метода Гаусса, использует свойство симметрии.

***Замечание* 3.** Метод квадратного корня может пользован также и для вычисления определителя матрицы A:

$$\det A = \prod_{i=1}^{n} d_{ii} s_{ii}^2. \tag{2.12}$$

§ 3. Метод прогонки решения систем линейных алгебраических уравнений с трехдиагональной матрицей

Заметим, что рассмотренный в § 1 метод Гаусса является универсальным прямым методом решения систем линейных алгебраических уравнений. В то же время матрица A системы в конкретных задачах часто имеет специальную структуру (например, является симметричной или имеет много нулевых элементов и т.п.). Как мы уже видели на примере метода квадратного корня, учет специфики задачи позволяет построить более экономичные алгоритмы по сравнению с универсальными.

В настоящем параграфе мы рассмотрим один из важнейших в приложениях случаев, когда матрица системы (1) является трехдиагональной.

Итак, пусть требуется найти решение следующей системы:

$$\begin{cases} c_0 y_0 - b_0 y_1 = f_0, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \ \ i = \overline{1, N-1}, \\ -a_N y_{N-1} + c_N y_N = f_N. \end{cases} \tag{3.1}$$

Обратим внимание на некоторые особенности в ее записи, связанные именно с ее дальнейшими приложениями (может быть, не очень скорыми). Во-первых, вектор неизвестных здесь – это вектор $y = (y_0, y_1, ..., y_N)^T$; во-вторых, нумерация его компонент начинается с нуля вместо традиционной – с единицы, а это означает, что в такой системе будет $(N+1)$ неизвестных.

Перейдем теперь к разработке алгоритма. Следуя идее метода Гаусса (в варианте схемы единственного деления), проведем исключение неизвестных в (3.1). Введем обозначения

$$\alpha_1 = \frac{b_0}{c_0}, \ \ \beta_1 = \frac{f_0}{c_0}$$

и перепишем (3.1) в виде

$$\begin{cases} y_0 - \alpha_1 y_1 = \beta_1, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \ \ i = \overline{1, N-1}, \\ -a_N y_{N-1} + c_N y_N = f_N. \end{cases} \tag{3.2}$$

Возьмем в получившейся системе (3.2) первые два уравнения:

$$\begin{cases} y_0 - \alpha_1 y_1 = \beta_1, \\ -a_1 y_0 + c_1 y_1 - b_1 y_2 = f_1. \end{cases}$$

Умножив первое уравнение на a_1 и складывая со вторым, получим:

$$(c_1 - \alpha_1 a_1) y_1 - b_1 y_2 = f_1 + a_1 \beta_1,$$

откуда

$$y_1 - \alpha_2 y_2 = \beta_2,$$

где

$$\alpha_2 = \frac{b_1}{c_1 - \alpha_1 a_1}, \ \ \beta_2 = \frac{f_1 + a_1 \beta_1}{c_1 - \alpha_1 a_1}.$$

На этом первый шаг гауссова исключения заканчивается, так как все остальные уравнения не содержат неизвестного y_0.

Теперь легко видеть, что после k-го шага прямого хода метода Гаусса получим систему для неизвестных y_k, $y_{k+1},...,y_N$

$$\begin{cases} y_k - \alpha_{k+1} y_{k+1} = \beta_{k+1}, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \quad i = \overline{k+1, N-1}, \\ -a_N y_{N-1} + c_N y_N = f_N \end{cases} \tag{3.3}$$

и формулы для нахождения неизвестных y_i с номерами $i \le k-1$:

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = k-1,...,0, \tag{3.4}$$

в которых коэффициенты α_i, β_i находятся по формулам

$$\alpha_{i+1} = \frac{b_i}{c_i - \alpha_i a_i}; \quad \beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - \alpha_i a_i}; \quad i = 1,2,...; \quad \alpha_1 = \frac{b_0}{c_{10}}; \quad \beta_1 = \frac{f_0}{c_0}. \tag{3.5}$$

Полагая в (3.3) $k = N-1$, получим систему для определения y_{N-1} и y_N

$$\begin{cases} y_{N-1} - \alpha_N y_N = \beta_N, \\ -a_N y_{N-1} + c_N y_N = f_N, \end{cases}$$

из которой найдем:

$$y_N = \beta_{N+1}; \quad y_{N-1} = \alpha_N y_N + \beta_N.$$

Объединяя эти равенства с (3.4), (3.5), получим окончательные формулы для нахождения неизвестных.

Итак, алгоритм имеет следующий вид:

1. Вычисляем ***прогоночные коэффициенты*** по формулам

$$\alpha_1 = \frac{b_0}{c_0}; \quad \alpha_{i+1} = \frac{b_i}{c_i - a_i \alpha_i}, \quad i = 1,2,....N-1; \tag{3.6}$$

$$\beta_1 = \frac{f_0}{c_0}; \quad \beta_{i+1} = \frac{f_i + a_i \beta_1}{c_i - a_i \alpha_i}, \quad i = 1,...,N. \tag{3.7}$$

2. Вычисляем решение по формулам

$$y_N = \beta_{N+1}; \quad y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = N-1,...,0. \tag{3.8}$$

Формулы (3.6) – (3.8) носят название ***метода прогонки***; (3.6), (3.7) – формулы ***прямой прогонки***; (3.8) – ***обратная прогонка***.

Так как значения y_i находятся последовательно от $(i+1)$-го к i-му, то (3.6) – (3.8) называют формулами ***правой прогонки***.

Элементарный подсчет количества арифметических операций в (3.6) – (3.8) показывает, что реализация метода прогонки требует выполнения $3N-1$ умножений, $2N+1$ делений, $3N-1$ сложений, т.е. всего $q_N = 8N-1$ операций, если не делать различий между действиями.

Аналогично (3.6) – (3.8) выводятся формулы ***левой прогонки***:

$$\begin{cases} \xi_N = \dfrac{a_N}{c_N}, \quad \xi_i = \dfrac{a_i}{c_i - \xi_{i+1} b_i}, \quad i = N-1,\ N-2,...,1, & (3.9) \\ \eta_N = \dfrac{f_N}{c_N}, \quad \eta_i = \dfrac{f_i + b_i \eta_{i+1}}{c_i - \xi_{i+1} b_i}, \quad i = N-1, N-2,...,0, & (3.10) \\ y_0 = \eta_0, \quad y_{i+1} = \xi_{i+1} y_i + \eta_{i+1}, \quad i = 0,1,...N-1. & (3.11) \end{cases}$$

Иногда оказывается удобным комбинировать правую и левую прогонки, получая так называемый ***метод встречных прогонок***. Этот метод целесообразно применять, если надо найти только одно неизвестное, например, y_m $(0 \le m \le N)$ или группу подряд идущих неизвестных.

Получим формулы метода встречных прогонок. Пусть $1 \le m \le N$ и по формулам (3.6), (3.7), (3.9), (3.10) найдены $\alpha_1, \alpha_2, ..., \alpha_m$; $\beta_1, \beta_2, ..., \beta_m$ и $\xi_N, ..., \xi_m$; $\eta_N, ..., \eta_m$. Выпишем формулы (3.8), (3.11) для обратного хода правой и левой прогонок для $i = m+1$. Получим:

$$\begin{cases} y_{m-1} = \alpha_m y_m + \beta_m, \\ y_m = \xi_m y_{m-1} + \eta_m. \end{cases}$$

Отсюда

$$y_m = \frac{\eta_m + \xi_m \beta_m}{1 - \xi_m \alpha_m}.$$

Используя найденное y_m, по формулам (3.8) для $i = m-1,...,0$ найдем последовательно $y_{m-1}, ..., y_0$, а по формулам (3.11) для $i = m, m+1, ..., N$ вычисляем остальные y: $y_{m+1}, ..., y_N$.

Итак, формулы метода встречных прогонок имеют вид

$$\begin{cases} \alpha_1 = \dfrac{b_0}{c_0}, \quad \alpha_{i+1} = \dfrac{b_i}{c_i - \alpha_i a_i}, \quad i = 1,...,m-1; \\ \beta_1 = \dfrac{f_0}{c_0}, \quad \beta_{i+1} = \dfrac{f_i + a_i \beta_i}{c_i - \alpha_i a_i}, \quad i = 1,...,m-1; \\ \xi_N = \dfrac{a_N}{c_N}, \quad \xi_i = \dfrac{a_i}{c_i - \xi_{i+1} b_i}, \quad i = N-1,...,m; \\ \eta_N = \dfrac{f_N}{c_N}, \quad \eta_i = \dfrac{f_i + b_i \eta_{i+1}}{c_i - \xi_{i+1} b_i}, \quad i = N-1,...,m \end{cases} \qquad (3.12)$$

для вычисления прогоночных коэффициентов и

$$\begin{cases} y_m = \dfrac{\eta_m + \xi_m \beta_m}{1 - \xi_m \alpha_m}, \quad y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = m-1,...,0, \\ y_{i+1} = \xi_{i+1} y_i + \eta_{i+1}, \quad i = m,...,N-1. \end{cases} \qquad (3.13)$$

для определения решения.

3.1. Обоснование метода прогонки

Выше были получены формулы метода прогонки без каких-либо предположений относительно коэффициентов системы (3.1). Остановимся теперь на вопросе о том, каким требованиям должны удовлетворять эти коэффициенты, чтобы метод мог быть применен и позволял получить решение задачи с достаточной точностью.

Так как расчетные формулы (3.6) – (3.8) метода прогонки содержат операции деления, то нужно гарантировать, чтобы знаменатели $c_i - \alpha_i a_i$ не обращались в нуль. В связи с этим будем говорить, что алгоритм метода прогонки ***корректен***, если $c_i - \alpha_i a_i \neq 0$, $i = 1,...,N$.

Далее, решение y_i находится по рекуррентной формуле (3.8). Эта формула может порождать накопление ошибок округления результатов арифметических операций. Действительно, пусть прогоночные коэффициенты α_i, β_i найдены точно, а при вычислении y_N допущена ошибка ε_N, т.е. найдено значение $\tilde{y}_N = y_N + \varepsilon_N$. Так как решение $\tilde{y}_i$ находится по формулам (3.8)

$$\tilde{y}_i = \alpha_{i+1}\tilde{y}_{i+1} + \beta_{i+1}, \quad i = N-1,...,0,$$

то погрешность $\varepsilon_i = \tilde{y}_i - y_i$ будет удовлетворять однородному уравнению

$$\varepsilon_i = \alpha_{i+1}\varepsilon_{i+1}, \quad i = N-1,...,0$$

с заданным ε_N. Отсюда следует, что если все α_i по модулю больше единицы, то может произойти сильное увеличение погрешности ε_0 и, если значение N достаточно велико, то полученное реальное решение $\tilde{y}_i$ будет значительно отличаться от искомого решения y_i.

Поэтому будем требовать, чтобы прогоночные коэффициенты α_i не превосходили по модулю единицы. Это достаточное условие гарантирует невозрастание погрешности ε_i в рассмотренной модельной ситуации.

Определение. Если выполнено условие $|\alpha_i| \leq 1$, $i = 1,...,N$, то будем говорить, что алгоритм правой прогонки ***устойчив***.

***Теорема* 1.** Пусть коэффициенты системы (3.1) удовлетворяют условиям

$$|c_0| > 0, \ |c_N| > 0,$$
$$|a_i| > 0, \ |b_i| > 0, \ i = 1,2,...,N-1,$$
$$|c_i| \geq |a_i| + |b_i|, \ i = 1,...,N-1, \tag{3.14}$$
$$|c_0| \geq |b_0|, \ |c_N| \geq |a_N|, \tag{3.15}$$

причем хотя бы в одном из условий (3.14) – (3.15) выполняется строгое неравенство (другими словами, матрица A имеет диагональное преобладание по строкам, причем хотя бы в одном случае строгое).

Тогда для алгоритма (3.6) – (3.8) метода прогонки имеют место неравенства

$$c_i - \alpha_i a_i \neq 0, \ |\alpha_i| \leq 1, \quad i = 1,...,N,$$

гарантирующие корректность и устойчивость метода.

Доказательство проведем по индукции.

Из условий теоремы 1 следует, что

$$0 \leq |\alpha_1| = \frac{|b_0|}{|c_0|} \leq 1.$$

Покажем, что из неравенства $|\alpha_i| \le 1$ $(i \le N-1)$ и условий теоремы 1 следуют неравенства

$$c_i - \alpha_i a_i \ne 0, \ |\alpha_{i+1}| \le 1, \ i \le N-1.$$

Имеем:

$$|c_i - \alpha_i a_i| \ge |c_i| - |\alpha_i| \cdot |a_i| \ge |b_i| + |a_i|(1-|\alpha_i|) \ge |b_i| > 0 \qquad (3.16)$$

и, следовательно,

$$c_i - \alpha_i a_i \ne 0, \ |\alpha_{i+1}| = \frac{|b_i|}{|c_i - \alpha_i a_i|} \le \frac{|b_i|}{|b_i|} \le 1, \ i \le N-1.$$

Осталось показать, что $c_N - \alpha_N a_N \ne 0$. Для того, чтобы это сделать, используем предположение, что хотя бы в одном из условий (3.14) – (3.15) имеет место строгое неравенство.

Возможны следующие случаи:

а) $|c_N| > |a_N|$.

Тогда, поскольку в соответствии с доказанным $|\alpha_N| \le 1$, то

$$|c_N - \alpha_N a_N| \ge |c_N| - |\alpha_N| \cdot |a_N| > |a_N|(1-|\alpha_N|) \ge 0;$$

б) строгое неравенство достигается в(3.14) для некоторого $i = i_0$, $1 \le i_0 \le N-1$.

Тогда в (3.16) имеем

$$|c_{i_0} - \alpha_{i_0} a_{i_0}| \ge |c_{i_0}| - |\alpha_{i_0}| \cdot |a_{i_0}| > |b_{i_0}| + |a_{i_0}|(1-|\alpha_{i_0}|) \ge |b_{i_0}|,$$

т.е. $|\alpha_{i_0+1}| < 1$ и, таким образом, для всех $i \ge i_0 + 1$ будет выполняться строгое неравенство $|\alpha_i| < 1$;

в) $|c_0| > |b_0|$.

Тогда неравенство $|\alpha_i| < 1$ выполняется, начиная с $i = 1$. ⊠

***Замечание* 1.** Условия теоремы 1 являются лишь достаточными. Их можно ослабить, разрешив некоторым из коэффициентов a_i или b_i обращаться в нуль. Так, например, если при некотором m, $1 \le m \le N-1$, окажется, что $a_m = 0$, то система (3.1) распадется на две системы:

$$\begin{cases} c_m y_m - b_m y_{m+1} = f_m, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \ i = m+1, ..., N-1, \\ -a_N y_{N-1} + c_N y_N = f_N, \end{cases}$$

для неизвестных $y_m, ..., y_N$ и

$$\begin{cases} c_0 y_0 - b_0 y_1 = f_0, \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \ i = 1, ..., m-2, \\ -a_{m-1} y_{m-2} + c_{m-1} y_{m-1} = f_{m-1} + b_{m-1} y_m, \end{cases}$$

для неизвестных $y_0, ..., y_{m-1}$,

и, следовательно, к каждой из них по отдельности можно применять алгоритм (3.6) – (3.8) (при выполнении условий теоремы 1). Более же аккуратные исследования показывают, что и для всей системы в рассматриваемом случае алгоритм также проходит.

***Замечание* 2.** Условия (3.14), (3.15) обеспечивают корректность и устойчивость также и методов левой и встречных прогонок.

***Замечание* 3.** При выполнении условий теоремы 1 система (3.1) имеет единственное решение при любой правой части.

Действительно, поскольку метод прогонки есть частный случай метода Гаусса (причем в варианте схемы единственного деления), то для матрицы A системы (3.1) справедлива теорема 1 из § 1 об LU -разложении, т.е. $A = LU$, где, как легко видеть,

$$L = \begin{pmatrix} c_0 & & & & & \\ -a_1 & \Delta_1 & & & & \\ & -a_2 & \Delta_2 & & & \\ & & \ddots & & & \\ & & & -a_N & \Delta_N \end{pmatrix}, \quad U = \begin{pmatrix} 1 & -\alpha_1 & & & & \\ & 1 & -\alpha_2 & & & \\ & & & \ddots & & \\ & & & & 1 & -\alpha_N \\ & & & & & 1 \end{pmatrix},$$

где

$$\Delta_i = c_i - \alpha_i a_i , \ \ i = 1,..., N .$$

Поэтому

$$\det A = \det L \cdot \det U = c_1 \cdot \Delta_1 \cdot \ldots \cdot \Delta_N \neq 0 .$$

3.2. Метод циклической прогонки

Помимо рассмотренных выше разновидностей метода прогонки существуют и другие ее варианты. Так, если матрица рассматриваемой задачи является трехдиагональной, но не обладает свойством диагонального доминирования, то для ее решения может быть применен метод ***немонотонной прогонки***, базирующийся на использовании схемы Гаусса с выбором главного элемента по строке.

Остановимся более подробно на еще одной модификации алгоритма – методе ***циклической прогонки***. Пусть требуется решить систему линейных алгебраических уравнений вида

$$-a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i , \ \ i = 0, \pm 1, \pm 2, \ldots , \tag{3.17}$$

коэффициенты и правая часть которой периодичны с периодом N , т.е. удовлетворяют условиям

$$a_i = a_{i+N} , \ \ b_i = b_{i+N} , \ \ c_i = c_{i+N} , \ \ f_i = f_{i+N} \tag{3.18}$$

(такая задача может, например, получиться в случае, когда с помощью сеточного алгоритма отыскивается периодическое решение обыкновенного дифференциального уравнения второго порядка).

При выполнении условий (3.18) решение задачи (3.17), если оно существует, также будет периодическим с периодом N , т.е.

$$y_i = y_{i+N} . \tag{3.19}$$

Учитывая (3.19), достаточно найти решения y_i , например, при $i = 0,..., N-1$. В этом случае задачу (3.17) – (3.19) можно записать следующим образом:

$$\begin{cases} -a_0 y_{N-1} + c_0 y_0 - b_0 y_1 = f_0, \\ \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i, \ \ i = 1,...,N-1, \\ y_N = y_0. \end{cases} \tag{3.20}$$

(3.21)

Условие (3.21) мы добавили к (3.20) для того, чтобы можно было из уравнения системы при $i = N-1$ не исключать неизвестное y_N, заменяя его на y_0. Это позволяет сохранить единый вид для уравнений (3.20) при $i = 1,...,N-1$.

Если ввести векторы $Y = (y_0,...,y_{N-1})^T$ и $F = (f_0,...,f_{N-1})^T$, то систему (3.20), (3.21) можно записать в матричном виде

$$AY = F,$$

где

$$A = \begin{pmatrix} c_0 & -b_0 & 0 & 0 & \cdots & 0 & 0 & -a_0 \\ -a_1 & c_1 & -b_1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & -a_2 & c_2 & -b_2 & \ldots & 0 & 0 & 0 \\ & & & \ldots & \ldots & & & \\ 0 & 0 & 0 & 0 & \ldots & c_{N-3} & -b_{N-3} & 0 \\ 0 & 0 & 0 & 0 & \ldots & -a_{N-2} & c_{N-2} & -b_{N-2} \\ -b_{N-1} & 0 & 0 & 0 & \ldots & 0 & -a_{N-1} & c_{N-1} \end{pmatrix}.$$

Присутствие отличных от нуля коэффициентов a_0 и b_{N-1} не позволяет решить эту систему «штатным» методом прогонки.

Введем обозначение для оператора, задающего левые части уравнения:

$$Pv_i = -a_i v_{i-1} + c_i v_i - b_i v_{i+1}.$$

Будем искать решение задачи (3.20), (3.21) в виде

$$y_i = u_i + y_0 v_i, \ \ i = 0,...,N, \tag{3.22}$$

где u_i – решение неоднородной задачи с однородными граничными условиями

$$\begin{cases} Pu_i = f_i, \ \ i = 1,...,N-1, \\ u_0 = 0, \ \ u_N = 0 \end{cases} \tag{3.23}$$

а v_i – решение однородной задачи с неоднородными условиями

$$\begin{cases} Pv_i = 0, \ \ i = 1,...,N-1, \\ u_0 = 1, \ \ u_N = 1. \end{cases} \tag{3.24}$$

Так как оператор P линейный, т.е.

$$Py_i = Pu_i + y_0 Pv_i,$$

то из (3.23), (3.24) следует, что при описанном выборе u и v выполняются условия (3.20) при $i = 1,...,N-1$. Выполнено также и уравнение (3.21). Подставляя (3.22) в первое из уравнений (3.20), получим:

$$-a_0(u_{N-1} + y_0 v_{N-1}) + c_0(u_0 + y_0 v_0) - b_0(u_1 + y_0 v_1) = f_0 .$$

Отсюда находим y_0:

$$y_0 = \frac{f_0 + a_0 u_{N-1} + b_0 u_1}{c_0 - a_0 v_{N-1} - b_0 v_1} . \tag{3.25}$$

Таким образом, если выбрать y_0 по формуле (3.25), то решение (3.20), (3.21) можно найти по формулам (3.22).

Остановимся теперь на решении систем (3.23) и (3.24). Они являются частными случаями трехдиагональных систем, для которых мы выше построили алгоритм метода прогонки. Запишем его в этом случае:

$$\begin{aligned} &u_N = 0, \quad u_i = \alpha_{i+1} u_{i+1} + \beta_{i+1}, \quad i = N-1,...,1, \\ &v_N = 1, \quad v_i = \alpha_{i+1} v_{i+1} + \gamma_{i+1}, \quad i = N-1,...,1, \end{aligned} \tag{3.26}$$

где прогоночные коэффициенты α_i, β_i, γ_i находятся по следующим формулам

$$\begin{aligned} &\alpha_1 = 0, \quad \alpha_{i+1} = \frac{b_i}{c_i - \alpha_i a_i}, \quad i = 1,2,...,N-1; \\ &\beta_1 = 0, \quad \beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - \alpha_i a_i}, \quad i = 1,2,...,N; \\ &\gamma_1 = 1, \quad \gamma_{i+1} = \frac{a_i \gamma_i}{c_i - \alpha_i a_i}, \quad i = 1,2,...,N. \end{aligned} \tag{3.27}$$

Преобразуем выражение, полученное нами для вычисления y_0. Из (3.26) имеем:

$$u_{N-1} = \alpha_N u_N + \beta_N = \beta_N ; \quad v_{N-1} = \gamma_N + \alpha_N .$$

Поэтому, учитывая свойство периодичности коэффициентов ($f_0 = f_N$, $a_0 = a_N$, $b_0 = b_N$, $b_0 = b_N$), будем иметь:

$$y_0 = \frac{f_N + a_N \beta_N + b_N u_1}{c_N - \alpha_N a_N - a_N \gamma_N - b_N v_1} = \frac{\beta_{N+1} + \alpha_{N+1} u_1}{1 - \gamma_{N+1} - \alpha_{N+1} v_1} .$$

Итак, построен алгоритм, который носит название ***циклической прогонки***:

$$\begin{aligned} &\alpha_2 = \frac{b_1}{c_1}, \quad \beta_2 = \frac{f_1}{c_1}, \quad \gamma_2 = \frac{a_1}{c_1}, \\ &\alpha_{i+1} = \frac{b_i}{c_i - \alpha_i a_i}, \quad \beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - \alpha_i a_i}, \quad \gamma_{i+1} = \frac{a_i \gamma_i}{c_i - \alpha_i a_i}, \quad i = 2,...,N, \end{aligned} \tag{3.28}$$

$$u_{N-1} = \beta_N, \;\; v_{N-1} = \alpha_N + \gamma_N,$$
$$u_i = \alpha_{i+1}u_{i+1} + \beta_{i+1}, \;\; v_i = \alpha_{i+1}v_{i+1} + \gamma_{i+1}, \;\; i = N-2,...,1, \qquad (3.29)$$
$$y_0 = \frac{\beta_{N+1} + \alpha_{N+1}u_1}{1 - \gamma_{N+1} - \alpha_{N+1}v_1}, \;\; y_i = u_i + y_0 v_i, \;\; i = 1,...,N-1.$$

Исследуем теперь вопрос о корректности и устойчивости алгоритма (3.28), (3.29). Имеет место утверждение, аналогичное теореме 1.

***Теорема* 2.** Пусть коэффициенты системы (3.20), (3.21) удовлетворяют условиям

$$|a_i| > 0, \;\; |b_i| > 0, \;\; |c_i| \geq |a_i| + |b_i|, \;\; i = 1,...,N \qquad (3.30)$$

и существует хотя бы одно i_0, $1 \leq i_0 \leq N$ такое, что $|c_{i_0}| > |a_{i_0}| + |b_{i_0}|$.

Тогда

$$c_i - \alpha_i a_i \neq 0, \;\; |\alpha_i| \leq 1, \;\; |\alpha_i| + |\gamma_i| \leq 1, \;\; i = 2,...,N; \;\; 1 - \gamma_{N+1} - \alpha_{N+1}v_1 \neq 0.$$

Доказательство.

Так как α_i, β_i, γ_i – прогоночные коэффициенты метода правой прогонки, примененного к решению задач (3.23), (3.24), для которого выполнены условия теоремы 1, то из нее следует справедливость неравенств

$$c_i - \alpha_i a_i \neq 0, \;\; |\alpha_i| \leq 1, \;\; i = 2,3,...,N;$$
$$|c_i - \alpha_i a_i| \geq |c_i| - |\alpha_i| \cdot |a_i| \geq |b_i| > 0.$$

Далее, из условий теоремы 2 следует, что $|a_1| + |b_1| \leq |c_1|$ и, следовательно, $|\alpha_2| + |\gamma_2| \leq 1$ (см. первую строку формул (3.28)). Отсюда, применяя метод индукции, получим неравенства

$$|\alpha_i| + |\gamma_i| \leq 1, \;\; i = 2,3,...,N,$$

так как

$$|\alpha_{i+1}| + |\gamma_{i+1}| = \frac{|b_i| + |a_i| \cdot |\gamma_i|}{|c_i - \alpha_i a_i|} \leq \frac{|a_i| + |b_i| - |a_i|(1 - |\gamma_i|)}{|c_i| - |\alpha_i| \cdot |a_i|} \leq \frac{|a_i| + |b_i| - |a_i| \cdot |\alpha_i|}{|c_i| - |\alpha_i| \cdot |a_i|} \leq \frac{|c_i| - |\alpha_i| \cdot |a_i|}{|c_i| - |\alpha_i| \cdot |a_i|} = 1.$$

Заметим, что $|c_{i_0}| > |a_{i_0}| + |b_{i_0}|$ и, следовательно, $|\alpha_{i_0+1}| + |\gamma_{i_0+1}| < 1$. Отсюда следует, что для $i \geq i_0 + 1$ имеет место строгое неравенство $|\alpha_i| + |\gamma_i| < 1$ (и в частности, так как $i_0 \leq N$, то $|\alpha_{N+1}| + |\gamma_{N+1}| < 1$).

Покажем, что $1 - \gamma_{N+1} - \alpha_{N+1}v_1 \neq 0$. Из (3.29) имеем:

$$|v_{N-1}| \leq |\alpha_N| + |\gamma_N| \leq 1.$$

Отсюда по индукции получаем:

$$|v_i| \leq |\alpha_{i+1}| \cdot |v_{i+1}| + |\gamma_{i+1}| \leq |\alpha_{i+1}| + |\gamma_{i+1}| \leq 1.$$

В частности, $|v_1| \leq 1$. Поэтому

$$|1 - \gamma_{N+1} - \alpha_{N+1}v_1| \geq 1 - |\gamma_{N+1}| - |\alpha_{N+1}| \cdot |v_1| \geq 1 - |\gamma_{N+1}| - |\alpha_{N+1}| > 0.$$

⊠

§ 4. Обусловленность систем линейных алгебраических уравнений

При использовании численных методов для решения тех или иных математических задач необходимо различать свойства самой задачи и свойства алгоритма, предназначенного для ее решения.

Для каждой математической задачи принято рассматривать вопрос о ее ***корректности*** (см., например, метод прогонки).

Определение* 1.** Говорят, что задача поставлена ***корректно, если:

1) решение задачи существует и единственно;
2) решение задачи непрерывно зависит от входных данных (т.е. ***устойчиво*** по входным данным).

Рассмотрим вопрос о корректности исходной задачи на примере системы линейных алгебраических уравнений (1):

$$Ax = f\ .$$

Для системы (1) первое требование в Определении 1 будет выполнено, если $\det A \neq 0$. В этом случае можно определить обратную матрицу A^{-1} и записать решение в виде

$$x = A^{-1} f\ . \tag{4.1}$$

Второе же требование в Определении 1 применительно к рассматриваемой системе нуждается в некоторой детализации. Входными данными в задаче (1), очевидно, являются компоненты f_i вектора f и элементы a_{ij} матрицы A. Предположим, что A и f заданы с некоторой погрешностью. Тогда вместо системы (1) мы практически имеем «возмущенную» систему

$$\widetilde{A}\widetilde{x} = \widetilde{f}\ . \tag{4.2}$$

Естественной будет постановка вопроса о том, как связана погрешность решения $\delta x = \widetilde{x} - x$ с погрешностью $\delta f = \widetilde{f} - f$ или $\delta A = \widetilde{A} - A$.

Сообразно описанным ситуациям различают:

а) ***устойчивость по правой части*** (когда возмущается только вектор f);

б) ***коэффициентную устойчивость*** (когда возмущается только матрица A).

Прежде чем дать количественные характеристики подобных связей, напомним некоторые определения.

Определение* 2. *Нормой вектора x будем называть число $\|x\|$, удовлетворяющее условиям:

1) $\|x\| \geq 0$ для всех $x \in R^n$;

2) $\|x\| = 0$ тогда и только тогда, когда $x = 0$;

3) $\|\alpha x\| = |\alpha| \cdot \|x\|$ для всех чисел $\alpha \in R$ и векторов $x \in R^n$;

4) $\|x + y\| \leq \|x\| + \|y\|$ для произвольных $x, y \in R^n$ (неравенство треугольника).

Существует ряд способов задания нормы векторов. Наиболее употребительны из них:

1) ***первая*** или ***кубическая*** норма, задаваемая формулой $\|x\|_{\text{I}} = \max\limits_{1 \leq i \leq n} |x_i|$;

2) ***вторая*** или ***октаэдрическая*** норма, задаваемая формулой $\|x\|_{\text{II}} = \sum\limits_{i=1}^{n} |x_i|$;

3) ***третья*** или ***сферическая*** норма, задаваемая формулой $\|x\|_{\text{III}} = \sqrt{(x, x)} = \sqrt{\sum\limits_{i=1}^{n} x_i^2}$.

Замечание. Вообще говоря, все три указанных выше способа задания векторной нормы являются частными случаями так называемой p-нормы вектора, определяемой формулой

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

(для $p = \infty$, $p = 1$ и $p = 2$ соответственно).

Определение* 3. *Нормой матрицы A называют число $\|A\|$, удовлетворяющее условиям:

1) $\|A\| \geq 0$ для всех матриц A из рассматриваемого пространства H;

2) $\|A\| = 0$ тогда и только тогда, когда $A = 0$;

3) $\|\alpha A\| = |\alpha| \cdot \|A\|$ для всех чисел $\alpha \in R$ и матриц $A \in H$;

4) $\|A + B\| \leq \|A\| + \|B\|$ для всех матриц $A, B \in H$ (неравенство треугольника);

5) $\|AB\| \leq \|A\| \cdot \|B\|$ для всех матриц $A, B \in H$ (свойство мультипликативности).

Норма матрицы, как и норма вектора, может быть определена различными способами. Если, например, рассматривать матрицу A как n^2 мерный вектор с вещественными компонентами, то очевидно следующее обобщение сферической (заметим, что ее также называют ***нормой Фробениуса*** или ***евклидовой***)

$$\|A\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2}$$

и кубической

$$\|A\| = n \max_{1 \leq i, j \leq n} |a_{ij}|$$

норм.

***Определение* 4.** Если для любой матрицы A и любого вектора x выполнено неравенство

$$\|Ax\| \leq \|A\| \cdot \|x\|, \tag{4.3}$$

то говорят, что норма матрицы ***согласована*** с данной нормой вектора.

Если в (4.3) определить

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|},$$

то введенная таким образом норма матрицы A будет наименьшей из всех норм, согласованных с данной нормой вектора. Эту норму будем называть ***подчиненной*** данной норме вектора.

В частности, подчиненная кубическая норма матрицы имеет вид

$$\|A\|_{\mathrm{I}} = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|.$$

Действительно, в этом случае

$$\|Ax\|_{\mathrm{I}} = \max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \leq \max_{1 \leq j \leq n} |x_j| \cdot \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}| = \|x\|_{\mathrm{I}} \cdot \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|$$

и поэтому

$$\frac{\|Ax\|}{\|x\|} \le \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}| . \tag{4.4}$$

В то же время, здесь знак «=» имеет место, если выбрать координаты $x_1,...,x_n$ вектора x таким образом, чтобы выполнялись равенства $|x_1| = \ldots = |x_n|$ и $a_{pk}x_k = |a_{pk}x_k|$, $(k = 1,...,n)$, где p – то значение индекса i, при котором достигается максимум в сумме $\sum_{j=1}^{n} |a_{ij}|$. Поэтому правая часть в (4.4) в точности равна $\sup\limits_{x\neq 0} \dfrac{\|Ax\|}{\|x\|}$.

Аналогично может быть показано, что октаэдрическая норма матрицы имеет вид

$$\|A\|_{\text{II}} = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}| , \tag{4.5}$$

а сферическая –

$$\|A\|_{\text{III}} = \sqrt{\lambda_{\max}\left(AA^T\right)} . \tag{4.6}$$

Упражнение. Доказать формулы (4.5), (4.6).

Теперь мы можем перейти к изучению количественных оценок, связывающих погрешности δf, δA с δx и тем самым расшифровать второе требование в определении корректности.

Сначала предположим, что в матрицу A возмущений не вносится, т.е. $\delta A = 0$. Тогда

$$A(x + \delta x) = f + \delta f .$$

Отсюда, учитывая (1), получаем:

$$A\delta x = \delta f ,$$

т.е.

$$\delta x = A^{-1}\delta f$$

и, следовательно,

$$\|\delta x\| \le \left\|A^{-1}\right\| \cdot \|\delta f\| . \tag{4.7}$$

Неравенство (4.7) устанавливает непрерывную зависимость решения от правой части, т.е. показывает, что условие $\|\delta f\| \to 0$ влечет за собой и $\|\delta x\| \to 0$.

В (4.7) входят нормы абсолютных погрешностей решения и правой части. В практических приложениях более качественной является связь между нормами относительных погрешностей $\dfrac{\|\delta x\|}{\|x\|}$ и $\dfrac{\|\delta f\|}{\|f\|}$.

Так как в силу (1) $f = Ax$, то отсюда получаем неравенство

$$\|f\| \le \|A\| \cdot \|x\| .$$

Перемножая почленно последнее неравенство и (4.7), будем иметь:

$$\|\delta x\| \cdot \|f\| \le \|A\| \cdot \left\|A^{-1}\right\| \cdot \|\delta f\| \cdot \|x\|,$$

откуда

$$\frac{\|\delta x\|}{\|x\|} \le \|A\| \cdot \left\|A^{-1}\right\| \cdot \frac{\|\delta f\|}{\|f\|}. \tag{4.8}$$

Пусть теперь $\delta f = 0$, $\delta A \neq 0$. Сформулирует вначале некоторое вспомогательное утверждение.

Лемма. Пусть C – квадратная матрица, удовлетворяющая условию $\|C\| < 1$, и E – единичная матрица. Тогда существует матрица $(E + C)^{-1}$, причем

$$\left\|(E + C)^{-1}\right\| \le \frac{1}{1 - \|C\|}. \tag{4.9}$$

Доказательство.

Для любого вектора x имеем:

$$\|(E + C)x\| = \|x + Cx\| \ge \|x\| - \|Cx\| \ge \|x\| - \|C\| \cdot \|x\| = (1 - \|C\|)\|x\|.$$

Поэтому если $(E + C)x = 0$, то последняя оценка примет вид

$$0 \ge (1 - \|C\|)\|x\|$$

или, так как $\|C\| < 1$,

$$\|x\| \le 0.$$

Следовательно, $x = 0$. Таким образом, система с матрицей $(E + C)$ имеет единственное решение, а значит, матрица $(E + C)^{-1}$ существует.

Введем обозначение $(E + C)x = y$ (т.е. $x = (E + C)^{-1} y$), получим для вектора y неравенство

$$\|y\| \ge (1 - \|C\|)\left\|(E + C)^{-1} x\right\|,$$

откуда следует:

$$\frac{\left\|(E + C)^{-1} y\right\|}{\|y\|} \le \frac{1}{1 - \|C\|}.$$

Так как правая часть последнего неравенства не зависит от y, то $\left\|(E + C)^{-1}\right\| \le \frac{1}{1 - \|C\|}$.

Теперь предположим, что выполняется неравенство ⊠

$$\|\delta A\| < \left\|A^{-1}\right\|^{-1}. \tag{4.10}$$

Так как при $\delta f = 0$ имеем систему

$$(A + \delta A)(x + \delta x) = f,$$

то отсюда

$$Ax + A\delta x + \delta A x + \delta A \delta x = f$$

и

$$(A + \delta A)\delta x = -\delta A x\,,$$

т.е.

$$\delta x = -(A + \delta A)^{-1}\delta A x = -\left(E + A^{-1}\delta A\right)^{-1} A^{-1}\delta A x\,,$$

откуда следует, что

$$\|\delta x\| \le \left\|\left(E + A^{-1}\delta A\right)^{-1}\right\| \cdot \left\|A^{-1}\right\| \cdot \|\delta A\| \cdot \|x\|\,.$$

Так как в силу (4.10)

$$\left\|A^{-1}\delta A\right\| \le \left\|A^{-1}\right\| \cdot \|\delta A\| < 1\,,$$

то матрица $E + A^{-1}\delta A$ по лемме 1 обратима и выполняется оценка (4.9):

$$\left\|\left(E + A^{-1}\delta A\right)^{-1}\right\| \le \frac{1}{1 - \left\|A^{-1}\right\| \cdot \|\delta A\|}\,.$$

Поэтому

$$\frac{\|\delta x\|}{\|x\|} \le \frac{\left\|A^{-1}\right\| \cdot \|A\|}{1 - \left\|A^{-1}\right\| \cdot \|A\| \cdot \dfrac{\|\delta A\|}{\|A\|}} \cdot \frac{\|\delta A\|}{\|A\|}\,. \tag{4.11}$$

Таким образом, (4.11) означает коэффициентную устойчивость системы (1).

Упражнение. Доказать, что в условиях возмущения и правой части, и матрицы коэффициентов при выполнении условия (4.10) имеет место следующая оценка полной погрешности приближенного решения:

$$\frac{\|\delta x\|}{\|x\|} \le \frac{\|A\| \cdot \left\|A^{-1}\right\|}{1 - \|A\| \cdot \left\|A^{-1}\right\| \dfrac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta f\|}{\|f\|} \right).$$

Входящее в оценки (4.8), (4.11) число

$$\nu(A) = \|A\| \cdot \left\|A^{-1}\right\| \tag{4.12}$$

называется ***числом обусловленности*** матрицы A (другое обозначение $\operatorname{cond} A$).

Отметим следующие свойства величины $\nu(A)$.

1^0.

$$\nu(A) \ge \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|} \ge 1\,, \tag{4.13}$$

где $|\lambda_{\max}(A)|$ и $|\lambda_{\min}(A)|$ - наибольшее и наименьшее по модулю собственное значение матрицы A соответственно.

Действительно, пусть y – собственный вектор матрицы A, отвечающий наибольшему по модулю ее собственному значению. Тогда по определению имеем:

$$Ay = \lambda_{\max}(A)y\,,$$

откуда непосредственно следует неравенство

$$|\lambda_{\max}(A)|\cdot\|y\| \le \|A\|\cdot\|y\|$$

или

$$|\lambda_{\max}(A)| \le \|A\|, \tag{4.14}$$

т.е. ***спектральный радиус*** (это – наибольшее по модулю собственное значение) матрицы не превосходит любой из ее норм, подчиненных некоторой норме вектора..

Поскольку $\lambda_{\min}^{-1}(A)$ является максимальным по модулю собственным значением матрицы A^{-1}, то для него справедлива аналогичная (4.14) оценка

$$\left|\lambda_{\min}^{-1}(A)\right| \le \left\|A^{-1}\right\|.$$

Перемножая две последние оценки, получаем (4.13):

$$1 \le \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|} \le \|A\|\cdot\left\|A^{-1}\right\| = \nu(A).$$

Для вещественной симметричной матрицы A и сферической нормы в первом из неравенств (4.13) имеет место знак «=», т.е.

$$\nu_{\text{III}}(A) = \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}. \tag{4.15}$$

Действительно, пусть $\{\mu_k\}_{k=1}^n$ – ортонормированый базис из собственных векторов матрицы A. Тогда, раскладывая произвольный вектор x по базису $\{\mu_k\}$, можем записать:

$$x = \sum_{k=1}^{n} c_k \mu_k .$$

Следовательно,

$$\|x\|^2 = \sum_{k=1}^{n} c_k^2 ,\ Ax = \sum_{k=1}^{n} c_k \lambda_k \mu_k ,\ \|Ax\|^2 = \sum_{k=1}^{n} c_k^2 \lambda_k^2 .$$

Таким образом,

$$\|Ax\|^2 \le |\lambda_{\max}(A)|^2 \cdot \|x\|^2 ,$$

т.е.

$$\|A\| \le |\lambda_{\max}(A)| .$$

Последнее неравенство совместно с (4.14) дает:

$$\|A\| = |\lambda_{\max}(A)| ,$$

что и завершает доказательство рассматриваемого утверждения.

2^0.

$$\nu(AB) \le \nu(A)\cdot\nu(B).$$

Данное свойство со всей очевидностью следует из п. 4 определения 3.

Матрицы с большим числом обусловленности $\nu(A)$ называются ***плохо обусловленными***. Соответствующие системы вида (1) также называются плохо обусловленными. Для плохо обусловленных систем линейных алгебраических уравнений недопустимо велика правая часть в неравенствах (4.8), (4.11) (особенно это касается неравенства (4.8)). Поэтому лишь очень малые погрешности входных данных гарантируют приемлемую относительную погрешность решения.

Рассмотрим некоторые примеры.

1^0. Пусть дана матрица

$$A = \begin{pmatrix} 3.0000 & -7.0001 \\ 3.0000 & -7.0000 \end{pmatrix}.$$

Воспользуемся для оценки числа обусловленности $\nu(A)$ формулой (4.13). Раскрывая определитель $\det(A - \lambda E)$, получаем характеристическое уравнение

$$(3-\lambda)(-7-\lambda) + 21.0003 = 0$$

или

$$\lambda^2 + 4\lambda + 0.0003 = 0,$$

откуда

$$\lambda_1 = -2 + \sqrt{1.9997} \approx -0.0001; \quad \lambda_2 = -2 - \sqrt{1.9997} \approx -3.9999.$$

Поэтому

$$\nu(A) \geq 39999 \approx 4 \cdot 10^4.$$

Рассмотрим теперь систему линейных алгебраических уравнений с матрицей A и точным решением $x = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$. Тогда, очевидно, $f = \begin{pmatrix} 0.9998 \\ 1 \end{pmatrix}$. Если же теперь взять $\tilde{f} = \begin{pmatrix} 1.0000 \\ 1.0000 \end{pmatrix}$, то решение новой системы будет иметь вид $\tilde{x} = \begin{pmatrix} 0.3333 \\ 0.0000 \end{pmatrix}$, т.е. незначительная погрешность в задании лишь одной компоненты вектора-столбца правых частей влечет за собой недопустимое искажение решения.

Полученный результат, между тем, довольно точно предсказывается оценкой (4.8). Действительно,

$$\delta f = \begin{pmatrix} 0.0002 \\ 0 \end{pmatrix}$$

и

$$\|\delta f\|_{\mathrm{I}} = 2 \cdot 10^{-4}, \ \|f\|_{\mathrm{I}} = 1, \ \|x\|_{\mathrm{I}} = 5;$$

$$\delta x = \begin{pmatrix} 4.6667 \\ 2 \end{pmatrix}$$

и

$$\|\delta x\|_{\mathrm{I}} = 4.6667.$$

Отсюда (поскольку $\nu(A) \approx 4 \cdot 10^4$),

$$\frac{\|\delta x\|}{\|x\|} \le \frac{4.6667}{5} \approx 0.93... \le 4\cdot 10^4 \cdot 2\cdot 10^{-4} = 8\,.$$

Таким образом, истинная величина относительной погрешности не превышает предсказываемой теорией.

Заметим, что может создаться впечатление, что причиной плохой обусловленности задачи является малость определителя $\det A$. Однако на самом деле это не совсем так, хотя последнее и может являться одной из причин.

Действительно, так как $\det A = \lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_n$, то, зафиксировав λ_n и рассмотрев последовательность матриц A_k таких, что $|\lambda_{1k}| \to 0$, получим:

$$\det A_k \to 0 \quad \text{и} \quad \nu(A_k) \to \infty\,.$$

Однако решающую роль все-таки играют не величины собственных значений сами по себе, а их разброс, о чем говорит следующий пример.

2^0. Рассмотрим последовательность симметричных матриц A_n, $n = 2m$, $m = 1,2,...$, обладающих следующим свойством: собственными значениями матрицы A_n являются числа λ_1 и λ_2, причем каждое из них имеет кратность m. Тогда

$$\det A_n = (\lambda_1 \cdot \lambda_2)^m,\ \ \nu(A_n) = \frac{|\lambda_2|}{|\lambda_1|}\,.$$

Таким образом, если $|\lambda_1 \cdot \lambda_2| < 1$, то $\det A_n \to 0$, в то время как число обусловленности $\nu(A)$ остается постоянным.

Классическим примером плохо обусловленной матрицы может служить ***матрица Гильберта*** $H = \left(h_{ij}\right)_{i,j=1}^{n}$, элементы которой определяются формулой $h_{ij} = \dfrac{1}{i+j-1}$. Для нее, например, при $n = 11$ $\nu(H) \approx 10^{16}$.

§ 5. Применение ортогональных преобразований для решения систем линейных алгебраических уравнений

Задача (1) характеризуется некоторым числом обусловленности $\nu(A)$, которое задано и не зависит от метода решения. Естественно потребовать, чтобы численный алгоритм не вносил дополнительных изменений в степень обусловленности исходной системы, т.е. обеспечивал достаточную вычислительную устойчивость. К сожалению, рассмотренные выше методы не всегда удовлетворяют указанному требованию. Например, так как в методе Гаусса с выбором главного элемента по столбцу переход от системы (1) к системе с верхней треугольной матрицей осуществляется по схеме (см. (1.14))

$$U = L_n L_{n-1} P_{n-1} ... L_1 P_1 A\,.$$

то на основании свойства чисел обусловленности справедливо неравенство

$$\nu(U) \le \nu(L_n)\cdot\nu(L_{n-1})\cdot\nu(P_{n-1})\cdot\ldots\cdot\nu(L_1)\cdot\nu(P_1)\cdot\nu(A)\,.$$

Поскольку для сферического числа обусловленности справедливы соотношения (покажите!)

$$\nu_{\text{III}}(P_k) = 1,$$
$$\nu_{\text{III}}(L_k) = \begin{cases} |a_{kk,k-1}|, & \text{если } |a_{kk,k-1}| > 1, \\ |a_{kk,k-1}|^{-1}, & \text{если } |a_{kk,k-1}| < 1, \end{cases}$$

то при существенном росте (или уменьшении) элементов матриц L_k в соответствии с оценками (4.8), (4.11) обратный ход в методе Гаусса может сопровождаться большой потерей точности. Такого нежелательного явления можно избежать, если осуществлять приведение матрицы A к верхнему треугольному виду с помощью ортогональных преобразований (здесь и далее будем иметь в виду случай вещественных матриц, хотя, очевидно, все сказанное остается в силе, если заменить слово «ортогональные» на слово «унитарные», и в комплексном случае).

Напомним, что матрица Q называется ортогональной, если выполняются равенства

$$QQ^T = Q^T Q = E.$$

Отсюда непосредственно имеем:

1^0. $Q^T = Q^{-1}$.

2^0. Произведение любого конечного числа ортогональных матриц есть матрица ортогональная.

3^0. $\nu_{\text{III}}(Q) = 1$.

Первые два свойства достаточно очевидны, а третье следует из того, что

$$\|Q\|_{\text{III}} = \sqrt{\lambda_{\max}\left(QQ^T\right)} = 1 = \left\|Q^{-1}\right\|.$$

Из свойства 3 непосредственно получаем:

$$\nu_{\text{III}}(QA) = \nu_{\text{III}}(A), \tag{5.1}$$

т.е. умножение матрицы A на некоторую ортогональную матрицу Q не меняет сферического числа обусловленности исходной матрицы.

Действительно, с одной стороны в силу свойства 2 числа обусловленности

$$\nu(AB) \le \nu(A)\cdot\nu(B),$$

а с другой, поскольку

$$\nu(A) \le \nu\left(B^{-1}\right)\cdot\nu(AB) = \nu(B)\cdot\nu(AB),$$

то

$$\nu(AB) \ge \frac{1}{\nu(B)}\nu(A).$$

Аналогично получаем неравенство

$$\nu(AB) \ge \frac{1}{\nu(A)}\nu(B).$$

Поэтому

$$\max\left\{\frac{1}{\nu(B)}\nu(A), \frac{1}{\nu(A)}\nu(B)\right\} \le \nu(AB) \le \nu(A)\cdot\nu(B).$$

Положив в последнем неравенстве $\nu(B)=1$, получим:

$$\max\left\{\frac{1}{\nu(A)},\ \nu(A)\right\}\le\nu(AB)\le\nu(A),$$

откуда и следует (5.1).

Справедлива

Теорема. Всякую невырожденную матрицу A можно представить в виде произведения ортогональной и верхней треугольной матриц, т.е.

$$A=QU,$$

где Q – ортогональная, а U – верхняя треугольная матрица.

Доказательство.

Так как матрица A^TA – симметричная и положительно определенная, то для нее справедливо разложение Холесского (см. (2.11)), т.е.

$$A^TA=U^TU,$$

где U – верхняя треугольная матрица. Теперь, записав матрицу A в виде

$$A=\left(AU^{-1}\right)U,$$

покажем, что матрица AU^{-1} является ортогональной. Действительно,

$$\left(AU^{-1}\right)\left(AU^{-1}\right)^T=AU^{-1}\left(U^{-1}\right)^TA^T=A\left(U^TU\right)^{-1}A^T=A\left(A^TA\right)^{-1}A^T=AA^{-1}\left(A^T\right)^{-1}A^T=E.$$

Установленное равенство завершает доказательство. ⊠

Таким образом, доказана возможность приведения системы (1) к эквивалентной системе с треугольной матрицей с помощью ортогональных преобразований, причем, как это следует из сказанного выше, характеристики вычислительного процесса будут определяться только свойствами исходной матрицы.

5.1. Метод отражений

Определим ***матрицу отражения*** (или матрицу ***Хаусхолдера***) следующим образом:

$$V=E-2\omega\omega^T,\tag{5.2}$$

где ω – вектор-столбец единичной длины в сферической норме (т.е. $(\omega,\omega)=\omega^T\omega=1$).

Матрица отражения обладает следующими свойствами:

1^0. Матрица V – симметричная.

Действительно, так как $\omega\omega^T=\left\{\omega_i\omega_j\right\}_{i,j=1}^n$, то

$$V_{ij}=\delta_{ij}-2\omega_i\omega_j=V_{ji}.$$

2^0. Матрица V – ортогональная.

Имеем:

$$VV^T = \left(E - 2\omega\omega^T\right)\cdot\left(E - 2\omega\omega^T\right) = E - 2E\omega\omega^T - 2\omega\omega^T E + 4\omega\left(\omega^T\omega\right)\omega^T =$$
$$= E - 4\omega\omega^T + 4\omega\omega^T = E.$$

3⁰. Матрица V оставляет без изменения любой вектор, ортогональный вектору ω, т.е. если $(x,\omega) = \omega^T x = 0$, то $Vx = x$.

Действительно,

$$Vx = x - 2\omega\omega^T x = x - 2\omega\left(\omega^T x\right) = x - 2\omega\cdot 0 = x.$$

4⁰. Матрица V переводит в противоположный любой вектор, коллинеарный вектору ω, т.е. если $x = \lambda\omega$, $\lambda \in R$, то $Vx = -x$.

Действительно,

$$Vx = x - 2\omega\omega^T x = \lambda\omega - 2\omega\omega^T\lambda\omega = \lambda\omega - 2\lambda\omega\left(\omega^T\omega\right) = -\lambda\omega = -x.$$

Решим теперь задачу о том, как с помощью матрицы отражения вида (5.2) перевести произвольный ненулевой вектор s в заданный вектор e единичной длины, т.е. как определить матрицу V и число α, чтобы имело место равенство

$$Vs = \alpha e. \tag{5.3}$$

Используя (5.2), получаем:

$$\left(E - 2\omega\omega^T\right)s = \alpha e.$$

Отсюда очевидным образом следует равенство

$$s - \alpha e = 2\omega\omega^T s = 2\omega\left(\omega^T s\right) = 2(\omega,s)\omega, \tag{5.4}$$

т.е.

$$\omega = \kappa(s - \alpha e), \text{ где } \kappa = \frac{1}{2(s,\omega)}, \tag{5.5}$$

Таким образом, искомый вектор ω коллинеарен вектору $s - \alpha e$. Подставляя (5.5) в (5.4), будем иметь:

$$s - \alpha e = 2(s,\kappa(s - \alpha e))\kappa(s - \alpha e)$$

или

$$s - \alpha e = 2\kappa^2(s, s - \alpha e)(s - \alpha e).$$

Последнее равенство равносильно равенству

$$\left[2\kappa^2(s, s - \alpha e) - 1\right](s - \alpha e) = 0$$

и, следовательно,

$$2\kappa^2(s, s - \alpha e) - 1 = 0.$$

Отсюда

$$\kappa = \frac{1}{\sqrt{2(s, s - \alpha e)}}. \tag{5.6}$$

Поскольку векторы s и e заданы, то осталось задать числовой параметр α. Сделаем это таким образом, чтобы выполнялось неравенство

$$(s, s-\alpha e) > 0.$$

Очевидно, условию удовлетворяет

$$\alpha = \sqrt{(s,s)}. \tag{5.7}$$

Действительно, тогда

$$(s, s-\alpha e) = (s,s) - \sqrt{(s,s)}(s,e) \geq \left[\text{неравенство Коши – Буняковского: } (s,e)^2 \leq (s,s)(e,e)\right] \geq$$
$$\geq (s,s) - \sqrt{(s,s)}\sqrt{(s,s)}\sqrt{(e,e)} = 0.$$

При этом знак равенства имеет место лишь в случае, когда $s = \lambda e$.

Таким образом, чтобы матрица отражения V, задаваемая формулой (5.2), удовлетворяла условию (5.3), в котором s и e – заданные векторы, необходимо положить

$$\begin{aligned} &\omega = \kappa(s - \alpha e), \\ &\alpha = \sqrt{(s,s)}, \\ &\kappa = \frac{1}{\sqrt{2(s, s-\alpha e)}}. \end{aligned} \tag{5.8}$$

Теперь задача разложения произвольной невырожденной матрицы A в произведение ортогональной и верхней треугольной может быть решена следующим образом (схема ***метода отражений***):

1-й этап:

Используя формулы (5.8), образуем матрицу отражения V_1 по векторам $s_1 = (a_{11}, a_{21}, ..., a_{n1})^T$ и $e_1 = (1, 0, ..., 0)^T$. Умножив исходную систему слева на матрицу V_1, получим систему

$$A^{(1)}x = f^{(1)}, \tag{5.9}$$

где в матрице $A^{(1)} = V_1 A$ все подиагональные элементы первого столбца равны нулю, а формулы для вычисления всех остальных ее элементов $a_{ij,1}$, а также элементов вектора $f^{(1)} = V_1 f$ ($f_{i,1}$) имеют вид

$$\begin{aligned} &a_{11,1} = \alpha_1; \\ &a_{ij,1} = a_{ij} - 2(b_j, \omega_1)\omega_{i,1}, \ \ i = 1, ..., n; \ \ j = 2, ..., n; \\ &f_{i,1} = f_i - 2(f, \omega_1)\omega_{i,1}, \ \ i = 1, ..., n, \end{aligned} \tag{5.10}$$

(здесь $b_j = (a_{1j}, ..., a_{nj})^T$ и $(b_j, \omega_1) = \sum_{p=1}^{n} b_{pj}\omega_{p,1}$) и непосредственно следуют из (5.2).

На втором этапе аналогично образуем матрицу V_2 по векторам $s_2 = (0, a_{22,1}, ..., a_{n2,1})^T$ и $e_2 = (0, 1, 0, ..., 0)^T$. Умножив (5.9) слева на матрицу V_2, перейдем к системе

$$A^{(2)}x = f^{(2)}, \tag{5.11}$$

в матрице $A^{(2)}$ которой первая строка совпадает с первой строкой матрицы $A^{(1)}$ (поскольку матрица V_2 – блочно-диагональная (!) и первый блок – размера 1×1 – единичная матрица) и все поддиагональные элементы второго столбца равны нулю. Полные формулы пересчета будут иметь вид

$$\begin{aligned}
&a_{1j,2} = a_{1j,1}, \;\; j = 1,...,n;\\
&a_{22,2} = \alpha_2;\\
&a_{ij,2} = a_{ij,1} - 2\left(b_j^{(1)}, \omega_2\right)\omega_{i,2}, \;\; i = 2,...,n; \;\; j = 3,...,n;\\
&f_{i,2} = f_{i,1} - 2\left(f^{(1)}, \omega_2\right)\omega_{i,2}, \;\; i = 2,...,n.
\end{aligned} \tag{5.12}$$

(аналогично здесь $b_j^{(1)} = \left(a_{1j,1},...,a_{nj,1}\right)^T$ и $\left(b_j^{(1)}, \omega_2\right) = \sum_{p=2}^{n} b_{pj}^{(1)}\omega_{p,2}$. Обратите внимание: суммирование здесь проводится в пределах от 2 до n, а не от единицы, как на первом этапе. Это происходит потому, что у вектора ω_2 первая компонента равна нулю. Поэтому иногда и у вектора $b_j^{(1)}$ первую компоненту также пллагают равной нулю).

По аналогии с описанным выше преобразования k-го этапа осуществляются с помощью матрицы отражения V_k, образованной по векторам $s_k = \left(0,...,0,a_{kk,k-1},...,a_{nk,k-1}\right)^T$ и $e_k = (0,...,0,1,0,...,0)^T$ (у последнего единственная отличная от нуля компонента стоит на k-м месте) и приводят при любом $1 \le k \le n-1$ к системе

$$A^{(k)}x = f^{(k)}, \tag{5.13}$$

где

$$\begin{aligned}
&a_{ij,k} = a_{ij,k-1}, \;\; i = 1,...,k-1; \;\; j = 1,...,n;\\
&a_{kk,k} = \alpha_k;\\
&a_{ik,k} = 0, \;\; i = k+1,...,n;\\
&a_{ij,k} = a_{ij,k-1} - 2\left(b_j^{(k-1)}, \omega_k\right)\omega_{i,k}, \;\; i = k,...,n; \;\; j = k+1,...,n;\\
&f_{i,k} = f_{i,k-1} - 2\left(f^{(k-1)}, \omega_k\right)\omega_{i,k}, \;\; i = k+1,...,n.
\end{aligned} \tag{5.14}$$

Как и выше, здесь $b_j^{(k-1)} = \left(a_{1k,k-1},...,a_{nk,k-1}\right)^T$ и $\left(b_j^{(k-1)}, \omega_k\right) = \sum_{p=k}^{n} b_{pj}^{(k-1)}\omega_{p,k}$.

После выполнения $(n-1)$ этапов будем иметь систему

$$A^{(n-1)}x = f^{(n-1)},$$

матрица которой – верхняя треугольная. Далее остается для нахождения неизвестных выполнить обратный ход, аналогичный обратному ходу метода Гаусса.

Упражнение. Выполнить подсчет количества арифметических операций в методе отражения.

5.2. Метод вращений

В изложенном выше методе отражений исключение неизвестных на каждом шаге производилось с помощью матриц отражения. Эта же задача может быть решена также и с помощью элементарных ортогональных матриц, которые носят название ***матриц вращения*** (или ***матриц Гивенса***) и имеют вид

$$T_{ij} = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cos\varphi & & & -\sin\varphi & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & \sin\varphi & & & \cos\varphi & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{pmatrix}, \qquad (5.15)$$

причем все необозначенные элементы – нулевые, а внличины $\cos\varphi$ и $\pm\sin\varphi$ расположены на пересечении строк и столбцов с номерами i и j.

Легко видеть, что, умножив произвольную невырожденную матрицу A слева на матрицу T_{ij}, получим новую невырожденную матрицу B, у которой элементы i-й и j-й строк определяются по формулам

$$\begin{cases} b_{ik} = a_{ik}\cos\varphi - a_{jk}\sin\varphi, \\ b_{jk} = a_{ik}\sin\varphi + a_{jk}\cos\varphi, \end{cases} \quad k = \overline{1,n}, \qquad (5.16)$$

а остальные элементы матрицы B такие же, как и соответствующие элементы матрицы A.

Если мы теперь захотим обратить в нуль элемент b_{js} матрицы B (что эквивалентно исключению из j-го уравнения системы (1) неизвестного x_s с помощью операции $T_{ij}Ax = T_{ij}f$), то необходимо во второй изформул (5.16) положить $k = s$. Тогда, прираняв правую ее часть к нулю, получим:

$$a_{is}\sin\varphi + a_{js}\cos\varphi = 0, \qquad (5.17)$$

откуда, если $a_{js} \neq 0$, имеем:

$$\operatorname{tg}\varphi = -\frac{a_{js}}{a_{is}}$$

и по соответствующим формулам тригонометрии персчитываем элементы матрицы вращений

$$\begin{aligned} \cos\varphi &= \frac{1}{\sqrt{1+\operatorname{tg}^2\varphi}} = \frac{a_{is}}{\sqrt{a_{is}^2 + a_{js}^2}}, \\ \sin\varphi &= \cos\varphi\cdot\operatorname{tg}\varphi = -\frac{a_{js}}{\sqrt{a_{is}^2 + a_{js}^2}}. \end{aligned} \qquad (5.18)$$

Если же $a_{js} = 0$, то из (5.17) следует $\sin\varphi = 0$ и, следовательно, $\cos\varphi = 1$, .т.е. матрица T_{ij} в этом случае совпадает с единичной.

Полученный результат позволяет сформулировать следующую схему ***метода вращений***:

1-й этап:

Последовательное выполнение умножений на матрицы вращений T_{12}, T_{13},...,T_{1n} и, следовательно, каждый раз пересчет элементов 1-й и 2-й, затем, 1-й и 3-й и так далее, 1-й и n-й строк по формулам типа (5.16), в которых компоненты вращения вычисляются по формулам типа (5.18):

$$\begin{cases} a_{1j}^{1,k} = a_{1j}^{1,k-1}\cos\varphi_{1,k} - a_{kj}^{1,k-1}\sin\varphi_{1,k}, \\ a_{kj}^{1,k} = a_{1j}^{1,k-1}\sin\varphi_{1,k} + a_{kj}^{1,k-1}\cos\varphi_{1,k}, \quad j = 1,...,n;\ k = 2,3,...,n, \\ a_{ij}^{1,k} = a_{ij}^{1,k-1}, \quad i \neq 1,\ i \neq k, \end{cases} \tag{5.19}$$

$$\cos\varphi_{1,k} = \frac{a_{11}^{1,k-1}}{\sqrt{\left(a_{11}^{1,k-1}\right)^2 + \left(a_{k1}^{1,k-1}\right)^2}}, \quad \sin\varphi_{1,k} = -\frac{a_{k1}^{1,k-1}}{\sqrt{\left(a_{11}^{1,k-1}\right)^2 + \left(a_{k1}^{1,k-1}\right)^2}}. \tag{5.20}$$

Заметим, однако, что формулы (5.20) можно несколько упростить. Действительно, к началу выполнения k-го шага первого этапа элементы k-й строки еще не менялись. Поэтому $a_{k1}^{1,k-1} = a_{k1}$, т.е. совпадает с соответствующим элементом исходной матрицы. Кроме того, применяя первую из формул (5.19) при $k = 2$, $j = 1$, получаем:

$$a_{11}^{1,2} = a_{11}\cdot\frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}} - a_{21}\cdot\left(-\frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}\right) = \sqrt{a_{11}^2 + a_{21}^2}.$$

Поэтому

$$\cos\varphi_{1,3} = \frac{\sqrt{a_{11}^2 + a_{21}^2}}{\sqrt{a_{11}^2 + a_{21}^2 + a_{31}^2}}, \quad \sin\varphi_{1,3} = -\frac{a_{31}}{\sqrt{a_{11}^2 + a_{21}^2 + a_{31}^2}}.$$

Продолжая аналогичные рассуждения, по индукции будем иметь:

$$\cos\varphi_{1,k} = \frac{\sqrt{a_{11}^2 + a_{21}^2 + \ldots + a_{k-1,1}^2}}{\sqrt{a_{11}^2 + a_{21}^2 + \ldots + a_{k1}^2}}, \quad \sin\varphi_{1,k} = -\frac{a_{k1}}{\sqrt{a_{11}^2 + a_{21}^2 + \ldots + a_{k1}^2}}. \tag{5.21}$$

По окончании первого этапа получаем систему

$$A^{(1)}x = T_1 f,$$

где

$$A^{(1)} = T_1 A, \quad T_1 = T_{1n}\cdot\ldots\cdot T_{12}.$$

В матрице $A^{(1)}$ этой системы все элементы первого столбца, кроме a_{11}^1, равны нулю.

На втором этапе точно таким же образом поступаем со вторым столбцом (последовательно с помощью матриц T_{23},...,T_{2n}) и так далее. После $(n-1)$-го шага получим систему

$$A^{(n-1)}x = T_{n-1}\cdot\ldots\cdot T_1 f,$$

матрица которой является верхней треугольной.

5.3. Метод ортогонализации

Рассмотрим еще одну идею, которая также позволяет решить задачу о треугольном разложении.

Перепишем систему (1) в виде

$$(a_i, y) = 0, \ \ i = 1,2,...,n, \tag{5.22}$$

где

$$a_i = (a_{i1},...,a_{in},-f_i)^T, \ \ i = 1,2,...,n,$$
$$y = (x_1,...,x_n,1)^T.$$

Равенства (5.22) означают, что решение системы (1) эквивалентно нахождению вектора y, имеющего единичную последнюю координату и ортогонального ко всем линейно независимым векторам $a_1,...,a_n$, т.е. к любому базису подпространства P_n, натянутого на $a_1,...,a_n$. Верно и обратное утверждение: ортогональность к любому базису P_n влечет за собой и ортогональность к $a_1,...,a_n$. Это позволяет указать следующий путь для вычисления вектора y: строим какой-либо ортогональный базис подпространства P_n и находим вектор z, ортогональный к этому базису.

Конкретный способ реализации этого пути может выглядеть следующим образом:

1) добавим к векторам $a_1,...,a_n$ еще один вектор, вместе с ними образующий линейно независимую систему. Таким вектором может быть $a_{n+1} = (0,...,0,1)^T$ (линейная независимость такой системы следует из того, что определитель матрицы, столбцами которой являются векторы $a_1,...,a_n,a_{n+1}$ равен определителю исходной матрицы A);

2) к системе $a_1,...,a_n,a_{n+1}$ применим процесс ортогонализации Грамма-Шмидта:

$$\begin{aligned} &v_1 = a_1, \ \ b_1 = \frac{v_1}{\sqrt{(v_1,v_1)}}, \\ &v_k = a_k - \sum_{i=1}^{k-1} c_{ki} v_i, \ \ c_{ki} = (a_k, b_i), \ \ b_k = \frac{v_k}{\sqrt{(v_k,v_k)}}, \ \ k = 2,...,n+1. \end{aligned} \tag{5.23}$$

Векторы $a_1,...,a_n$ линейно выражаются через $b_1,...,b_n$. Поэтому вектор v_{n+1} ортогонален ко всем векторам $a_1,...,a_n$ и, таким образом решение исходной линейной системы (1) можно вычислить по формуле

$$x_i = \frac{z_i}{z_{n+1}}, \ \ i = 1,...,n, \tag{5.24}$$

где z_i – компоненты вектора v_{n+1}. Заметим при этом, что формула (5.24) корректна, т.е. ее знаменатель z_{n+1} отличен от нуля. Действительно, предполагая, что $z_{n+1} = 0$, мы получим следующее: условия

$$(v_{n+1}, a_i) = 0, \ \ i = 1,...,n$$

приводят к системе

$$\sum_{j=1}^{n} a_{ij} z_j = 0, \ \ i = 1,...,n,$$

которая должна иметь ненулевое решение (в противном случае в ортонормированной системе присутствует нулевой вектор), что невозможно в силу исходного предположения о невырожденности матрицы A.

С точки зрения матричных операций нормировка первой строки матрицы A (т.е. построение вектора b_1) эквивалентна умножению слева матрицы A на диагональную матрицу

$$D_1 = \mathrm{diag}\left\{\frac{1}{\sqrt{(v_1, v_1)}}, 1, ..., 1\right\}.$$

В результате такого умножения получим матрицу $B_1 = D_1 A$.

Вычисление по формуле (5.23) вектора v_2 эквивалентно умножению слева матрицы B_1 на матрицу

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -c_{21} & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \dots & \dots & \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}, \quad c_{21} = (a_2, b_1).$$

Аналогично получаем: на k-м шаге процесса ортогонализации вычисление вектора v_k равносильно умножению матрицы, полученной после выполнения предыдущего шага на

$$M_k = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ & & \ddots & & & & & \\ 0 & 0 & & 1 & 0 & 0 & \dots & 0 \\ -c_{k1} & -c_{k2} & \dots & -c_{kk-1} & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ & & \dots & & \dots & \dots & \ddots & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{pmatrix}, \quad c_{kj} = (a_k, b_j), \quad j = 1, ..., k-1,$$

а его нормировка (т.е. получение вектора b_k) означает умножение слева матрицы последнего шага на диагональную матрицу

$$D_k = \mathrm{diag}\left\{1, ..., 1, \frac{1}{\sqrt{(v_k, v_k)}}, 1, ..., 1\right\}.$$

В результате после выполнения n этапов получим ортогональную матрицу

$$B_n = D_n M_n D_{n-1} M_{n-1} \dots D_2 M_2 D_1 A, \tag{5.25}$$

откуда следует, что

$$A = L B_n,$$

где $L = (D_n M_n \dots D_2 M_2 D_1)^{-1}$ – нижняя треугольная матрица, т.е. (5.25) – ортогонально-треугольное разложение.

ГЛАВА II

Итерационные методы решения систем линейных алгебраических уравнений

Напомним, что основное отличие итерационных методов от рассмотренных выше прямых состоит в том, что точное решение здесь может быть получено лишь в пределе некоторого бесконечного процесса приближений. Поэтому вначале освежим в памяти некоторые вопросы, связанные со сходимостью матричных последовательностей. Прежде всего, отметим, что имеет место

***Лемма* 1.** Для того чтобы имело место соотношение $A^m \underset{m\to\infty}{\to} 0$, необходимо и достаточно, чтобы все собственные значения матрицы A были по модулю меньше единицы.

Доказательство.

Известно, что с помощью преобразования подобия, не меняющего собственных значений матрицы, исходная матрица A всегда может быть приведена к канонической форме Жордана:

$$I = C^{-1}AC,$$

где C – матрица подобия, а I – квазидиагональная матрица

$$I = \operatorname{diag}\{I_{\tau_1}(\lambda_1),...,I_{\tau_r}(\lambda_r)\},$$

r – число канонических клеток Жордана вида

$$I_\tau(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & \dots & 0 & 0 \\ 1 & \lambda & 0 & \dots & 0 & 0 \\ 0 & 1 & \lambda & \dots & 0 & 0 \\ & & \dots & \dots & & \\ 0 & 0 & 0 & \dots & \lambda & 0 \\ 0 & 0 & 0 & \dots & 1 & \lambda \end{pmatrix}.$$

При этом r равно числу линейно независимых собственных векторов матрицы A и

$$\sum_{i=1}^{r} \tau_i = n.$$

Тогда

$$A = CIC^{-1}$$

и

$$A^m = CI^mC^{-1}.$$

Поэтому матрицы A^m и I^m стремятся к нулю при $m \to \infty$ одновременно, а поскольку

$$I^m = \operatorname{diag}\{I^m_{\tau_1}(\lambda_1),...,I^m_{\tau_r}(\lambda_r)\},$$

то для выполнения сходимости $A^m \underset{m\to\infty}{\to} 0$ достаточно установить лишь условия сходимости $I^m_\tau(\lambda) \underset{m\to\infty}{\to} 0$. Непосредственной проверкой несложно установить, что

$$I_\tau^m(\lambda)=\begin{pmatrix} \lambda^m & 0 & 0 & \dots & 0 & 0 \\ \frac{(\lambda^m)'}{1!} & \lambda^m & 0 & \dots & 0 & 0 \\ \frac{(\lambda^m)''}{2!} & \frac{(\lambda^m)'}{1!} & \lambda^m & \dots & 0 & 0 \\ & \dots & & \dots & & \\ \frac{(\lambda^m)^{(\tau-1)}}{(\tau-1)!} & \frac{(\lambda^m)^{(\tau-2)}}{(\tau-2)!} & \frac{(\lambda^m)^{(\tau-3)}}{(\tau-3)!} & \dots & \frac{(\lambda^m)'}{1!} & \lambda^m \end{pmatrix}.$$

Отсюда видно, что для сходимости $I_\tau^m(\lambda)\underset{m\to\infty}{\to}0$ необходимо и достаточно выполнение условия $|\lambda|<1$. ⊠

Сформулированный критерий неудобен для проверки, так как требует информации о собственных значениях матрицы A. Более удобным в этом плане является следующий ***достаточный*** признак:

***Лемма* 2.** Для того чтобы $A^m\underset{m\to\infty}{\to}0$, достаточно, чтобы хоть одна из норм матрицы A была меньше единицы.

Доказательство.

Чтобы установить, что $A^m\underset{m\to\infty}{\to}0$, по определению достаточно проверить, что $\|0-A^m\|\underset{m\to\infty}{\to}0$ хотя бы для одной из норм матрицы A. Имеем:

$$\|0-A^m\|=\|A^m\|\le\|A^{m-1}\|\cdot\|A\|\le\ldots\le\|A\|^m\underset{m\to\infty}{\to}0,$$

если $\|A\|<1$. ⊠

Теперь мы можем исследовать вопрос о поведении матричной геометрической прогрессии

$$E+A+A^2+\ldots+A^m+\ldots.$$

***Теорема* 1.** Для того чтобы ряд $E+A+A^2+\ldots+A^m+\ldots$ сходился, необходимо и достаточно, чтобы $A^m\underset{m\to\infty}{\to}0$. В этом случае матрица $E-A$ имеет обратную и

$$E+A+A^2+\ldots+A^m+\ldots=(E-A)^{-1}.$$

Доказательство.

Необходимость следует из того, что по определению сходимость матричной последовательности (ряда) равносильна сходимости n^2 скалярных последовательностей (рядов). Тогда условие стремления к нулю n-го члена есть необходимое условие сходимости любого числового ряда.

Достаточность. Если $A^m\underset{m\to\infty}{\to}0$, то, по Лемме 1, все собственные значения матрицы A по модулю меньше единицы. Тогда все собственные значения матрицы $E-A$ $\lambda_i(E-A)=1-\lambda_i(A)$ будут отличны от нуля. Следовательно,

$$\det(E-A)=\prod_{i=1}^{n}\lambda_i(E-A)\ne0.$$

Поэтому существует матрица $(E-A)^{-1}$.

Рассмотрим тождество

$$\left(E+A+A^2+\ldots+A^m\right)\cdot(E-A)=E-A^{m+1}.$$

Умножив его справа на матрицу $(E-A)^{-1}$, получим:

$$E+A+A^2+\ldots+A^m=(E-A)^{-1}-A^{m+1}(E-A)^{-1}$$

и поскольку $A^m \underset{m\to\infty}{\to} 0$, то, переходя к пределу при $m\to\infty$, получим:

$$E+A+A^2+\ldots+A^m+\ldots \underset{m\to\infty}{\to}(E-A)^{-1}.$$

$\boxtimes$

С учетом Леммы 1 критерий сходимости матричной геометрической прогрессии может быть сформулирован в виде

***Теорема* 2.** Для того чтобы ряд $E+A+A^2+\ldots+A^m+\ldots$ сходился, необходимо и достаточно, чтобы все собственные значения матрицы A были меньше единицы по модулю.

С учетом же Леммы 2, а также оценки спектрального радиуса матрицы можно дать более простой для проверки достаточный признак сходимости матричной геометрической прогрессии:

***Теорема* 3.** Если какая-либо норма матрицы A меньше единицы, то ряд

$$E+A+A^2+\ldots+A^m+\ldots$$

сходится.

Теперь мы готовы перейти к изучению материала, обозначенного в названии главы.

§ 1. Основные разновидности итерационных процессов

Объектом нашего изучения, как и в предыдущей главе, будет система линейных алгебраических уравнений

$$Ax=f \tag{1.1}$$

с неособенной матрицей A. При построении итерационных методов решения таких систем достаточно часто исходную систему (1.1) преобразуют к эквивалентному виду

$$x=Bx+b. \tag{1.2}$$

Если такое преобразование выполнено, то последовательность приближений $x^{(n)}$ к решению $x^{(*)}$ этой системы можно строить, например, по формулам

$$x^{(k+1)}=Bx^{(k)}+b,\ \ k=0,1,\ldots, \tag{1.3}$$

причем начальный вектор $x^{(0)}$ выбирают обычно из некоторых дополнительных соображений, хотя (именно в случае линейных систем) это можно сделать и произвольным образом (например, $x^{(0)}=b$).

Приведение системы (1.1) к виду (1.2) может быть осуществлено по-разному. Например, если C – некоторая неособенная матрица, то можно записать

$$x = x + C(f - Ax).$$

Тогда

$$B = E - CA, \;\; b = Cf$$

и алгоритм (1.3) примет вид

$$x^{(k+1)} = x^{(k)} + C\left(f - Ax^{(k)}\right), \;\; k = 0,1,\ldots. \tag{1.4}$$

Если преобразование, подобное описанному, проводить на каждом шаге итераций с новой, вообще говоря, матрицей C, то мы придем к алгоритму

$$x^{(k+1)} = x^{(k)} + C^{(k)}\left(f - Ax^{(k)}\right), \;\; k = 0,1,\ldots \tag{1.5}$$

или

$$x^{(k+1)} = B^{(k)}x^{(k)} + b^{(k)}, \;\; k = 0,1,\ldots. \tag{1.6}$$

Итерационные процессы вида (1.6) (или (1.5)) называют ***нестационарными*** в отличие от ***стационарных*** вида (1.3) (или (1.4)).

При построении итерационных процессов для решения системы (1.1) последнюю можно преобразовывать также и к виду

$$Px + Qx = b,$$

где

$$P + Q = CA, \;\; b = Cf,$$

а C – неособенная матрица.

В этом случае также можно строить итерационные алгоритмы:
стационарный вида

$$Px^{(k+1)} + Qx^{(k)} = b, \;\; k = 0,1,\ldots \tag{1.7}$$

и нестационарный вида

$$P^{(k)}x^{(k+1)} + Q^{(k)}x^{(k)} = b^{(k)}, \;\; k = 0,1,\ldots. \tag{1.8}$$

В отличие от ***явных*** итерационных процессов вида (1.3) – (1.6) процессы (1.7) – (1.8) называют ***неявными*** (поскольку для нахождения приближения $x^{(k+1)}$ нужно решать линейную систему с матрицей P или $P^{(k)}$). В качестве матрицы P, учитывая сказанное, как правило, берут треугольную или диагональную.

Все рассмотренные здесь итерационные процессы являются ***линейными*** и ***одношаговыми***, поскольку следующее приближение $x^{(k+1)}$ выражается только через предыдущее приближение $x^{(k)}$, причем это выражение осуществляется с помощью линейного отображения.

Можно, вообще говоря, рассматривать и итерационные процессы более общего вида, например

$$x^{(k+1)} = \varphi^{(k)}\left(x^{(k-p+1)},\ldots,x^{(k)}\right), \tag{1.9}$$

где $\varphi^{(k)}$ – некоторая функция, зависящая от матрицы A, вектора f и указанных начальных приближений. (1.9) – пример ***явного p-шагового (нелинейного) нестационарного*** итерационного процесса.

§ 2. Метод простой итерации

Будем считать, что система (1.1) преобразована к виду (1.2) (будем говорить: ***приведена к виду, удобному для итерации***). Тогда итерационный процесс вида (1.3)

$$x^{(k+1)} = Bx^{(k)} + b, \quad k = 0,1,\ldots, \tag{2.1}$$

обычно и называют методом простой итерации. Очевидно, это явный линейный одношаговый процесс. В силу линейности последовательность приближений всегда может быть построена. Если эта последовательность сходится, т.е. $x^{(k)} \underset{k\to\infty}{\to} x^{(*)}$, то это предельное значение $x^{(*)}$ будет решением нашей системы. Действительно, переходя к пределу в равенстве (2.1), получим:

$$x^{(*)} = Bx^{(*)} + b.$$

Поэтому возникает вопрос: когда последовательность приближений сходится? Ответ на этот вопрос дает следующая

***Теорема* 1.** (критерий сходимости) Для того чтобы метод простой итерации (2.1) сходился при любом начальном приближении $x^{(0)}$, необходимо и достаточно, чтобы все собственные значения матрицы B были по модулю меньше единицы.

Доказательство.

Достаточность. Выразим приближение любого номера через начальное:

$$\begin{aligned} x^{(k+1)} &= Bx^{(k)} + b = B\left(Bx^{(k-1)} + b\right) + b = B^2 x^{(k-1)} + (E+B)b = \cdots = \\ &= B^{k+1}x^{(0)} + \left(E + B + \ldots B^k\right)b. \end{aligned}$$

В условиях теоремы в силу Леммы 1 и Теоремы 1 из введения к текущей главе имеют место соотношения $B^{k+1} \underset{k\to\infty}{\to} 0$ и $E + B + B^2 + \ldots + B^k \underset{k\to\infty}{\to} (E-B)^{-1}$. Поэтому, переходя к пределу в полученном равенстве, получаем:

$$x^{(k+1)} \underset{k\to\infty}{\to} (E-B)^{-1}b = x^{(*)}.$$

Необходимость. Пусть $x^{(k)} \underset{k\to\infty}{\to} x^{(*)}$. Тогда

$$x^{(*)} = Bx^{(*)} + b$$

и

$$x^{(k)} - x^{(*)} = B\left(x^{(k-1)} - x^{(*)}\right) = B^2\left(x^{(k-2)} - x^{(*)}\right) = \ldots = B^k\left(x^{(0)} - x^{(*)}\right).$$

Переходя в этом равенстве к пределу и учитывая, что $x^{(0)}$ – произвольный вектор, имеем: $B^k \underset{k\to\infty}{\to} 0$, откуда по Лемме 1 и следует доказываемое утверждение. ⊠

Теорема 1,как это уже отмечалось относительно Леммы 1 из введения, не слишком удобна для практического применения. На основе свойства спектрального радиуса можно сформулировать более простой достаточный признак сходимости.

***Теорема* 2.** Для того чтобы метод простой итерации (2.1) сходился, достаточно, чтобы какая-либо норма матрицы B была меньше единицы.

Доказательство немедленно следует из того факта, что спектральный радиус матрицы не превосходит любой из ее норм, т.е. $|\lambda_i(B)| \le \|B\| < 1$, и применения Теоремы 1. ⊠

Следствие. Для того чтобы метод простой итерации (2.1) сходился, достаточно, чтобы для элементов b_{ij} матрицы B выполнялось одно из следующих услови1:

1) $\sum_{j=1}^{n}|b_{ij}|<1,\ \ i=1,...,n$ (тогда кубическая норма матрицы B меньше единицы);

2) $\sum_{i=1}^{n}|b_{ij}|<1,\ \ j=1,...,n$ (тогда октаэдрическая норма матрицы B меньше единицы).

Важным является также и вопрос о скорости сходимости итерационного процесса. Получить ответ на этот вопрос помогают оценки погрешности метода.

Справедлива

***Теорема* 3.** Если какая-либо норма матрицы B, согласованная с рассматриваемой нормой вектора, меньше единицы, то для погрешности метода простой итерации (2.1) справедлива оценка

$$\left\|x^{(*)}-x^{(л)}\right\|\le\|B\|^k\left\|x^{(0)}\right\|+\frac{1}{1-\|B\|}\|B\|^k\|b\|. \qquad (2.2)$$

Доказательство.

Так как

$$x^{(k)}=B^kx^{(0)}+\left(E+B+B^2+\ldots+B^{k-1}\right)b,$$

а с другой стороны

$$x^{(*)}=\left(E+B+B^2+\ldots+B^{k-1}+\ldots\right)b,$$

то, вычитая эти равенства, имеем:

$$x^{(*)}-x^{(k)}=\left(B^k+B^{k+1}+\ldots\right)b-B^kx^{(0)}$$

и поскольку $\|B\|<1$, то

$$\left\|x^{(*)}-x^{(k)}\right\|\le\|B\|^k(1+\|B\|+\ldots)\|b\|+\|B\|^k\left\|x^{(0)}\right\|=\|B\|^k\left\|x^{(0)}\right\|+\frac{1}{1-\|B\|}\|B\|^k\|b\|.$$

⊠

Если $x^{(0)}=b$, то оценка (2.2) упрощается. Действительно, тогда

$$\left\|x^{(*)}-x^{(k)}\right\|=\left\|\left(B^{k+1}+B^{k+2}+\ldots\right)b\right\|\le\frac{\|B\|^{k+1}\|b\|}{1-\|B\|}.$$

Так как

$$x^{(*)}-x^{(k)}=B\left(x^{(*)}-x^{(k-1)}\right),$$

то

$$\left\|x^{(*)}-x^{(k)}\right\|\le\|B\|\cdot\left\|x^{(*)}-x^{(k-1)}\right\|$$

или

$$\left\|\varepsilon^{(k)}\right\|\le\|B\|\cdot\left\|\varepsilon^{(k-1)}\right\|.$$

Это неравенство позволяет сопоставить погрешности приближенного решения на двух соседних шагах итерационного процесса и сделать вывод, что метод простой итерации сходится со скоростью геометрической прогрессии со знаменателем, равным $\|B\|$.

2.1. Основные приемы приведения систем к виду, удобному для итерации

Как мы уже отмечали выше, один из способов состоит в том, что исходную систему линейных алгебраических уравнений (1.1) преобразуют к виду

$$x = x + C(f - Ax). \tag{2.3}$$

Следовательно, успех такой процедуры (в смысле обеспечения сходимости и скорости последней) зависит от выбора матрицы C. Легко видеть, что в случае выполнения равенства $C = A^{-1}$ мы сразу получили бы точное решение исходной системы. Пользуясь этим, иногда проводят «грубое» обращение матрицы A, не заботясь о точности, и берут полученную матрицу в качестве матрицы C. Но такая процедура, как правило, очень трудоемка (хотя полученный итерационный процесс и может быть использован, например, для ***уточнения*** решений, получаемых по методу Гаусса). Поэтому чаще обращают не всю матрицу A, а лишь ее часть, т.е. представляют матрицу A в виде

$$A = R + S,$$

где R легко обращается и, что самое существенное, содержит основную информацию о матрице A. Тогда (2.3) примет вид

$$x = x + R^{-1}(f - (R+S)x) = -R^{-1}Sx + R^{-1}f = Bx + b. \tag{2.4}$$

Чаще всего указанный способ используют, когда матрица A имеет диагональное доминирование. Тогда можно записать

$$\begin{cases} x_1 + \dfrac{a_{12}}{a_{11}}x_2 + \ldots + \dfrac{a_{1n}}{a_{11}}x_n = \dfrac{f_1}{a_{11}}, \\ \qquad\qquad \ldots \\ \dfrac{a_{n1}}{a_{nn}}x_1 + \dfrac{a_{n2}}{a_{nn}}x_2 + \ldots + x_n = \dfrac{f_n}{a_{nn}}. \end{cases}$$

Теперь, выражая из i-го уравнения неизвестное x_i через остальные, получим канонический вид

$$\begin{cases} x_1 = 0\cdot x_1 - \dfrac{a_{12}}{a_{11}}x_2 - \ldots - \dfrac{a_{1n}}{a_{11}}x_n + \dfrac{f_1}{a_{11}}, \\ x_2 = -\dfrac{a_{21}}{a_{22}}x_1 + 0\cdot x_2 - \ldots - \dfrac{a_{2n}}{a_{22}}x_n + \dfrac{f_2}{a_{22}}, \\ \qquad\qquad \ldots \\ x_n = -\dfrac{a_{n1}}{a_{nn}}x_1 - \dfrac{a_{n2}}{a_{nn}}x_2 - \ldots + 0\cdot x_n + \dfrac{f_n}{a_{nn}}, \end{cases} \tag{2.5}$$

в котором

$$B = \begin{pmatrix} 0 & -\dfrac{a_{12}}{a_{11}} & \ldots & -\dfrac{a_{1n}}{a_{11}} \\ -\dfrac{a_{21}}{a_{22}} & 0 & \ldots & -\dfrac{a_{2n}}{a_{22}} \\ \ldots & \ldots & & \\ -\dfrac{a_{n1}}{a_{nn}} & -\dfrac{a_{n2}}{a_{nn}} & \ldots & 0 \end{pmatrix}, \quad b = \begin{pmatrix} \dfrac{f_1}{a_{11}} \\ \vdots \\ \dfrac{f_n}{a_{nn}} \end{pmatrix}. \tag{2.6}$$

Иногда для сокращения количества операций деления этот процесс организуют иначе: вместо $x_1,...,x_n$ находят

$$\begin{cases} a_{11}x_1 = f_1 - a_{12}x_2 - \ldots - a_{1n}x_n, \\ \qquad\qquad \ldots \\ a_{nn}x_n = f_n - a_{n1}x_1 - \ldots - a_{n,n-1}x_{n-1}. \end{cases} \tag{2.7}$$

Такой метод преобразования к каноническому виду ((2.5) или (2.7)) называют ***методом Якоби*** в явной или неявной форме. Он, очевидно, укладывается в общую схему (2.4), где

$$R = \mathrm{diag}\{a_{11},...,a_{nn}\}.$$

Как уже говорилось, рассчитывать на сходимость метода Якоби можно лишь в случае наличия диагонального доминирования у матрицы A исходной системы (другие условия мы сформулируем несколько позже). Если же это не так, то в качестве матрицы R можно выделить блочно-диагональную матрицу, например,

$$R = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & \ldots \\ a_{21} & a_{22} & 0 & 0 & \ldots \\ 0 & 0 & a_{33} & a_{34} & \ldots \\ 0 & 0 & a_{43} & a_{44} & \ldots \\ & & & & \ddots \end{pmatrix}.$$

Очевидно, явное нахождение R^{-1} здесь не представляет сложности, и итерационный процесс может быть легко построен по формулам (2.4).

С целью упрощения преобразований в (2.4) в качестве R иногда берут скалярную матрицу: $R = \alpha E$. Тогда

$$R^{-1} = \frac{1}{\alpha}E \qquad \text{и} \qquad B = E - \frac{1}{\alpha}A.$$

Если матрица A исходной системы – вещественная, положительная и симметричная, то несложно указать такой выбор скалярного параметра α, при котором итерационный процесс заведомо оказывается сходящимся. Для этого достаточно положить $\alpha = \|A\|$,т.е.

$$B = E - \frac{1}{\|A\|}A,$$

где $\|\cdot\|$ – какая-либо из норм матрицы.

Действительно, тогда собственные значения матрицы B $\lambda_i(B)$ могут быть вычислены по формуле

$$\lambda_i(B) = 1 - \frac{\lambda_i(A)}{\|A\|}, \quad i = 1,...,n,$$

причем $\lambda_i(A) > 0$, поскольку матрица A – положительная. А так как $|\lambda_i(A)| \le \|A\|$, то, очевидно, $0 \le \lambda_i(B) < 1$, т.е. соответствующий итерационный процесс будет сходящимся в силу Теоремы 1.

В общем случае матрица A симметричной не является. Однако линейную систему с любой неособенной вещественной матрицей A можно привести к эквивалентной системе с симметричной положительной матрицей с помощью ***трансформации Гаусса***:

умножая систему слева на матрицу A^T, переходим к системе

$$A^T Ax = A^T f\ . \tag{2.8}$$

Следовательно, к (2.8) можно применять описанный выше прием приведения к каноническому виду, который, таким образом, является достаточно универсальным.

§ 3. Метод Зейделя

Пусть линейная система (1.1) приведена к виду, удобному для итерации (1.2) и для нее записан итерационный процесс метода простой итерации (2.1). Распишем его в координатной форме:

$$x_i^{(k+1)} = \sum_{j=1}^{n} b_{ij} x_j^{(k)} + b_i\ ,\ i = 1,...,n;\ \ k = 0,1,...\ .$$

Очевидно, координаты вектора $x^{(k+1)}$ можно находить в любом порядке независимо друг от друга. Таким образом, метод простой итерации не использует две существенные возможности для улучшения алгоритма:
1) выбор порядка вычисления координат;
2) найденные уже «улучшенные» значения координат не принимают участия в вычислениях.

Зафиксируем пока порядок вычисления компонент вектора $x^{(k+1)}$ (а именно: в порядке возрастания номеров). Тогда для удовлетворения второго требования процесс, очевидно, нужно организовать следующим образом:

$$\begin{cases} x_1^{(k+1)} = b_{11}x_1^{(k)} + b_{12}x_2^{(k)} + \ldots + b_{1,n-1}x_{n-1}^{(k)} + b_{1n}x_n^{(k)} + b_1\ , \\ x_2^{(k+1)} = b_{21}x_1^{(k+1)} + b_{22}x_2^{(k)} + \ldots + b_{2,n-1}x_{n-1}^{(k)} + b_{2n}x_n^{(k)} + b_2\ , \\ \quad \ldots\ldots \\ x_n^{(k+1)} = b_{n1}x_1^{(k+1)} + b_{n2}x_2^{(k+1)} + \ldots + b_{n,n-1}x_{n-1}^{(k+1)} + b_{nn}x_n^{(k)} + b_n\ , \quad k = 0,1,... \end{cases} \tag{3.1}$$

В более компактной форме его можно записать так:

$$x_i^{(k+1)} = \sum_{j=1}^{i-1} b_{ij} x_j^{(k+1)} + \sum_{j=i}^{n} b_{ij} x_j^{(k)} + b_i\ ,\ \ i = 1,...,n;\ \ k = 0,1,... \tag{3.2}$$

Процесс (3.1), (3.2) называется ***итерационным процессом Зейделя***. Проведем исследование его сходимости. Для этих целей более удобной будет матричная форма записи. Вводя в рассмотрение матрицы

$$H = \begin{pmatrix} 0 & 0 & \ldots & 0 & 0 \\ b_{21} & 0 & \ldots & 0 & 0 \\ b_{31} & b_{32} & \ldots & 0 & 0 \\ & & \ldots & & \\ b_{n1} & b_{n2} & \ldots & b_{n,n-1} & 0 \end{pmatrix},\quad F = \begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1,n-1} & b_{1n} \\ 0 & b_{22} & \cdots & b_{2,n-1} & b_{2n} \\ 0 & 0 & \ldots & b_{3,n-1} & b_{3n} \\ & & \ldots & & \\ 0 & 0 & \ldots & 0 & b_{nn} \end{pmatrix},$$

(3.1) запишем в виде

$$x^{(k+1)} = Hx^{(k+1)} + Fx^{(k)} + b, \quad k = 0,1,...$$

или

$$(E - H)x^{(k+1)} = Fx^{(k)} + b, \quad k = 0,1,... .$$

Таким образом, метод Зейделя есть неявный одношаговый метод вида (1.7), где $P = E - H$, причем $\det P = 1$.

Так как P – невырожденная матрица, то метод Зейделя можно переписать в виде

$$x^{(k+1)} = (E - H)^{-1} Fx^{(k)} + (E - H)^{-1} b, \quad k = 0,1,.... .$$

Следовательно, метод Зейделя эквивалентен методу простой итерации с матрицей $B = (E - H)^{-1} F$. Это означает, что если исходную систему линейных алгебраических уравнений привести к виду, удобному для итерации

$$x = (E - H)^{-1} Fx + (E - H)^{-1} b \ ,$$

то получим метод Зейделя. Данный факт позволяет просто сформулировать критерий сходимости метода Зейделя:

***Теорема* 1.** Для того чтобы метод Зейделя сходился, необходимо и достаточно, чтобы все собственные значения матрицы $(E - H)^{-1} F$ были по модулю меньше единицы.

Таким образом, сходимость метода Зейделя связана с корнями уравнения

$$\det\left((E - H)^{-1} F - \lambda E\right) = 0 .$$

Упростим его:

$$\begin{aligned}\det\left((E - H)^{-1} F - \lambda E\right) &= \det\left((E - H)^{-1}(E - H)\left((E - H)^{-1} F - \lambda E\right)\right) = \\ &= \det\left((E - H)^{-1}(F + \lambda H - \lambda E)\right) = \det(E - H)^{-1} \cdot \det(F + \lambda H - \lambda E) = \\ &= \det(F + \lambda H - \lambda E).\end{aligned}$$

Следовательно, справедлива

***Теорема* 2.** Для того чтобы метод Зейделя сходился при любом начальном приближении $x^{(0)}$, необходимо и достаточно, чтобы все корни уравнения

$$\det(F + \lambda H - \lambda E) = 0$$

были по модулю меньше единицы.

Заметим, что области сходимости метода простой итерации и метода Зейделя ***различны***, так как связаны с корнями ***различных*** уравнений (см. Теорему 2 из § 2).

***Теорема* 3.** Для того чтобы метод Зейделя сходился, достаточно, чтобы какая-либо норма матрицы $(E - H)^{-1} F$ была меньше единицы.

Однако имеет место и более простой признак (по существу совпадающий с признаком сходимости метода простой итерации).

***Теорема* 4.** Для того чтобы метод Зейделя сходился, достаточно, чтобы выполнялось условие

$$\|B\|_I = \max_{1 \le i \le n} \sum_{j=1}^{n} |b_{ij}| < 1 ,$$

либо

$$\|B\|_{II} = \max_{1\le j\le n} \sum_{i=1}^{n} |b_{ij}| < 1 .$$

Доказательство.
Проверим лишь первое условие. Пусть

$$\sum_{j=1}^{n} |b_{ij}| < 1, \;\; i = 1,...,n .$$

Покажем, что при выполнении этого условия никакое значение λ^*, по модулю не меньшее единицы, не может быть корнем уравнения

$$\det(F + \lambda H - \lambda E) = 0 .$$

Предположив противное (т.е. $|\lambda^*| \ge 1$), имеем:

$$|\lambda^*| \cdot |b_{i1}| + \ldots + |\lambda^*| \cdot |b_{i,i-1}| + |b_{ii+1}| + \ldots + |b_{in}| \le |\lambda^*| \cdot \left(\sum_{j=1}^{n} |b_{ij}| - |b_{ii}| \right) <$$
$$< |\lambda^*| \cdot (1 - |b_{ii}|) = |\lambda^*| - |\lambda^*| \cdot |b_{ii}| \le |\lambda^*| - |b_{ii}| \le |\lambda^* - b_{ii}| = |b_{ii} - \lambda^*|, \;\; i = 1,...,n .$$

Таким образом, при сделанном предположении матрица $F + \lambda H - \lambda E$ имеет диагональное преобладание по строкам, а по сему является невырожденной.

Следовательно, значения λ^*, по модулю не меньшие единицы, не могут быть корнями указанного уравнения, а значит, в силу Теоремы 2 метод Зейделя сходится.

⊠

§ 4. Элементы общей теории одношаговых итерационных процессов

Выше мы изучили два наиболее часто используемых представителя итерационных методов решения систем линейных алгебраических уравнений: метод простой итерации и метод Зейделя, а также рассмотрели общую классификацию итерационных методов.

Сейчас же заметим, что все рассмотренные выше одношаговые итерационные процессы могут быть записаны в следующем виде:

$$B \frac{x^{(k+1)} - x^{(k)}}{\tau_{k+1}} + Ax^{(k)} = f, \;\; k = 0,1,... , \qquad (4.1)$$

где B – невырожденная $n \times n$-матрица, τ_{k+1} – последовательность числовых итерационных параметров.

Действительно, стационарный метод (1.4) имеет вид (4.1). Для этого его следует переписать его в виде

$$C^{-1}\left(x^{(k+1)} - x^{(k)}\right) + Ax^{(k)} = f .$$

Отсюда следует, что $B = C^{-1}$ и для всех k $\tau_{k+1} = 1$.

Аналогично для процесса (1.7) имеем:

$$Px^{(k+1)} + Qx^{(k)} = b \Leftrightarrow P\left(x^{(k+1)} - x^{(k)}\right) + (P+Q)x^{(k)} = b \Leftrightarrow$$
$$\Leftrightarrow \; P\left(x^{(k+1)} - x^{(k)}\right) + CAx^{(k)} = Cf \Leftrightarrow C^{-1}P\left(x^{(k+1)} - x^{(k)}\right) + Ax^{(k)} = f .$$

Таким образом, здесь $B = C^{-1}P$ и вновь для всех k $\tau_{k+1} = 1$.

В некотором смысле представление (4.1) (называемое ***канонической формой двухслойных итерационных процессов***) оказывается более удобной для исследования сходимости итерационных процессов с общей функциональной точки зрения. В особенности это касается случая, когда матрица A исходной системы обладает такими свойствами как симметрия и положительная определенность (т.е. $A = A^T > 0$). Напомним, что это свойство $A > 0$ равносильно выполнению ***скалярного*** (!) неравенства

$$(Ax, x) > 0 \text{ для всех } x \in R^n, \quad x \neq 0. \tag{4.2}$$

В связи с этим отметим также, что при $A > 0$ векторную норму, порождаемую матрицей A (обозначение $\|\cdot\|_A$), которая определяется формулой

$$\|x\|_A = \sqrt{(Ax, x)} \tag{4.3}$$

и называется энергетической нормой вектора.

Кроме того, матричное неравенство $C > D$ равносильно неравенству $C - D > 0$ и, в соответствии с (4.2) означает выполнение неравенства

$$(Cx, x) > (Dx, x), \quad x \in R^n, \ x \neq 0.$$

Теперь рассмотрим итерационный процесс вида (4.1), в котором при всех k итерационный параметр τ_{k+1} принимает постоянное значение, равное τ:

$$B\frac{x^{(k+1)} - x^{(k)}}{\tau} + Ax^{(k)} = f. \tag{4.4}$$

Тогда справедлива

***Теорема* 1.** Пусть A – симметричная положительно определенная матрица, B – положительно определенная матрица. Тогда при выполнении условия

$$B > \frac{\tau}{2}A \tag{4.5}$$

метод (4.4) сходится.

Доказательство.

Рассмотрим погрешность k-го приближения: $z^{(k)} = x^{(k)} - x^{(*)}$, где $x^{(*)}$ – точное решение исходной системы, т.е.

$$Ax^{(*)} = f$$

или

$$B\frac{x^{(*)} - x^{(*)}}{\tau} + Ax^{(*)} = f.$$

Вычитая из последнего равенства равенство (4.4), получим задачу для погрешности:

$$B\frac{z^{(k+1)} - z^{(k)}}{\tau} + Az^{(k)} = 0. \tag{4.6}$$

Наша задача – доказать, что при некотором выборе нормы выполняется соотношение

$$\left\|z^{(k)}\right\|\underset{k\to\infty}{\to}0.$$

Представим вектор $z^{(k)}$ в виде

$$z^{(k)}=\frac{z^{(k+1)}+z^{(k)}}{2}-\frac{\tau}{2}\frac{z^{(k+1)}-z^{(k)}}{\tau}.$$

Тогда (4.6) перепишется в виде

$$\left(B-\frac{\tau}{2}A\right)\frac{z^{(k+1)}-z^{(k)}}{\tau}+\frac{1}{2}A\left(z^{(k+1)}+z^{(k)}\right)=0. \tag{4.7}$$

Умножим (4.7) скалярно на вектор $2\left(z^{(k+1)}-z^{(k)}\right)$. Так как $A=A^T$ (что равносильно выполнению соотношения $(Ax,y)=(x,Ay)$), то в результате получим:

$$\left(A\left(z^{(k+1)}+z^{(k)}\right),\left(z^{(k+1)}-z^{(k)}\right)\right)=\left\|z^{(k+1)}\right\|_A^2-\left\|z^{(k)}\right\|_A^2$$

и, таким образом, имеем ***основное энергетическое тождество***

$$2\tau\left(\left(B-\frac{\tau}{2}A\right)\frac{z^{(k+1)}-z^{(k)}}{\tau},\frac{z^{(k+1)}-z^{(k)}}{\tau}\right)+\left\|z^{(k+1)}\right\|_A^2=\left\|z^{(k)}\right\|_A^2. \tag{4.8}$$

Отсюда, поскольку $B>\frac{\tau}{2}A$ и $\tau>0$, получаем цепочку неравенств:

$$\left\|z^{(k+1)}\right\|_A^2\le\left\|z^{(k)}\right\|_A^2\le\ldots\le\left\|z^{(0)}\right\|_A^2.$$

Таким образом, последовательность $\left\|z^{(k)}\right\|_A^2=\left(Az^{(k)},z^{(k)}\right)$ не возрастает, ограничена снизу (нулем) и потому сходится. Переходя в (4.8) к пределу, получим:

$$\lim_{k\to\infty}\left(\left(B-\frac{\tau}{2}A\right)\frac{z^{(k+1)}-z^{(k)}}{\tau},\frac{z^{(k+1)}-z^{(k)}}{\tau}\right)=0 \tag{4.9}$$

и, значит (поскольку скалярный параметр τ и матрицы B и A являются постоянными и $B-\frac{\tau}{2}A>0$),

$$\lim_{k\to\infty}\left\|z^{(k+1)}-z^{(k)}\right\|^2=0. \tag{4.10}$$

Действительно, условие $C>0$ равносильно неравенству $(Cx,x)>0$, а последнее означает, что существует постоянная $\delta>0$ такая, что справедлива оценка снизу: $(Cx,x)\ge\delta\|x\|^2$. В качестве такой константы, как легко видеть, в случае симметричной матрицы C можно взять ее наименьшее собственное значение $\lambda_{\min}(C)$, а в случае, когда C свойством симметрии не обладает – $\lambda_{\min}\left(\frac{C+C^T}{2}\right)$ (покажите!).

Таким образом,

$$\left(\left(B-\frac{\tau}{2}A\right)\frac{z^{(k+1)}-z^{(k)}}{\tau},\frac{z^{(k+1)}-z^{(k)}}{\tau}\right)\geq\frac{\delta}{\tau^2}\left\|z^{(k+1)}-z^{(k)}\right\|^2.$$

Переходя в этом неравенстве к пределу и учитывая (4.9), получаем (4.10).

Теперь, учитывая условие $A>0$ (и, таким образом, существует $A^{\frac{1}{2}}>0$), перепишем (4.6) в виде

$$A^{\frac{1}{2}}z^{(k)}=-A^{-\frac{1}{2}}B\frac{z^{(k+1)}-z^{(k)}}{\tau}.$$

Отсюда получим:

$$\left\|A^{\frac{1}{2}}z^{(k)}\right\|^2=\left(A^{\frac{1}{2}}z^{(k)},A^{\frac{1}{2}}z^{(k)}\right)=\left(Az^{(k)},z^{(k)}\right)=\left\|z^{(k)}\right\|_A^2=\left\|-A^{-\frac{1}{2}}B\frac{z^{(k+1)}-z^{(k)}}{\tau}\right\|^2\leq$$

$$\leq\left(\left\|A^{\frac{1}{2}}\right\|\cdot\|B\|\cdot\frac{\left\|z^{(k+1)}-z^{(k)}\right\|}{\tau}\right)^2=\left\|A^{-1}\right\|\cdot\|B\|^2\cdot\frac{\left\|z^{(k+1)}-z^{(k)}\right\|^2}{\tau^2}.$$

Наконец, переходя в последнем соотношении к пределу при $k\to\infty$, получаем:

$$\left\|z^{(k)}\right\|_A^2\underset{k\to\infty}{\to}0.$$

⊠

Применим Теорему 1 к исследованию рассмотренных ранее итерационных алгоритмов.

Вначале проделаем это на примере метода Якоби (см. каноническое представление (2.7)). Напомним, что в общем случае достаточным условием сходимости является наличие диагонального доминирования у матрицы A исходной системы. Вводя в рассмотрение диагональную матрицу $D=\mathrm{diag}\{a_{11},...,a_{nn}\}$ (это – главная диагональ матрицы A), перепишем метод Якоби в виде

$$D\left(x^{(k+1)}-x^{(k)}\right)+Ax^{(k)}=f.$$

Таким образом, исследуемый алгоритм имеет каноническое представление (4.4), в котором $B=D$ и $\tau=1$. Таким образом, в случае симметричной положительно определенной матрицы A условие (4.5) примет вид

$$D>\frac{1}{2}A$$

или

$$2D>A. \tag{4.11}$$

Отметим, что полученное условие, вообще говоря, оказывается не очень удобным для проверки, но более слабым по сравнению с общим.

Теперь исследуем с этих же позиций построенный на основе канонического представления (2.5) метод Зейделя. Представив матрицу $A=A^T>0$ в виде $A=L+D+U$ (здесь L – нижний треугольник матрицы A, D – ее главная диагональ и U – верхний треугольник), приведем алгоритм к каноническому виду (4.4):

$$(L+D)\left(x^{(k+1)}-x^{(k)}\right)+Ax^{(k)}=f. \tag{4.12}$$

Тогда условие (4.5) примет вид

$$L + D > \frac{1}{2} A . \tag{4.13}$$

Хотя внешне оно также кажется не очень удобным для проверки, как и (4.11), но на его основе легко получается

***Теорема* 2.** Если $A = A^T > 0$, то метод Зейделя (4.12) сходится.

Доказательство.
Заметим, что из условия симметричности матрицы A следует равенство $L^T = U$. Поэтому $(Lx, x) = \left(x, L^T x\right) = (x, Ux)$ и из условия (4.13) получаем:

$$\begin{aligned}\left(\left(D + L - \frac{1}{2} A\right)x, x\right) &= (Dx, x) + (Lx, x) - \frac{1}{2}((L + D + U)x, x) = (Dx, x) + (Lx, x) - \\ &- \frac{1}{2}((Lx, x) + (Dx, x) + (Ux, x)) = (Dx, x) + (Lx, x) - \frac{1}{2}((Dx, x) + (Lx, x) + (Lx, x)) = \\ &= \frac{1}{2}(Dx, x).\end{aligned}$$

Теперь остается заметить, что из условия $A = A^T > 0$ следует $D = D^T > 0$ (докажите!), что и завершает доказательство.

⊠

Таким образом, мы получили более сильный по сравнению с установленным ранее результат.

§ 5. Итерационные методы вариационного типа

При построении итерационных методов решения линейных систем могут быть использованы также и другие идеи анализа. Так, например, в случае системы с вещественной симметричной положительной матрицей A проблема решения системы линейных алгебраических уравнений эквивалентна проблеме нахождения минимума функции ошибки

$$E(x) = (Az, z), \tag{5.1}$$

где

$$z = x^{(*)} - x .$$

Однако функция $E(z)$ не может быть вычислена, если неизвестно точное решение исходной системы. В то же время, нетрудно указать такую функцию, которая лишь постоянным слагаемым отличается от функции ошибки, но может быть вычислена без знания точного решения. Тогда за убыванием функции ошибки можно следить по убыванию такой функции. Построим ее:

$$\begin{aligned}E\left(x^{(k)}\right) &= \left(Ax^{(k)}, x^{(k)}\right) = \left(x^{(k)}, Ax^{(k)}\right) = \left(x^{(*)} - x^{(k)}, Ax^{(*)} - Ax^{(k)}\right) = \\ &= \left(x^{(*)} - x^{(k)}, f - Ax^{(k)}\right) = \left(x^{(*)}, f\right) - \left(x^{(k)}, f\right) - \left(x^{(*)}, Ax^{(k)}\right) + \left(x^{(k)}, Ax^{(k)}\right) = \\ &= \left[\left(x^{(*)}, Ax^{(k)}\right) = \left(Ax^{(*)}, x^{(k)}\right) = \left(f, x^{(k)}\right)\right] = \left(Ax^{(k)}, x^{(k)}\right) - 2\left(x^{(k)}, f\right) + \left(x^{(*)}, f\right) = \\ &= F\left(x^{(k)}\right) + \left(x^{(*)}, f\right),\end{aligned} \tag{5.2}$$

где

$$F(x) = (Ax, x) - 2(f, x). \tag{5.3}$$

Здесь $F(x)$ не зависит от точного решения и может быть эффективно вычислена без знания последнего. Второе же слагаемое в функции ошибки ($(x^{(*)}, f)$) не зависит от приближенного решения и потому не влияет на положение точки экстремума.

Таким образом, задача решения системы (1.1) может быть заменена задачей минимизации функционала (5.3). Следовательно, на этом пути могут быть сконструированы многие новые итерационные алгоритмы. Один из простейших – метод покоординатного спуска. Он является общим методом минимизации функций многих переменных. Опишем его подробнее.

Итак, нам необходимо искать минимум функционала

$$F(x) = (Ax, x) - 2(f, x) = \sum_{i,j=1}^{n} a_{ij} x_i x_j - 2\sum_{i=1}^{n} f_i x_i = F(x_1, x_2, ..., x_n).$$

Будем считать, что нам известно некоторое начальное приближение к решению:

$$x^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right)^T.$$

Согласно идее метода покоординатного спуска будем на первом шаге минимизировать не функцию $F(x_1, ..., x_n)$, а функцию одной переменной $F\left(x_1, x_2^{(0)}, ..., x_n^{(0)}\right)$. Интересующее же нас значение переменной x_1 должно быть решением уравнения

$$\frac{\partial}{\partial x_1} F\left(x_1, x_2^{(0)}, ..., x_n^{(0)}\right) = 2\left(a_{11}x_1 + a_{12}x_2^{(0)} + \ldots + a_{1n}x_n^{(0)} - f_1\right) = 0.$$

Решая это линейное по x_1 уравнение, находим значение $x_1^{(1)}$, которое и минимизирует нашу функцию. На следующем шаге будем минимизировать функцию $F\left(x_1^{(1)}, x, x_3^{(0)} ..., x_n^{(0)}\right)$ и в результате решения уравнения

$$\frac{\partial}{\partial x_2} F\left(x_1^{(1)}, x_2, x_3^{(0)} ..., x_n^{(0)}\right) = 2\left(a_{21}x_1^{(1)} + a_{12}x_2 + a_{23}x_3^{(0)} + \ldots + a_{2n}x_n^{(0)} - f_2\right) = 0$$

получим значение $x_2^{(1)}$, минимизирующее ее.

Проделав рассмотренные преобразования поочередно со всеми координатами вплоть до последней, найдем в результате вектор $x^{(1)}$ нового приближения. Принимая $x^{(1)}$ за начальное приближение, можем повторить весь процесс сначала. Тем самым будет построен следующий итерационный процесс:

$$\begin{cases} a_{11}x_1^{(k+1)} + a_{12}x_2^{(k)} + \ldots + a_{1n}x_n^{(k)} = f_1, \\ a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} + \ldots + a_{2n}x_n^{(k)} = f_2, \\ \quad \ldots\ldots \\ a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \ldots + a_{nn}x_n^{(k+1)} = f_n, \end{cases} \qquad k = 0, 1, \ldots.$$

Очевидно, это есть один из вариантов метода Зейделя.

Дадим сейчас векторную форму вывода описанного алгоритма, которая, с другой стороны, позволит указать и на возможные обобщения.

Итак, при известном приближении $x^{(k+1,i-1)} = \left(x_1^{(k+1,i-1)},\dots x_{i-1}^{(k+1,i-1)}, x_i^{(k)},\dots,x_n^{(k)}\right)^T$ следующее приближение будем выбирать так, чтобы изменять лишь одну i-ю компоненту вектора $x^{(k)}$, т.е.

$$x^{(k+1,i)} = \left(x_1^{(k+1,i-1)},\dots,x_{i-1}^{(k+1,i-1)}, x_i^{(k+1,i)}, x_{i+1}^{(k)},\dots,x_n^{(k)}\right)^T.$$

Для этого, очевидно, достаточно положить

$$x^{(k+1,i)} = x^{(k+1,i-1)} - \alpha e_i, \quad \alpha = \text{const}, \quad e_i = \left(0,\dots,\underset{i}{1},\dots 0\right)^T, \tag{5.4}$$

где α будем выбирать из условия минимума функционала (5.3). Подставляя (5.4) в (5.3), будем иметь:

$$\begin{aligned} F\left(x^{(k+1,i)}\right) &= \left(A\left(x^{(k+1,i-1)} - \alpha e_i\right), x^{(k+1,i-1)} - \alpha e_i\right) - 2\left(f, x^{(k+1,i-1)} - \alpha e_i\right) = \\ &= \left(Ax^{(k+1,i-1)}, x^{(k+1,i-1)}\right) - 2\alpha\left(Ax^{(k+1,i-1)}, e_i\right) + \alpha^2 (Ae_i, e_i) - 2\left(f, x^{(k+1,i-1)}\right) + 2\alpha(f, e_i). \end{aligned}$$

Заметим теперь, что $(Ae_i, e_i) = a_{ii}$, и объединим также коэффициенты при первой степени α в один. При этом введем обозначение $Ax^{(k)} - f = r^{(k)}$. Тогда получим:

$$\begin{aligned} F\left(x^{(k+1,i)}\right) &= \left(Ax^{(k+1,i-1)}, x^{(k+1,i-1)}\right) - 2\left(f, x^{(k+1,i-1)}\right) - 2\alpha\left(r^{(k+1,i-1)}, e_i\right) + \alpha^2 a_{ii} = \\ &= F\left(x^{(k+1,i-1)}\right) - 2\alpha\left(r^{(k+1,i-1)}, e_i\right) + \alpha^2 a_{ii} = \varphi(\alpha). \end{aligned}$$

Полученное выражение представляет собой квадратный трехчлен по переменной α с положительным (напомним, у положительной матрицы все диагональные элементы положительны) старшим коэффициентом. Поэтому

$$\varphi'(\alpha) = 2\alpha a_{ii} - 2\left(r^{(k+1,i-1)}, e_i\right) = 0.$$

Отсюда, учитывая, что $\left(r^{(k)}, e_i\right) = r_i^{(k)}$, находим значение α, доставляющее на данном шаге минимум функционалу (5.3):

$$\alpha_0 = \frac{r_i^{(k+1,i-1)}}{a_{ii}}.$$

Именно при этом значении параметра α i-я компонента вектора невязки $r_i^{(k+1)}$ обращается в нуль, что, в частности, и характеризует метод Зейделя.

Однако, вообще говоря, при построении очередного приближения не обязательно каждый раз минимизировать функционал (5.3). Достаточно лишь требовать его убывания. Тогда, положив в (5.4)

$$\alpha = \omega \frac{r_i^{(k+1,i-1)}}{a_{ii}}, \tag{5.5}$$

получим алгоритм, который в литературе называют ***методом релаксации*** (при $\omega = 1$ говорят о ***полной*** релаксации, в противном случае – о ***неполной***, при $\omega < 1$ – ***нижней***, а при $\omega > 1$ – ***верхней***):

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} - \omega\sum_{j=1}^{i-1}\frac{a_{ij}}{a_{ii}}x_j^{(k+1)} - \omega\sum_{j=i+1}^{n}\frac{a_{ij}}{a_{ii}}x_j^{(k)} + \omega\frac{f_i}{a_{ii}},\quad i=1,\dots,n;\ \ k=0,1,\dots. \qquad (5.6)$$

(5.6) – метод релаксации в координатной форме. Исследуем его сходимость, используя Теорему 1 из предыдущего параграфа. Записывая (5.6) в матричной форме и преобразуя к каноническому виду (4.4), получаем:

$$(D+\omega L)\frac{x^{(k+1)}-x^{(k)}}{\omega} + Ax^{(k)} = f\ . \qquad (5.7)$$

Таким образом, $B = D+\omega L$, $\tau = \omega$ и имеет место

***Теорема* 1.** Пусть $A = A^T > 0$ и $0 < \omega < 2$. Тогда метод релаксации сходится.

Доказательство.

Проверим неравенство $B > \frac{\tau}{2}A$, которое в нашем случае примет вид $D+\omega L > \frac{\omega}{2}A$. Так как $L^T = U$ и $D > 0$, то получаем:

$$\left(\left(D+\omega L-\frac{\omega}{2}(L+D+U)\right)x,x\right) = \left(\left(\left(1-\frac{\omega}{2}\right)D+\frac{\omega}{2}L-\frac{\omega}{2}U\right)x,x\right) =$$
$$= \left(1-\frac{\omega}{2}\right)(Dx,x)+\frac{\omega}{2}(Lx,x)-\frac{\omega}{2}(Ux,x) = \left(1-\frac{\omega}{2}\right)(Dx,x) > 0.$$

⊠

5.1. Метод наискорейшего (градиентного) спуска

Другим возможным обобщением описанного итерационного процесса является метод градиентного спуска, идея которого состоит в том, чтобы спускаться к точке минимума функционала (5.3) не по всем направлениям последовательно, а в направлении наиболее его быстрого убывания. Как известно, направлением наиболее быстрого убывания функции F является направление, противоположное ее градиенту

$$\operatorname{grad} F = \left(\frac{\partial F}{\partial x_1},\dots,\frac{\partial F}{\partial x_n}\right)^T.$$

По аналогии с (5.4) положим

$$x^{(k+1)} = x^{(k)} - \alpha \operatorname{grad} F\left(x^{(k)}\right). \qquad (5.8)$$

Так как

$$\operatorname{grad} F\left(x^{(k)}\right) = 2\left(Ax^{(k)} - f\right) = 2r^{(k)},$$

(здесь, как и выше, $r^{(k)}$ – вектор невязки), то (5.8) эквивалентно равенству

$$x^{(k+1)} = x^{(k)} - 2\alpha r^{(k)}. \qquad (5.9)$$

Подставляя (5.9) в (5.3), получим:

$$F\left(x^{(k+1)}\right)=\varphi(\alpha)=\left(A\left(x^{(k)}-2\alpha r^{(k)}\right),x^{(k)}-2\alpha r^{(k)}\right)-2\left(f,x^{(k)}-2\alpha r^{(k)}\right)=$$
$$=\left(Ax^{(k)},x^{(k)}\right)-4\alpha\left(Ax^{(k)},r^{(k)}\right)+4\alpha^2\left(Ar^{(k)},r^{(k)}\right)-2\left(f,x^{(k)}\right)+4\alpha\left(f,r^{(k)}\right)=F\left(x^{(k)}\right)- \quad (5.10)$$
$$-4\alpha\left(Ax^{(k)}-f,r^{(k)}\right)+4\alpha^2\left(Ar^{(k)},r^{(k)}\right)=F\left(x^{(k)}\right)-4\alpha\left(r^{(k)},r^{(k)}\right)+4\alpha^2\left(Ar^{(k)},r^{(k)}\right).$$

Тогда

$$\varphi'(\alpha)=8\alpha\left(Ar^{(k)},r^{(k)}\right)-4\left(r^{(k)},r^{(k)}\right)=0,$$

откуда

$$\alpha_0=\frac{1}{2}\frac{\left(r^{(k)},r^{(k)}\right)}{\left(Ar^{(k)},r^{(k)}\right)}.$$

Таким образом, алгоритм метода градиентного спуска (5.9) принимает вид

$$x^{(k+1)}=x^{(k)}-\frac{\left(r^{(k)},r^{(k)}\right)}{\left(Ar^{(k)},r^{(k)}\right)}r^{(k)},\quad k=0,1,... \quad (5.11)$$
$$r^{(k)}=Ax^{(k)}-f.$$

***Теорема* 2.** Пусть $A=A^T>0$ и $\min\limits_{1\le i\le n}\{\lambda_i(A)\}=m$, $\max\limits_{1\le i\le n}\{\lambda_i(A)\}=M$. Тогда метод наискорейшего спуска сходится со скоростью геометрической прогрессии со знаменателем $q=\dfrac{M-m}{M+m}$.

Доказательство.

В силу соотношений (5.1), (5.2), (5.10), (5.11) имеем:

$$E\left(x^{(k+1)}\right)-E\left(x^{(k)}\right)=F\left(x^{(k+1)}\right)-F\left(x^{(k)}\right)=-4\alpha\left(r^{(k)},r^{(k)}\right)+4\alpha^2\left(Ar^{(k)},r^{(k)}\right)=$$

$$-4\left(r^{(k)},r^{(k)}\right)\cdot\frac{1}{2}\frac{\left(r^{(k)},r^{(k)}\right)}{\left(Ar^{(k)},r^{(k)}\right)}+4\left(Ar^{(k)},r^{(k)}\right)\cdot\left(\frac{1}{2}\frac{\left(r^{(k)},r^{(k)}\right)}{\left(Ar^{(k)},r^{(k)}\right)}\right)^2=-\frac{\left(r^{(k)},r^{(k)}\right)^2}{\left(Ar^{(k)},r^{(k)}\right)}$$

и

$$E\left(x^{(k)}\right)=\left(A\left(x^{(*)}-x^{(k)}\right),\left(x^{(*)}-x^{(k)}\right)\right)=\left(Ax^{(*)}-Ax^{(k)},A^{-1}\left(Ax^{(*)}-Ax^{(k)}\right)\right)=\left(A^{-1}r^{(k)},r^{(k)}\right).$$

Следовательно,

$$\frac{E\left(x^{(k+1)}\right)}{E\left(x^{(k)}\right)}=1-\frac{\left(r^{(k)},r^{(k)}\right)^2}{E\left(x^{(k)}\right)\cdot\left(Ar^{(k)},r^{(k)}\right)}=1-q_k\ ,$$

где

$$q_k=\frac{\left(r^{(k)},r^{(k)}\right)^2}{\left(A^{-1}r^{(k)},r^{(k)}\right)\left(Ar^{(k)},r^{(k)}\right)}.$$

Оценим снизу величину q_k.

Пусть $\lambda_1\le\lambda_2\le\ldots\le\lambda_n$ – собственные значения матрицы A, причем $0<m\le\lambda_i\le M$. Разложим вектор $r^{(k)}$ по соответствующей этим собственным значениям ортонормированной системе $\{u_i\}_{i=1}^n$ собственных векторов матрицы A:

$$r^{(k)}=\sum_{i=1}^n c_i u_i\ .$$

Тогда

$$Ar^{(k)}=\sum_{i=1}^n c_i\lambda_i u_i,\quad A^{-1}r^{(k)}=\sum_{i=1}^n c_i\lambda_i^{-1}u_i$$

и, таким образом,

$$\left(r^{(k)},r^{(k)}\right)=\sum_{i=1}^{n}c_i^2,\ \ \left(Ar^{(k)},r^{(k)}\right)=\sum_{i=1}^{n}c_i^2\lambda_i,\ \ \left(A^{-1}r^{(k)},r^{(k)}\right)=\sum_{i=1}^{n}c_i^2\lambda_i^{-1}.$$

Теперь заметим, что если a_i – некоторые положительные числа и γ_i – числа, удовлетворяющие неравенствам $0<m\le\gamma_i\le M$, то справедливо неравенство

$$\frac{\sum_{i=1}^{n}\gamma_i a_i\cdot\sum_{i=1}^{n}\gamma_i^{-1}a_i}{\left(\sum_{i=1}^{n}a_i\right)^2}\le\frac{1}{4}\left[\sqrt{\frac{M}{m}}+\sqrt{\frac{m}{M}}\right]^2. \tag{5.12}$$

Действительно, вводя обозначения

$$\alpha_i=\frac{a_i}{\sum_{i=1}^{n}a_i}\qquad\text{и}\qquad\gamma_i=\sqrt{mM}\,\delta_i,$$

перепишем (5.12) в виде

$$\sum_{i=1}^{n}\delta_i\alpha_i\cdot\sum_{i=1}^{n}\alpha_i\frac{1}{\delta_i}\le\frac{1}{4}\left[\sqrt{\frac{M}{m}}+\sqrt{\frac{m}{M}}\right]^2. \tag{5.13}$$

Отметим, что при этом

$$\sum_{i=1}^{n}\alpha_i=1\qquad\text{и}\qquad\sqrt{\frac{m}{M}}\le\delta_i\le\sqrt{\frac{M}{m}}.$$

Пользуясь неравенством Коши, из (5.13) получим:

$$\sum_{i=1}^{n}\delta_i\alpha_i\cdot\sum_{i=1}^{n}\alpha_i\frac{1}{\delta_i}\le\left[\frac{\sum_{i=1}^{n}\alpha_i\delta_i+\sum_{i=1}^{n}\alpha_i\frac{1}{\delta_i}}{2}\right]^2=\frac{1}{4}\left[\sum_{i=1}^{n}\alpha_i\left(\delta_i+\frac{1}{\delta_i}\right)\right]^2.$$

С другой стороны, функция $g(\delta)=\delta+\frac{1}{\delta}$ наибольшее значение на отрезке $\left[\sqrt{\frac{m}{M}};\sqrt{\frac{M}{m}}\right]$, равное $\sqrt{\frac{m}{M}}+\sqrt{\frac{M}{m}}$, принимает на концах указанного отрезка и поэтому $\delta_i+\frac{1}{\delta_i}\le\sqrt{\frac{m}{M}}+\sqrt{\frac{M}{m}}$, $i=1,2,...,n$, откуда следует (5.13) (а значит, и (5.12)).

Из (5.12) непосредственно получаем:

$$q_k\ge\frac{4}{\left[\sqrt{\frac{m}{M}}+\sqrt{\frac{M}{m}}\right]^2}$$

и, следовательно,

$$\frac{E\left(x^{(k+1)}\right)}{E\left(x^{(k)}\right)}=1-q_k\le\left(\frac{M-m}{M+m}\right)^2<1,$$

т.е.

$$E\left(x^{(k+1)}\right)\le\left(\frac{M-m}{M+m}\right)^2E\left(x^{(k)}\right).$$

Отсюда, рекуррентно опускаясь по k к начальному приближению, имеем:

$$E\left(x^{(k)}\right)\le\left(\frac{M-m}{M+m}\right)^{2k}E\left(x^{(0)}\right)$$

или, так как

$$E(x)=(Ax,x)\ge m(x,x),$$

$$\left\|x^{(*)}-x^{(k)}\right\|^2 \le \frac{E\left(x^{(0)}\right)}{m}\left(\frac{M-m}{M+m}\right)^{2k} \underset{k\to\infty}{\to} 0 .$$

$\boxtimes$

§ 6. Оптимизация скорости сходимости простейших итерационных процессов

Рассмотрим явный стационарный одношаговый итерационный процесс

$$\frac{x^{(k+1)}-x^{(k)}}{\tau}+Ax^{(k)}=f . \tag{6.1}$$

Очевидно, (6.1) эквивалентен методу простой итерации с матрицей $B=E-\tau A$, и, таким образом, его сходимость зависит от собственных значений

$$\lambda_i(B)=1-\tau\lambda_i(A) . \tag{6.2}$$

Пусть в (6.1) матрица $A=A^T>0$ и, кроме того, известны границы ее спектра (такие же, как в Теореме 2): $m\le\lambda_i(A)\le M$, $i=1,2,...,n$. Тогда скорость сходимости итерационного процесса (6.1) будет характеризоваться величиной

$$\rho(\tau)=\max_{m\le\lambda\le M}|1-\tau\lambda| . \tag{6.3}$$

Рассмотрим задачу минимизации $\rho(\tau)$ за счет выбора итерационного параметра τ. Очевидно, при $\tau\le 0$ $\rho(\tau)\ge 1$. При $0<\tau<\frac{1}{M}$ функция $\rho_1(\tau)=1-\tau\lambda$ положительна и монотонно убывает по переменной λ на отрезке $[m;M]$ и поэтому для таких значений τ имеем:

$$\rho(\tau)=1-\tau m,\ \ \tau\in\left(0;\frac{1}{M}\right).$$

При $\tau>\frac{1}{M}$ величина $1-\tau M$ становится отрицательной, и модуль ее растет с ростом τ. Следовательно, при некотором $\tau=\tau_0$ наступит момент, когда

$$1-\tau_0 m=-(1-\tau_0 M) \text{ и тогда } \rho(\tau_0)=|1-\tau_0 m| .$$

Отсюда

$$\tau_0=\frac{2}{M+m}, \text{ причем } \rho(\tau_0)=\frac{M-m}{M+m} . \tag{6.4}$$

Покажем, что τ_0 является искомым оптимальным параметром, при котором величина (6.3) принимает минимальное значение. Действительно, если $\tau<\tau_0$, то

$$\rho(\tau)=1-\tau m>1-\tau_0 m=\rho(\tau_0) ;$$

если же $\tau>\tau_0$, то

$$\rho(\tau)\ge|1-\tau M|=M\tau-1\ge M\tau_0-1=\rho(\tau_0) .$$

Таким образом, параметр τ_0, задаваемый формулой (6.4), является искомым, т.е. обеспечивает максимальную скорость сходимости итерационного процесса (6.1).

Аналогичную задачу можно ставить и для k-шаговых процессов.

РАЗДЕЛ II

Вычисление собственных значений и собственных векторов матриц

В предыдущем разделе мы вкратце познакомились лишь с одной из основных групп вычислительных задач линейной алгебры – с задачами численного нахождения решений систем линейных алгебраических уравнений. Сейчас мы рассмотрим другую важную группу таких задач, порождаемую так называемой проблемой собственных значений.

Напомним, что собственными значениями квадратной матрицы A называются такие числа λ, при которых для некоторого вектора $x \neq 0$ имеет место равенство

$$Ax = \lambda x . \tag{1}$$

Любой ненулевой вектор x, удовлетворяющий равенству (1), называется собственным вектором, соответствующим собственному значению λ. Очевидно, что все собственные векторы матрицы определяются с точностью до постоянного множителя.

В предыдущем разделе мы убедились в том, насколько ценной может быть информация о собственных значениях матрицы (см. критерий сходимости метода простой итерации, матричной геометрической прогрессии и т.п.) в вычислительной математике. В то же время она играет очень важную роль также и в других областях знаний (физика, астрономия, механика и пр.): там, где присутствуют колебательные явления и вибрации.

Как известно, однородная система линейных алгебраических уравнений (1) имеет нетривиальные решения только тогда, когда определитель ее матрицы равен нулю, т.е.

$$\det(A-\lambda E) = \begin{vmatrix} a_1-\lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22}-\lambda & \dots & a_{2n} \\ & \dots & \dots & \\ a_{n1} & a_{n2} & \dots & a_{nn}-\lambda \end{vmatrix} = 0 . \tag{2}$$

Левая часть этого равенства (многочлен относительно λ) называется ***характеристическим полиномом***, а само уравнение – ***характеристическим*** или ***вековым*** уравнением. Очевидно,

$$\det(A-\lambda E) = (-1)^n\left(\lambda^n - p_1\lambda^{n-1} - \dots - p_{n-1}\lambda - p_n\right) = (-1)^n P_n(\lambda).$$

Полином $P_n(\lambda)$ обычно называют ***собственным*** многочленом матрицы. Совокупность ***всех*** собственных значений $\lambda_1, \lambda_2, ..., \lambda_n$ (каждое выписано столько раз, какова его кратность) называется ***спектром*** матрицы.

Проблему нахождения собственных значений и собственных векторов матрицы A можно разбить на три естественных этапа:

1) построение собственного многочлена матрицы A;
2) решение уравнения $P_n(\lambda) = 0$ и нахождение собственных значений $\lambda_i(A)$;
3) отыскание нетривиальных решений однородных систем

$$Ax_i = \lambda_i x, \ i = 1, ..., n$$

(отыскание собственных векторов).

Каждый из этапов представляет собой достаточно сложную вычислительную задачу (хотя существуют подходы, которые позволяют обойтись без некоторых из этих этапов).

Как и методы решения систем линейных алгебраических уравнений, методы решения проблемы собственных значений подразделяют на прямые и итерационные.

В то же время сама проблема может подразделяться на ***полную*** (находятся все собственные значения и (или) собственные векторы) и ***частичную*** (находятся собственные значения с какими-то заданными свойствами). Чаще всего полную проблему решают прямыми либо специальными итерационными, а частичную – итерационными методами.

ГЛАВА III

Прямые методы решения полной проблемы собственных значений

§ 1. Метод А.Н. Крылова

Рассмотрим произвольный ненулевой вектор $c^{(0)}$, согласованный по размерности с исходной квадратной матрицей A. Очень часто в качестве $c^{(0)}$ берут, например, вектор $(1,0,...,0)^T$. По этому вектору будем составлять последовательность векторов

$$c^{(1)} = Ac^{(0)},\ \ c^{(2)} = Ac^{(1)} = A^2c^{(0)},....$$

до тех пор, пока не встретим первый вектор (например, $c^{(m)} = A^m c^{(0)}$), который будет являться линейной комбинацией предыдущих линейно независимых векторов, т.е. пока не станет справедливым равенство

$$q_1c^{(m-1)} + q_2c^{(m-2)} + \ldots + q_mc^{(0)} = c^{(m)},$$

причем среди коэффициентов q_i, $i = 1,...,m$, есть хотя бы один, отличный от нуля.

Очевидно, что $m \le n$, где n – размерность вектора $c^{(0)}$, поскольку в n-мерном пространстве система из более чем n векторов всегда линейно зависима.

Для того чтобы практически определить число m и найти коэффициенты $q_1,...,q_m$ соответствующей линейной комбинации, можно поступить следующим образом. Запишем предельно возможную $(m = n)$ линейную комбинацию

$$q_1c^{(n-1)} + q_2c^{(n-2)} + \ldots + q_nc^{(0)} = c^{(n)}$$

в координатном виде:

$$\begin{cases} q_1c_1^{(n-1)} + \ldots + q_nc_1^{(n-1)} = c_1^{(n)}, \\ \quad\quad\quad\ldots\ldots \\ q_1c_n^{(n-1)} + \ldots + q_nc_n^{(n-1)} = c_n^{(n)}. \end{cases} \qquad (1.1)$$

Для определения коэффициентов $q_1,...,q_n$ получаем неоднородную линейную систему из n уравнений. Определитель Δ этой будет отличен от нуля лишь в случае линейной независимости векторов $c^{(n-1)},...,c^{(0)}$. Только в этом случае $(m = n)$ система имеет единственное решение $q_1,...,q_n$. Чтобы выяснить, отличен ли от нуля определитель Δ, обычно систему (1.1) решают методом Гаусса.

Если все n шагов прямого хода метода Гаусса выполнимы и система приводится к треугольному виду, то это свидетельствует о том, что $\Delta \neq 0$ и векторы $c^{(n-1)},...,c^{(0)}$ линейно независимы. Тогда из системы

$$\begin{cases} q_1 + b_{12}q_2 + \ldots + b_{1n}q_n = g_1, \\ \qquad q_3 + \ldots + b_{2n}q_n = g_2, \\ \qquad\quad \ldots\ldots \\ \qquad\qquad\quad q_n = g_n \end{cases}$$

обратным ходом метода Гаусса мы единственным образом последовательно находим все коэффициенты $q_n,...,q_1$ рассматриваемой линейной комбинации.

Если же выполнимы только m шагов прямого хода метода Гаусса, то линейно независимыми будут только m первых векторов $c^{(0)}, c^{(1)},...,c^{(m-1)}$. Записав соответствующую линейную комбинацию

$$q_1c^{(m-1)} + q_2c^{(m-2)} + \ldots + q_mc^{(0)} = c^{(m)}$$

покоординатно и выбрав (например, по методу Гаусса) из этих n линейных алгебраических уравнений m линейно независимых, мы найдем коэффициенты $q_1,...,q_m$ разыскиваемой линейной комбинации.

Построенная линейная комбинация и будет тем алгебраическим образом, по виду которого можно непосредственно записать либо собственный многочлен матрицы, либо его делитель.

Рассмотрим сначала случай $m = n$. В этом случае коэффициенты $q_1,...,q_n$ равны соответствующим коэффициентам $p_1,...,p_n$ собственного многочлена

$$P(\lambda) = \lambda^n - p_1\lambda^{n-1} - \ldots - p_n,$$

т.е.

$$q_i = p_i, \quad i = 1,...,n.$$

Действительно, на основании теоремы Гамильтона-Кели (матрица является корнем своего собственного многочлена)

$$P(A) = A^n - p_1A^{n-1} - \ldots - p_nE = 0.$$

Умножая это равенство на вектор $c^{(0)}$ и принимая во внимание, что $A^ic^{(0)} = c^{(i)}$, $i = 1,...,n$, получаем:

$$p_1c^{(n-1)} + p_2c^{(n-2)} + \ldots + p_nc^{(0)} = c^{(n)}. \tag{1.2}$$

С другой стороны

$$q_1c^{(n-1)} + q_2c^{(n-2)} + \ldots + q_nc^{(0)} = c^{(n)}. \tag{1.3}$$

Вычитая эти равенства, имеем:

$$(p_1 - q_1)c^{(n-1)} + (p_2 - q_2)c^{(n-2)} + \ldots + (p_n - q_n)c^{(0)} = 0.$$

Так как векторы $c^{(0)},...,c^{(n-1)}$ линейно независимы, то последнее равенство возможно лишь в случае, если $p_i = q_i$, $i = 1,...,n$.

Таким образом, при $m = n$ по виду построенной нами линейной комбинации можно непосредственно записать собственный многочлен $P(\lambda)$ матрицы A. Решая уравнение

$$P(\lambda) = 0,$$

мы находим все собственные значения матрицы A (каким образом это можно сделать, мы узнаем в следующем семестре).

В случае $m < n$ построенная линейная комбинация имеет вид

$$q_1 c^{(m-1)} + q_2 c^{(m-2)} + \ldots + q_m c^{(0)} = c^{(m)}$$

или, учитывая равенства $c^{(i)} = A^i c^{(0)}$,

$$\left(A^m - q_1 A^{m-1} - \ldots - q_m E\right) c^{(0)} = 0,$$

т.е.

$$\varphi(A) c^{(0)} = 0,$$

где

$$\varphi(\lambda) = \lambda^m - q_1 \lambda^{m-1} - \ldots - q_m.$$

Следовательно, найдя коэффициенты $q_1,...,q_m$, мы тем самым построим многочлен $\varphi(\lambda)$, который будет являться минимальным аннулирующим вектор $c^{(0)}$ многочленом матрицы A (если бы существовал многочлен $g(\lambda)$, удовлетворяющий условию $g(A)c^{(0)} = 0$ и имеющий степень, меньшую, чем степень многочлена $\varphi(\lambda)$, то это противоречило бы линейной независимости векторов $c^{(0)},...,c^{(m-1)}$). Этот многочлен является делителем любого другого аннулирующего вектор $c^{(0)}$ многочлена (и в частности, собственного многочлена матрицы A).

Таким образом, в случае $m < n$ по виду построенной линейной комбинации мы сможем записать не сам собственный многочлен $P(\lambda)$, а лишь его делитель $\varphi(\lambda)$. Решив уравнение

$$\varphi(\lambda) = 0,$$

мы найдем лишь часть собственных значений матрицы A. Изменяя исходный вектор $c^{(0)}$ и повторяя описанный процесс, можно найти и недостающие собственные значения.

1.1. Вычисление собственных векторов матрицы по методу Крылова

После того как собственное значение λ_i матрицы A вычислено, задача нахождения принадлежащих ему собственных векторов нашей матрицы сводится, вообще говоря, к решению следующей однородной системы:

$$\left(A - \lambda_i E\right) x = 0.$$

Но зачастую промежуточные результаты вычислений, выполняемых при нахождении собственных значений матрицы, могут быть с успехом использованы и для вычисления соответствующих собственных векторов. Это, как правило, позволяет значительно сократить

затраты вычислительного труда. Такая возможность предоставляется и описанным выше алгоритмом метода Крылова.

Пусть известен корень λ_i минимального аннулирующего вектор $c^{(0)}$ многочлена

$$\varphi(\lambda) = \lambda^m - q_1\lambda^{m-1} - \ldots - q_m$$

матрицы A (независимо от случая $m = n$ или $m < n$). Собственный вектор $x^{(i)}$ будем искать в виде линейной комбинации векторов $c^{(0)},...,c^{(m-1)}$:

$$x^{(i)} = \beta_{i1}c^{(m-1)} + \beta_{i2}c^{(m-2)} + \ldots + \beta_{im}c^{(0)}. \qquad (1.4)$$

Коэффициенты β_{ij}, $j = 1,...,m$, необходимо выбрать так, чтобы удовлетворить условию

$$Ax^{(i)} = \lambda_i x^{(i)}. \qquad (1.5)$$

Умножим линейную комбинацию (1.4) слева на матрицу A, учитывая при этом равенства $c^{(j)} = Ac^{(j-1)}$, $j = 1,...,m$, и требования (1.5). В итоге получим:

$$\lambda_i\left(\beta_{i1}c^{(m-1)} + \beta_{i2}c^{(m-2)} + \ldots + \beta_{im}c^{(0)}\right) = \beta_{i1}c^{(m)} + \beta_{i2}c^{(m-1)} + \ldots + \beta_{im}c^{(1)}. \qquad (1.6)$$

Так как

$$\varphi(A)c^{(0)} = 0,$$

т.е.

$$c^{(m)} = q_1c^{(m-1)} + q_2c^{(m-2)} + \ldots + q_mc^{(0)},$$

то (1.6) можно переписать в виде

$$\lambda_i\sum_{j=1}^{m}\beta_{ij}c^{(m-j)} = \beta_{i1}\sum_{j=1}^{m}q_jc^{(m-j)} + \sum_{j=2}^{m}\beta_{ij}c^{(m-j+1)}$$

или

$$\left(q_m\beta_{i1} - \lambda_i\beta_{im}\right)c^{(0)} + \left(q_{m-1}\beta_{i1} + \beta_{im} - \lambda_i\beta_{im-1}\right)c^{(1)} +$$
$$+\left(q_{m-2}\beta_{i1} + \beta_{im-1} - \lambda_i\beta_{im-2}\right)c^{(2)} + \ldots + \left(q_1\beta_{i1} + \beta_{i2} - \lambda_i\beta_{i1}\right)c^{(m-1)} = 0.$$

В силу линейной независимости векторов $c^{(0)},...,c^{(m-1)}$ все коэффициенты последней линейной комбинации должны быть равны нулю, т.е.

$$\begin{cases} q_m\beta_{i1} - \lambda_i\beta_{im} = 0, \\ q_{m-1}\beta_{i1} + \beta_{im} - \lambda_i\beta_{im-1} = 0, \\ q_{m-2}\beta_{i1} + \beta_{im-1} - \lambda_i\beta_{im-2} = 0, \\ \quad\ldots\ldots \\ q_1\beta_{i1} + \beta_{i2} - \lambda_i\beta_{i1} = 0. \end{cases}$$

Из этих равенств, начиная с последнего, последовательно находим:

$$\beta_{i2} = (\lambda_i - q_1)\beta_{i1},$$
$$\beta_{i3} = (\lambda_i^2 - q_1\lambda_i - q_2)\beta_{i1},$$
$$\ldots\ldots$$
$$\beta_{im} = (\lambda_i^{m-1} - q_1\lambda_i^{m-2} - \ldots - q_{m-1})\beta_{i1},$$
$$(\lambda_i^m - q_1\lambda_i^{m-1} - \ldots - q_m)\beta_{i1} = 0.$$

При этом заметим, что последнее из полученных равенств справедливо для любых конечных значений β_{i1}, так как равносильно соотношению $\varphi(\lambda_i)\beta_{i1} = 0$, а λ_i является корнем уравнения $\varphi(\lambda) = 0$.

Полагая, например, $\beta_{i1} = 1$, мы для нахождения искомых коэффициентов β_{ij} будем иметь следующие расчетные формулы:

$$\begin{cases} \beta_{i1} = 1, \\ \beta_{i2} = \lambda_i - q_1, \\ \beta_{i3} = \lambda_i^2 - q_1\lambda_i - q_2, \\ \quad\ldots\ldots \\ \beta_{im} = \lambda_i^{m-1} - q_1\lambda_i^{m-2} - \ldots - q_{m-1}. \end{cases} \tag{1.7}$$

Если данному собственному значению $\lambda_i(A)$ принадлежит несколько собственных векторов, то для их отыскания можно повторить весь процесс, исходя из различных начальных векторов $c^{(0)}$.

§ 2. Метод Данилевского

Известно, что преобразование подобия не изменяет собственного многочлена матрицы. В самом деле, если $B = S^{-1}AS$, то

$$\det(B - \lambda E) = \det(S^{-1}AS - \lambda S^{-1}ES) = \det S^{-1} \cdot \det(A - \lambda E) \cdot \det S = \det(A - \lambda E).$$

Естественно поэтому пытаться преобразованиями подобия привести матрицу A к такой форме, для которой собственный многочлен выписывался бы непосредственно. Для этого удобна ***каноническая форма Фробениуса***:

$$\Phi = \begin{pmatrix} p_1 & p_2 & p_3 & \cdots & p_{n-1} & p_n \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ & & \ldots & \ldots & & \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{pmatrix}. \tag{2.1}$$

Отметим, что в литературе встречаются и другие варианты формы Фробениуса, в которых коэффициенты p_i являются элементами, например, не первой, а последней строки или, соответственно, первого или последнего столбца, с надлежащим образом расположенными единичными элементами. С точки зрения практических потребностей все эти пред-

ставления равноценны, поэтому мы будем далее везде под формой Фробениуса понимать (2.1). Замечательной особенностью данной канонической формы является то, что числа p_i, $(i=1,...,n)$ являются коэффициентами многочлена $P(\lambda)$. Действительно,

$$\det(\Phi-\lambda E)=\det(A-\lambda E)=\begin{vmatrix} p_1-\lambda & p_2 & p_3 & \dots & p_{n-1} & p_n \\ 1 & -\lambda & 0 & \dots & 0 & 0 \\ 0 & 1 & -\lambda & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ & & & \dots & & \\ 0 & 0 & 0 & \dots & 1 & -\lambda \end{vmatrix}=\begin{bmatrix}\text{разложим по элементам}\\ \text{первого столбца}\end{bmatrix}=$$

$$=(p_1-\lambda)\cdot(-\lambda)^{n-1}-\begin{vmatrix} p_2 & p_3 & \dots & p_{n-1} & p_n \\ 1 & -\lambda & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \dots & & \\ 0 & 0 & \dots & 1 & -\lambda \end{vmatrix}=(p_1-\lambda)\cdot(-\lambda)^{n-1}-p_2(-\lambda)^{n-2}+$$

$$+\begin{vmatrix} p_3 & p_4 & \dots & p_{n-1} & p_n \\ 1 & -\lambda & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \dots & & \\ 0 & 0 & \dots & 1 & -\lambda \end{vmatrix}=\dots=(p_1-\lambda)\cdot(-\lambda)^{n-1}-p_2(-\lambda)^{n-2}+p_3(-\lambda)^{n-3}-\dots+(-1)^{n+1}p_n=$$

$$=(-1)^n\left(\lambda^n-p_1\lambda^{n-1}-p_2\lambda^{n-2}-\dots-p_n\right)=(-1)^nP_n(\lambda).$$

Таким образом, основная задача сводится к разысканию нужной нам матрицы подобия S. А.М. Данилевский предложил строить эту матрицу последовательно с помощью $(n-1)$ преобразований подобия, переводящих строки матрицы A, начиная с последней, в соответствующие строки матрицы Φ.

2.1. Построение собственного многочлена матрицы

Итак, пусть задана квадратная матрица

$$A=\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n-1} & a_{2n} \\ & & \dots & & \\ a_{n-11} & a_{n-12} & \dots & a_{n-1n-1} & a_{n-1n} \\ a_{n1} & a_{n2} & \dots & a_{nn-1} & a_{nn} \end{pmatrix}.$$

В зависимости от ее элементов в методе Данилевского различают два случая: регулярный и нерегулярный. Рассмотрим вначале, что предполагает алгоритм в регулярном случае.

Пусть $a_{nn-1} \neq 0$. Тогда $(n-1)$-й столбец матрицы A разделим на a_{nn-1}, а потом с помощью полученного столбца обратим в нуль все оставшиеся элементы n-й строки. Для этого достаточно элементы $(n-1)$-го столбца умножить на a_{ni} и вычесть из соответствующих элементов i-го столбца (сравните с действиями в прямом ходе метода Гаусса!). Описанный процесс эквивалентен умножению матрицы A справа на матрицу

$$M_{n-1} = \begin{pmatrix} 1 & \dots & 0 & 0 & 0 \\ & \dots & \dots & & \\ 0 & \dots & 1 & 0 & 0 \\ -\dfrac{a_{n1}}{a_{nn-1}} & \dots & -\dfrac{a_{nn-2}}{a_{nn-1}} & \dfrac{1}{a_{nn-1}} & -\dfrac{a_{nn}}{a_{nn-1}} \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix}.$$

Полученная матрица AM_{n-1} не будет подобна исходной матрице A и, чтобы сохранить подобие, нужно AM_{n-1} слева умножить на матрицу M_{n-1}^{-1}. Такая матрица существует, так как $\det M_{n-1} = \dfrac{1}{a_{nn-1}} \neq 0$, и несложно проверить, что

$$M_{n-1}^{-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & \dots & \dots & & \\ a_{n1} & a_{n2} & \dots & a_{nn-1} & a_{nn} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

Заметим, что преобразование $M_{n-1}^{-1}AM_{n-1}$ не изменяет последней строки в матрице AM_{n-1}, и она останется такой, как того требует форма Фробениуса. Этим заканчивается первый этап преобразования. В результате его получится матрица

$$A_1 = M_{n-1}^{-1}AM_{n-1} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n-1}^{(1)} & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \dots & a_{2n-1}^{(1)} & a_{2n}^{(1)} \\ & \dots & \dots & & \\ a_{n-11}^{(1)} & a_{n-12}^{(1)} & \dots & a_{n-1n-1}^{(1)} & a_{n-1n}^{(1)} \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}.$$

Второй шаг преобразования аналогичен первому и состоит в приведении предпоследней строки матрицы A_1 к виду Фробениуса при условии неизменности последней ее строки. Предположим, что элемент $a_{n-1n-2}^{(1)}$ в A_1 отличен от нуля. Тогда нужные преобразования можно записать в виде

$$A_2 = M_{n-2}^{-1}A_1M_{n-2} = M_{n-2}^{-1}M_{n-1}^{-1}AM_{n-1}M_{n-2} = \begin{pmatrix} a_{11}^{(2)} & \dots & a_{1n-2}^{(2)} & a_{1n-1}^{(2)} & a_{1n}^{(2)} \\ & \dots & \dots & & \\ a_{n-21}^{(2)} & \dots & a_{n-2n-2}^{(2)} & a_{n-2n-1}^{(2)} & a_{n-2n}^{(2)} \\ 0 & \dots & 1 & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix},$$

где

$$M_{n-2}=\begin{pmatrix} 1 & \dots & 0 & 0 & 0 & 0 \\ & \dots & \dots & & & \\ -\dfrac{a^{(1)}_{n-11}}{a^{(1)}_{n-2n-1}} & \dots & -\dfrac{a^{(1)}_{n-1n-3}}{a^{(1)}_{n-2n-1}} & \dfrac{1}{a^{(1)}_{n-2n-1}} & -\dfrac{a^{(1)}_{n-1n-1}}{a^{(1)}_{n-2n-1}} & -\dfrac{a^{(1)}_{n-1n}}{a^{(1)}_{n-2n-1}} \\ 0 & \dots & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 \end{pmatrix};$$

$$M^{-1}_{n-2}=\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \dots & \dots & \\ a^{(1)}_{n-11} & a^{(1)}_{n-11} & \dots & a^{(1)}_{n-1n-1} & a^{(1)}_{n-1n} \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

Правило построения матриц M_{n-2} и M^{-1}_{n-2} по матрице A_1 аналогично правилу построения матриц M_{n-1} и M^{-1}_{n-1} по матрице A. Эта аналогия сохраняется на всех последующих шагах. Таким образом, в регулярном случае, когда $a_{nn-1}\neq 0$, $a^{(1)}_{n-1n-1}\neq 0,\dots,a^{(n-1)}_{21}\neq 0$, после выполнения $(n-1)$ шагов преобразования матрица A будет приведена к каноническому виду Фробениуса:

$$A_{n-1}=M^{-1}_1M^{-1}_2\dots M^{-1}_{n-1}AM_{n-1}\dots M_2M_1=S^{-1}AS=$$

$$=\begin{pmatrix} a^{(n-1)}_{11} & a^{(n-1)}_{12} & \dots & a^{(n-1)}_{1n-1} & a^{(n-1)}_{1n} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \dots & & \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}=\begin{pmatrix} p_1 & p_2 & \dots & p_{n-1} & p_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \dots & & \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}=\Phi.$$

По первой строке матрицы Φ составляется собственный многочлен матрицы A:

$$P(\lambda)=\lambda^n-p_1\lambda^{n-1}-\dots-p_n.$$

Рассмотрим теперь нерегулярный случай. Будем считать, что преобразования Данилевского доведены до строки номера k и элемент $a^{(n-k)}_{kk-1}$ равен нулю.

Дальнейшее преобразование может выполняться по двум вариантам, в зависимости от того, будет ли в k-й строке левее $a^{(n-k)}_{kk-1}$ элемент, отличный от нуля, или такого элемента нет.

Допустим сначала, что $a^{(n-k)}_{ki}\neq 0$ $(i<k-1)$. Тогда вычисления легко могут быть сведены к регулярному случаю. Для этого достаточно поменять местами столбцы с номерами $k-1$ и i, а чтобы такое преобразование было преобразованием подобия, нужно поменять местами и соответствующие строки. Вспоминая об элементарных матрицах перестановок, соответствующее преобразование можем записать в виде $P_{ik-1}A_{n-k}P_{ik-1}$, где P_{ik-1} – элемен-

тарная матрица перестановок. Записанное преобразование есть преобразование подобия, поскольку $P_{ik-1}P_{ik-1} = E$. После перестановки следующий шаг может быть выполнен так же, как и в регулярном случае.

Пусть теперь левее элемента $a_{kk-1}^{(n-k)}$ не найдется элемента, отличного от нуля, т.е. имеют место равенства $a_{ki}^{(n-k)} = 0$, $i = 1,...,k-1$. В этом случае матрица A_{n-k} имеет следующий вид:

$$A_{n-k} = \left(\begin{array}{ccc:ccc} a_{11}^{(n-k)} & \dots & a_{1k-1}^{(n-k)} & a_{1k}^{(n-k)} & \dots & a_{1n-1}^{(n-k)} & a_{1n}^{(n-k)} \\ & \dots & & & \dots & & \\ a_{k-1k-1}^{(n-k)} & \dots & a_{k-1k-1}^{(n-k)} & a_{k-1k}^{(n-k)} & \dots & a_{k-1n-1}^{(n-k)} & a_{1n}^{(n-k)} \\ \hdashline 0 & \dots & 0 & a_{kk}^{(n-k)} & \dots & a_{kn-1}^{(n-k)} & a_{kn}^{(n-k)} \\ 0 & \dots & 0 & 1 & \dots & 0 & 0 \\ & \dots & & & \dots & & \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 \end{array}\right) = \left(\begin{array}{c:c} B_{n-k} & C_{n-k} \\ \hdashline 0 & \Phi_{n-k} \end{array}\right),$$

где матрицы B_{n-k}, C_{n-k}, Φ_{n-k}, 0 в A_{n-k} отмечены вертикальными прямыми. В этом случае, согласно теореме Лапласа о разложении определителя имеет место равенство, в правой части которого индексы у E означают порядки единичной матрицы:

$$\det(A_{n-k} - \lambda E) = \det(B_{n-k} - \lambda E_{k-1}) \cdot \det(\Phi_{n-k} - \lambda E_{n-k+1}) = 0 .$$

Так как Φ_{n-k} имеет канонический вид Фробениуса, ее собственный многочлен выписывается по элементам первой строки. Остается привести к каноническому виду Фробениуса лишь матрицу B_{n-k}, имеющую порядок $k-1 < n$, и, следовательно, задача преобразования упрощается.

Можно подсчитать, что для приведения указанным способом квадратной матрицы A порядка n к форме Фробениуса в регулярном случае необходимо выполнить $O(n^3)$ операций умножения и деления, т.е. метод Данилевского является одним из самых экономичных прямых методов. Однако, как и все прямые методы, он очень чувствителен к ошибкам округлений. Чтобы повысить надежность вычислений, можно предложить вариант алгоритма с выбором главного элемента, по своей сути аналогичный методу Гаусса с выбором главного элемента: на место ведущего на k-м шаге элемента $a_{kk-1}^{(n-k)}$, на который производится деление на $(n-k+1)$-м шаге, ставится максимальный по модулю элемент матрицы A_{n-k}, стоящий левее и (или) выше его. Это очевидным образом можно сделать с помощью элементарных матриц перестановок точно так же, как это было сделано в первом варианте нерегулярного случая.

Для окончательного контроля вычислений полезно первый коэффициент p_1, полученный из формы Фробениуса, сравнивать со следом исходной матрицы A, поскольку

$$p_1 = \lambda_1 + \lambda_2 + \ldots + \lambda_n = a_{11} + a_{22} + \ldots + a_{nn} = \mathrm{Sp}\, A .$$

2.2. Вычисление собственных векторов матрицы

Метод Данилевского при нахождении собственных векторов позволяет обойтись без решения соответствующих однородных систем вида

$$Ax = \lambda_i x, \quad i = 1,...,n .$$

Собственные векторы при преобразовании подобия, вообще говоря, меняются, однако справедлива следующая

Лемма. Если x – собственный вектор матрицы A, соответствующий собственному значению λ, а y – собственный вектор матрицы $\Phi = S^{-1}AS$, соответствующий тому же собственному значению λ, то $x = Sy$.

Доказательство.
Так как y – собственный вектор матрицы Φ, то справедливо равенство

$$\Phi y = \lambda y,$$

которое, учитывая вид матрицы Φ, равносильно равенству

$$S^{-1}ASy = \lambda y.$$

Умножая последнее равенство слева на матрицу S, получаем:

$$ASy = \lambda Sy,$$

т.е. Sy – собственный вектор матрицы A, соответствующий собственному значению λ.

⊠

Нахождение же собственных векторов матрицы Фробениуса является простой задачей. В самом деле, переписывая систему $\Phi y = \lambda y$ в координатном виде, имеем:

$$\begin{cases} p_1 y_1 + p_2 y_2 + \ldots + p_{n-1} y_{n-1} + p_n y_n = \lambda y_1, \\ y_1 = \lambda y_2, \\ \quad \ldots\ldots \\ y_{n-1} = \lambda y_n. \end{cases} \tag{2.2}$$

Так как собственный вектор определяется с точностью до постоянного множителя, то можно положить $y_n = 1$. Тогда

$$y_{n-1} = \lambda, \;\; y_{n-2} = \lambda^2, \ldots, y_1 = \lambda^{n-1}.$$

Первое же уравнение системы (2.2) становится тривиальным и может быть использовано для контроля вычислений.

Таким образом, если λ известно, то $y = \left(\lambda^{n-1}, \ldots, \lambda, 1\right)^T$ – собственный вектор матрицы Фробениуса и, следовательно, $x = Sy$.

Если нахождение собственных значений производилось по методу Данилевского, то матрица S непосредственно выписывается в регулярном случае и в первом варианте нерегулярного. Например, в первом случае $S = M_{n-1}M_{n-2}\ldots M_2 M_1$. Так как матрицы M_i только одной строкой отличаются от единичной, то естественно произведение $x = Sy = M_{n-1}M_{n-2}\ldots M_2 M_1 y$ находить, умножая вектор y последовательно на матрицы $M_1, M_2, \ldots, M_{n-1}$ слева. При этом умножении будет, очевидно, изменяться лишь i-я компонента вектора.

При втором варианте нерегулярного случая метода Данилевского использовать данный прием нельзя.

§ 3. Методы, основанные на использовании формул Ньютона

3.1. Метод Леверье

Метод хронологически является одним из первых, предложенных для решения рассматриваемой проблемы. Несмотря на большой объем работы, обусловленный вычислительной схемой метода, он давно получил признание как один из универсальных и простых по логике алгоритмов построения собственного многочлена матрицы.

Идея метода Леверье основана на использовании хорошо известных из алгебры формул Ньютона

$$kp_k = S_k - p_1 S_{k-1} - p_2 S_{k-2} - \ldots - p_{k-1} S_1, \quad k = 1,...,n, \tag{3.1}$$

связывающих коэффициенты $p_1,...,p_n$ собственного многочлена

$$P_n(\lambda) = \lambda^n - p_1 \lambda^{n-1} - \ldots - p_n$$

матрицы A с симметрическими функциями

$$S_k = \sum_{i=1}^{n} \lambda_i^k, \quad k = 1,...,n,$$

его корней, т.е. собственных значений $\lambda_1,...,\lambda_n$ этой матрицы. Если значения S_k известны, то формулы Ньютона (3.1) позволяют последовательно вычислять коэффициенты собственного многочлена матрицы:

$$\begin{cases} p_1 = S_1, \\ p_2 = \dfrac{1}{2}(S_2 - p_1 S_1), \\ p_3 = \dfrac{1}{3}(S_3 - p_1 S_2 - p_2 S_1), \\ \quad \ldots\ldots \\ p_n = \dfrac{1}{n}(S_n - p_1 S_{n-1} - p_2 S_{n-2} - \ldots - p_{n-1} S_1). \end{cases} \tag{3.2}$$

Величины же S_k принципиально нетрудно подсчитать по исходной матрице A. В самом деле, так как собственными значениями матрицы A^k являются $\lambda_1^k,...,\lambda_n^k$, то

$$S_k = \lambda_1^k + \lambda_2^k + \ldots + \lambda_n^k = \operatorname{Sp} A^k = \sum_{i=1}^{n} a_{ii}^{(k)}.$$

Таким образом, для нахождения собственного многочлена $P(\lambda)$ матрицы A нужно по этой матрице составить матрицы $A^2, A^3,..., A^n$, подсчитать следы $S_k = \operatorname{Sp} A^k, \ k = 1,...,n$ этих матриц и по формулам (3.2) получить коэффициенты нужного нам многочлена.

Если учесть, что для вычисления следа матрицы нужно знать не все ее элементы, а только диагональные, то можно ограничиться составлением матриц A^k лишь при $k = 2,3,...,m$, где $m = \left[\dfrac{n+1}{2}\right]$. Следы же матриц $A^{m+1},...,A^n$ теперь можно находить, минуя

вычисление недиагональных элементов этих матриц. Это можно сделать, например, по формулам типа

$$a_{ii}^{(p)} = \sum_{k=1}^{n} a_{ik}^{(s)} \cdot a_{ki}^{(p-s)},$$

где $p > m$, а $s < m$.

Однако и при учете сделанного замечания метод Леверье остается очень трудоемким, так как связан с многократным перемножением матриц.

Упражнение. Подсчитайте количество операций, необходимых для его реализации.

3.2. Метод Фаддеева

Интересное видоизменение метода Леверье было предложено Фаддеевым. Оно не только позволяет вычислять коэффициенты собственного многочлена матрицы, но и дает возможность эффективно находить матрицу, обратную данной, а также может быть использовано и для получения собственных векторов матрицы.

Предлагается вместо последовательности $A^1, A^2, ..., A^n$ находить другую матричную последовательность $A_1, A_2, ..., A_n$, построенную следующим образом:

$$\begin{aligned}
&A_1 = A, && \operatorname{Sp} A_1 = q_1, && B_1 = A_1 - q_1 E,\\
&A_2 = A \cdot B_1, && \frac{1}{2}\operatorname{Sp} A_2 = q_2, && B_2 = A_2 - q_2 E,\\
&&\ldots\ldots&&&\\
&A_{n-1} = A \cdot B_{n-2}, && \frac{1}{n-1}\operatorname{Sp} A_{n-1} = q_{n-1}, && B_{n-1} = A_{n-1} - q_{n-1} E,\\
&A_n = A \cdot B_{n-1}, && \frac{1}{n}\operatorname{Sp} A_n = q_n, && B_n = A_n - q_n E.
\end{aligned} \tag{3.3}$$

При этом справедлива

Теорема.

1. $q_i = p_i$, $i = 1, ..., n$;
2. $B_n = 0$;
3. если матрица A неособенная, то $A^{-1} = \dfrac{1}{p_n} B_{n-1}$.

Доказательство.
Первое утверждение докажем по индукции. Имеем:
при $i = 1$

$$p_1 = \operatorname{Sp} A = \operatorname{Sp} A_1 = q_1.$$

Предположив, что выполняются равенства $q_i = p_i$, $i = 1, ..., k-1$, получим:
согласно алгоритму построения

$$A_k = A^k - q_1 A^{k-1} - \ldots - q_{k-1} A, \tag{3.4}$$

откуда согласно индуктивному предположению следует:

$$A_k = A^k - p_1 A^{k-1} - \ldots - p_{k-1} A.$$

Следовательно,

$$kq_k = \operatorname{Sp} A_k = \operatorname{Sp} A^k - p_1 \operatorname{Sp} A^{k-1} - p_2 \operatorname{Sp} A^{k-2} - \ldots - \operatorname{Sp} A =$$

$$= S_k - p_1 S_{k-1} - p_2 S_{k-2} - \ldots - p_{n-1} S_1 \overset{(3.1)}{=} kp_k ,$$

т.е.

$$q_k = p_k ;$$

Используя (3.3), (3.4), далее получаем:

$$B_n = A_n - q_n E = A^n - p_1 A^{n-1} - \ldots - p_{n-1} A - p_n E .$$

Многочлен же, стоящий в правой части этого равенства, согласно теореме Гамильтона-Кели равен нулю.

Наконец, с учетом доказанного из (3.3) получаем:

$$A_n = p_n E$$

и поскольку

$$A_n = AB_{n-1},$$

то

$$AB_{n-1} = p_n E .$$

Таким образом,

$$A^{-1} = \frac{1}{p_n} B_{n-1} .$$

⊠

Метод Фаддеева, как мы уже отмечали, позволяет также находить собственные векторы матрицы A.

Рассмотрим матрицу

$$Q(\lambda) = \lambda^{n-1} E + \lambda^{n-2} B_1 + \lambda^{n-3} B_2 + \ldots + \lambda B_{n-2} + B_{n-1} .$$

Если все собственные значения исходной матрицы A различны, то матрицы $Q(\lambda_i)$ $(i = 1,\ldots,n)$ – ненулевые. И в этом случае любой ненулевой столбец матрицы $Q(\lambda_i)$ может быть принят в качестве собственного вектора матрицы A, соответствующего собственному значению λ_i.

Действительно,

$$(\lambda_i E - A)Q(\lambda_i) = (\lambda_i E - A)(\lambda_i^{n-1} E + \lambda_i^{n-2} B_1 + \lambda_i^{n-3} B_2 + \ldots + \lambda_i B_{n-2} + B_{n-1}) =$$

$$= \lambda_i^n E + \lambda_i^{n-1}(B_1 - A) + \lambda_i^{n-2}(B_2 - AB_1) + \ldots + \lambda_i (B_{n-1} - AB_{n-2}) - AB_{n-1} =$$

$$= (\lambda_i^n - p_1 \lambda_i^{n-1} - p_2 \lambda_i^{n-2} - \ldots - p_n) E = 0,$$

так как по построению (см. (3.3))

$$B_k - AB_{k-1} = -q_k E = -p_k E, \quad k = 1,\ldots,n .$$

Из полученного равенства

$$(\lambda_i E - A)Q(\lambda_i) = 0$$

следует, что

$$(\lambda_i E - A)x = 0$$

или

$$Ax = \lambda_i x,$$

где x – любой столбец матрицы $Q(\lambda_i)$.

Очевидно, нет необходимости при этом строить всю матрицу $Q(\lambda)$. Достаточно ограничиться одним ее столбцом.

В случае кратных собственных значений задача нахождения соответствующих собственных векторов усложняется. Наряду с матрицей $Q(\lambda)$ здесь может понадобиться привлекать к рассмотрению также матрицы, полученные путем дифференцирования ее по переменной λ.

§ 4. Другие подходы к построению собственного многочлена

Помимо рассмотренных выше подходов к построению собственного многочлена существуют и другие.

Так, например, из курса алгебры известно, что произвольный алгебраический многочлен степени n зависит от $(n+1)$ параметров-коэффициентов и потому однозначно определяется своими значениями в $(n+1)$ различных точках. Поэтому, зная значения собственного многочлена $P_n(\lambda)$, скажем, в точках $\lambda_1^*,...,\lambda_n^*$ (старший коэффициент здесь уже фиксирован и равен единице!), для определения его коэффициентов можем записать систему линейных алгебраических уравнений (позже мы узнаем, как записать многочлен сразу, не решая системы)

$$\begin{cases} (\lambda_1^*)^n - p_1(\lambda_1^*)^{n-1} - \ldots - p_n = P_n(\lambda_1^*), \\ \ldots\ldots \\ (\lambda_n^*)^n - p_1(\lambda_n^*)^{n-1} - \ldots - p_n = P_n(\lambda_n^*). \end{cases} \qquad (4.1)$$

Сделанное замечание позволяет предложить серию так называемых ***интерполяционных*** алгоритмов построения собственного многочлена, суть которых состоит в том, что мы, задавая в характеристическом уравнении (2) n ***произвольных различных*** значений переменной λ ($\lambda_1^*,...,\lambda_n^*$) и вычисляя n ***числовых*** определителей $D(\lambda_i^*) = \det(A - \lambda_i^* E)$, $i = 1,...,n$, далее коэффициенты собственного многочлена находим из системы (4.1), в которой $P_n(\lambda_i^*) = (-1)^n D(\lambda_i^*)$, $i = 1,...,n$. Задачу вычисления определителей при этом можно решать, например, с помощью метода Гаусса за $O(n^3)$ алгебраических операций на один определитель. Проблема при этом состоит в том, что общая трудоемкость такого алгоритма будет составлять $O(n^4)$операций (примерно такая же, как и в методах Леверье и Фаддеева).

Однако объем вычислительной работы может быть значительно уменьшен путем предварительного преобразования исходной матрицы A с помощью преобразований подобия к некоторому специального виду. В частности, в качестве цели таких преобразований может быть выбрана верхняя почти треугольная матрица (так называемая ***верхняя форма Хессенберга***), у которой отличные от нуля элементы могут располагаться лишь в верхнем треугольнике и на главной поддиагонали (заметим, что в случае симметрии исходной матрицы ее форма Хессенберга представляет из себя трехдиагональную матрицу).

В дальнейшем мы увидим, что матрицы в форме Хессенберга имеют и другие применения, а пока рассмотрим более подробно вопрос о вычислении определителей матрицы Хессенберга.

Итак, пусть матрица A является верхней почти треугольной, т.е. $a_{ij} = 0$ при $i > j+1$:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n-2} & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n-2} & a_{2n-1} & a_{2n} \\ 0 & a_{32} & a_{33} & \dots & a_{3n-2} & a_{3n-1} & a_{3n} \\ 0 & 0 & a_{43} & \dots & a_{4n-2} & a_{4n-1} & a_{4n} \\ & & & \dots & & & \\ 0 & 0 & 0 & \dots & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ 0 & 0 & 0 & \dots & 0 & a_{nn-1} & a_{nn} \end{pmatrix}.$$

Вычисление определителя такой матрицы удобно проводить методом Гаусса, учитывая большое количество нулей. При этом каждый этап исключения сводится к преобразованию только двух строк. Поэтому расчетные формулы принимают вид

$$\begin{cases} b_{kj} = \dfrac{a_{kj,k-1}}{a_{kk,k-1}}, \\ a_{k+1\,j,k} = a_{k+1\,j,k-1} - a_{k+1k,k-1} \cdot b_{kj}, \end{cases} \qquad \begin{matrix} j = k+1,...,n; \\ k = 1,...,n-1; \end{matrix} \qquad (4.2)$$

$$\det A = a_{11,0} \cdot a_{22,1} \cdot \ldots \cdot a_{nn,n-1}.$$

Таким образом, определитель вычисляется за $O(n^2)$ операций. Поскольку нас в настоящий момент интересует определитель $\det(A - \lambda E)$, то для его вычисления достаточно в формулах (4.2) заменить a_{ii} на $a_{ii} - \lambda$.

Еще проще может быть вычислено значение собственного многочлена матрицы A при фиксированном значении λ для трехдиагональной матрицы. Обозначим главный минор m-го порядка матрицы $A - \lambda E$ через $D_m(\lambda)$:

$$D_m(\lambda) = \left(\begin{array}{ccccccc} a_{11} - \lambda & a_{12} & \dots & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} - \lambda & \dots & 0 & 0 & 0 & 0 \\ & & \dots & & & & \\ 0 & 0 & \dots & a_{m-3m-3} - \lambda & a_{m-3m-2} & 0 & 0 \\ 0 & 0 & \dots & a_{m-2m-3} & a_{m-2m-2} - \lambda & 0 & 0 \\ \hdashline 0 & 0 & \dots & 0 & a_{m-1m-2} & a_{m-1m-1} - \lambda & a_{m-1m} \\ \hdashline 0 & 0 & \dots & 0 & 0 & a_{mm-1} & a_{mm} - \lambda \end{array}\right).$$

Разложим такой минор по элементам последней строки. В ней всего два ненулевых элемента, так что получим:

$$D_m(\lambda) = (a_{mm} - \lambda) \cdot D_{m-1}(\lambda) - a_{mm-1} \cdot B_{mm-1}(\lambda), \qquad (4.3)$$

где через $B_{mm-1}(\lambda)$ обозначен минор, дополняющий элемент a_{mm-1}:

$$B_{mm-1}(\lambda)=\left(\begin{array}{ccccc:c} a_{11}-\lambda & a_{12} & \dots & 0 & 0 & 0 \\ a_{21} & a_{22}-\lambda & \dots & 0 & 0 & 0 \\ & & \dots & & & \\ 0 & 0 & \dots & a_{m-3m-3}-\lambda & a_{m-3m-2} & 0 \\ 0 & 0 & \dots & a_{m-2m-3} & a_{m-2m-2}-\lambda & 0 \\ \hdashline 0 & 0 & \dots & 0 & a_{m-1m-2} & a_{m-1m} \end{array}\right).$$

Этот минор содержит в последнем столбце только один ненулевой элемент a_{m-1m}. Поэтому, разлагая $B_{mm-1}(\lambda)$ по элементам последнего столбца, имеем:

$$B_{mm-1}(\lambda)=a_{m-1m}\cdot D_{m-2}(\lambda). \tag{4.4}$$

Подставляя (4.4) в (4.3), получим рекуррентное соотношение

$$D_m(\lambda)=(a_{mm}-\lambda)\cdot D_{m-1}(\lambda)-a_{mm-1}\cdot a_{m-1m}\cdot D_{m-2}(\lambda). \tag{4.5}$$

Для начала расчета по формуле (4.5) достаточно положить

$$D_0(\lambda)=1,\ \ D_1(\lambda)=a_{11}-\lambda\ .$$

Легко видеть, что в этом случае объем вычислительной работы для подсчета одного определителя составит $O(n)$ арифметических операций.

Рассмотрим теперь некоторые из алгоритмов приведения матрицы к форме Хессенберга.

4.1. Метод отражений

Ранее мы рассматривали вариант этого метода применительно к решению систем линейных алгебраических уравнений.

Напомним, что матрицу отражений V мы определили следующим образом:

$$V=E-2\omega\omega^T,$$

где ω – произвольный вектор-столбец единичной длины. При этом задача перевода произвольного заданного вектора s в вектор, коллинеарный заданному вектору e единичной длины решалась по формулам

$$\begin{cases} Vs=\alpha e: \\ \omega=\kappa(s-\alpha e), \\ \alpha=\sqrt{(s,s)},\ \ \kappa=\dfrac{1}{\sqrt{2(s,s-\alpha e)}}. \end{cases} \tag{4.6}$$

Покажем, как для произвольной матрицы A можно подобрать такую конечную последовательность отражений, чтобы она приводила к верхней почти треугольной форме. Как и в случае линейных систем, очередное отражение должно уничтожать самый длинный ненулевой столбец в нижней части матрицы A.

Будем считать, что уже уничтожен $k-1$ столбец. Опишем очередной шаг алгоритма: разобьем матрицу A на клетки следующим образом:

$$A = \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right),$$

причем здесь A_1 – верхняя почти треугольная матрица размерности $k \times k$, а в прямоугольной клетке A_3 только последний столбец отличен от нуля. Сделаем отражение с помощью вектора ω_k, у которого первые k компонент нулевые. Тогда очевидно, что если матрицу отражения V_k разбить на клетки того же размера, что и матрицу A, то недиагональные клетки будут нулевыми, т.е.

$$V_k = \left(\begin{array}{c|c} E & 0 \\ \hline 0 & W \end{array}\right).$$

Следовательно, искомое преобразование подобия имеет вид (используем правило перемножения клеточных матриц):

$$B = V_q^{-1} A V_q = \left(\begin{array}{c|c} E & 0 \\ \hline 0 & W \end{array}\right) \cdot \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right) \cdot \left(\begin{array}{c|c} E & 0 \\ \hline 0 & W \end{array}\right) = \left(\begin{array}{c|c} A_1 & A_2 W \\ \hline W A_3 & W A_4 W \end{array}\right). \tag{4.7}$$

Левая верхняя клетка результирующей матрицы B имеет нужную нам форму – верхнюю почти треугольную, а у клетки $B_3 = WA_3$ только элементы последнего столбца пока отличны от нуля. С помощью формул (4.6), полагая в них

$$s = (a_{kk+1}, ..., a_{kn})^T, \quad e = (1, 0, ..., 0)^T,$$

можем теперь выполнить намеченное уничтожение. Выпишем сейчас конкретные формулы для определения вектора ω:

$$\begin{aligned}
&\alpha = \alpha_k = b_{kk+1} = \left(\sum_{i=k+1}^{n} a_{ki}^2 \right)^{\frac{1}{2}}; \\
&s - \alpha e = \begin{cases} a_{kk+1} - \alpha_k, & i = k+1, \\ a_{ki}, & i = k+2, ..., n; \end{cases} \\
&(s, s - \alpha e) = a_{kk+1}(a_{kk+1} - \alpha_k) + \sum_{i=k+2}^{n} a_{ki}^2 = \\
&\qquad = \sum_{i=k+1}^{n} a_{ki}^2 - \alpha_k a_{kk+1} = b_{kk+1}(b_{kk+1} - a_{kk+1}); \\
&\kappa = \frac{1}{\sqrt{2 b_{kk+1}(b_{kk+1} - a_{kk+1})}}; \\
&\omega_i = \begin{cases} \kappa(a_{kk+1} - \alpha_k), & i = k+1, \\ \kappa a_{ki}, & i = k+2, ..., n. \end{cases}
\end{aligned} \tag{4.8}$$

Последовательно полагая $k = 1,2,...,n-2$, определяя соответствующие векторы ω_k и производя отражения, мы приведем произвольную матрицу A к верхней почти треугольной форме. Заметим, что если исходная матрица A симметрична, то результирующая матрица будет трехдиагональной.

Рассмотрим вопрос об экономной организации вычислений. Основное число операций уходит на перемножение матричных клеток в (4.7). Поэтому отметим, что

1) клетка A_1 не меняется;

2) в клетке WA_3 имеется только один ненулевой элемент – b_{kk+1};

3) при нахождении остальных двух клеток умножение необходимо организовывать следующим образом (сравните формулы метода отражения для линейных систем):

$$A_4 W = A_4\left(E - 2\omega\omega^T\right) = A_4 - 2(A_4\omega)\omega^T,$$

т.е. вместо перемножения двух матриц нужно дважды перемножить матрицу на вектор.

4.2. Прямой метод вращений

Точно так же, как и при решении систем линейных алгебраических уравнений для приведения матрицы к верхнему почти треугольному (трехдиагональному) виду может быть использован и метод вращений. Напомним, что элементарная матрица вращений T_{kl} имеет вид

$$T_{kl} = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cos\varphi & & -\sin\varphi & & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & \sin\varphi & & \cos\varphi & & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{pmatrix}.$$

Одно подобное преобразование вращения состоит из двух преобразований. Первое из них заключается в построении матрицы $B = AT_{kl}$.

При этом столбцы матрицы B совпадают со столбцами матрицы A, за исключением столбцов с номерами k и l, связанных со столбцами матрицы A следующим образом:

$$\begin{aligned} B_k &= \cos\varphi \cdot A_k + \sin\varphi \cdot A_l; \\ B_l &= -\sin\varphi \cdot A_k + \cos\varphi \cdot A_l. \end{aligned} \qquad (4.9)$$

Второе преобразование связано с построением матрицы

$$C = T_{kl}^T B = T_{kl}^T A T_{kl}.$$

Строки матрицы C совпадают со строками матрицы B, за исключением k-й и l-й:

$$C^k = \cos\varphi \cdot B^k + \sin\varphi \cdot B^l,$$
$$C^l = -\sin\varphi \cdot B^k + \cos\varphi \cdot B^l. \qquad (4.10)$$

Очевидно, всегда можно подобрать угол поворота таким образом, чтобы уничтожить (обнулить) элемент c_{lk-1}. Для этого достаточно положить

$$\cos\varphi = \frac{a_{kk-1}}{\sqrt{a_{kk-1}^2 + a_{lk-1}^2}}, \quad \sin\varphi = \frac{a_{lk-1}}{\sqrt{a_{kk-1}^2 + a_{lk-1}^2}}. \qquad (4.11)$$

На основании данного факта выберем следующую стратегию уничтожения элементов. На первом этапе уничтожим элементы первого столбца, начиная с третьего, т.е. элементы в позициях $(3,1), (4,1),...,(n,1)$. Для этого используем матрицы $T_{23}, T_{24}, ..., T_{2n}$. Ясно, что ранее уничтоженные элементы первого столбца в позициях $(i,1)$ при дальнейших преобразованиях T_{2j} $(j > i)$ меняться не будут.

На втором этапе с помощью подобных преобразований вращения $T_{34}, T_{35}, ..., T_{3n}$ уничтожим элементы второго столбца, начиная с четвертого. При этом преобразования второго этапа не изменяют элементов в позициях $(i,1)$, $i = 2,...,n$, полученных после первого этапа. Действительно, преобразования (4.9) не изменяют первый столбец, а из преобразований (4.10) следует, что элементы c_{k1} и c_{l1} равны нулю, поскольку являются линейными комбинациями нулевых элементов b_{ki} и b_{li}.

Продолжая аналогичным образом алгоритм, через $\frac{(n-1)(n-2)}{2}$ подобных преобразований вращения приведем матрицу A к верхнему почти треугольному виду. Если исходная матрица симметрична, то указанная стратегия приведет ее к трехдиагональному виду.

ГЛАВА IV

Итерационные методы решения полной проблемы собственных значений

§ 1. Метод Якоби (итерационный метод вращений)

Несмотря на свою быстроту, описанные выше прямые методы не вполне удовлетворительны, например, по следующим показателям: 1) их алгоритм состоит из разнородных частей: преобразования исходной матрицы; вычисления корней многочлена; нахождения собственных векторов. Кроме того, формулы, как правило, не упрощаются для некоторых употребительных специальных форм матриц, наиболее важных в приложениях.

Поэтому разработан и используется ряд итерационных методов, в общем случае более медленных, но обладающих какими-то частными преимуществами.

Для эрмитовых матриц наиболее известен (мы здесь рассматриваем его для случая вещественных симметричных матриц) итерационный матриц вращений, предложенный Якоби в 1846 г.

Идеологической основой метода Якоби является известное из курса алгебры утверждение о том, что любая симметричная матрица ортогонально подобна диагональной, т.е. существует ортогональная матрица Q такая, что $A = Q^T \Lambda Q$ (при этом столбцы матрицы Q представляют собой собственные векторы матрицы A).

Метод основан на подборе такой бесконечной последовательности элементарных вращений, которая в пределе преобразует симметричную матрицу A в диагональную.

Справедлива

Теорема. Евклидова норма матрицы не меняется при подобном преобразовании вращения.

Доказательство.

По определению

$$\|A\|^2 = \sum_{i,j=1}^{n} a_{ij}^2 . \tag{1.1}$$

Рассмотрим матрицу вращений T_{kl}, $k < l$ (см. выше), и образуем матрицу $B = AT_{kl}$. Как это уже отмечалось в предыдущем параграфе, матрица B отличается от матрицы A только элементами k-го и l-го столбцов, причем

$$\begin{cases} b_{ik} = a_{ik}\cos\varphi + a_{il}\sin\varphi, \\ b_{il} = -a_{ik}\sin\varphi + a_{il}\cos\varphi, \end{cases} \quad i = 1,...,n; \\ b_{ij} = a_{ij}, \quad j \neq k, \ j \neq l. \tag{1.2}$$

Тогда

$$b_{ik}^2 + b_{il}^2 = (a_{ik}\cos\varphi + a_{il}\sin\varphi)^2 + (-a_{ik}\sin\varphi + a_{il}\cos\varphi)^2 = a_{ik}^2 + a_{il}^2$$

и поскольку элементы остальных столбцов не изменяются, то

$$\|B\| = \|AT_{kl}\| = \|A\|.$$

Аналогично образуем матрицу $C = T_{kl}^T B = T_{kl}^T AT_{kl}$, отличающуюся от матрицы B только строками с номерами k и l, причем

$$\begin{cases} c_{ki} = b_{ki}\cos\varphi + b_{li}\sin\varphi, \\ c_{li} = -b_{ki}\sin\varphi + b_{li}\cos\varphi, \end{cases} \quad i = 1,...,n; \\ c_{ji} = b_{ji}, \quad j \neq k, \ j \neq l. \tag{1.3}$$

Здесь снова, очевидно, будет выполняться равенство

$$c_{ki}^2 + c_{li}^2 = b_{ki}^2 + b_{li}^2 .$$

Таким образом,

$$\|C\| = \|T_{kl}^T B\| = \|B\| = \|A\|. \qquad \boxtimes$$

Перейдем теперь к рассмотрению алгоритма. Разобьем сумму, входящую в (1.1), на две части:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2 = \sum_{i=1}^{n} a_{ii}^2 + \sum_{i=1}^{n}\sum_{j=1,\ j\neq i}^{n} a_{ij}^2 = S(A) + t^2(A).$$

Как показано в Теореме, внедиагональные элементы a_{ik}, a_{il}, a_{ki}, a_{li} меняются таким образом, что сохраняются неизменными величины $a_{ik}^2 + a_{il}^2$ и $a_{ki}^2 + a_{li}^2$, $i \neq k$, $i \neq l$.

Кроме этих элементов внедиагональным является также элемент a_{kl}. Поэтому изменение величины $t^2(A)$ полностью определяется изменением элемента a_{kl}^2. Для того чтобы максимально уменьшить $t^2(A)$ за одно вращение, выберем угол поворота таким образом, чтобы уничтожить элемент a_{kl}. Из формул (1.2), (1.3) получаем:

$$\begin{aligned} c_{kl} &= b_{kl}\cos\varphi + b_{ll}\sin\varphi = \\ &= (-a_{kk}\sin\varphi + a_{kl}\cos\varphi)\cos\varphi + (-a_{lk}\sin\varphi + a_{ll}\cos\varphi)\sin\varphi = [a_{kl} = a_{lk}] = \\ &= a_{kl}\cos 2\varphi + \frac{1}{2}(a_{ll} - a_{kk})\sin 2\varphi. \end{aligned} \qquad (1.4)$$

Полагая в (1.4) $c_{kl} = 0$, получим:

$$\operatorname{tg} 2\varphi = \frac{2a_{kl}}{a_{kk} - a_{ll}}.$$

Выбирая $\varphi \in \left[-\frac{\pi}{4}; \frac{\pi}{4}\right]$ и пользуясь формулами половинного аргумента

$$\cos\varphi = \pm\sqrt{\frac{1+\cos 2\varphi}{2}}, \quad \sin\varphi = \pm\sqrt{\frac{1-\cos 2\varphi}{2}},$$

а также выражая косинус через тангенс, будем иметь:

$$\cos\varphi = \sqrt{\frac{1}{2}\left(1 + \frac{1}{\sqrt{1+\mu^2}}\right)}, \quad \sin\varphi = \operatorname{sign}\mu\sqrt{\frac{1}{2}\left(1 - \frac{1}{\sqrt{1+\mu^2}}\right)}, \quad \mu = \operatorname{tg} 2\varphi.$$

Нетрудно убедиться, что при таком вращении уменьшается сумма квадратов недиагональных элементов матрицы A. Действительно, сумма недиагональных элементов всех строк, кроме k-й и l-й, остается неизменной, учитывая равенства $b_{ik}^2 + b_{il}^2 = a_{ik}^2 + a_{il}^2$. Для k-й же и l-й строк выполняются равенства $c_{ki}^2 + c_{li}^2 = b_{ki}^2 + b_{li}^2$. При этом для всех $i \neq k$ и $i \neq l$ $b_{ki} = a_{ki}$, $b_{li} = a_{li}$. Поэтому $c_{ki}^2 + c_{li}^2 = a_{ki}^2 + a_{li}^2$, $i \neq k$, $i \neq l$. Остаются два равенства:

$$\begin{aligned} c_{kk}^2 + c_{lk}^2 &= a_{kk}^2 + a_{lk}^2, \\ c_{kl}^2 + c_{ll}^2 &= a_{kl}^2 + a_{ll}^2, \end{aligned}$$

или, так как $c_{kl} = c_{lk} = 0$,

$$c_{kk}^2 + c_{ll}^2 = a_{kk}^2 + a_{ll}^2 + 2a_{kl}^2.$$

Таким образом,

$$\begin{aligned} t^2(C) &= \sum_{i=1}^{n}\sum_{j=1,\ j\neq i}^{n} c_{ij}^2 = \sum_{i=1}^{n}\sum_{j=1,\ j\neq i}^{n} a_{ij}^2 - 2a_{kl}^2 = t^2(A) - 2a_{kl}^2, \\ S(C) &= \sum_{i=1}^{n} c_{ii}^2 = \sum_{i=1}^{n} a_{ii}^2 + 2a_{kl}^2 = S(A) + 2a_{kl}^2. \end{aligned} \qquad (1.5)$$

Итак, при каждом вращении $S(A)$ увеличивается, а $t^2(A)$, соответственно, уменьшается. Если подобрать такую последовательность вращений, чтобы $t^2(A) \to 0$, то все недиагональные элементы после достаточного числа поворотов станут пренебрежимо малыми и

матрица A преобразуется в диагональную. Диагональные же элементы полученной диагональной матрицы и будут искомыми собственными значениями. Но уничтожить все недиагональные элементы за конечное число поворотов нельзя, ибо, в отличие от прямого метода вращений, здесь при очередном повороте ранее уничтоженный элемент может снова стать ненулевым.

Какой же именно недиагональный элемент целесообразно аннулировать при очередном повороте? Конечно, если уничтожать максимальный по модулю недиагональный элемент, то скорость убывания величины $t^2(A)$ будет наибольшей. При ручных расчетах такая стратегия будет наилучшей. Однако в компьютерном варианте нахождение наибольшего по модулю недиагонального элемента требует $\frac{n(n-1)}{2}$ сравнений всех недиагональных элементов, что существенно снижает эффективность алгоритма. С другой стороны, если аннулировать элементы в заранее заданном порядке, то сходимость будет слишком медленной.

Наиболее выгодным оказалось уничтожение так называемого ***оптимального*** элемента.

Составим суммы квадратов элементов недиагональных элементов строк:

$$\sigma_i = \sum_{j=1,\ j\neq i}^{n} a_{ij}^2,\ \ i = 1,...,n.$$

Выберем из этих сумм наибольшую, а в ней выберем наибольший по модулю элемент (его и называют оптимальным). Именно его и будем уничтожать при очередном повороте. Его нахождение требует всего $2n-1$ сравнений. Если оптимальным будет элемент a_{kl}, то при вращении изменяться только суммы σ_k и σ_l, а именно:

$$\sigma_k^{(1)} = \sigma_k + \left(a_{kk}^{(1)}\right)^{(2)} - a_{kk}^2 - a_{kl}^2,$$
$$\sigma_l^{(1)} = \sigma_l + \sigma_k - \sigma_k^{(1)},$$

т.е. для нахождения оптимального элемента на следующем шаге необходимо пересчитывать только σ_k и σ_l. Описанная стратегия выбора оптимального элемента позволяет судить о скорости сходимости метода вращений.

Действительно, согласно (1.5)

$$t^2(A_{m+1}) = t^2(A_m) - 2\left(a_{kl}^{(m)}\right)^2. \tag{1.6}$$

По определению же оптимального элемента

$$\left(a_{kl}^{(m)}\right)^2 \geq \frac{1}{n-1}\sigma_k^{(m)} \geq \frac{1}{n(n-1)}t^2(A_m). \tag{1.7}$$

Подставляя (1.7) в (1.6), получаем оценку

$$t^2(A_{m+1}) \leq \left(1 - \frac{2}{n(n-1)}\right)t^2(A_m) \leq \ldots \leq \left(1 - \frac{2}{n(n-1)}\right)^{m+1} t^2(A_0),$$

из которой следует, что метод вращений с выбором оптимального элемента всегда сходится.

§ 2. QR -алгоритм

Пусть $\det A \neq 0$. Тогда (см. § 5, Гл. I) матрица A может быть представлена в виде

$$A = QR, \tag{2.1}$$

где Q – ортогональная матрица, а R – верхняя треугольная.

Учитывая (2.1), построим последовательность матриц A_k по следующему правилу: матрицу A_k разлагаем в произведение ортогональной и верхней треугольной матриц, т.е. представляем в виде

$$A_k = Q_{k+1}R_{k+1},$$

а затем полагаем

$$A_{k+1} = R_{k+1}Q_{k+1}. \tag{2.2}$$

Поскольку

$$A_{k+1} = R_{k+1}Q_{k+1} = Q_{k+1}^{-1}Q_{k+1}R_{k+1}Q_{k+1} = Q_{k+1}^{-1}A_kQ_{k+1},$$

то все матрицы A_k подобны между собой и подобны исходной матрице A.

Последовательность $A_1, A_2, ..., A_k, ...$ будем называть последовательностью QR-алгоритма.

Справедлива

Теорема. Пусть невырожденная матрица A имеет различные вещественные собственные значения

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|, \tag{2.3}$$

и пусть также существует LU-разложение матрицы подобия Q^{-1}, т.е.

$$Q^{-1} = LU,$$

где

$$A = Q\Lambda Q^{-1}, \quad \Lambda = \operatorname{diag}\{\lambda_1, ..., \lambda_n\}.$$

Тогда последовательность A_k QR-алгоритма сходится к верхней треугольной матрице.

Доказательство.
Пусть

$$P_k = Q_1Q_2\ldots Q_k, \; U_k = R_k\ldots R_2R_1,$$

где матрицы Q_k и R_k определены в последовательности QR-алгоритма, т.е.

$$\begin{aligned} &A = Q_1R_1, \; A_1 = R_1Q_1 = Q_1^{-1}AQ_1, \\ &A_1 = Q_2R_2, \; A_2 = R_2Q_2 = Q_2^{-1}Q_1^{-1}AQ_1Q_2, \\ &\qquad \ldots\ldots \\ &A_{k-1} = Q_kR_k, \; A_k = R_kQ_k = Q_k^{-1}\ldots Q_1^{-1}AQ_1\ldots Q_k. \end{aligned} \tag{2.4}$$

Тогда

$$\begin{aligned} &P_kU_k = P_{k-1}Q_kR_kU_{k-1} = \left[Q_kR_k = A_{k-1}\right] = P_{k-1}A_{k-1}U_{k-1} = \\ &= \left[A_{k-1} = Q_{k-1}^{-1}\ldots Q_1^{-1}AQ_1\ldots Q_{k-1} = P_{k-1}^{-1}AP_{k-1}\right] = P_{k-1}P_{k-1}^{-1}AP_{k-1}U_{k-1} = A\cdot\left(P_{k-1}U_{k-1}\right) = \\ &= A\cdot\left(A\cdot\left(P_{k-2}U_{k-2}\right)\right) = \ldots = A^k. \end{aligned}$$

С другой стороны, так как $A = Q\Lambda Q^{-1}$, то

$$A^k = Q\Lambda Q^{-1} = \left[Q^{-1} = LU\right] = Q\Lambda^k LU .$$

Таким образом,

$$P_k U_k = Q\Lambda^k LU = \left(Q\Lambda^k L\Lambda^{-k}\right)\Lambda^k U = G_k\Lambda^k U .$$

Отсюда непосредственно следует, что матрица

$$H_k = P_k^{-1}Q\Lambda^k L\Lambda^{-k} = U_k U^{-1}\Lambda^{-k}$$

является верхней треугольной как произведение верхних треугольных матриц U_k и U^{-1} и диагональной матрицы Λ^{-k}. Из последних двух формул получаем также:

$$P_k = G_k\Lambda^k UU_k^{-1} = G_k H_k^{-1},$$

т.е.

$$P_k^{-1} = H_k G_k^{-1}.$$

Подставляя полученные для матрицы P_k и обратной к ней выражения в формулы (2.4), задающие алгоритм, будем иметь:

$$\begin{aligned} A_k &= P_k^{-1}AP_k = P_k^{-1}Q\Lambda Q^{-1}P_k = H_k\left(G_k^{-1}Q\Lambda Q^{-1}G_k\right)H_k^{-1} = [\text{разворачиваем } G_k] = \\ &= H_k\left(\left(Q\Lambda^k L\Lambda^{-k}\right)^{-1}Q\Lambda Q^{-1}\left(Q\Lambda^k L\Lambda^{-k}\right)\right)H_k^{-1} = H_k\left(\Lambda^k L^{-1}\Lambda^{-k}Q^{-1}Q\Lambda Q^{-1}Q\Lambda^k L\Lambda^{-k}\right)H_k^{-1} = \\ &= H_k\left(\Lambda^k L^{-1}\Lambda L\Lambda^{-k}\right)H_k^{-1}. \end{aligned} \qquad (2.5)$$

Проанализируем полученное выражение. Прежде всего, отметим, что матрица $B = L^{-1}\Lambda L$ является нижней треугольной, причем $b_{ii} = \lambda_i$. Поэтому матрица $C_k = \Lambda^k B\Lambda^{-k}$ также будет нижней треугольной, причем

$$c_{ij}^{(k)} = b_{ij}\cdot\left(\frac{\lambda_i}{\lambda_j}\right)^k,\ i \ge j .$$

Из последнего соотношения в силу условия (2.3) имеем:

$$\lim_{k\to\infty} c_{ij}^{(k)} = \begin{cases} 0, & \text{если } i > j, \\ \lambda_{\mathrm{i}}, & \text{если } i = j. \end{cases}$$

Таким образом,

$$\lim_{k\to\infty} C_k = E .$$

Остается изучить поведение матриц H_k и H_k^{-1} при $k \to \infty$. Для этого рассмотрим составляющие их сомножители. Матрицы P_k и P_k^{-1} являются ограниченными, так как, согласно алгоритму, они ортогональные, матрицы Q и Q^{-1} – постоянные, $l_{ii} = 1$ (согласно теореме об LU-разложении возможен выбор единичной диагонали либо у верхней треугольной матрицы, либо у нижней). Поэтому

$$\lim_{k\to\infty}\Lambda^k L\Lambda^{-k} = \lim_{k\to\infty}\Lambda^k L^{-1}\Lambda^{-k} = E .$$

Следовательно, элементы матриц H_k и H_k^{-1} ограничены при любом значении k. Поэтому, переходя к пределу в (2.5) при $k \to \infty$, получаем требуемое, так как матрицы H_k и H_k^{-1} являются верхними треугольными.

⊠

Таким образом, искомые собственные значения матрицы A (с требуемой точностью) будут находиться на диагонали матрицы A_k при некотором значении k. Изложенный алгоритм носит название ***QR-алгоритм в ортодоксальной форме***.

Заметим, что сформулированные в теореме условия сходимости могут быть ослаблены. Простейшее из таких ослаблений состоит в том, что в условии (2.3) достаточно только одного строго неравенства.

Практическое же применение QR-алгоритма связано со следующими необходимыми элементами:

1) ***Уменьшение объема работы на одну итерацию***. Это достигается путем преобразования матрицы A к верхней форме Хессенберга. При этом несложно показать, что верхняя форма Хессенберга инвариантна по отношению к QR-итерациям (покажите!). Заметим, что в последнем случае для получения QR-разложения достаточно $O(n^2)$ арифметических операций, т.е. затраты на одну итерацию QR-алгоритма в случае использования верхней почти треугольной формы будут составлять $O(n^2)$операций вместо $O(n^3)$ в общем случае;
2) ***Уменьшение количества итераций***. Это достигается путем ускорения сходимости метода за счет организации сдвигов (***QR-алгоритм со сдвигом***). При этом вместо формул (2.2) алгоритм задается формулами

$$\begin{aligned} &A_0 = A, \\ &A_k - \mu E = Q_k R_k, \\ &A_{k+1} = R_k Q_k + \mu E, \quad k = 0,1,... \end{aligned} \tag{2.6}$$

где μ - специальным образом подобранный скалярный параметр (сдвиг). Он, вообще говоря, может быть и переменным.

Формулы (2.6) также позволяют построить последовательность $A_1, A_2, ..., A_k, ...$, все матрицы которой подобны исходной матрице A и потому сохраняют ее спектр. Действительно,

$$A_{k+1} = R_k Q_k + \mu E = Q_k^{-1} Q_k R_k Q_k + Q_k^{-1} Q_k \mu E = Q_k^{-1} \left(Q_k R_k + \mu E \right) Q_k = Q_k^{-1} A_k Q_k .$$

Выбирая надлежащим образом параметр сдвига μ, можно в значительной степени ускорить сходимость исходного алгоритма, которая в обычном режиме определяется отношением $\dfrac{|\lambda_2|}{|\lambda_1|}$ (геометрическая прогрессия с таким знаменателем). Возможна, например, такая организация работы: после m итераций обычного QR-алгоритма в качестве параметра сдвига берут элемент, стоящий в позиции (n,n) матрицы A_m (это – некоторое приближение к наименьшему по модулю собственному значению матрицы A). Следующие m_1 этапов организуют со сдвигом по формулам (2.6). В результате в позиции (n,n) матрицы A_{m+m_1} будет стоять достаточно хорошее приближение к наименьшему по модулю собственному значению матрицы A. Далее стандартный QR-алгоритм можно применять к матрице $(n-1)$ порядка, полученной из A_{m+m_1} путем вычеркивания последнего столбца и последней строки и т.д.

Замечание. Существуют более совершенные видоизменения QR-алгоритма. Например, так называемый ***обобщенный QR-алгоритм с мультисдвигом Релея*** (см. [11]) позволяет добиться ***кубической*** сходимости в отличие от линейной в базовом варианте.

ГЛАВА V

Методы решения частичной проблемы собственных значений

§ 1. Степенной метод вычисления наибольшего по модулю собственного значения и соответствующего ему собственного вектора

Все методы, предлагаемые для решения частичной проблемы собственных значений (под этим понимается нахождение одного или нескольких собственных значений и соответствующих собственных векторов матрицы A) итерационны и в основном используют структурные свойства матрицы.

Здесь мы рассмотрим некоторые наиболее простые случаи степенного метода. В частности, мы будем предполагать, что элементарные делители матрицы A линейны. Такое предположение наверняка выполняется в двух важных частных случаях:

1) A – симметричная;

2) собственные значения матрицы различны.

Матрицу A, как и во всех предыдущих случаях, будем предполагать вещественной.

Перенумеруем все собственные значения матрицы A в порядке невозрастания модулей:

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n| .$$

С учетом наложенных требований матрица A обладает полной системой собственных векторов.

Возьмем произвольный вектор $y^{(0)} = \left(y_1^{(0)}, y_2^{(0)}, \ldots, y_n^{(0)}\right)^T$ и построим рекуррентную последовательность векторов по правилу

$$y^{(k)} = Ay^{(k-1)}, \quad k = 1, 2, \ldots .$$

Чтобы выяснить, каким будет $y^{(k)}$ при больших значениях k, разложим $y^{(0)}$ по базису из собственных векторов матрицы A $\{x_1, x_2, \ldots, x_n\}$:

$$y^{(0)} = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n . \tag{1.1}$$

Здесь α_i – некоторые числа, среди которых могут быть равные нулю. Предположим, что $\alpha_1 \neq 0$. Если $\alpha_1 = 0$ и это условие каким-то образом выявлено, то выбор начального вектора $y^{(0)}$ следует изменить и добиться такого положения, чтобы $\alpha_1 \neq 0$.

Приняв во внимание, что $A^k x_i = \lambda_i^k x_i$, из (1.1) получим:

$$y^{(k)} = \alpha_1 \lambda_1^k x_1 + \alpha_2 \lambda_2^k x_2 + \ldots + \alpha_n \lambda_n^k x_n . \tag{1.2}$$

Рассмотрим теперь некоторые частные случаи:

1^0. Наибольшее по модулю собственное значение вещественное и простое, т.е.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n| .$$

Тогда при больших значениях k в правой части (1.2) первое слагаемое будет, очевидно, доминирующим. Для нахождения λ_1 векторное равенство (1.2) удобно записать в координатной форме:

$$y_i^{(k)} = \alpha_1 \lambda_1^k x_{i1} + \alpha_2 \lambda_2^k x_{i2} + \ldots + \alpha_n \lambda_n^k x_{in}. \quad (1.3)$$

Отсюда

$$y_i^{(k)} = \alpha_1 \lambda_1^k x_{i1} \left[1 + \frac{\alpha_2}{\alpha_1 x_{i1}} \left(\frac{\lambda_2}{\lambda_1} \right)^k x_{i2} + \ldots + \frac{\alpha_n}{\alpha_1 x_{i1}} \left(\frac{\lambda_n}{\lambda_1} \right)^k x_{in} \right], \ i = 1, \ldots, n.$$

Так как λ_1 – максимальное по модулю, то $\left| \frac{\lambda_i}{\lambda_1} \right| < 1$, $i = 1, \ldots, n$, и при достаточно больших значениях k величина $\left(\frac{\lambda_i}{\lambda_1} \right)^k$ будет малой и ей можно пренебречь, т.е. можно записать соотношения

$$y_i^{(k)} = \alpha_1 x_{i1} \lambda_1^k \left(1 + O\left(\left| \frac{\lambda_2}{\lambda_1} \right| \right)^k \right)$$

и аналогично

$$y_i^{(k+1)} = \alpha_1 x_{i1} \lambda_1^{k+1} \left(1 + O\left(\left| \frac{\lambda_2}{\lambda_1} \right| \right)^{k+1} \right), \ i = 1, \ldots, n.$$

Взяв отношение i-х координат двух соседних векторов, будем иметь:

$$\frac{y_i^{(k+1)}}{y_i^{(k)}} = \frac{\alpha_1 x_{i1} \lambda_1^{k+1} \left(1 + O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^{k+1} \right) \right)}{\alpha_1 x_{i1} \lambda_1^k \left(1 + O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right) \right)} = \lambda_1 + O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right), \ i = 1, \ldots, n. \quad (1.4)$$

При достаточно больших значениях k с принятой точностью будет выполняться приближенное равенство

$$\lambda_1 \approx \frac{y_i^{(k+1)}}{y_i^{(k)}}. \quad (1.5)$$

Правая часть (1.5) зависит от номера i взятой составляющей, и если эта часть будет иметь одинаковые значения при всяких i в пределах принятой точности, то это является некоторой гарантией того, что итерационный процесс сошелся с нужной точностью.

Собственный вектор x_1 находится просто: запишем (1.2) в виде

$$y^{(k)} = \alpha_1 \lambda_1^k x_1 \left(1 + O\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right) \right),$$

откуда, отбрасывая бесконечно малую, получим:

$$y^{(k)} \approx \alpha_1 \lambda_1^k x_1.$$

Так как собственный вектор находится с точностью до постоянного множителя, то можно считать, что $y^{(k)}$ является собственным вектором, соответствующим собственному значению λ_1.

Замечание. В процессе вычислений, очевидно, координаты вектора $y^{(k)}$ при $|\lambda_1|>1$ будут сильно расти, а при $|\lambda_1|<1$ – сильно убывать; и то, и другое плохо с вычислительной точки зрения. Поэтому реальный вычислительный процесс обычно проводят с нормировкой вектора $y^{(k)}$. Нормировку можно выбрать любую. Тогда получим параллельно последовательность нормированных векторов $z^{(k)}=\rho_k y^{(k)}$, где ρ_k – нормирующий множитель. В этом случае нужно помнить, что для вычисления λ_1 нужно брать отношение компонент векторов $Az^{(k)}$ и $z^{(k)}$, т.е. алгоритм будет выглядеть следующим образом:
Выбрать единичный вектор $z^{(0)}$. Затем для $k=1,2,...$ выполнить:

1) сформировать вектор
$$y^{(k)}=Az^{(k-1)}; \qquad (*)$$
2) нормировать построенный вектор: $z^{(k)}=\dfrac{y^{(k)}}{\|y^{(k)}\|}$;
3) подвергнуть $z^{(k)}$ тесту на сходимость.

2^0. Наибольшее по модулю собственное значение вещественное и кратное, т.е.

$$\lambda_1=\lambda_2=\ldots=\lambda_r$$

и

$$|\lambda_1|>|\lambda_{r+1}|\geq|\lambda_{r+2}|\geq\ldots\geq|\lambda_n|.$$

В этом случае разложение (1.2) принимает вид

$$y^{(k)}=\lambda_1^k\left(\alpha_1x_1+\ldots+\alpha_rx_r\right)+\alpha_{r+1}\lambda_{r+1}^kx_{r+1}+\ldots+\alpha_n\lambda_n^kx_n \qquad (1.6)$$

или в координатной форме

$$y_i^{(k)}=\lambda_1^k\left(\alpha_1x_{i1}+\ldots+\alpha_rx_{ir}\right)+\alpha_{r+1}\lambda_{r+1}^kx_{ir+1}+\ldots+\alpha_n\lambda_n^kx_{in}. \qquad (1.7)$$

Очевидно, по аналогии с предыдущим случаем можем записать (в предположении, что $\alpha_1x_1+\ldots+\alpha_rx_r\neq0$)

$$y_i^{(k)}=\lambda_1^k\left(\alpha_1x_{i1}+\ldots+\alpha_rx_{ir}\right)\left[1+O\left(\left|\frac{\lambda_{r+1}}{\lambda_1}\right|^k\right)\right],$$

$$y_i^{(k+1)}=\lambda_1^{k+1}\left(\alpha_1x_{i1}+\ldots+\alpha_rx_{ir}\right)\left[1+O\left(\left|\frac{\lambda_{r+1}}{\lambda_1}\right|^{k+1}\right)\right],$$

т.е. видим, что построенная последовательность вновь ведет себя как геометрическая прогрессия со знаменателем, на этот раз, $\left|\dfrac{\lambda_{r+1}}{\lambda_1}\right|$. Поэтому

$$\frac{y_i^{(k+1)}}{y_i^{(k)}}=\lambda_1+O\left(\left|\frac{\lambda_{r+1}}{\lambda_1}\right|^k\right),\ \ i=1,...,n. \qquad (1.8)$$

Таким образом, при достаточно больших k с принятой точностью будет

$$\lambda_1 \approx \frac{y_i^{(k+1)}}{y_i^{(k)}},$$

причем последнее отношение не зависит от номера участвующих в нем компонент.

Отметим, что сама формула (1.8) не дает возможности судить о кратности собственного значения λ_1. Как и в первом случае, в качестве собственного вектора, соответствующего собственному значению λ_1, приближенно можно взять вектор $y^{(k)}$. Исходя из различных начальных векторов $y^{(0)}$, мы, вообще говоря, придем к различным собственным векторам, что дает возможность вычислить другие собственные вектора, соответствующие собственному значению λ_1.

Практически в случае $\mathbf{2}^0$, как и в случае $\mathbf{1}^0$, видна покоординатная сходимость последовательностей $y^{(k)}$ и $\lambda_1^k = \frac{y_i^{(k+1)}}{y_i^{(k)}}$.

3^0. Два наибольших по модулю собственных значения вещественны и противоположны по знаку:

$$\lambda_1 = -\lambda_2 \qquad \text{и} \qquad |\lambda_1| > |\lambda_3| \geq \ldots |\lambda_n|.$$

Тогда разложение (1.2) будет иметь вид

$$y^{(k)} = \lambda_1^k\left(\alpha_1 x_1 + \alpha_2 x_2\right) + \alpha_3\lambda_3^k x_3 + \ldots + \alpha_n\lambda_n^k x_n$$

или, в зависимости от четности числа k,

$$y^{(2k)} = \lambda_1^{2k}\left(\alpha_1 x_1 + \alpha_2 x_2\right) + \alpha_3\lambda_3^{2k} x_3 + \ldots + \alpha_n\lambda_n^{2k} x_n,$$
$$y^{(2k+1)} = \lambda_1^{2k+1}\left(\alpha_1 x_1 - \alpha_2 x_2\right) + \alpha_3\lambda_3^{2k+1} x_3 + \ldots + \alpha_n\lambda_n^{2k+1} x_n.$$

Отсюда видно, что векторы $y^{(2k)}$ и $y^{(2k+1)}$ одновременно нельзя использовать для определения λ_1, ибо у этих векторов различные главные части, а именно: $\lambda_1^{2k}\left(\alpha_1 x_1 + \alpha_2 x_2\right)$ у первого и $\lambda_1^{2k+1}\left(\alpha_1 x_1 - \alpha_2 x_2\right)$ у второго. Однако в этом случае мы, очевидно, сможем определить λ_1^2, используя тот факт, что у последовательностей только с четными, либо только с нечетными номерами главные части уже будут одинаковы:

$$\frac{y_i^{(2k+2)}}{y_i^{(2k)}} = \lambda_1^2\left(1 + O\left(\left|\frac{\lambda_3}{\lambda_1}\right|^{2k}\right)\right)$$

или

$$\frac{y_i^{(2k+1)}}{y_i^{(2k-1)}} = \lambda_1^2\left(1 + O\left(\left|\frac{\lambda_3}{\lambda_1}\right|^{2k-1}\right)\right),$$

откуда

$$\lambda_1 \approx \pm\sqrt{\frac{y_i^{(2k+2)}}{y_i^{(2k)}}}, \quad \lambda_2 \approx \mp\sqrt{\frac{y_i^{(2k+2)}}{y_i^{(2k)}}}. \tag{1.9}$$

В вычислительной таблице заметна покоординатная сходимость отдельно векторов с четными и нечетными номерами и соответствующих отношений.

Так как при достаточно больших k

$$y^{(2k)} \approx \lambda_1^{2k}\left(\alpha_1 x_1 + \alpha_2 x_2\right),$$
$$y^{(2k+1)} \approx \lambda_1^{2k+1}\left(\alpha_1 x_1 - \alpha_2 x_2\right),$$

то, рассматривая записанную пару равенств как систему относительно x_1 и x_2, получим:

$$y^{(2k+1)} + \lambda_1 y^{(2k)} \approx 2\lambda_1^{2k+1}\alpha_1 x_1,$$
$$y^{(2k+1)} - \lambda_1 y^{(2k)} \approx -2\lambda_1^{2k+1}\alpha_2 x_2.$$

Таким образом, первую комбинацию можно взять за собственный вектор, соответствующий λ_1, а вторую – λ_2.

4^0. Два наибольших по модулю собственных значения образуют комплексно-сопряженную пару:

$$\lambda_1 = r(\cos\theta + i\sin\theta),\ \ \lambda_2 = r(\cos\theta - i\sin\theta),$$
$$|\lambda_1| > |\lambda_3| \geq \ldots \geq |\lambda_n|.$$

Если собственные значения вещественной матрицы комплексно-сопряженные, то и отвечающие им собственные векторы должны иметь комплексно-сопряженные координаты.

Действительно, в нашем случае векторы $y^{(0)}$ и $y^{(k)}$ – вещественные, поэтому в разложении (1.3) два первых слагаемых должны представлять собой комплексно-сопряженные числа, т.е.

$$\alpha_1 x_{j1} = R_j\left(\cos\varphi_j + i\sin\varphi_j\right),$$
$$\alpha_2 x_{j2} = R_j\left(\cos\varphi_j - i\sin\varphi_j\right).$$

Запишем разложение j-х координат для трех последовательных итераций (при достаточно больших значений k):

$$\begin{aligned} y_j^{(k)} &= 2R_j r^k \cos\left(k\theta + \varphi_j\right) + O\left(|\lambda_3|^k\right), \\ y_j^{(k+1)} &= 2R_j r^{k+1} \cos\left((k+1)\theta + \varphi_j\right) + O\left(|\lambda_3|^{k+1}\right), \\ y_j^{(k+2)} &= 2R_j r^{k+2} \cos\left((k+2)\theta + \varphi_j\right) + O\left(|\lambda_3|^{k+2}\right). \end{aligned} \qquad (1.10)$$

Составим теперь определитель

$$\begin{aligned} I_j^{(k)} &= \begin{vmatrix} y_j^{(k)} & y_j^{(k+1)} \\ y_j^{(k+1)} & y_j^{(k+2)} \end{vmatrix} = \\ &= 4R_j^2 r^{2k+2}\left[\cos\left((k+2)\theta + \varphi_j\right)\cos\left(k\theta + \varphi_j\right) - \cos^2\left((k+1)\theta + \varphi_j\right) + r^k O\left(|\lambda_1|^k\right)\right] = \\ &= -4R_j^2 r^{2k+2}\sin^2\theta + r^k O\left(|\lambda_3|^k\right). \end{aligned}$$

Аналогично

$$I_j^{(k-1)} = -4R_j^2 r^{2k}\sin^2\theta + r^{k-1} O\left(|\lambda_3|^{k-1}\right).$$

Тогда, очевидно, при достаточно больших k будет справедливо приближенное равенство

$$r^2 \approx \frac{I_j^{(k)}}{I_j^{(k-1)}} = \frac{y_j^{(k)} y_j^{(k+2)} - \left(y_j^{(k+1)}\right)^2}{y_j^{(k-1)} y_j^{(k+1)} - \left(y_j^{(k)}\right)^2}. \qquad (1.11)$$

Далее из (1.10) получаем также:

$$y_j^{(k+2)} + r^2 y_j^{(k)} = 2R_j r^{k+2}\left[\cos\left((k+2)\theta + \varphi_j\right) + \cos\left(k\theta + \varphi_j\right)\right] + O\left(|\lambda_3|^k\right) =$$
$$= 4R_j r^{k+2} \cos\left((k+1)\theta + \varphi_j\right)\cos\theta + O\left(|\lambda_3|^k\right) = 2r^{k+1}\cos\theta + O\left(|\lambda_3|^k\right).$$

Поэтому при больших k будем иметь:

$$\cos\theta \approx \frac{y_j^{(k+2)} + r^2 y_j^{(k)}}{2r y_j^{(k+1)}}. \qquad (1.12)$$

Таким образом, и модуль, и аргумент интересующих нас собственных значений будут определены, т.е. λ_1 и λ_2 будут известны. Следовательно, из (1.2) получаем:

$$y^{(k)} = \alpha_1 \lambda_1^k x_1 + \alpha_2 \lambda_2^k x_2 + O\left(|\lambda_3|^k\right),$$
$$y^{(k+1)} = \alpha_1 \lambda_1^{k+1} x_1 + \alpha_2 \lambda_2^{k+1} x_2 + O\left(|\lambda_3|^{k+1}\right),$$

откуда по аналогии со случаем **3⁰** находим:

$$y^{(k+1)} - \lambda_2 y^{(k)} \approx \alpha_1 \lambda_1^k \left(\lambda_1 - \lambda_2\right) x_1,$$
$$y^{(k+1)} - \lambda_1 y^{(k)} \approx \alpha_1 \lambda_2^k \left(\lambda_2 - \lambda_1\right) x_2,$$

и, следовательно, первую комбинацию можно взять за первый собственный вектор, а вторую – за второй.

Характерным внешним признаком итерационного процесса является колебательный характер последовательности приближений.

5⁰. Наибольшее по модулю собственное значение вещественно и находится в жордановом ящике второго порядка.

При этом будем считать, что другим собственным значениям матрицы A соответствуют линейные элементарные делители и все собственные значения по модулю распределены следующим образом:

$$|\lambda_1| > |\lambda_3| \geq \ldots |\lambda_n|.$$

В этом случае при решении задачи о вычислении собственного значения λ_1 удобно использовать вместо базиса из собственных векторов ***канонический базис Жордана***. Пусть векторы $x_1, x_2, \ldots, x_n$ образуют указанный базис. Тогда воздействие матрицы A на векторы этого базиса происходит по формулам

$$\begin{cases} Ax_1 = \lambda_1 x_1 + x_2, \\ Ax_2 = \lambda_1 x_2, \\ Ax_3 = \lambda_3 x_3, \\ \ldots \\ Ax_n = \lambda_n x_n. \end{cases}$$

Значит,

$$\begin{cases} A^k x_1 = \lambda_1^k x_1 + k\lambda_1^{k-1} x_2, \\ Ax_2 = \lambda_1^k x_2, \\ Ax_3 = \lambda_3^k x_3, \\ \dots \\ Ax_n = \lambda_n^k x_n. \end{cases} \tag{1.13}$$

Покажем теперь, каким образом может быть найдено собственное значение λ_1 при указанном распределении собственных значений матрицы A. Пусть $y^{(0)}$ – начальный вектор. Разложим его по векторам канонического базиса:

$$y^{(0)} = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n .$$

Используя формулы (1.13), получим:

$$y^{(k)} = \lambda_1^k \left(\alpha_1 x_1 + \alpha_2 x_2\right) + \alpha_2 k \lambda_2^{k-1} x_2 + \alpha_3 \lambda_3^k x_3 + \dots + \alpha_n \lambda_n^k x_n \tag{1.14}$$

или в координатной форме

$$y_i^{(k)} = \lambda_1^k \left(\alpha_1 x_{i1} + \alpha_2 x_{i2}\right) + \alpha_2 k \lambda_2^{k-1} x_{i2} + \alpha_3 \lambda_3^k x_{i3} + \dots + \alpha_n \lambda_n^k x_{in} . \tag{1.15}$$

Предположим, что при некотором значении i имеет место неравенство $\alpha_2 x_{i2} \neq 0$. Тогда отношение компонент $y_i^{(k+1)}$ и $y_i^{(k)}$ может быть представлено в виде

$$\frac{y_i^{(k+1)}}{y_i^{(k)}} = \lambda_1 \left(1 + O\left(\frac{1}{k}\right)\right).$$

Полученная формула показывает, что

$$\frac{y_i^{(k+1)}}{y_i^{(k)}} \underset{k\to\infty}{\to} \lambda_1 .$$

Однако из-за наличия множителя k во втором слагаемом (1.14) сходимость будут медленнее, чем сходимость любой геометрической прогрессии со знаменателем, меньше единицы.

Используя (1.14), легко найти собственный вектор x_2, отвечающий собственному значению λ_1. Действительно, имеем:

$$\begin{cases} y^{(k)} = \lambda_1^k (\alpha_1 x_1 + \alpha_2 x_2) + \alpha_2 k \lambda_1^{k-1} x_2 + O\left(|\lambda_3|^k\right), \\ y^{(k+1)} = \lambda_1^{k+1} (\alpha_1 x_1 + \alpha_2 x_2) + \alpha_2 (k+1) \lambda_1^k x_2 + O\left(|\lambda_3|^{k+1}\right). \end{cases}$$

Отсюда получаем:

$$\frac{1}{\lambda_1^k}\left(y^{(k+1)} - \lambda_1 y^{(k)}\right) = \alpha_2 x_2 + O\left(\left|\frac{\lambda_3}{\lambda_1}\right|^k\right).$$

Таким образом, в качестве собственного вектора, отвечающего собственному значению λ_1, можно приближенно взять вектор

$$\frac{1}{\lambda_1^k}\left(y^{(k+1)} - \lambda_1 y^{(k)}\right).$$

Замечание. В случаях **4⁰**, **5⁰**, а также и в случае **1⁰** при условии, что отношение $\left|\frac{\lambda_2}{\lambda_1}\right|$ близко к единице, более удобно искать не сами собственные значения, а коэффициенты

приведенного многочлена второй степени, корнями которого эти собственные значения являются. Это можно сделать, например, следующим образом.

Пусть λ_1 и λ_2 являются корнями многочлена

$$\lambda^2 + p\lambda + q = 0 \tag{1.16}$$

и k настолько велико, что

$$y^{(k)} \approx \alpha_1 \lambda_1^k x_1 + \alpha_2 \lambda_2^k x_2 .$$

Тогда

$$y^{(k+1)} + py^{(k)} + qy^{(k-1)} \approx \alpha_1 \lambda_1^{k-1} x_1 \left(\lambda_1^2 + p\lambda_1 + q\right) + \alpha_2 \lambda_2^{k-1} x_2 \left(\lambda_1^2 + p\lambda_1 + q\right) = 0 .$$

Здесь приближенное равенство справедливо с точностью до $O\left(|\lambda_3|^k\right)$.

Переходя к координатной форме записи и используя два соседних значения k, получаем систему для определения коэффициентов многочлена (1.16)

$$\begin{cases} py_i^{(k)} + qy_i^{(k-1)} \approx -y_i^{(k+1)}, \\ py_i^{(k+1)} + qy_i^{(k)} \approx -y_i^{(k+2)}, \end{cases}$$

откуда получаем:

$$\begin{aligned} p &\approx -\frac{y_i^{(k+1)} y_i^{(k)} - y_i^{(k+2)} y_i^{(k-1)}}{\left(y_i^{(k)}\right)^2 - y_i^{(k+1)} y_i^{(k-1)}}, \\ q &\approx -\frac{y_i^{(k)} y_i^{(k+2)} - \left(y_i^{(k+1)}\right)^2}{\left(y_i^{(k)}\right)^2 - y_i^{(k+1)} y_i^{(k-1)}}. \end{aligned} \tag{1.17}$$

§ 2. Видоизменения степенного метода

Изложенный в предыдущем параграфе степенной метод может быть усовершенствован либо в смысле ускорения сходимости соответствующих итерационных последовательностей, либо в смысле приспособления его к решению других задач, например, нахождения минимального по модулю собственного значения и т.п.

2.1. Метод обратных итераций

Этот метод предназначен для нахождения наименьшего по модулю собственного значения матрицы A. Пусть

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_{n-1}| > |\lambda_n| .$$

Так как собственные значения матрицы A^{-1} (ν_i) связаны с собственными значениями матрицы A соотношением $\nu_i = \dfrac{1}{\lambda_i}$, то следовательно

$$\nu_{\max} = \nu_n = \frac{1}{\lambda_{\min}} = \frac{1}{\lambda_n} .$$

Поэтому для того чтобы найти минимальное по модулю собственное значение матрицы A, достаточно найти ν_n, т.е. применить описанный выше алгоритм степенного метода (случай $\mathbf{1^0}$) к матрице A^{-1}. При этом, очевидно, нет необходимости обращать матрицу A. Достаточно лишь в формулах (*) заменить пункт 1) на
1′) решить относительно $y^{(k)}$ систему

$$Ay^{(k)} = z^{(k-1)}.$$

Тогда, как легко видеть, все, о чем говорилось в § 1, остается в силе, и построенная последовательность будет сходиться со скоростью геометрической прогрессии со знаменателем $\left|\frac{\nu_{n-1}}{\nu_n}\right| = \left|\frac{\lambda_n}{\lambda_{n-1}}\right|$ к наименьшему по модулю собственному значению матрицы A; по аналогии с описанным выше правилом находится и соответствующий собственный вектор.

Заметим, что в описываемом алгоритме целесообразным может оказаться построение LU-разложения матрицы A при выполнении первой итерации. Тогда все последующие итерации могут быть реализованы за $O(n^2)$ арифметических операций.

В случае симметричной матрицы A как в случае степенного метода, так и в случае метода обратных итераций можно ускорить сходимость за счет очень простого приема: вместо отношения компонент векторов $y^{(k+1)}$ и $y^{(k)}$ составляют отношение $\frac{\left(y^{(k+1)}, y^{(k)}\right)}{\left(y^{(k)}, y^{(k)}\right)}$. Тогда, учитывая, что собственные векторы матрицы A образуют ортонормированный базис в R^n, разлагая вектор $y^{(0)}$ по этому базису, получаем:

$$y^{(k)} = \sum_{i=1}^{n} \alpha_i \lambda_i^k x_i ,$$

и, значит,

$$\frac{\left(y^{(k+1)}, y^{(k)}\right)}{\left(y^{(k)}, y^{(k)}\right)} = \frac{\sum_{i=1}^{n} \alpha_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^{n} \alpha_i^2 \lambda_i^{2k}} = \lambda_1 \left(1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)\right).$$

Отсюда следует справедливость приближенной формулы

$$\lambda_1 \approx \frac{\left(y^{(k+1)}, y^{(k)}\right)}{\left(y^{(k)}, y^{(k)}\right)},$$

причем погрешность последней по сравнению со штатной погрешностью степенного метода возводится в квадрат.

2.2. Итерации со сдвигом

Как мы видели ранее, скорость сходимости «штатного» степенного метода (как и метода обратных итераций) определяется отношением типа $\left|\frac{\lambda_2}{\lambda_1}\right|$ (если $|\lambda_2|$ – второе по величине собственное значение матрицы A). Поэтому, если применить указанные алгоритмы к матрице $A - \sigma E$ вместо матрицы A (см. также QR-алгоритм со сдвигом), то новые коэффициенты сходимости будут иметь следующий вид:

для степенного метода:

$$\rho_\sigma = \frac{\max\limits_{j \neq s} |\lambda_j - \sigma|}{|\lambda_s - \sigma|},$$

где

$$|\lambda_s - \sigma| = \max_i |\lambda_i - \sigma|;$$

для метода обратной итерации:

$$\nu_\sigma = \frac{|\lambda_t - \sigma|}{\min\limits_{j \neq t} |\lambda_j - \sigma|},$$

где

$$|\lambda_t - \sigma| = \min_i |\lambda_i - \sigma|.$$

Параметр σ принимает, в принципе, любое значение, а s может принимать значения лишь из множества $\{1, n\}$ (действительно, поскольку $\lambda_1 - \sigma \geq \lambda_2 - \sigma \geq \ldots \geq \lambda_n - \sigma$, то экстремальными значениями соответствующих модулей могут быть либо первое, либо последнее). Следовательно, степенной метод может сходиться либо к λ_1, либо к λ_n. Выбор значения параметра σ, который минимизирует значение ρ_σ, есть $\frac{\lambda_2 + \lambda_n}{2}$ в первом случае и $\frac{\lambda_1 + \lambda_{n-1}}{2}$ во втором. При этом скорость сходимости, вообще говоря, остается скоростью сходимости геометрической прогрессии, только с несколько лучшим по сравнению с базовым методом знаменателем.

В то же время, напротив, $\nu_\sigma \to 0$ при $\sigma \to \lambda_j$ для всех значений j. Поэтому метод обратных итераций со сдвигом может сходиться к любому собственному значению исходной матрицы, причем очень быстро, если σ выбрано удачно. Более того, величина сдвига может быть сделана переменной (изменяющейся на каждой итерации). Исследуем этот случай более подробно в предположении симметрии исходной матрицы A.

Рассмотрим некоторый начальный вектор $y^{(0)}$ и построим итерационные последовательности $\{\mu_k\}$ и $\{y^{(k)}\}$, каждый член которых определяется по формулам

$$\mu_k = \frac{\left(Ay^{(k-1)}, y^{(k-1)}\right)}{\left(y^{(k-1)}, y^{(k-1)}\right)}, \quad \left(A - \mu_k E\right) y^{(k)} = y^{(0)}. \tag{2.1}$$

Выясним смысл величины μ_k и свойства последовательности $\{\mu_k\}$. С этой целью, как и ранее, запишем разложение вектора $y^{(0)}$ по собственным векторам матрицы A:

$$y^{(0)} = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n, \tag{2.2}$$

причем

$$\left(x_i, x_j\right) = \delta_i^j$$

(отсюда, в частности, следует, что в случае $y^{(0)} = \alpha_i x_i$ выполняется равенство $\mu\left(y^{(0)}\right) = \lambda_i$).

Тогда

$$y^{(k)} = \left(A - \mu_k E\right)^{-1} y^{(0)} = \frac{\alpha_1}{\lambda_1 - \mu_k} x_1 + \frac{\alpha_2}{\lambda_2 - \mu_k} x_2 + \ldots + \frac{\alpha_n}{\lambda_n - \mu_k} x_n,$$

поскольку

$$\left(A-\mu_k E\right)^{-1}x_i=\frac{1}{\lambda_i-\mu_k}x_i.$$

Следовательно, для μ_{k+1} получим:

$$\mu_{k+1}=\frac{\left(Ay^{(k)},y^{(k)}\right)}{\left(y^{(k)},y^{(k)}\right)}=\frac{\dfrac{\alpha_1^2}{\left(\lambda_1-\mu_k\right)^2}\lambda_1+\dfrac{\alpha_2^2}{\left(\lambda_2-\mu_k\right)^2}\lambda_2+\ldots+\dfrac{\alpha_n^2}{\left(\lambda_n-\mu_k\right)^2}\lambda_n}{\dfrac{\alpha_1^2}{\left(\lambda_1-\mu_k\right)^2}+\dfrac{\alpha_2^2}{\left(\lambda_2-\mu_k\right)^2}+\ldots+\dfrac{\alpha_n^2}{\left(\lambda_n-\mu_k\right)^2}}. \qquad (2.3)$$

Отсюда следует: если сдвинутое собственное значение $\nu_1=\lambda_1-\mu_k$ мало, то в числителе и знаменателе за счет дроби $\dfrac{\alpha_1^2}{\left(\lambda_1-\mu_k\right)^2}$ выделится главный член. Поэтому главная часть всей дроби (2.3) также должна быть близка к λ_1. Рассмотрим разность

$$\mu_{k+1}-\lambda_1=\frac{\dfrac{\alpha_2^2}{\left(\lambda_2-\mu_k\right)^2}\left(\lambda_2-\lambda_1\right)+\ldots+\dfrac{\alpha_n^2}{\left(\lambda_n-\mu_k\right)^2}\left(\lambda_n-\lambda_1\right)}{\dfrac{\alpha_1^2}{\left(\lambda_1-\mu_k\right)^2}+\dfrac{\alpha_2^2}{\left(\lambda_2-\mu_k\right)^2}+\ldots+\dfrac{\alpha_n^2}{\left(\lambda_n-\mu_k\right)^2}}.$$

Из этого равенства имеем:

$$\left|\mu_{k+1}-\lambda_1\right|=\left|\mu_k-\lambda_1\right|^2\cdot\frac{\dfrac{\alpha_2^2}{\left(\lambda_2-\mu_k\right)^2}\left|\lambda_2-\lambda_1\right|+\ldots+\dfrac{\alpha_n^2}{\left(\lambda_n-\mu_k\right)^2}\left|\lambda_n-\lambda_1\right|}{\alpha_1^2+\left(\lambda_1-\mu_k\right)^2\cdot\left[\dfrac{\alpha_2^2}{\left(\lambda_2-\mu_k\right)^2}+\ldots+\dfrac{\alpha_n^2}{\left(\lambda_n-\mu_k\right)^2}\right]}=O\left(\left(\lambda_1-\mu_k\right)^2\right). \qquad (2.4)$$

На основании (2.4) можно записать:

$$\begin{aligned}\left|\mu_{k+1}-\lambda_1\right|&\le q\left(\mu_k-\lambda_1\right)^2\le q\left(q\left(\mu_k-\lambda_1\right)^2\right)^2\le\ldots\le\\&\le q^{1+2+\ldots+2^{k-1}}\left(\mu_k-\lambda_1\right)^{2^k}=q^{2^k-1}\left(\mu_k-\lambda_1\right)^{2^k}=\frac{1}{q}\left[q\left(\mu_k-\lambda_1\right)\right]^{2^k}\end{aligned} \qquad (2.5)$$

(здесь $q>0$ – некоторая константа).

Оценка (2.5) показывает, что в случае удачного выбора $y^{(0)}$ (т.е. при выполнении неравенства $q\left|\mu_1-\lambda_1\right|<1$) последовательность $\left\{\mu_k\right\}$ сходится и

$$\lim_{k\to\infty}\mu_k=\lambda_1,$$

причем сходимость квадратичная. В аналогичном случае степенной метод дает лишь сходимость со скоростью геометрической прогрессии.

Однако, как показывают исследования, в общем случае можно добиться большего, а именно: кубической сходимости.

Рассмотрим алгоритм, отличающийся от (2.1) лишь зависимостью от номера итерации в правой части. Добавив процедуру нормировки, сформулируем его в следующем виде:

Выбрать единичный вектор $z^{(0)}$, затем для $k = 0,1,...$ повторить:

1. Вычислить

$$\mu_k = \frac{\left(Az^{(k)}, z^{(k)}\right)}{\left(z^{(k)}, z^{(k)}\right)};$$

2. Если матрица $\left(A - \mu_k E\right)$ вырождена, то найти из системы

$$\left(A - \mu_k E\right) z^{(k+1)} = 0$$

единичный вектор и остановиться.

В противном случае

а) решить систему

$$\left(A - \mu_k E\right) y^{(k+1)} = z^{(k)}$$

относительно $y^{(k+1)}$;

б) нормировать вектор $y^{(k+1)}$, т.е. вычислить

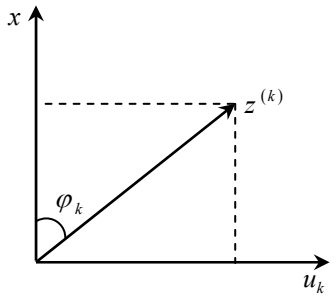
$$z^{(k+1)} = \frac{y^{(k+1)}}{\left\| y^{(k+1)} \right\|};$$

в) подвергнуть вектор $z^{(k+1)}$ тесту на сходимость.

Последовательность $\left\{z^{(k)}\right\}$ называют ***последовательностью Релея***.

В предположении, что последовательность Релея сходится, исследуем ее локальную скорость сходимости. В данном случае (как, впрочем, и в остальных) поведение последовательности удобнее всего описывать с помощью так называемых углов ошибки. Поясним, что это такое.

На k-м шаге рассмотрим однозначно определенную плоскость, содержащую вектор $z^{(k)}$ и собственный вектор x, к которому по предположению сходится наша последовательность. Пусть u_k – единичный вектор этой плоскости, ортогональный вектору x (см. рис.). По мере продолжения алгоритма плоскость поворачивается вокруг фиксированной оси x. Очевидно, для вектора $z^{(k)}$ можно записать разложение



$$z^{(k)} = x\cos\varphi_k + u_k \sin\varphi_k . \qquad (2.6)$$

Умножив равенство (2.6) на матрицу $\left(A - \mu_k E\right)^{-1}$, получим:

$$y^{(k+1)} = \left(A - \mu_k E\right)^{-1} x\cos\varphi_k + \left(A - \mu_k E\right)^{-1} u_k \sin\varphi_k = x\frac{\cos\varphi_k}{\lambda - \mu_k} + \left(A - \mu_k E\right)^{-1} u_k \sin\varphi_k . \qquad (2.7)$$

Прежде всего, заметим, что вектор $\left(A - \mu_k E\right)^{-1} u_k$ по-прежнему ортогонален вектору x. Действительно,

$$\left(x, \left(A - \mu_k E\right)^{-1} u_k\right) = \left(\left(A - \mu_k E\right)^{-1} x, u_k\right) = \left(\frac{1}{\lambda - \mu_k} x, u_k\right) = \frac{1}{\lambda - \mu_k}\left(x, u_k\right) = 0 .$$

Поэтому, переписав (2.7) в виде

$$y^{(k+1)} = x\frac{\cos\varphi_k}{\lambda - \mu_k} + \frac{\left(A - \mu_k E\right)^{-1} u_k}{\left\|\left(A - \mu_k E\right)^{-1} u_k\right\|} \cdot \left\|\left(A - \mu_k E\right)^{-1} u_k\right\| \sin\varphi_k ,$$

мы фактически получаем разложение типа (2.6) на $(k+1)$-м шаге. Так как вектор $y^{(k+1)}$ отличается от вектора $z^{(k+1)}$ лишь нормирующим множителем, то

$$z^{(k+1)} = x\frac{\cos\varphi_k}{(\lambda-\mu_k)\left\|y^{(k+1)}\right\|} + \frac{(A-\mu_k E)^{-1}u_k}{\left\|(A-\mu_k E)^{-1}u_k\right\|}\cdot\frac{\left\|(A-\mu_k E)^{-1}u_k\right\|}{\left\|y^{(k+1)}\right\|}\sin\varphi_k,$$

или

$$z^{(k+1)} = x\cos\varphi_{k+1} + u_{k+1}\sin\varphi_{k+1},$$

где

$$\cos\varphi_{k+1} = \frac{\cos\varphi_k}{(\lambda-\mu_k)\left\|y^{(k+1)}\right\|},\ u_{k+1} = \frac{(A-\mu_k E)^{-1}u_k}{\left\|(A-\mu_k E)^{-1}u_k\right\|},\ \sin\varphi_{k=1} = \frac{\left\|(A-\mu_k E)^{-1}u_k\right\|}{\left\|y^{(k+1)}\right\|}\sin\varphi_k.$$

Отсюда имеем:

$$\operatorname{tg}\varphi_{k+1} = \frac{\sin\varphi_{k+1}}{\cos\varphi_{k+1}} = (\lambda-\mu_k)\left\|(A-\mu_k E)^{-1}u_k\right\|\cdot\operatorname{tg}\varphi_k. \tag{2.8}$$

С другой стороны, из (2.6) следует:

$$\mu_k = \frac{(Az^{(k)},z^{(k)})}{(z^{(k)},z^{(k)})} = \frac{(A(x\cos\varphi_k + u_k\sin\varphi_k),x\cos\varphi_k + u_k\sin\varphi_k)}{(x\cos\varphi_k + u_k\sin\varphi_k,x\cos\varphi_k + u_k\sin\varphi_k)} =$$

$$= \frac{(\lambda x\cos\varphi_k + Au_k\sin\varphi_k,x\cos\varphi_k + u_k\sin\varphi_k)}{\cos^2\varphi_k + \sin^2\varphi_k} = \lambda\cos^2\varphi_k + (Au_k,u_k)\sin^2\varphi_k.$$

Поэтому

$$\lambda-\mu_k = \lambda\sin^2\varphi_k - (Au_k,u_k)\sin^2\varphi_k = [(u,u_k)=1] = (\lambda-\mu(u_k))\sin^2\varphi_k \tag{2.9}$$

и, следовательно,

$$\operatorname{tg}\varphi_{k+1} = (\lambda-\mu(u_k))\cdot\left\|(A-\mu_k E)^{-1}u_k\right\|\cdot\operatorname{tg}^3\varphi_k\cdot\cos^2\varphi_k. \tag{2.10}$$

Теперь напомним, что при всех k вектор u_k остается в инвариантном подпространстве, ортогональном вектору x (будем обозначать это подпространство $x^\perp$). Поэтому, обозначив через $\left[(A-\mu_k E)^{-1}\right]^\perp$ сужение оператора $(A-\mu_k E)^{-1}$ на подпространство $x^\perp$, получим:

$$\left\|(A-\mu_k E)^{-1}u_k\right\| = \left\|\left[(A-\mu_k E)^{-1}\right]^\perp u_k\right\| \le \left\|\left[(A-\mu_k E)^{-1}\right]^\perp\right\| = \frac{1}{\min\limits_{\lambda_i\ne\lambda}|\lambda_i-\mu_k|}. \tag{2.11}$$

Пусть

$$\gamma = \min_{\lambda_i\ne\lambda}|\lambda_i-\lambda|.$$

Так как по предположению последовательность $z^{(k)}$ сходится к x, то $\varphi_k\to 0$ при $k\to\infty$ и, значит, как это следует из (2.9), $\mu_k\to\lambda$. Поэтому для достаточно больших k выполняется неравенство $|\lambda_i-\mu_k|\ge\frac{\gamma}{2}$ для всех значений i. Теперь из (2.10), (2.11) с учетом последнего замечания получаем неравенство

$$\left|\operatorname{tg}\varphi_{k+1}\right| \le \rho\left|\operatorname{tg}\varphi_k\right|^3,$$

которое означает кубическую сходимость рассматриваемого алгоритма, называемого также RQI (Raileigh Quotient Iteration).

При этом дополнительные исследования (см. [9]) позволяют установить, что предельное значение коэффициента ρ не превышает единицы, а также тот факт, что почти во всех случаях имеет место глобальная сходимость итераций с отношением Релея.

2.3. Метод λ-разности

Рассмотрим сейчас применительно к степенному методу возможность нахождения следующих по величине модуля собственных значений матрицы A.

Пусть

$$|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \ldots |\lambda_n| .$$

Будем считать также, что в разложении начального вектора $y^{(0)}$ по собственным векторам матрицы A

$$y^{(0)} = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n$$

коэффициенты α_1 и α_2 отличны от нуля.

Пусть построена степенная последовательность $y^{(0)}, y^{(1)}, \ldots, y^{(k)}, \ldots$ и по какому-либо методу вычислено значение λ_1. Введем обозначение

$$\Delta_\lambda y^{(k)} = y^{(k+1)} - \lambda_1 y^{(k)} \quad (k = 0,1,\ldots) . \tag{2.12}$$

Величину $\Delta_\lambda y^{(k)}$ будем называть λ-разностью от $y^{(k)}$.

Рассмотрим разложение

$$y^{(k)} = \alpha_1 \lambda_1^k x_1 + \alpha_2 \lambda_2^k x_2 + \ldots + \alpha_n \lambda_n^k x_n . \tag{2.13}$$

С его помощью для (2.12) (записывая последнее в координатной форме) получаем:

$$\Delta_{\lambda_1} y_s^{(k)} = \alpha_2 (\lambda_2 - \lambda_1) \lambda_2^k x_{s2} + \ldots + \alpha_n (\lambda_n - \lambda_1) \lambda_n^k x_{sn}$$

и

$$\Delta_{\lambda_1} y_s^{(k-1)} = \alpha_2 (\lambda_2 - \lambda_1) \lambda_2^{k-1} x_{s2} + \ldots + \alpha_n (\lambda_n - \lambda_1) \lambda_n^{k-1} x_{sn} .$$

Если k достаточно велико, то в выражениях для $\Delta_{\lambda_1} y_s^{(k)}$ и $\Delta_{\lambda_1} y_s^{(k-1)}$ доминирующими будут члены, соответствующие λ_2^k и λ_2^{k-1}. Значит,

$$\lambda_2 \approx \frac{\Delta_{\lambda_1} y_s^{(k)}}{\Delta_{\lambda_1} y_s^{(k-1)}} . \tag{2.14}$$

Заметим, что если мы определяли λ_1 по формуле $\lambda_1 \approx \frac{y_s^{(k+1)}}{y_s^{(k)}}$, то λ_2 целесообразно вычислять по формуле $\lambda_2 \approx \frac{\Delta_{\lambda_1} y_s^{(m)}}{\Delta_{\lambda_1} y_s^{(m-1)}}$, где $m < k$ и m является наименьшим из чисел, при котором преимущество λ_2^m над следующими членами вида λ_i^m уже начинает сказываться. Эта формула имеет то преимущество перед (2.12), что нам не приходится вычитать близкие друг к другу числа, в то время как $y_s^{(k+1)} - \lambda_1 y_s^{(k)} \approx 0$ и $y_s^{(k)} - \lambda_1 y_s^{(k-1)} \approx 0$.

В качестве собственного вектора матрицы A, отвечающего λ_2, приближенно можно взять вектор $\Delta_{\lambda_1} y^{(k)}$.

Теоретически возможно метод λ-разности применять и к вычислению следующих по величине модуля собственных значений, однако результаты будут еще менее надежными, чем в случае λ_2. Причины этого явления кроются в том, что названные вычисления связаны с операцией уничтожения главной части в линейных выражениях вида (2.13), что, как правило, влечет за собой потерю значащих цифр.

2.4. Возведение матрицы в степень

Для построения высоких итераций вектора иногда целесообразно возвести данную матрицу в степень. Наиболее просто вычисляются последовательные степени матрицы A: A^2, A^4, A^8,.... Однако возведение матрицы в квадрат, очевидно, по объему работы равносильно образованию n итераций вектора, так что образование матрицы A^{2^k} равносильно построению nk итераций. Соответственно, вычисление $A^{2^k} y^{(0)}$ равносильно вычислению $kn+1$ итераций и поэтому выигрыш в объеме работы получается, если $kn < 2^k$, т.е. если число итераций, необходимых для получения нужной точности, превышает $n\log_2 n$.

Можно ограничиться вычислением некоторой фиксированной степени матрицы A и затем составлять итерации посредством вычисленной степени. Например, вычислив A^8, можно найти $A^8 y^{(0)}$, затем $A^{16} y^{(0)}$ и т.д.

Степени матрицы A могут быть использованы и непосредственно для нахождения наибольшего по модулю собственного значения, если оно простое и вещественное.

Действительно, имеем:

$$\left|\lambda_1\right| \approx \sqrt[2^k]{\operatorname{Sp} A^{2^k}}, \tag{2.15}$$

поскольку

$$\operatorname{Sp} A^m = \lambda_1^m + \lambda_2^m + \ldots + \lambda_n^m$$

и, следовательно,

$$\sqrt[m]{\operatorname{Sp} A^m} = \lambda_1 \sqrt[m]{1 + \left(\frac{\lambda_2}{\lambda_1}\right)^m + \ldots + \left(\frac{\lambda_n}{\lambda_1}\right)^m} = \lambda_1 + O\left(\frac{1}{m}\left|\frac{\lambda_2}{\lambda_1}\right|^m\right).$$

Отметим, что несколько более удобной по сравнению с (2.15) является формула

$$\lambda_1 \approx \frac{\operatorname{Sp} A^{2^k+1}}{\operatorname{Sp} A^{2^k}}. \tag{2.16}$$

Вообще говоря, метод следов может быть распространен также на случай кратных и комплексных корней.

2.5. Применение степенного метода к отысканию нескольких собственных значений

Ранее мы рассмотрели несколько случаев, когда наибольшее по модулю собственное значение не изолировано, т.е. когда имелось другое собственное значение, равное ему по модулю. Процедура несколько усложнялась по сравнению с исходной, однако позволяла решить задачу. В соответствующем замечании мы говорили о возможности ее обобщения, позволяющей с единых позиций строить итерационный процесс. Сейчас же мы с помощью изложенной там идеи приспособим итерационный процесс для нахождения нескольких собственных значений.

Пусть элементарные делители матрицы A взаимно просты и

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_r| > |\lambda_{r+1}| \geq \ldots \geq |\lambda_n|.$$

Как и ранее, выбрав начальный вектор $y^{(0)}$, запишем его разложение по базису из собственных векторов матрицы A:

$$y^{(0)} = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_r x_r + \alpha_{r+1} x_{r+1} + \ldots + \alpha_n x_n,$$

причем будем считать, что все α_i, $i = 1,...,n$, отличны от нуля.

Пусть также

$$t^r + \beta_1 t^{r-1} + \ldots + \beta_r = (t - \lambda_1)\ldots(t - \lambda_r) -$$

полином, корнями которого являются искомые собственные значения $\lambda_1,...,\lambda_r$.

Тогда имеет место равенство

$$y^{(k+r)} + \beta_1 y^{(k+r-1)} + \ldots + \beta_r y^{(k)} \approx 0. \qquad (2.17)$$

Действительно,

$$\begin{aligned}
y^{(k)} &= \alpha_1 \lambda_1^k x_1 + \alpha_2 \lambda_2^k x_2 + \ldots + \alpha_r \lambda_r^k x_r + \alpha_{r+1} \lambda_{r+1}^k x_{r+1} + \ldots + \alpha_n \lambda_n^k x_n, \\
y^{(k+1)} &= \alpha_1 \lambda_1^{k+1} x_1 + \alpha_2 \lambda_2^{k+1} x_2 + \ldots + \alpha_r \lambda_r^{k+1} x_r + \alpha_{r+1} \lambda_{r+1}^{k+1} x_{r+1} + \ldots + \alpha_n \lambda_n^{k+1} x_n, \\
&\ldots\ldots \\
y^{(k+r)} &= \alpha_1 \lambda_1^{k+r} x_1 + \alpha_2 \lambda_2^{k+r} x_2 + \ldots + \alpha_r \lambda_r^{k+r} x_r + \alpha_{r+1} \lambda_{r+1}^{k+r} x_{r+1} + \ldots + \alpha_n \lambda_n^{k+r} x_n,
\end{aligned}$$

Складывая эти соотношения с коэффициентами, указанными в (2.17), получаем:

$$\begin{aligned}
y^{(k+r)} + \beta_1 y^{(k+r-1)} + \ldots + \beta_r = {} & \alpha_1 x_1 \lambda_1^k \left(\lambda_1^r + \beta_1 \lambda_1^{r-1} + \ldots + \beta_r\right) + \alpha_2 x_2 \lambda_2^k \left(\lambda_2^r + \beta_1 \lambda_2^{r-1} + \ldots + \beta_r\right) + \\
& + \ldots + \alpha_r x_r \lambda_r^k \left(\lambda_r^r + \beta_1 \lambda_r^{r-1} + \ldots + \beta_r\right) + \alpha_{r+1} x_{r+1} \lambda_{r+1}^k \left(\lambda_{r+1}^r + \beta_1 \lambda_{r+1}^{r-1} + \ldots + \beta_r\right) + \\
& + \ldots + \alpha_n x_1 \lambda_n^k \left(\lambda_n^r + \beta_1 \lambda_n^{r-1} + \ldots + \beta_r\right) \approx 0
\end{aligned}$$

Векторное равенство (2.17) равносильно системе из n равенств для соответствующих компонент. Взяв какие-либо r из них, получим систему из r линейных алгебраических уравнений относительно $\beta_1, \beta_2,...,\beta_r$. Пусть, например, выбраны первые r компонент. Тогда будем иметь:

$$\begin{cases}
y_1^{(k+r)} + \beta_1 y_1^{(k+r-1)} + \ldots + \beta_r y_1^{(k)} \approx 0, \\
\quad \ldots\ldots \\
y_r^{(k+r)} + \beta_1 y_r^{(k+r-1)} + \ldots + \beta_r y_r^{(k)} \approx 0.
\end{cases} \qquad (2.18)$$

Отсюда, например, по правилу Крамера, получим для коэффициентов β_i приближенные значения

$$\beta_1 = \frac{\begin{vmatrix} -y_1^{(k+r)} & y_1^{(k+r-2)} & \dots & y_1^{(k)} \\ & \dots & \dots & \\ -y_r^{(k+r)} & y_r^{(k+r-2)} & \dots & y_r^{(k)} \end{vmatrix}}{\begin{vmatrix} y_1^{(k+r-1)} & y_1^{(k+r-2)} & \dots & y_1^{(k)} \\ & \dots & \dots & \\ y_r^{(k+r-1)} & y_r^{(k+r-2)} & \dots & y_r^{(k)} \end{vmatrix}}, \dots, \beta_r = \frac{\begin{vmatrix} y_1^{(k+r-1)} & \dots & y_1^{(k+1)} & -y_1^{(k+r)} \\ & \dots & \dots & \\ y_r^{(k+r-1)} & \dots & y_1^{(k+1)} & -y_r^{(k+r)} \end{vmatrix}}{\begin{vmatrix} y_1^{(k+r-1)} & y_1^{(k+r-2)} & \dots & y_1^{(k)} \\ & \dots & \dots & \\ y_r^{(k+r-1)} & y_r^{(k+r-2)} & \dots & y_r^{(k)} \end{vmatrix}} \quad . \quad (2.19)$$

Несложно показать, что все равенства (2.19) справедливы с точностью $O\left(\left|\frac{\lambda_{r+1}}{\lambda_r}\right|^k\right)$.

Заметим, что вместо r различных компонент векторов $y^{(k+r)}, \dots, y^{(k)}$ можно брать какую-либо одну компоненту из наборов $y^{(k+r)}, y^{(k+r-1)}, \dots, y^{(k)}$; $y^{(k+r+1)}, y^{(k+r)}, \dots, y^{(k+1)}$; $\dots$; $y^{(k+2r-1)}, y^{(k+2r-2)}, \dots, y^{(k+r-1)}$.

При практических вычислениях на самом деле нет необходимости использовать правило Крамера. Систему обычно решают одним из известных способов. Отметим, что хороший результат может быть получен, если первые r собственных значений близки по модулю, а $(r+1)$-е собственное значение сильно отрывается от них. Если же это не так, то система (2.18) будет очень плохо обусловлена.

Теоретически указанным способом можно построить весь собственный полином (вернее – минимальный аннулирующий вектор $y^{(0)}$ полином), приняв $r = n$. В этом случае мы придем к методу Крылова, выполненному исходя из вектора $A^k y^{(0)}$. Конечно, в этом случае нужно брать $k = 0$, чтобы не вычислять лишней работы, к тому же с ростом k ухудшается и обусловленность метода Крылова.

В случае

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_r| > |\lambda_{r+1}| \geq \dots \geq |\lambda_n|$$

можно несколько изменить описанный процесс. Именно, в этом случае достаточно вычислить лишь свободные члены последовательных многочленов

$$t - \lambda_1, \ (t-\lambda_1)(t-\lambda_2), \dots, (t-\lambda_1)\dots(t-\lambda_r),$$

так как эти числа с точностью до знака равны произведениям соответствующих собственных значений. Расчетные формулы на этот раз будут иметь вид

$$\lambda_1 \approx \frac{y_1^{(k+1)}}{y_1^{(k)}}; \quad \lambda_1\lambda_2 \approx \frac{\begin{vmatrix} y_1^{(k+2)} & y_1^{(k+1)} \\ y_2^{(k+2)} & y_2^{(k+1)} \end{vmatrix}}{\begin{vmatrix} y_1^{(k+1)} & y_1^{(k)} \\ y_2^{(k+1)} & y_2^{(k)} \end{vmatrix}}; \dots; \quad \lambda_1 \dots \lambda_r = \frac{\begin{vmatrix} y_1^{(k+r)} & \dots & y_1^{(k+1)} \\ & \dots & \\ y_r^{(k+r)} & \dots & y_r^{(k+1)} \end{vmatrix}}{\begin{vmatrix} y_1^{(k+r-1)} & \dots & y_1^{(k)} \\ & \dots & \\ y_r^{(k+r-1)} & \dots & y_r^{(k)} \end{vmatrix}} \quad . \quad (2.20)$$

Заметим, что определение произведения даже двух первых собственных значений наталкивается на препятствие в виде исчезновения значащих цифр, так как при достаточно больших k строки определителей становятся почти пропорциональными.

Литература

1. Воеводин В.В. Вычислительные основы линейной алгебры. – М.: Наука. – 1987.
2. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры. – Физматгиз. – 1963.
3. Крылов В.И., Бобков В.В., Монастырный П.И. Вычислительные методы высшей математики. – Т. 1. – Мн.: Вышэйшая школа. – 1972.
4. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука. – 1989.
5. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. – М.: Наука. – 1978.
6. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Наука. – 1987.
7. Калиткин Н.Н. Численные методы. – М.: Наука. – 1978.
8. Уилкинсон Дж.Х. Алгебраическая проблема собственных значений. – М.: Наука. – 1970.
9. Парлетт Б. Симметричная проблема собственных значений. – М.: Мир. – 1983.
10. Крылов В.И., Бобков В.В., Монастырный П.И. Вычислительные методы. – Т. 1. – М.: Наука. – 1976.
11. Тыртышников Е.Е. Методы численного анализа. – М.: ИЦ «Академия». – 2007.