

Вычислительные методы алгебры
Конспект по 3 курсу специальности «прикладная
математика»
(лектор А. М. Будник)

Оглавление

1	Введение.	3
1.1	Математическое моделирование и вычислительные эксперименты.	3
1.2	Структура погрешности. Корректность.	3
1.3	Элементы теории погрешностей.	4
2	Методы решения СЛАУ.	6
2.1	Нормы вектора и матриц.	6
2.2	Обусловленность СЛАУ.	7
2.3	Метод Гаусса (метод последовательного исключения неизвестных).	8
2.3.1	Схема единственного деления.	9
2.3.2	Метод Гаусса с выбором главного элемента.	10
2.3.3	Вычисление определителей и обратных матриц.	11
2.4	LU-разложение.	11
2.4.1	Теорема существования и единственности.	11
2.4.2	Схема единственного деления и LU -разложение.	13
2.4.3	Метод Гаусса с выбором главного элемента и LU -разложение.	13
2.5	Метод квадратного корня.	15
2.6	Метод прогонки.	16
2.6.1	Алгоритм метода прогонки.	17
2.6.2	Метод прогонки и метод Гаусса.	18
2.6.3	Обоснование метода прогонки.	19
2.7	Методы основанные на ортогональных преобразованиях.	21
2.7.1	QR-разложение.	21
2.7.2	Метод отражений.	22
2.7.3	Метод вращений.	24
2.8	Метод простой итерации.	26
2.8.1	Классификация итерационных процессов. Основные определения.	26
2.8.2	Сходимость матричной геометрической прогрессии.	28
2.8.3	Условия сходимости метода простой итерации.	29
2.8.4	Метод Якоби как частный случай метода простой итерации.	30
2.9	Метод Зейделя.	31
2.9.1	Описание и сходимость метода Зейделя.	32
2.9.2	Метод Гаусса-Зейделя.	34
2.10	Итерационные методы вариационного типа.	34
2.10.1	Решение СЛАУ и точка минимума квадратичного функционала ошибки.	34
2.10.2	Метод покоординатного спуска.	36
2.10.3	Метод градиентного спуска.	37
2.10.4	Метод релаксации.	39
2.10.5	Метод минимальных невязок.	40

3	Методы решения проблемы собственных значений (ПСЗ).	42
3.1	Общие положения ПСЗ.	42
3.2	Метод Данилевского.	44
3.2.1	Построение собственного многочлена.	44
3.2.2	Вычисление собственных векторов матриц.	46
3.3	Метод Крылова.	47
3.3.1	Построение собственного многочлена.	48
3.3.2	Вычисление собственных векторов матрицы.	50
3.4	Итерационный метод вращений решения ППСЗ.	51
3.4.1	Предварительные сведения.	51
3.4.2	Алгоритм метода.	52
3.4.3	Сходимость итерационного процесса.	54
3.5	Степенной метод.	55
3.5.1	Нахождение наибольшего по модулю собственного значения и соответствующего собственного вектора.	55
3.5.2	Нахождение наименьшего по модулю собственного значения (метод обратных итераций).	58
3.5.3	Случай кратного доминирующего собственного значения.	58
3.5.4	Случай двух наибольших собственных значений, отличающихся знаками.	59
3.5.5	Нахождение собственного значения второго по величине	60
3.6	Методы построения собственного многочлена основанные на использовании формулы Ньютона.	60
3.6.1	Метод Леверье.	60
3.6.2	Метод Фадеева.	61
3.7	Построения собственного многочлена, использующие диагональные преобразования.	63
3.7.1	Преобразование матрицы по методу отражений.	63
3.7.2	Преобразование матрицы по методу вращений.	65
3.8	QR-метод (алгоритм).	65

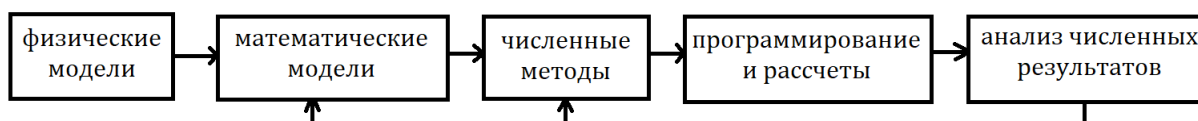
Глава 1

Введение.

1.1 Математическое моделирование и вычислительные эксперименты.

- *Математическое моделирование* — описание реальных явлений, процессов и объектов с помощью математических уравнений.
- *Вычислительный эксперимент* — исследование естественных научных проблем средствами научной математики.

Схема вычислительного эксперимента



В основе вычислительного эксперимента лежит решение уравнений математической модели численными методами. Под численными методами понимаются методы, работающие с приближенными данными.

1.2 Структура погрешности. Корректность.

Формально любую задачу можно определить следующим образом: если требуется определить некоторую величину y по величине x , то символически задачу можно записать в виде

$$y = A(x),$$

где A — некоторый оператор, действующий на величину x , а y и x — числа, совокупности чисел, функции и так далее. Если оператор A настолько сложный, что решение не удастся явно выписать, то задачу решаем приближенно с помощью любого метода. При этой замене возникает погрешность.

Причины возникновения погрешностей в решении задач:

1. математическое описание неточное;
2. исходные данные заданы приближенно;
3. метод не является точным;

4. при вводе данных, выполнении математических операций и выводе производится округление из-за ограниченности расчетной сетки.

Запишем погрешности, соответствующие причинам:

1. погрешность математической модели;
 2. неустранимая погрешность;
 3. погрешность метода;
 4. вычислительная погрешность.
- Задача $y = A(x)$ называется **корректно поставленной**, если
 1. \forall входных данных $\exists!$ решение y ;
 2. решение y устойчиво по входным данным.
 - Под **устойчивостью решения** y по входным данным x понимается непрерывная зависимость погрешности Δx решения от погрешности Δy входных данных (то есть если $\Delta x \rightarrow 0$, то $\Delta y \rightarrow 0$).
 - При невыполнении хотя бы одного из этих условий задача называется **некорректной**.

1.3 Элементы теории погрешностей.

Пусть x — точное, а x^* — приближенное значение некоторой величины. Тогда число $\varepsilon = x - x^*$ — погрешность приближенного значения x^* .

Модуль величины погрешности оценивается

$$|x - x^*| = |\varepsilon| \leq \Delta(x^*),$$

где $\Delta(x^*)$ — **абсолютная погрешность**.

В свою очередь,

$$\left| \frac{x - x^*}{x^*} \right| = \left| \frac{\varepsilon}{x^*} \right| \leq \frac{\Delta(x^*)}{|x^*|} = \delta(x^*),$$

где $\delta(x^*)$ — **относительная погрешность**.

Основная задача теории погрешностей — указание области неопределенности результатов.

Рассмотрим задачу теории погрешностей. Пусть имеется функция

$$y = f(x_1, x_2, \dots, x_n).$$

Прямая задача теории погрешностей заключается в определении абсолютной погрешности вычисления функции по абсолютной погрешности определения аргументов. Обозначим

$$x^* = (x_1^*, x_2^*, \dots, x_n^*)$$

— приближенное значение аргументов функции f , для которой известны $\Delta(x_i^*)$, $i = \overline{1, n}$;

$$x = (x_1, x_2, \dots, x_n)$$

— точное значение аргументов. Воспользуемся формулой конечных приращений Лагранжа

$$f(x) = f(x^*) + \sum_{i=1}^n \frac{\partial f(x^* + \theta(x - x^*))}{\partial x_i} \cdot (x_i - x_i^*), \quad 0 < \theta < 1.$$

С помощью нее можно осуществить простую оценку погрешности результата. Учитывая, что

$$|x_i - x_i^*| \leq \Delta(x_i^*),$$

мы можем записать оценку погрешности в виде

$$|f(x) - f(x^*)| \leq \sum_{i=1}^n \sup_x \left| \frac{\partial f^*}{\partial x_i} \right| \Delta(x_i^*) = \Delta(f^*) \quad (1)$$

Эта оценка дает возможность вычислить погрешность результата по $\Delta(x_i^*)$.

Обратная задача теории погрешностей заключается в определении погрешности аргументов по известной погрешности функции. Различают 2 подхода решения обратной задачи:

1. *принцип равных влияний*

Мы будем предполагать, что если

$$\left| \frac{\partial f^*}{\partial x_i} \right| \Delta(x_i^*) = const, \forall i = \overline{1, n},$$

то можно сделать оценку $\Delta(x_i^*)$ из формулы (1):

$$\Delta(x_i^*) \leq \frac{\Delta(f^*)}{n \left| \frac{\partial f^*}{\partial x_i} \right|}, \forall i = \overline{1, n}.$$

2. *принцип равных погрешностей*

Мы будем предполагать, что если $\Delta(x_i^*) = const, \forall i = \overline{1, n}$, то

$$\Delta(x_k^*) \leq \frac{\Delta(f^*)}{\sum_{i=1}^n \left| \frac{\partial f^*}{\partial x_i} \right|}, \forall k = \overline{1, n}.$$

Глава 2

Методы решения СЛАУ.

Выделим 4 основных задачи линейной алгебры:

1. решение СЛАУ $Ax = b$, где A — квадратная матрица, x — искомый вектор, b — заданный вектор;
2. вычисление определителя квадратной матрицы $|A| = \det A$;
3. нахождение обратной матрицы A^{-1} ;
4. проблемы собственных значений, то есть определение собственных значений λ_i и собственных векторов x_i

2.1 Нормы вектора и матриц.

• **Нормой вектора** $x = (x_1, \dots, x_n)$ называется действительное число $\|x\|$, удовлетворяющее условиям

1. $\|x\| \geq 0 \forall x \neq 0; \|x\| = 0 \Leftrightarrow x = 0$
2. $\|cx\| = |c| \cdot \|x\|, c = \text{const}$
3. $\|x + y\| \leq \|x\| + \|y\|$

Основные векторные нормы:

1. кубическая: $\|x\|_1 = \max_i |x_i|$;
2. октаэдрическая $\|x\|_2 = \sum_{i=1}^n |x_i|$;
3. сферическая $\|x\|_3 = \left(\sum_{i=1}^n |x_i| \right)^{\frac{1}{2}}$.

• **Нормой матрицы** A размерности $n \times n$ с элементами a_{ij} называется число $\|A\|$, удовлетворяющее условиям

1. $\|A\| \geq 0 \forall A \neq 0; \|A\| = 0 \Leftrightarrow A = 0$
2. $\|cA\| = |c| \cdot \|A\|, c = \text{const}$
3. $\|A + B\| \leq \|A\| + \|B\|$
4. $\|AB\| \leq \|A\| \cdot \|B\|$

- Если выполняется неравенство

$$\|Ax\| \leq \|A\| \cdot \|x\|,$$

то норма матрицы называется **согласованной** с данной нормой вектора.

- Если при условии $\|Ax\| \leq \|A\| \cdot \|x\|$ выполняется также

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

то норма матрицы называется **подчиненной** данной норме вектора.

Основные матричные нормы:

1. $\|A\|_1 = \max_i \sum_{j=1}^n |a_{ij}|;$
2. $\|A\|_2 = \max_j \sum_{i=1}^n |a_{ij}|;$
3. $\|A\|_3 = \sqrt{\Lambda}, \Lambda = \max \lambda(A^*A).$

2.2 Обусловленность СЛАУ.

Рассмотрим СЛАУ в векторной форме $Ax = b$, где A — квадратная матрица. Данная задача будет являться корректно поставленной, если

1. \forall входных данных $\exists!$ решение, то есть, в нашем случае, $|A| \neq 0$;
2. решение устойчиво по входным данным.

Для рассматриваемой задачи входными данными являются компоненты вектора $b = (b_1, \dots, b_n)^T$ и коэффициенты a_{ij} , $i, j = \overline{1, n}$ матрицы A . Обозначим через $\delta b, \delta A$ возмущения вносимые по входным данным.

Тогда рассмотрим СЛАУ

$$(A + \delta A)x_B = b + \delta b,$$

где $x_B = x + \delta x$ — решение возмущенной СЛАУ. Представим решение исходной системы как

$$x = A^{-1}b.$$

Варьируем это равенство (операция подобная дифференцированию):

$$\delta x = \delta A^{-1}b + A^{-1}\delta b.$$

В частности, $\delta E = 0$, т.к. E — константная матрица. Мы можем утверждать, что

$$E = AA^{-1} \Rightarrow \delta A \cdot A^{-1} = -A\delta A^{-1} \Rightarrow \delta A^{-1} = -A^{-1}\delta AA^{-1}.$$

Подставим в рассмотренное выше равенство и получим формулу для выражения возмущения δx :

$$\delta x = A^{-1}(\delta b - \delta Ax).$$

Оценим обе части

$$\|\delta x\| \leq \|A^{-1}\| \cdot (\|\delta b\| + \|\delta A\| \cdot \|x\|).$$

Это неравенство устанавливает непрерывность полученного решения от входных данных, то есть

$$\|\delta b\| \rightarrow 0, \|\delta A\| \rightarrow 0 \Rightarrow \|\delta x\| \rightarrow 0.$$

Таким образом, из данного неравенства следует корректность поставленной задачи.

Заметим, что в нашем случае $\|\delta x\|$ — аналог абсолютной погрешности. Тогда $\frac{\|\delta x\|}{\|x\|}$ — аналог относительной погрешности. Найдем оценку относительной погрешности:

$$\begin{aligned} \frac{\|\delta x\|}{\|x\|} &\leq \|A^{-1}\| \cdot \|A\| \cdot \left(\frac{\|\delta b\|}{\|A\| \cdot \|x\|} + \frac{\|\delta A\|}{\|A\|} \right) \leq \left[\|A\| \cdot \|x\| \geq \|b\| \right] \leq \underbrace{\|A^{-1}\| \cdot \|A\|}_{\nu(A)} \cdot \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) = \\ &= \nu(A) \cdot \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right). \end{aligned}$$

• Число $\nu(A)$ называется **числом обусловленности матрицы A** (т.ж. обозначается $\text{cond } A$).

Из оценок абсолютной и относительной погрешностей видно, что чем больше $\nu(A)$, тем больше погрешность решения системы.

• Матрица с большим числом обусловленности называется **плохо обусловленной**.

• Системы с плохо обусловленной матрицей называются **плохо обусловленными системами**.

Замечания.

1. Если $\|A^{-1}\| \cdot \|\delta A\| < 1$, то вместо полученной ранее оценки относительной погрешности можно получить более точную оценку

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\nu(A)}{1 - \nu(A) \frac{\|\delta A\|}{\|A\|}} \cdot \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right).$$

2. Необходимым условием плохой обусловленности является $|A| \approx 0$. Но оно не является достаточным.
3. $\nu(A) \geq \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \geq 1$.
4. $\nu(AB) \leq \nu(A)\nu(B)$.
5. Число обусловленности определяется не только матрицей, но и выбором нормы. Например, в сферической норме

$$A = A^T \Rightarrow \nu(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

2.3 Метод Гаусса (метод последовательного исключения неизвестных).

Методы численного решения СЛАУ делятся на 2 группы:

1. точные (прямые) методы;
2. итерационные методы.

Точные методы представляют собой способы решения СЛАУ за конечное число шагов. Итерационные методы являются приближенными методами, позволяющими получить решения систем с помощью бесконечно сходящихся процессов.

Метод Гаусса основан на последовательном переходе с помощью элементарных преобразований от исходной системы к эквивалентной ей системе с верхней треугольной матрицей.

2.3.1 Схема единственного деления.

Запишем систему в координатном виде:

[illegible]

Прямой ход. Предположим, что $a_{11} \neq 0$. Разделим первое уравнение системы (1) на a_{11} и получим:

$$x_1 + c_{11}x_2 + \dots + c_{1n}x_n = q_1 \quad (2),$$

где $c_{1j} = \frac{a_{1j}}{a_{11}}$, $j = \overline{2, n}$, $q_1 = \frac{b_1}{a_{11}}$. С помощью уравнения (2) исключим x_1 из всех остальных уравнений, начиная со 2-го. Таким образом, первое уравнение не поменяется, а все остальные примут вид

[illegible]

где $a_{ij}^1 = a_{ij} - c_{1j}a_{i1}$, $b_i^1 = b_i - q_1a_{i1}$, $i, j = \overline{2, n}$. Мы завершили первый шаг прямого хода.

- Элемент a_{11} называется **ведущим** элементом шага.

Далее считаем, что a_{22} ведущий элемент. Аналогично делим на него и исключаем x_2 и так далее до x_{nn} . Окончательно приходим к системе с верхней треугольной матрицей следующего вида

[illegible]

Для того, чтобы получить общие формулы вычисления матрицы (4) введем следующие обозначения:

1. $a_{kj}^0 = a_{kj}, k, j = \overline{1, n}$ — исходные элементы матрицы;
2. $b_k^0 = b_k, k = \overline{1, n}$ — компоненты вектора правых частей исходной матрицы.

Тогда

$$c_{kj} = \frac{a_{kj}^{k-1}}{a_{kk}^{k-1}}, \quad j = \overline{k+1, n}, \quad k = \overline{1, n-1},$$

$$a_{ij}^k = a_{ij}^{k-1} - a_{ik}^{k-1} c_{kj}, \quad i, j = \overline{k+1, n}, \quad k = \overline{1, n-1}, \quad (5)$$

$$q_k = \frac{b_k^{k-1}}{a_{kk}^{k-1}}, \quad k = \overline{1, n},$$

$$b_i^k = b_i^{k-1} - a_{ik}^{k-1} q_k, \quad i = \overline{k+1, n}, \quad k = \overline{1, n-1}. \quad (6)$$

Метод Гаусса завершен, числа a_{kk}^{k-1} , $k = \overline{1, n}$ — ведущие элементы.

Обратный ход. Обратный ход состоит в последовательном нахождении неизвестных

$$x_i = q_i - \sum_{j=i+1}^n c_{ij} x_j, \quad i = \overline{n-1, 1}. \quad (7)$$

$$x_n = q_n.$$

Формулы (5)-(7) составляют **алгоритм схемы единственного деления**. Сложность данного алгоритма $O(n^3) = \frac{2}{3}n^3 + O(n^2)$. В частности, все операции умножения и деления занимают $\frac{1}{3}n(n^2 + 3n - 1)$ действий.

2.3.2 Метод Гаусса с выбором главного элемента.

Недостаток схемы единственного деления состоит в том, что при его реализации накапливается вычислительная погрешность. Особенно, когда на каждом k -м шаге $a_{kk}^{k-1} \approx 0$. Для того, чтобы снизить темп роста погрешности используется видоизменение метода Гаусса — методом Гаусса с выбором главного элемента. Основная идея состоит в том, чтобы в качестве ведущего элемента на каждом этапе исключения выбирается наибольший по модулю элемент, который называется *главным*. Возможны две стратегии выбора главного элемента:

1. частичный выбор главного элемента:

(а) выбор главного элемента по столбцу.

На каждом k -ом шаге прямого хода в качестве ведущего выбирается максимальный по модулю элемент в неприведенной части столбца. После этого строка расширенной матрицы соответствующая главному элементу переставляется с k -ой строкой и производится перенумерация коэффициентов при неизвестных.

(б) выбор главного элемента по строке.

В данном случае мы будем выбирать на каждом шаге максимальный по модулю элемент соответствующий номеру шага.

2. полный выбор главного элемента

(а) выбор главного элемента по всей матрице.

Будем искать главный элемент на каждом шаге в неприведенной части матрицы размерности $(n-k) \times (n-k)$. Причем количество арифметических операций $Q(n) = n^3 + O(n^2)$, $n \rightarrow \infty$.

С вычислительной точки зрения наиболее экономичной является модификация 1а. Остальные варианты требуют дополнительного вектора. Причем в последнем случае необходимо осуществлять поиск максимального элемента в матрице.

2.3.3 Вычисление определителей и обратных матриц.

Выполнение преобразований матрицы A в процессе решения системы (1) позволяет без дополнительных затрат вычислить определитель этой матрицы, а также найти обратную матрицу.

$$|A| = (-1)^m \cdot a_{11}^0 \cdot a_{22}^1 \cdot \dots \cdot a_{nn}^{n-1}, \quad (8)$$

где m — количество перестановок, осуществленных при прямом ходе метода Гаусса. В схеме единственного деления можно считать $m = 0$.

Задача нахождения матрицы обратной матрице A эквивалентна задаче решения матричного уравнения

$$AX = E,$$

где $X = A^{-1}$ — искомая матрица. Если обозначить $x^{(1)}, \dots, x^{(n)}$ — столбцы матрицы X , то эта матрица может быть найдена по столбцам решения системы вида

$$Ax^{(j)} = \delta^{(j)}, \quad \delta^{(j)} = (\delta_{1j}, \dots, \delta_{nj})^T, \quad x^{(j)} = (x_{1j}, \dots, x_{nj})^T, \quad j = \overline{1, n} \quad (9)$$

Решение всех n систем доставит нам все столбцы обратной матрицы. Таким образом, задача нахождения обратной матрицы сводится к применению метода Гаусса n раз к системе вида (9).

2.4 LU-разложение.

В основе всех точных методов лежит возможность представления любой матрицы A в виде произведения двух треугольных матриц. Одну из них мы будем обозначать L — нижняя треугольная матрица. Вторую соответственно U — верхняя треугольная матрица. Тогда решение исходной системы

$$Ax = b$$

сводится к решению двух систем с треугольными матрицами

$$Ly = b, Ux = y. \quad (1)$$

2.4.1 Теорема существования и единственности.

• **LU-разложением** матрицы A n -ого порядка будем называть ее представление в виде произведения левой нижней треугольной матрицы L с ненулевыми диагональными элементами на правую верхнюю треугольную матрицу U с диагональными элементами равными единице. То есть

$$A = LU = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{12} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \begin{pmatrix} 1 & u_{12} & \dots & u_{1n} \\ 0 & 1 & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}, \quad (2)$$

$$l_{ii} \neq 0, \quad i = \overline{1, n}$$

Теорема. Любую квадратную матрицу A n -ого порядка, у которой все угловые диагональные миноры

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

отличны от нуля, можно представить причем единственным образом в виде LU -разложения.

◆ Доказательство проведем по индукции. При $n = 1$

$$l_{11} = a_{11} \neq 0$$

тогда утверждение очевидно.

Пусть теорема верна при $n = n - 1$. Докажем возможность LU -разложения для разложения матрицы $n \times n$. Представим матрицу A в блочном виде

$$A = \left(\begin{array}{c|c} A_{n-1} & w \\ \hline v & a_{nn} \end{array} \right) \quad (3)$$

$$A_{n-1} = \begin{pmatrix} a_{11} & \dots & a_{1n-1} \\ \vdots & \ddots & \vdots \\ a_{n-11} & \dots & a_{n-1n-1} \end{pmatrix}, w = \begin{pmatrix} a_{1n} \\ \vdots \\ a_{n-1n} \end{pmatrix}, v = (a_{n1}, \dots, a_{nn-1})$$

Найдем наше LU -разложение, записав матрицу A в виде

$$A = \left(\begin{array}{c|c} L_{n-1} & 0 \\ \hline l & l_{nn} \end{array} \right) \left(\begin{array}{c|c} U_{n-1} & u \\ \hline 0 & 1 \end{array} \right), \quad (4)$$

где $l = (l_{n1}, \dots, l_{nn-1})$, $u = \begin{pmatrix} u_{1n} \\ \vdots \\ u_{n-1n} \end{pmatrix}$. В разложении (4) l и u — неизвестные пока векторы.

Если мы найдем формулы, по которым можем их вычислить, то докажем теорему. Для этого умножим две матрицы из формулы (4) и приравняем их к матрице (3):

$$\left(\begin{array}{c|c} L_{n-1}U_{n-1} & L_{n-1}u \\ \hline lU_{n-1} & lu + l_{nn} \end{array} \right) = \left(\begin{array}{c|c} A_{n-1} & w \\ \hline v & a_{nn} \end{array} \right)$$

Отсюда получаем

$$L_{n-1}u = w,$$

$$lU_{n-1} = v,$$

$$lu + l_{nn} = a_{nn}.$$

По предположению матрицы L_{n-1} и U_{n-1} невырожденные, следовательно, существуют обратные матрицы. В итоге можем вычислить

$$u = L_{n-1}^{-1}w,$$

$$l = vU_{n-1}^{-1},$$

$$l_{nn} = a_{nn} - lu.$$

Для того, чтобы завершить доказательство существования, нужно доказать, что $l_{nn} \neq 0$. Доказательство данного утверждения проведем, используя равенство (4):

$$|A| = \left| \begin{array}{c|c} L_{n-1} & 0 \\ \hline l & l_{nn} \end{array} \right| \cdot \left| \begin{array}{c|c} U_{n-1} & u \\ \hline 0 & 1 \end{array} \right| = |L_{n-1}| \cdot l_{nn} \cdot |U_{n-1}|.$$

По условию теоремы $|A| \neq 0$. Далее $|U_{n-1}| = 1$, а $|L_{n-1}| \neq 0$ по индуктивному предположению. Следовательно, $l_{nn} \neq 0$. Мы доказали существование LU -разложения.

Покажем, что такое разложение единственно. Докажем от противного, пусть

$$A = L_1 U_1 = L_2 U_2.$$

Если умножим обе части последнего равенства слева на L_1^{-1} , а справа на U_2^{-1} . В итоге получим

$$L_1^{-1} L_2 = U_1 U_2^{-1}.$$

Получаем слева нижнюю треугольную матрицу, а справа верхнюю треугольную матрицу. А такое возможно лишь когда обе матрицы диагональны. Следовательно, $U_1 U_2^{-1} = E \Rightarrow L_1^{-1} L_2 = E \Rightarrow L_1 = L_2 \Rightarrow U_1 = U_2$. \square

2.4.2 Схема единственного деления и LU -разложение.

Легко показать, что на каждом шаге исключения каждой переменной x_k мы фактически производим умножение матрицы A слева на матрицу L_k вида

$$L_k = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \frac{1}{a_{kk}^{k-1}} & \dots & 0 \\ 0 & \dots & -\frac{a_{k+1k}^{k-1}}{a_{kk}^{k-1}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & -\frac{a_{nk}^{k-1}}{a_{kk}^{k-1}} & \dots & 1 \end{pmatrix} \quad (5)$$

В итоге этих умножений мы приходим к

$$L_n \dots L_1 A x = L_n \dots L_1 b \quad (6).$$

Таким образом, у нас произошло $A = LU$ где

$$L = L_1^{-1} \dots L_n^{-1}, \quad U = L_n \dots L_1 A,$$

причем на главной диагонали матрицы L находятся ведущие элементы матрицы A , т.е. $a_{11}^0, \dots, a_{nn}^{n-1}$.

$$L_k^{-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & a_{kk}^{k-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & a_{nk}^{k-1} & \dots & 1 \end{pmatrix}$$

2.4.3 Метод Гаусса с выбором главного элемента и LU -разложение.

• **Матрицей перестановок** P называется матрица $n \times n$, каждая строка и каждый столбец которой содержит ровно одну единицу и $n - 1$ нулей.

• **Элементарной матрицей перестановок** P_{km} называется матрица, полученная из единичной матрицы перестановкой k -ой и m -ой строк.

Любая элементарная матрица перестановок является матрицей перестановок.

Свойства элементарной матрицы перестановок:

1. Произведение любого числа элементарных матриц перестановок есть матрица перестановок, причем необязательно элементарная.
2. Если рассмотреть матрицу $P_{km}A$, где A — произвольная $n \times n$ матрица, то результат перемножения — матрица A , k -ая и m -ая строки которой меняются местами.
3. Результат произведения AP_{km} — матрица A , k -ый и m -ый столбцы которой меняются местами.

С учетом введенных определений можно получить матричное представление метода Гаусса с выбором главного элемента:

1. по столбцу;

вместо (6) запишем

$$L_n L_{n-1} P_{n-1} \dots L_1 P_1 A x = L_n L_{n-1} P_{n-1} \dots L_1 P_1 b \quad (8).$$

В формуле (8) P_1, \dots, P_{n-1} — элементарная матрица перестановок такая, что $P_k = P_{km}$, $k < m \leq n$, $k = \overline{1, n}$, и L_k $k = \overline{1, n}$ — матрицы вида (5).

Введем следующие обозначения:

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

Причем $L_{k-i}^{(i)}$ — нижняя треугольная матрица, которая отличается от $L_{k-i}^{(i-1)}$ перестановкой двух элементов в $(k-i)$ -ом столбце ниже диагонали.

Формулу (8) перепишем через L :

$$L_n L_{n-1} L_{n-2}^{(1)} L_{n-3}^{(2)} \dots L_1^{(n-2)} P A x = \dots P b, \quad (9)$$

где $P = P_{n-1} \dots P_1$ — результирующая матрица перестановок. Таким образом, получим

$$P A x = P b.$$

Следовательно, решение можно найти по схеме единственного деления, но применить ее к системе матриц PA и Pb .

Поскольку $|A| \neq 0$, то $\exists P$ такое, что $PA = LU$. Тогда

$$|PA| = |LU| = |L| \cdot |U| = l_n \dots l_{nk}.$$

Так как матрицы PA и A отличаются только перестановкой, то справедлива формула (8).

2.5 Метод квадратного корня.

Рассмотрим матрицу $Ax = b$, где $A = A^*$ ($a_{ij} = \bar{a}_{ji}$). В вещественном случае будет просто симметрическая матрица. Предположим, что все главные миноры матрицы A ненулевые, то есть выполнены условия теоремы об LU -разложении. В таком случае \exists невырожденная верхняя треугольная матрица $S = (s_{ij})$ с вещественными положительными элементами на главной диагонали и \exists диагональная матрица D с вещественными над диагональю значениями по модулю равными 1, то есть $D(d_{ii} = \pm 1)$. Тогда

$$A = S^*DS, \quad (1)$$

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

Если матрица A является положительно определенной ($(Ax, x) > 0, \forall x \neq 0$), то $D = E$ и тогда вместо разложения (1) запишем разложение

$$A = S^*S. \quad (2)$$

• Разложение матрицы в виде (1) или (2) называется **разложением Холецкого**.

Для того, чтобы решить систему $Ax = b$, надо найти LU -разложение, а потом используя разложение (1) или (2) разбить систему на две. Приравняем элементы матриц A и S^*DS и получим систему уравнений

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

Решая эту систему, мы получим общие формулы

$$\begin{aligned} d_{ii} &= \text{sign}(a_{ii} - \sum_{k=1}^{i-1} |s_{ki}|^2 d_{kk}), \quad i = \overline{1, n}, \\ s_{ii} &= \sqrt{\left| a_{ii} - \sum_{k=1}^{i-1} |s_{ki}|^2 d_{kk} \right|}, \quad i = \overline{1, n} \\ s_{ij} &= \frac{a_{ij} - \sum_{k=1}^{i-1} \bar{s}_{ki} d_{kk} s_{kj}}{s_{ii} d_{ii}}, \quad i < j, \quad i, j = \overline{1, n}. \end{aligned} \quad (3)$$

Если матрица A вещественная и положительно определенная, то матрица S также вещественная. Следовательно,

$$A = A^T > 0 \Rightarrow S^* = S^T \Rightarrow D = E \Rightarrow A = S^T S.$$

Отсюда

$$\begin{aligned} s_{11} &= \sqrt{a_{11}}, \\ s_{1j} &= \frac{a_{1j}}{s_{11}}, \quad j = \overline{2, n}, \\ s_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} s_{ki}^2}, \quad 1 < i \leq n, \end{aligned} \quad (4)$$

$$s_{ij} = \begin{cases} \frac{a_{ij} - \sum_{k=1}^{i-1} s_{ki}s_{kj}}{s_{ii}}, & i < j \\ 0, & i > j. \end{cases}$$

Таким образом, решение системы сводится к решению двух систем уравнений с треугольными матрицами:

$$\begin{cases} S^* D y = b, \\ S x = y. \end{cases} \quad (5)$$

Если матрица вещественная, то есть $A = A^T > 0$, а значит $D = E$ и $A = S^T S$, то получим две системы

$$\begin{cases} S^T y = b, \\ S x = y \end{cases} \quad (6)$$

Если эти системы — системы линейных уравнений, но с треугольными матрицами, то можно сразу выписывать решения. Причем первая система из (5) и (6) — нижняя треугольная, а вторая — верхняя треугольная.

Выпишем алгоритм решения системы (6) для примера:

1. решаем первое уравнение, предполагая, что элементы вычислены по формулам (4), тогда получим

$$y_1 = \frac{b_1}{s_{11}}, \quad y_i = \frac{b_i - \sum_{k=1}^{i-1} s_{ki}y_k}{s_{ii}}, \quad i > 1; \quad (7)$$

2. аналогично для второго уравнения

$$x_n = \frac{y_n}{s_{nn}}, \quad x_i = \frac{y_i - \sum_{k=i+1}^n s_{ik}x_k}{s_{ii}}, \quad i < n. \quad (8)$$

Замечание.

1. Сложность алгоритма $Q(n) = \frac{1}{3}n^3 + O(n^2)$;
2. Определитель матрицы можно вычислить как

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

2.6 Метод прогонки.

Рассмотрим систему с трехдиагональной матрицей, то есть с матрицей, которая имеет главную диагональ и две параллельные ей, одна из которых выше главной диагонали, а вторая — ниже.

Предположение о ленточной структуре матрицы позволяет вычислять решение системы с количеством операций $Q(n) = O(n)$. Метод прогонки является частным случаем метода Гаусса для систем с трехдиагональной матрицей.

2.6.1 Алгоритм метода прогонки.

В трехдиагональной матрице ненулевые элементы находятся на главной диагонали и на побочных двух, то есть $a_{ij} \neq 0$, $j = i - 1, i, i + 1$, а все остальные элементы нулевые, то есть $a_{ij} = 0$, $i - j > 1$, $j - i > 1$. Введем обозначения

$$a_{ii-1} = -a_i, \quad a_{ii} = c_i, \quad a_{ii+1} = -b_i,$$

а в столбце неоднородности

$$b_i = f_i.$$

Тогда перепишем

$$\begin{cases} c_0 x_0 - b_0 x_1 = f_0, \\ -a_i x_{i-1} + c_i x_i - b_i x_{i+1} = f_i, \quad i = \overline{1, n-1} \\ -a_n x_{n-1} + c_n x_n = f_n. \end{cases} \quad (1)$$

Будем искать решения системы (1) в виде (2):

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}, \quad i = \overline{n-1, 0}, \quad (2)$$

где α_i и β_i — неизвестные пока коэффициенты. Подставим (2) в (1) для $i = \overline{1, n-1}$ и получим

$$-a_i(\alpha_i x_i + \beta_i) + c_i x_i - b_i x_{i+1} = f_i,$$

следовательно,

$$x_i = \frac{b_i}{c_i - \alpha_i a_i} x_{i+1} + \frac{f_i + \beta_i a_i}{c_i - \alpha_i a_i}. \quad (3)$$

Сравнивая формулы (2) и (3), получим

$$\alpha_{i+1} = \frac{b_i}{c_i - \alpha_i a_i}; \quad \beta_{i+1} = \frac{f_i + \beta_i a_i}{c_i - \alpha_i a_i}. \quad (4)$$

Из первого уравнения системы (1) получаем

$$x_0 = \frac{b_0}{c_0} + \frac{f_0}{c_0}.$$

Значит, как следует из формулы (2), при $i = 0$ получаем

$$\alpha_1 = \frac{b_0}{c_0}, \quad \beta_1 = \frac{f_0}{c_0}. \quad (5)$$

Уравнения (4) и (5) дают возможность выписать α_i и β_i для $i = \overline{1, n}$. И можем использовать (2) для нахождения x_i . Сравнивая уравнение (1) с уравнением (2) при $i = n - 1$, получим

$$x_{n-1} = \alpha_n x_n + \beta_n.$$

Сравнивая это и значение из (5), получаем

$$x_n = \frac{f_n + \beta_n a_n}{c_n - \alpha_n a_n}. \quad (6)$$

Объединяя все формулы, окончательно получим алгоритм решения системы (1):

$$\alpha_{i+1} = \frac{b_i}{c_i - \alpha_i a_i}, \quad i = 1, 2, \dots, n-1, \quad \alpha_1 = \frac{b_0}{c_0} \quad (7)$$

$$\beta_{i+1} = \frac{f_i + \beta_i a_i}{c_i - \alpha_i a_i}, \quad i = 1, 2, \dots, n, \quad \beta_1 = \frac{f_0}{c_0} \quad (8)$$

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}, \quad i = n-1, \dots, 0, \quad x_n = \beta_{n+1} \quad (9)$$

- Формулы (7) – (9) принято называть **методом правой прогонки**. Формулы (7), (8) – **прямая прогонка**, (9) – **обратная прогонка**.

Этих формул достаточно для того, чтобы решить систему (1).

В заключении отметим количество операций

$$Q(n) \approx 8n.$$

Аналогично формулам правой прогонки можно построить метод левой прогонки, а также метод встречной прогонки. Комбинация левой и правой прогонок дают нам формулу метода встречной прогонки.

2.6.2 Метод прогонки и метод Гаусса.

Покажем, что метод прогонки является частным случаем схемы единственного деления. Для этого нужно показать, что эти формулы обеспечивают прямой ход и обратный ход схемы единственного деления.

Первое уравнение системы (1) с учетом формулы (5) может быть записано как

$$x_0 = \alpha_1 x_1 + \beta_1.$$

Если x_0 выражено из первого уравнения, то мы должны исключить x_0 из всех уравнений, находящихся ниже первого уравнения. Легко видеть, что x_0 встречается только в одном уравнении. Поэтому исключение нужно провести из одного уравнения:

$$-a_1 x_0 + c_1 x_1 - b_1 x_2 = f_1.$$

В итоге мы получим формулу

$$x_1 = \alpha_2 x_2 + \beta_2.$$

И так далее

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}.$$

Таким образом, после осуществления метода прямой прогонки мы приведем нашу систему к верхнему треугольному виду. В итоге у нас получится матрица вида

$$Ux = \begin{pmatrix} 1 & -\alpha_1 & 0 & \dots & 0 \\ 0 & 1 & -\alpha_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 - \alpha_n \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \\ \beta_{n+1} \end{pmatrix} \quad (10)$$

Таким образом, мы осуществили прямой ход метода Гаусса (сравнить с системой (4) из параграфа 3). Обратный ход осуществляется по формулам (9) (сравните с формулой (7) из параграфа 3). Так мы установили связь между алгоритмом методов прогонки и схемой единственного деления.

2.6.3 Обоснование метода прогонки.

Для обоснования метода прогонки мы должны убедиться в том, что этот метод корректен. Доказывать корректность мы будем из определения корректности.

Теорема (о корректности метода прогонки). *Пусть коэффициенты системы (1) удовлетворяют условиям*

$$|c_0| > 0, |c_n| > 0, |a_i| > 0, |b_i| > 0, i = \overline{1, n-1},$$

$$|c_i| \geq |a_i| + |b_i|, i = \overline{1, n-1}, \quad (11)$$

$$|c_0| \geq |b_0|, |c_n| \geq |a_n|, \quad (12)$$

причем хотя бы в одном из условий (11), (12) (условия диагонального преобладания) выполняется строго неравенство. Тогда для алгоритма (7) – (9) метода правой прогонки имеют место неравенства

$$c_i - \alpha_i a_i \neq 0, |\alpha_i| \leq 1, i = \overline{1, n},$$

обеспечивающие корректность метода.

◆ Доказательство проведем по индукции. Согласно формулам (5) и (12)

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

Покажем, что из неравенства

$$|\alpha_i| \leq 1, \forall i \leq n-1$$

и выполнения условий теоремы будет следовать, что

$$c_i - \alpha_i a_i \neq 0, |\alpha_{i+1}| \leq 1, \forall i \leq n-1.$$

Рассмотрим

$$|c_i - \alpha_i a_i| \geq |c_i| - |\alpha_i| \cdot |a_i| \geq [(11)] \geq |b_i| + |a_i| \cdot (1 - |\alpha_i|) \geq |b_i| > 0.$$

Теперь

$$|\alpha_{i+1}| = \frac{|b_i|}{|c_i - \alpha_i a_i|} \leq \frac{|b_i|}{|b_i|} \leq 1, \forall i \leq n-1.$$

Осталось показать, что

$$c_n - \alpha_n a_n \neq 0, i = \overline{1, n}.$$

Для этого используем предположение, что хотя бы для одного значения i хотя бы в одном из неравенств (11), (12) у нас выполняется строгое неравенство. Возможны следующие случаи:

1. пусть $|c_n| > |a_n|$. Тогда рассматриваем

$$|c_n - \alpha_n a_n| \geq |c_n| - |\alpha_n| \cdot |a_n| \geq |c_n| - |a_n| > 0.$$

2. пусть при $i = i_0, 1 \leq i_0 \leq n-1$ неравенство (11) становится строгим. Тогда рассмотрим

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

Тогда

$$|\alpha_{i_0+1}| < 1 \Rightarrow \forall i \geq i_0 + 1 \quad |a_i| < 1 \Rightarrow |c_n - \alpha_n a_n| \geq |c_n| - |\alpha_n| \cdot |a_n| > 0.$$

3. пусть $|c_0| > |b_0|$. Следовательно,

$$|\alpha_1| < 1 \Rightarrow |\alpha_i| < 1, \quad i = \overline{2, n} \Rightarrow \text{приходим к случаю 2.}$$

Таким образом, при выполнении условий (11), (12) задача (1) разрешима, то есть метод применим.

Докажем устойчивость расчета по рекуррентным формулам (9). Пусть в формуле (9) при $i = i_0 + 1$ вместо x_{i_0+1} мы получили

$$\tilde{x}_{i_0+1} = x_{i_0+1} + \delta_{i_0+1}.$$

По формулам (9) вместо

$$x_{i_0} = \alpha_{i_0+1}x_{i_0+1} + \beta_{i_0+1}$$

мы получим

$$\tilde{x}_{i_0} = \alpha_{i_0+1}(x_{i_0+1} + \delta_{i_0+1}) + \beta_{i_0+1}.$$

Посчитаем погрешность, которая допущена:

$$\delta_{i_0} = \tilde{x}_{i_0} - x_{i_0} = \alpha_{i_0+1}\delta_{i_0+1}.$$

Отсюда имеем оценку

$$|\delta_{i_0}| \leq |\alpha_{i_0+1}\delta_{i_0+1}| = |\alpha_{i_0+1}| \cdot |\delta_{i_0+1}| \leq |\delta_{i_0+1}|.$$

Из последнего неравенства следует, что погрешность при переходе к следующему шагу не возрастает. Таким образом, мы доказали теорему. \square

Замечание. Почему же проверка условия

$$\Delta_i = c_i - \alpha_i a_i \neq 0, \quad i = \overline{1, n}$$

достаточна для того, чтобы гарантировать отличие от нуля определителя системы (1). Оказывается, что можно показать, что при применении метода прогонки, имеет место LU-разложение. Очевидно, что LU-разложение обеспечивается с помощью U матрицы из формулы (10) и матрицы

$$L = \begin{pmatrix} c_0 & 0 & \dots & 0 \\ -a_1 & \Delta_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -a_n & \Delta_n \end{pmatrix}.$$

Таким образом, для матрицы $A = LU$

$$|A| = |LU| = |L| \cdot |U| = |L| \neq 0.$$

Значит в системе (1) матрица является невырожденной.

Замечание. Теорема дает достаточное условие корректности метода. Поэтому даже при нарушении условий (11), (12) метод может быть устойчивым. Кроме того некоторые из коэффициентов a_i , b_i могут обращаться в 0, то есть можно ослабить условия.

2.7 Методы основанные на ортогональных преобразованиях.

Вернемся к задаче

$$Ax = b, \quad (1)$$

где A — матрица $n \times n$. Ранее мы доказали, что

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

Теперь посмотрим, как изменилось число обусловленности $\nu(U)$ по сравнению с $\nu(A)$. По свойствам числа обусловленности

$$\nu(U) \leq \nu(L_n) \nu(L_{n-1}) \nu(P_{n-1}) \dots \nu(L_1) \nu(P_1) \nu(A).$$

Для сферической нормы легко показать, что

$$\nu(P_k) = 1.$$

Значит на число обусловленности влияет $\nu(L_k)$:

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

Таким образом, при существенном росте либо уменьшении элементов матрицы L_k обратный ход метода Гаусса может сопровождаться большой потерей точности. Поэтому были предложены методы основанные на ортогональных преобразованиях, которые не влияют на изменение числа обусловленности. Теория этих методов сводится к тому, что мы будем рассматривать не LU -разложение матрицы, а QR -разложение.

2.7.1 QR-разложение.

Далее будем рассматривать вещественные матрицы, хотя всё сказанное переносится и на комплексный случай.

• Матрица Q называется **ортогональной**, если для нее выполняется

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

Из определения ортогональных матриц следуют свойства:

1. $Q^T = Q^{-1}$;
2. произведение любого конечного числа ортогональных матриц есть матрица ортогональная, то есть

$$\tilde{Q} = \prod_{i=1}^n Q_i$$

3. $\nu_{III}(Q) = 1$;
4. $\nu_{III}(QA) = \nu_{III}(A)$.

• **QR-разложением матрицы A** будем называть ее представление в виде произведения

$$A = QR,$$

где Q — ортогональная, а R — верхняя треугольная.

Теорема (о существовании QR-разложения). *Всякую невырожденную матрицу A можно представить в виде QR-разложения.*

♦ Рассмотрим матрицу $A^T A$. Известно, что эта матрица является симметрической и положительно определенной, т.е. $A^T A > 0$. Следовательно, для такой матрицы мы можем записать разложение Холецкого, которое в данном случае будет иметь вид:

$$A^T A = R^T R,$$

R — верхняя треугольная матрица. Сделаем некоторые преобразования:

$$A = (AR^{-1})R,$$

докажем, что можно считать $AR^{-1} = Q$, то есть, что AR^{-1} — ортогональная матрица. Для того, чтобы доказать, что она является ортогональной, надо показать, что

$$(AR^{-1})(AR^{-1})^T = E.$$

Сделаем следующие преобразования

$$(AR^{-1})(AR^{-1})^T = AR^{-1}(R^{-1})^T A^T = A(R^T R)^{-1} A^T = A(A^T A)^{-1} A^T = AA^{-1}(A^T)^{-1} A^T = E.$$

□

Рассмотрим методы, основанные на ортогональных преобразованиях.

2.7.2 Метод отражений.

• **Матрицей отражения (Хаусхолдера)** называют матрицу вида

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

где ω — вектор-столбец единичной длины в сферической норме, то есть $(\omega, \omega) = \omega^T \omega = 1$.

Свойства матрицы отражений:

1. V — симметричная матрица;
2. V — ортогональная матрица;
3. матрица V оставляет без изменений любой вектор ортогональный вектору ω , то есть если $\forall x (x, \omega) = \omega^T x = 0$, то $Vx = x$;
4. матрица V переводит в противоположный любой вектор коллинеарный вектору ω , то есть если $x = \lambda\omega$, $\lambda \in \mathbb{R}$, то $Vx = -x$.

Можно показать, что \forall вектора $s \neq 0$ и заданного вектора e единичной длины имеет место равенство

$$Vs = \alpha e, \quad (3)$$

где V — матрица отражений (2), в которой

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

Алгоритм метода отражения рассмотрим поэтапно, а затем запишем общую формулу.

Первый этап. Используя формулу (4), образуем матрицу отражения V^1 по векторам

$$s^1 = (a_{11}, a_{21}, \dots, a_{n1})^T, \quad e^1 = (1, 0, \dots, 0)^T.$$

Умножим слева исходную систему (1) на матрицу V^1 . После умножения у нас получится система

$$A^1 x = b^1, \quad A^1 = V^1 A, \quad b^1 = V^1 b.$$

Легко видеть, что в матрице A^1 все элементы, стоящие ниже элемента a_{11} , равны 0. Запишем формулы, отражающие, что произойдет с элементами матрицы A :

$$\begin{cases} a_{11}^1 = \alpha_1, \\ a_{ij}^1 = a_{ij} - 2(g_j, \omega^1) \omega_i^1, \quad i = \overline{1, n}, \quad j = \overline{2, n}, \\ b_i^1 = b_i - 2(b, \omega^1) \omega_i^1, \quad i = \overline{1, n}, \end{cases} \quad (5)$$

где

$$g_j = (a_{1j}, a_{2j}, \dots, a_{nj})^T.$$

Скалярное произведение определим как

$$(g_j, \omega^1) = \sum_{p=1}^n g_{pj} \omega_p^1.$$

Второй этап. Строим матрицу V^2 по векторам

$$s^1 = (0, a_{22}^1, \dots, a_{n2}^1)^T, \quad e^2 = (0, 1, \dots, 0)^T.$$

Тогда мы образуем систему

$$A^2 x = b^2, \quad A^2 = V^2 A^1, \quad b^2 = V^2 b^1.$$

Легко увидеть, что в A^2 первая строка будет совпадать с A^1 , а остальные элементы будут изменены по предыдущему алгоритму.

Алгоритм. На любом k -ом шаге, $1 \leq k \leq n - 1$ мы строим векторы

$$s^k = (0, \dots, 0, a_{kk}^k - 1, \dots, a_{nk}^{k-1})^T, \quad e^k = (0, \dots, 0, 1, 0, \dots, 0)^T$$

образуем матрицу V^k и преобразуем систему к виду

$$A^k x = b^k. \quad (6)$$

Запишем, как меняются коэффициенты от шага в шагу:

$$\begin{cases} a_{ij}^k = a_{ij}^{k-1}, & i = \overline{1, k-1}, j = \overline{1, n}, \\ a_{kk}^k = \alpha_k, \\ a_{ij}^k = a_{ij}^{k-1} - 2(g_j^{k-1}, \omega^k) \omega_i^k, & i = \overline{k, n}, j = \overline{k+1, n}, \\ b_i^k = b_i^{k-1}, & i = \overline{1, k-1}, \\ b_i^k = b_i^{k-1} - 2(b^{k-1}, \omega^k) \omega_i^k, & i = \overline{k, n}. \end{cases} \quad (7)$$

В формуле (7)

$$\begin{aligned} g_j^{k-1} &= (a_{1j}^{k-1}, \dots, a_{nj}^{k-1}), \\ (g_j^{k-1}, \omega^k) &= \sum_{p=k}^n g_{pj}^{k-1} \omega_p^k, \\ (b^{k-1}, \omega^k) &= \sum_{p=k}^n b_p^{k-1} \omega_p^k. \end{aligned}$$

После выполнения всех $n - 1$ этапа мы получим матрицу

$$A^{n-1}x = b^{n-1},$$

причем матрица A^{n-1} является верхней треугольной. Далее после приведения к такому виду, чтобы записать решение, нам нужно записать формулы аналогичные обратному ходу метода Гаусса.

Число операций будет равно $Q(n) = O(n^3)$. В отличие от метода Гаусса количество умножений здесь будет в 2 раза больше. Однако, в отличие от метода Гаусса, число обусловленности мы не изменяем.

2.7.3 Метод вращений.

• *Матрицей вращения (Гивенса) называется матрица вида*

$$T_{ij} = \begin{matrix} & & i & & j & & \\ & & & & & & \\ & & & & & & \\ i & \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \cos \varphi & \dots & -\sin \varphi & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ j & \begin{pmatrix} 0 & \dots & \sin \varphi & \dots & \cos \varphi & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \end{pmatrix} \end{matrix}. \quad (8)$$

Если мы умножим матрицу A на матрицу T_{ij} слева, то мы получим матрицу $B = T_{ij}A$, у которой элементы i -ой и j -ой строк будут изменены

$$\begin{cases} b_{ik} = a_{ik} \cos \varphi - a_{jk} \sin \varphi, \\ b_{jk} = a_{ik} \sin \varphi + a_{jk} \cos \varphi, \end{cases} \quad k = \overline{1, n}, \quad (9)$$

а остальные элементы не изменятся.

Запишем формулы, обеспечивающие выбор φ . Предположим, что нам нужно исключить

x_s неизвестную из j -ого уравнения, то есть коэффициент при этом значении должен быть равен нулю. Для этого нам необходимо в формуле (9) положить $k = s$ и приравнять этот элемент к нулю. То есть

$$a_{is} \sin \varphi + a_{js} \cos \varphi = 0.$$

Получили уравнение с одной неизвестной. Очевидно

$$\operatorname{tg} \varphi = -\frac{a_{js}}{a_{is}}, \quad a_{is} \neq 0.$$

Отсюда

$$\begin{cases} \cos \varphi = \frac{a_{is}}{\sqrt{a_{is}^2 + a_{js}^2}}, \\ \sin \varphi = -\frac{a_{js}}{\sqrt{a_{is}^2 + a_{js}^2}}. \end{cases} \quad (10)$$

Заметим, что в формуле (10) a_{is} может быть равным нулю. Но если $a_{js} = 0$, то $T_{ij} = E$.

Запишем алгоритм метода вращения.

Первый этап. Последовательно будем умножать исходную систему на матрицы

$$T_{12}, T_{13}, \dots, T_{1n}.$$

При этом первая строка получится такая, какая получится, а все остальные строки будут без переменной x_1 . Итак у нас будет количество умножений $n - 1$ и для каждого k -ого умножения запишем формулу

$$\cos \varphi_{1,k} = \frac{a_{11}^{1,k-1}}{\sqrt{(a_{11}^{1,k-1})^2 + (a_{k1}^{1,k-1})^2}}, \quad \sin \varphi_{1,k} = -\frac{a_{k1}^{1,k-1}}{\sqrt{(a_{11}^{1,k-1})^2 + (a_{k1}^{1,k-1})^2}}, \quad k = \overline{2, n} \quad (11)$$

Для пересчета ненулевых элементов запишем формулу

$$\begin{cases} a_{1j}^{1,k} = a_{1j}^{1,k-1} \cos \varphi_{1,k} - a_{kj}^{1,k-1} \sin \varphi_{1,k}, & j = \overline{1, n}, \quad k = \overline{2, n}, \\ a_{kj}^{1,k} = a_{1j}^{1,k-1} \sin \varphi_{1,k} + a_{kj}^{1,k-1} \cos \varphi_{1,k}, & j = \overline{1, n}, \quad k = \overline{2, n}, \\ a_{ij}^{1,k} = a_{ij}^{1,k-1}, & i \neq 1, \quad i \neq k. \end{cases} \quad (12)$$

$$\begin{cases} b_1^{1,k} = b_1^{1,k-1} \cos \varphi_{1,k} - b_k^{1,k-1} \sin \varphi_{1,k}, \\ b_k^{1,k} = b_1^{1,k-1} \sin \varphi_{1,k} + b_k^{1,k-1} \cos \varphi_{1,k}, \\ b_p^{1,k} = b_p^{1,k-1}, & p \neq 1, p \neq k \end{cases} \quad (13)$$

Данная формула соответствует первому этапу. По окончании первого этапа мы получаем систему

$$A^1 x = b^1, \quad A^1 = T_{1n} \dots T_{13} T_{12} A = T_1 A, \quad b^1 = T_1 b.$$

Все элементы ниже главной диагонали в полученной матрице будут равны нулю.

Второй этап. Далее во втором этапе нам нужно домножить на матрицу

$$T_2 = T_{2n} \dots T_{23}.$$

И так далее.

n-1 этап. Получим систему

$$A^{n-1} x = b^{n-1}, \quad A^{n-1} = T_{n-1} T_{n-2} \dots T_1 A, \quad b^{n-1} = T_{n-1} T_{n-2} \dots T_1 b.$$

матрица которой является верхней треугольной. Для того, чтобы завершить этот метод необходимо осуществить обратный ход метода Гаусса.

Метод вращения является более трудоемким, чем метод отражения. Количество операций этого метода $Q(n) = O(n^3)$, но количество умножений будет в 2 раза больше, чем у метода вращений.

2.8 Метод простой итерации.

Рассмотрим задачу

$$Ax = b \quad (1)$$

и решение системы (1) мы будем искать как предел бесконечного вычислительного процесса. Когда строится последовательность $x^0, x^1, \dots, x^k, \dots$ — последовательность приближения к решению x^* , мы должны обеспечить выполнение условия

$$\lim_{k \rightarrow \infty} \|x^k - x^*\| = 0.$$

В наших обозначениях $0, 1, \dots, k, \dots$ — номер итерации. И каждый элемент является k -ой итерацией, или k -ым приближением к искомому значению x^* .

2.8.1 Классификация итерационных процессов. Основные определения.

Чтобы построить последовательность x^k исходную систему приведем к эквивалентному виду: будем рассматривать систему

$$x = Bx + g \quad (2)$$

где B — матрица размерности $n \times n$, g — это вектор. То, что системы (1) и (2) эквивалентны, значит, что найдя решение системы (2), мы решим систему (1). По системе (2) легко построить итерационную последовательность: мы имеем рекуррентное соотношение

$$x^{k+1} = Bx^k + g, \quad k = 0, 1, \dots, \quad (3)$$

x^0 — заданное начальное приближение и, вообще говоря, его можно задать произвольно.

• Формулу (3), мы будем называть **методом простой итерации**.

Вычислим B и g . Если предположить, что матрицу A мы можем представить в виде двух матриц $C + D = A$, причем для определенности $|C| \neq 0$, тогда очевидно

$$B = -C^{-1}D, \quad g = C^{-1}b.$$

Можно предложить другой часто используемый способ. Исходную систему записывают в виде

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

По-прежнему мы предполагаем, что $|C| \neq 0$. Тогда

$$B = E - CA, \quad g = CB$$

и мы получим вид (2).

Если матрицу C в любом из вариантов выбирать на каждой итерации, то мы придем к формуле

$$x^{k+1} = B_k x^k + g^k, \quad k = 0, 1, \dots$$

$$B_k = E - C_k A, \quad g = C_k b. \quad (4)$$

• Итерационные процессы, когда матрица B не зависит от номера итерации называются **стационарными**. А итерационные процессы вида (4), где B зависит от номера итерации называются **нестационарными**.

Метод простой итерации (3) является стационарным.

При построении итерационных процессов систему (1) можно привести к виду

$$Px + Qx = g.$$

Для того, чтобы преобразование было эквивалентным,

$$P + Q = CA, \quad g = Cb, \quad |C| \neq 0,$$

где матрицу C мы можем сами выбирать. Теперь мы можем построить итерационный процесс по аналогии с (3) и (4):

$$Px^{k+1} + Qx^k = g, \quad k = 0, 1, \dots, \quad (5)$$

— это стационарный процесс. Можем также построить нестационарный процесс:

$$P_k x^{k+1} + Q_k x^k = g^k, \quad k = 0, 1, \dots \quad (6)$$

• Процессы (5) и (6) называются **неявными**, так как для определения x^{k+1} нам нужно обращать матрицу P (в случае (5)) или обращать на каждом шаге матрицы P_k (в случае (6)). Соответственно способы (3) и (4) называются **явными**.

Причем (5) — стационарный неявный, а (6) — нестационарный неявный процессы.

Запишем еще одну формулу, позволяющую построить итерационный процесс:

$$B_k \frac{x^{k+1} - x^k}{\tau_{k+1}} + Ax^k = b, \quad k = 0, 1, \dots \quad (7)$$

где $|B_k| \neq 0$, а $\tau_{k+1} \in \mathbb{R}$ — итерационные параметры.

• Формула (7) называется **канонической формой** одношагового итерационного процесса.

Формула (7) — это фактически частный случай формулы (6).

Таким образом, все указанные методы (3)-(7) являются одношаговыми и линейными.

Тогда метод простой итерации (3) является явным линейным стационарным одношаговым процессом.

Можно также строить и многошаговые, и нелинейные методы.

Замечание. Если последовательность x^k сходится, то она сходится к решению системы (2). Действительно, если $x^k \rightarrow x^*$, $k \rightarrow \infty$, то переходя к пределу в формуле (3), получим

$$x^* = Bx^* + g,$$

то есть x^* — решение системы (2). Значит при условии, что мы построили итерационный процесс сходящийся, мы тем самым найдем итерационный процесс, который будет являться решением задачи (1).

Займемся выяснением условий сходимости последовательности x^k .

2.8.2 Сходимость матричной геометрической прогрессии.

Выясним условия, при которых будет сходиться матричная геометрическая прогрессия вида

$$E + A + A^2 + \dots + A^m + \dots \quad (8)$$

В формуле (8) следует иметь ввиду, что в данном случае показатель — это уже степень перемножения матриц. Оказывается, что от поведения этой матричной геометрической прогрессии зависит сходимость x^k . Приведем три леммы без доказательства.

Лемма. Для того, чтобы $A^m \rightarrow 0$, $m \rightarrow \infty$, необходимо и достаточно, чтобы все собственные значения матрицы A были по модулю меньше 1, то есть $|\lambda_i(A)| < 1$.

Лемма. Для того, чтобы $A^m \rightarrow 0$, $m \rightarrow \infty$, достаточно, чтобы хотя одна из ее норм была меньше 1, то есть $\|A\| < 1$.

Лемма. Модуль каждого собственного значения матрицы не превосходит любой из ее норм, то есть $|\lambda_i(A)| \leq \|A\|$.

Имея эти утверждения, докажем теорему о сходимости матричной геометрической прогрессии.

Теорема. Для того, чтобы ряд (8) сходиллся необходимо и достаточно, чтобы $A^m \rightarrow 0$, $m \rightarrow \infty$. При этом матрица $E - A$ обратима, то есть $|E - A| \neq 0$, и сумма ряда (8) равна

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

♦ \Rightarrow) Необходимость следует из того, что по определению сходимость матричной последовательности эквивалентна сходимости n^2 соответствующих числовых рядов из элементов этих матриц. Тогда условие стремления к нулю m -ого члена есть необходимое условие сходимости любого числового ряда.

\Leftarrow) Если по условию $A^m \rightarrow 0$, то по лемме 1 $|\lambda_i(A)| < 1$. Тогда

$$\lambda_i(E - A) = 1 - \lambda_i(A) \neq 0.$$

Следовательно, если собственные значения матрицы $E - A$, то

$$|E - A| = \prod_{i=1}^n \lambda_i(E - A) \neq 0.$$

Тогда $\exists (E - A)^{-1}$.

Для доказательства сходимости (8) рассмотрим тождество

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

Теперь это тождество мы умножим справа на $(E - A)^{-1}$, тогда

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

Так как по условию $A^m \rightarrow 0$, то $A^{m+1} \rightarrow 0$, то есть, переходя к пределу при $m \rightarrow \infty$, мы получим

$$E + A + \dots + A^m \rightarrow (E - A)^{-1}, \quad m \rightarrow \infty.$$

□

С учетом леммы 1 теорему 1 можно сформулировать в следующем виде

Теорема. Для сходимости ряда (8) необходимо и достаточно, чтобы все собственные значения матрицы A были по модулю меньше единицы, то есть $|\lambda_i(A)| < 1$.

С учетом леммы 2 можно сформулировать достаточный признак сходимости в виде теоремы.

Теорема. Если какая-либо норма матрицы A меньше единицы, то есть, $\|A\| < 1$, то ряд (8) сходится.

2.8.3 Условия сходимости метода простой итерации.

Пользуясь результатами пункта (8.2), можно доказать теорему.

Теорема. Для сходимости метода простой итерации (3) при любом начальном приближении x^0 необходимо и достаточно, чтобы все собственные значения матрицы B были по модулю меньше единицы, то есть $|\lambda_i(B)| < 1$.

◆ \Rightarrow) Для этого преобразуем формулу метода

$$x^{k+1} = Bx^k + g = B(Bx^{k-1} + g) + g = B^2x^{k-1} + (E + B)g = \dots = B^{k+1}x^0 + (E + B + B^2 + \dots + B^k)g.$$

Так как по условию все собственные значения B меньше единицы, то в силу леммы 1

$$B^{k+1} \rightarrow 0,$$

а по теореме 1 ряд $E + B + B^2 + \dots + B^k$ сходится и $E + B + B^2 + \dots + B^k \rightarrow (E - B)^{-1}$. Следовательно,

$$x^{k+1} \rightarrow (E - B)^{-1}g = x^*.$$

То есть итерационная последовательность сходится при любом начальном приближении к конечному решению.

\Leftarrow) По условию $x^k \rightarrow x^*, k \rightarrow \infty, \forall x^0$. Рассмотрим

$$x^k - x^{k+1} = B(x^* - x^k) = B^2(x^* - x^{k-1}) = \dots = B^{k+1}(x^* - x^0).$$

В полученном равенстве перейдем к пределу при $k \rightarrow \infty$:

$$x^k - x^* = B^{k+1}(x^* - x^0),$$

значит $B^{k+1} \rightarrow 0$, следовательно, по лемме 1 все собственные значения матрицы B будут по модулю меньше единицы, то есть $|\lambda_i(B)| < 1$. □

Теорема. Для того, чтобы метод простой итерации сходиллся достаточно, чтобы какая-либо норма матрицы B была меньше единицы, то есть $\|B\| < 1$.

◆ Если норма матрицы B меньше единицы, то все собственные значения этой матрицы будут меньше единицы по модулю по лемме 3. Тогда по теореме 4, исходя из критерия, у нас следует сходимость метода (3) для любого начального приближения. \square

Теорема. Если какая-либо норма матрицы B , согласованная с нормой вектора x , меньше единицы, то есть $\|B\| < 1$, то для погрешности метода простой итерации справедлива оценка

$$\|x^* - x^k\| \leq \|B\|^k \|x^0\| + \frac{\|B\|^k}{1 - \|B\|} \|g\|. \quad (9)$$

◆ По теореме 4

$$x^k = Bx^{k-1} + q = \dots = B^k x^0 + (E + B + \dots + B^{k-1})q.$$

Так как $\|B\| < 1$, то по построению

$$x^* = (E - B)^{-1}q = (E + B + \dots + B^k + \dots)q.$$

$$x^* - x^k = (B^k + B^{k+1} + \dots)q - B^k x^0.$$

Перейдем к нормам в последнем равенстве, пользуясь свойствами норм,

$$\|x^* - x^k\| = \|(B^k + B^{k+1} + \dots)g - B^k x^0\| \leq (9).$$

☒

Замечания.

1. Формула (9) упростится, если мы в качестве начального приближения выберем свободный вектор $x^0 = q$, тогда

$$\|x^* - x^k\| \leq \frac{\|B\|^{k+1}}{1 - \|B\|} \|g\|. \quad (10)$$

2. Из формулы (10) можно найти номер k нужной итерации, которая обеспечит требуемую точность ε заданную заранее. Можно получить оценку количества итераций

$$k \geq \frac{\lg \varepsilon + \lg(1 - \|B\|) - \lg \|g\|}{\lg \|B\|}.$$

3. Метод простой итерации сходится со скоростью геометрической прогрессии со знаменателем равным норме матрицы B , то есть $\|B\| < 1$. Это следует из оценки

$$\|x^* - x^k\| \leq \|B\|^k \|x^* - x^0\|.$$

2.8.4 Метод Якоби как частный случай метода простой итерации.

Проблема, которая была в методе простой итерации — привести систему к эквивалентному каноническому виду (2). Сейчас рассмотрим частный случай метода простой итерации. Он отличается специальным способом приведения системы (1) к каноническому виду (2). Из каждого i -го уравнения системы (1) выражаем x_i . Тогда система примет следующий вид:

[illegible]

Получим систему с матрицей и векторов

$$B = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \dots & -\frac{a_{1n-1}}{a_{11}} & -\frac{a_{1n}}{a_{11}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \dots & -\frac{a_{nn-1}}{a_{nn}} & 0 \end{pmatrix}, g = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}. \quad (11)$$

Таким образом, **методом Якоби** называется метод простой итерации (3) с матрицей B и вектором g (11).

Сформулируем достаточное условие сходимости метода Якоби.

Теорема. *Метод Якоби для системы (1) сходится, если элементы a_{ij} матрицы A удовлетворяют одному из условий*

1.

$$\sum_{j=1, j \neq i}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1, \quad i = 1, \dots, n;$$

2.

$$\sum_{i=1, i \neq j}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1, \quad j = 1, \dots, n;$$

3.

$$\sum_{i,j=1, i \neq j}^n \left(\frac{a_{ij}}{a_{ii}} \right)^2 < 1.$$

◆ Доказывается, используя результат предыдущего пункта. □

Условия теоремы означают, что для сходимости метода Якоби достаточно, чтобы матрица A была строго диагонально доминирующей.

2.9 Метод Зейделя.

Метод простой итерации (3) из предыдущего параграфа в координатной форме можно записать как

$$x_i^{k+1} = \sum_{j=1}^n b_{ij} x_j^k + g_i, \quad i = 1, \dots, n, \quad k = 0, 1, \dots; \quad x_0$$

где i — координата искомого вектора. (x_0 означает, что x_0 задано) Укажем две возможности:

1. Выбор порядка вычисления координат (нестационарный метод простой итерации). Вводится функция ошибки $\delta_i^{k+1} = x_i^{k+1} - x_i^k$.
2. Использование вычисленных координат вектора x^{k+1} для вычисления других координат этого вектора. То есть улучшенное значение координат используется при уточнении остальных координат (стационарный метод Зейделя).

Опишем второй метод и рассмотрим условия его сходимости.

2.9.1 Описание и сходимость метода Зейделя.

Фиксируем порядок вычисления компонент вектора x^{k+1} в порядке возрастания номеров от 1 до n и при вычислении каждой последующей координаты будем использовать уточненные ранее предыдущей координаты. Запишем это в виде формулы

[illegible]

Запишем это в более компактном виде

$$x_i^{k+1} = \sum_{j=1}^{i-1} b_{ij} x_j^{k+1} + \sum_{j=i}^n b_{ij} x_j^k + g_i, \quad i = \overline{1, n}, k = 0, 1, \dots; \quad x_0 \quad (2)$$

Итерационный метод (1) или (2) называется **методом Зейделя**. Для исследования его сходимости запишем формулу (2) в матричном виде. Введем в рассмотрение матрицы

$$H = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ b_{21} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn-1} & 0 \end{pmatrix}, \quad F = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n-1} & b_{1n} \\ 0 & b_{22} & \dots & b_{2n-1} & b_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & b_{nn} \end{pmatrix}$$

Очевидно $B = H + F$. Тогда вместо формулы (2) мы можем записать метод Зейделя в матричном виде

$$x^{k+1} = Hx^{k+1} + Fx^k + q, \quad k = 0, 1, \dots$$

Данный метод можно легко представить как метод простой итерации. Следовательно, можем переписать эту формулу в виде

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

Таким образом, метод Зейделя — стационарный неявный одношаговый метод (см. формулу (5) из п.8) с матрицей

$$P = E - H, \quad |P| = 1.$$

Матрица P обратима, тогда мы можем записать метод Зайделя как метод простой итерации с матрицей специального вида

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

Таким образом, метод Зейделя эквивалентен методу простой итерации, примененному к системе

$$x = (E - H)^{-1}Fx + (E - H)^{-1}g$$

Значит все результаты, касающиеся сходимости, переносятся на этот метод.

Теорема. Для того, чтобы метод Зейделя сходился, необходимо и достаточно, чтобы все собственные значения матрицы $(E - H)^{-1}F$ были по модулю меньше единицы, то есть $|\lambda((E - H)^{-1}F)| < 1$.

Поскольку вычисление всех собственных значений связано с определителем вида

$$|(E - H)^{-1}F - \lambda E| = 0,$$

то мы это уравнение упростим.

$$\left| (E - H)^{-1}F - \lambda E \right| = \left| (E - H)(E - H)^{-1}((E - H)^{-1}F - \lambda E) \right| = \dots = |F + \lambda H - \lambda E| = 0.$$

Переформулируем теорему следующим образом

Теорема. Для сходимости метода Зейделя при любом начальном приближении x_0 необходимо и достаточно, чтобы все корни уравнения

$$|F + \lambda H - \lambda E| = 0$$

были по модулю меньше единицы.

Сформулируем достаточный признак сходимости метода Зейделя, причем сформулируем его непосредственно для элементов матрицы B .

Теорема. Для того, чтобы метод Зейделя сходился, достаточно выполнения одного из условий

1.

$$\max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}| < 1;$$

2.

$$\max_{1 \leq j \leq n} \sum_{i=1}^n |b_{ij}| < 1.$$

◆ Проведем доказательство условия 1. Рассмотрим

$$\sum_{j=1}^n |b_{ij}| < 1, \quad i = 1, \dots, n.$$

Воспользуемся теоремой 2 и покажем, что при выполнении этого условия все корни данного уравнения будут по модулю меньше единицы.

От противного. Пусть $\exists \lambda : |\lambda| \geq 1$. Рассмотрим матрицу

$$F + \lambda H - \lambda E = \begin{pmatrix} b_{11} - \lambda & b_{12} & \dots & b_{1n} \\ \lambda b_{21} & b_{22} - \lambda & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda b_{n1} & \lambda b_{n2} & \dots & b_{nn} - \lambda \end{pmatrix}$$

Докажем, что определитель этой матрицы отличен от нуля в предположении, что $|\lambda| > 1$. Для любого i от 1 до n рассмотрим сумму модулей недиагональных элементов любой строки. В i -ой строке

$$|\lambda| \cdot |b_{i1}| + \dots + |\lambda| \cdot |b_{ii-1}| + |b_{ii+1}| + \dots + |b_{in}| \leq |\lambda| \left(\sum_{j=1}^n |b_{ij}| - |b_{ii}| \right) < |\lambda|(1 - |b_{ii}|) \leq |\lambda - b_{ii}|.$$

Таким образом, при сделанном предположении матрица имеет строго преобладание диагональных элементов по строкам. Следовательно, определитель этой матрицы

$$|F + \lambda H - \lambda E| \neq 0.$$

Но тогда λ не является корнем характеристического уравнения. \(\square\)

2.9.2 Метод Гаусса-Зейделя.

Аналогично методу Якоби мы можем построить частный случай метода Зейделя в следующем виде

$$x_i^{k+1} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}, \quad i = \overline{1, n}, \quad k = 0, 1, \dots; \quad x^0 \quad (3)$$

Для того, чтобы записать это в матричном виде, представим

$$A = L + D + U,$$

где

$$L = \begin{pmatrix} 0 & \dots & 0 & 0 \\ a_{21} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}, \quad D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} 0 & a_{12} & \dots & a_{1n-1} & a_{1n} \\ 0 & 0 & \dots & a_{2n-1} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{n-1n} \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Теперь можем записать матричное представление метода Гаусса-Зейделя:

$$x^{k+1} = -(D + L)^{-1} U x^k + (D + L)^{-1} b.$$

Фактически мы записали метод простой итерации с заданной в специальном виде матрицей.

Теорема. Метод Гаусса-Зейделя сходится при любом начальном приближении x^0 тогда и только тогда, когда все корни уравнения

$$\left| U + \lambda(D + L) \right| = \begin{vmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ \lambda a_{21} & \lambda a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda a_{n1} & \lambda a_{n2} & \dots & \lambda a_{nn} \end{vmatrix} = 0$$

по модулю меньше единицы.

Теорема. Если исходная система (1) обладает матрицей A , являющейся симметрической и положительно определенной, то есть $A = A^T > 0$, то метод Гаусса-Зейделя сходится.

2.10 Итерационные методы вариационного типа.

2.10.1 Решение СЛАУ и точка минимума квадратичного функционала ошибки.

Для простоты изложения мы будем строить методы для системы

$$Ax = b, \quad A = A^T > 0, \quad a_{ij}, b_i, x_i \in \mathbb{R}. \quad (1)$$

Наряду с задачей (1) будем рассматривать функционал

$$F(x) = (Ax, x) - 2(b, x)$$

которая называется **квадратичным функционалом ошибки**, являющийся многочленом второй степени от x . Очевидно, что он сопряжен с системой (1) посредством наличия тех же коэффициентов, то есть

$$F(x) = \sum_{i,j=1}^n a_{ij}x_{ij} - 2 \sum_{i=1}^n b_i x_i. \quad (2)$$

Ввиду того, что A положительно определенная, многочлен (2) имеет единственный минимум. Покажем, что решение задачи

$$x^* = A^{-1}b$$

доставляет минимум функционала (2). Если мы установим соответствие между (1) и (2), то минимум функционала (2), т.е. решение задачи минимизации функционала (2), доставит нам решение исходной задачи. Для этого докажем две теоремы.

Теорема. *Если в некоторой точке $x = (x_1, \dots, x_n)$ многочлен (2) имеет минимум, то координаты этой точки удовлетворяют системе (1).*

◆ Рассмотрим произвольную точку x и придадим этой точке приращение $\alpha\Delta x$, где α — это произвольный численный множитель, а Δx — произвольный вектор. Рассмотрим

$$\begin{aligned} F(x + \alpha\Delta x) &= (Ax + \alpha A\Delta x, x + \alpha\Delta x) - 2(b, x + \alpha\Delta x) = \\ &= (Ax, x) + \alpha(Ax, \Delta x) + \alpha(A\Delta x, x) + \alpha^2(A\Delta x, \Delta x) - 2(b, x) - 2\alpha(b, \Delta x) = \\ &= F(x) + \alpha^2(A\Delta x, \Delta x) + 2\alpha(Ax - b, \Delta x). \end{aligned}$$

$$\Delta F(x) = F(x + \alpha\Delta x) - F(x) = \alpha^2(A\Delta x, \Delta x) + 2\alpha(Ax - b, \Delta x). \quad (3)$$

Если x — точка минимума (а по условию она является таковой), то должно выполняться равенство

$$(Ax - b, \Delta x) = 0, \quad \forall \Delta x.$$

Действительно, если мы предположим противное, что

$$\exists \Delta x : (Ax - b, \Delta x) \neq 0,$$

то мы получим противоречие. Потому что при достаточно малых α слагаемое $2\alpha(Ax - b, \Delta x)$ становится главным. Значит $\Delta F(x)$ изменит знак при изменении знака α . То есть точка x не будет являться точкой минимума. Следовательно,

$$(Ax - b, \Delta x) = 0, \quad \forall \Delta x \iff Ax - b = 0,$$

то есть x — решение системы (1). □

Теорема. *Если x — решение уравнения (1), то многочлен (2) имеет в точке x абсолютный минимум во всем пространстве.*

◆ По условию x — решение. Значит второе слагаемое в (3) равно нулю. В итоге

$$\Delta F(x) = \alpha^2(A\Delta x, \Delta x) > 0, \quad \forall \Delta x \neq 0.$$

Таким образом, $\Delta F(x) = 0 \iff \Delta x = 0$. Значит x есть абсолютный минимум в пространстве определения функционала (2). □

Теоремы показывают, что решение системы (1) и нахождение минимума функции $F(x)$ — задачи равносильные.

2.10.2 Метод покоординатного спуска.

Пусть $x^0 = (x_1^0, \dots, x_n^0)$ — начальное приближение в точке минимума функции (2). Рассмотрим функцию одного аргумента $F(x_1, x_2^0, \dots, x_n^0)$. Для отыскания ее минимума нам необходимо, чтобы производная обращалась в ноль. А для этого

$$\frac{\partial}{\partial x_1} F(x_1, x_2^0, \dots, x_n^0) = 2(a_{11}x_1 + a_{12}x_2^0 + \dots + a_{1n}x_n^0 - b_n) = 0.$$

Находим x_1^1 . Составляем уравнение

$$\frac{\partial}{\partial x_2} F(x_1^1, x_2, \dots, x_n^0) = 0.$$

И так далее.

$$\frac{\partial}{\partial x_n} F(x_1^1, x_2^1, \dots, x_n) = 0.$$

В итоге получим n условий для вычисления производной функции F . Тогда из этих условий

$$\left\{ \begin{array}{l} x_1^1 = -\frac{a_{12}}{a_{11}}x_2^0 - \dots - \frac{a_{1n}}{a_{11}}x_n^0 + \frac{b_1}{a_{11}}, \\ x_2^1 = -\frac{a_{21}}{a_{22}}x_1^1 - \frac{a_{23}}{a_{22}}x_3^0 - \dots - \frac{a_{2n}}{a_{22}}x_n^0 + \frac{b_2}{a_{22}}, \\ \dots\dots\dots \\ x_n^1 = -\frac{a_{n1}}{a_{nn}}x_1^1 - \dots - \frac{a_{nn-1}}{a_{nn}}x_{n-1}^1 + \frac{b_n}{a_{nn}}. \end{array} \right.$$

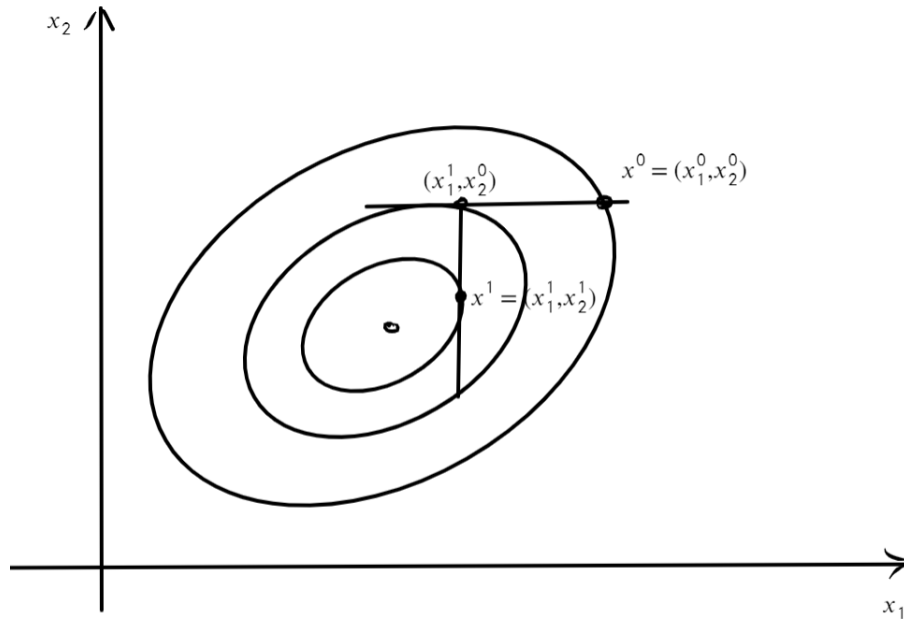
Если заменить теперь

[illegible]

то полученная формула совпадает с методом Гаусса-Зейделя. Таким образом, метод покоординатного спуска является одним из частных случаев метода Зейделя.

Изобразим геометрически как мы находим решение. Возьмем $n = 2$. Это значит, что

$$F(x_1, x_2) = a_{11}(x_1)^2 + 2a_{12}x_1x_2 + a_{22}(x_2^2) - 2(b_1x_1 + b_2x_2).$$



2.10.3 Метод градиентного спуска.

Этот метод может быть применен для нахождения точки минимума любого функционала, для которого имеет смысл понятие градиента.

• **Градиентом функции** $F(x_1, \dots, x_n)$ называется вектор с координатами

$$\text{grad } F(x) = \left(\frac{\partial F}{\partial x_1}, \dots, \frac{\partial F}{\partial x_n} \right)^T.$$

Пусть известно начальное приближение $x^0 = (x_1^0, \dots, x_n^0)$. Из этой точки будем двигаться в направлении $-\text{grad } F(x^0)$. Тогда путь, по которому мы будем перемещаться, имеет вид уравнения

$$x = x^0 - t \text{grad } F(x^0), \quad t \geq 0.$$

Будем двигаться до тех пор, пока функция F не достигнет своего минимума. Но для того, чтобы сформулировать это математически, нам нужно следить за изменением. Введем обозначение

$$\varphi(t) = F(x^0 - t \text{grad } F(x^0)).$$

То есть мы останавливаемся в тот момент t , когда

$$\varphi'(t) = \frac{d\varphi(t)}{dt} = 0.$$

Выясним, какой вид имеет градиент в случае (2)

$$\text{grad } F(x) = 2(Ax - b).$$

Учитывая, что

$$\varphi(t) = (Ax, x) - 2(b, x), \quad x = x^0 - 2t(Ax^0 - b).$$

Продифференцируем эту функцию по t :

$$\begin{aligned} \frac{d}{dt}(Ax, x) &= \left(A \frac{dx}{dt}, x \right) + \left(Ax, \frac{dx}{dt} \right) = 2 \left(\frac{dx}{dt}, Ax \right) = 2(-\text{grad } F(x^0), Ax) = \\ &= \dots = 8t(Ax^0 - b) - 4(Ax^0 - b, Ax^0). \end{aligned}$$

$$\frac{d}{dt}(b, x) = -2(Ax^0 - b, b).$$

Теперь вычислим

$$\varphi'(t) = 8t(Ax^0 - b, A(Ax^0 - b)) - 4(Ax^0 - b, Ax^0 - b).$$

А мы ищем

$$\varphi'(t) = 0.$$

То есть

$$8t(Ax^0 - b, A(Ax^0 - b)) - 4(Ax^0 - b, Ax^0 - b) = 0$$

и отсюда

$$t = \frac{(Ax^0 - b, Ax^0 - b)}{2(Ax^0 - b, A(Ax^0 - b))}.$$

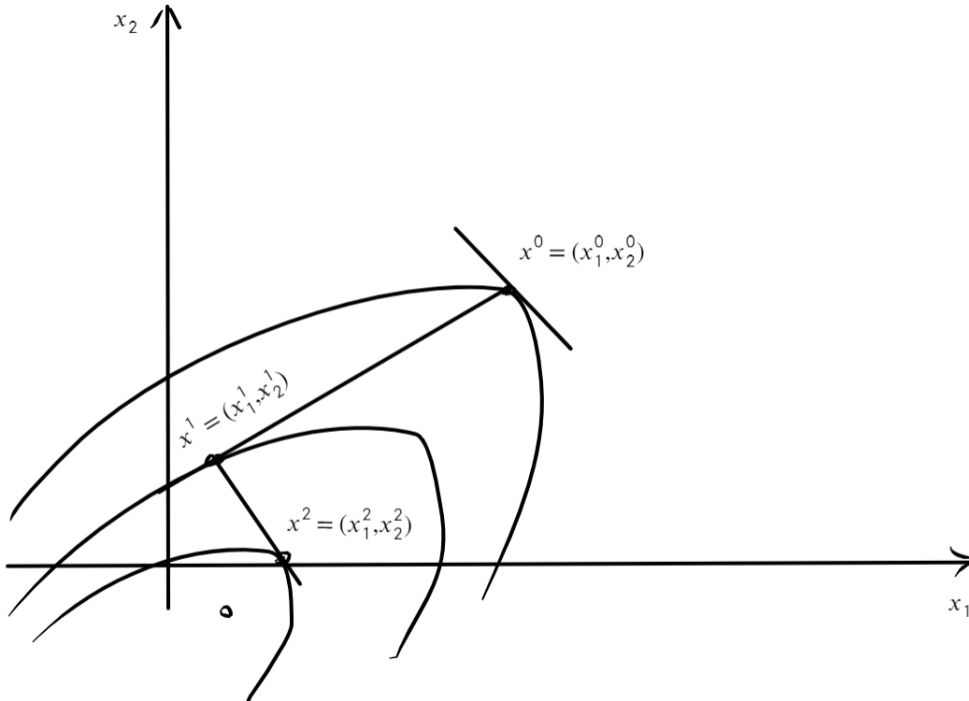
Если мы нашли значение t , то мы можем уточнить вектор x :

$$x^1 = x^0 - \frac{(Ax^0 - b, Ax^0 - b)}{2(Ax^0 - b, A(Ax^0 - b))} \cdot 2(Ax^0 - b).$$

Из x^1 точно так же переходим к x^2 . Легко видеть, что эта формула не изменится, а она будет носить рекуррентный вид. Отсюда запишем **общий метод итерационного процесса градиентного спуска**. Для этого введем обозначение невязки $r = Ax - b$. Тогда

$$x^{k+1} = x^k - \frac{(r^k, r^k)}{(Ar^k, r^k)} r^k, \quad r^k = Ax^k - b. \quad (4)$$

Геометрический смысл строится аналогично методу покоординатного спуска.



Покажем сходимость метода.

Теорема. Если $A = A^T > 0$, то метод градиентного спуска (4) сходится и для погрешности справедлива оценка

$$\|x^k - x^*\| \leq \rho_0^k \|x^0 - x^*\|, \quad \rho_0 = \frac{1 - \lambda_{\min}(A)}{1 + \lambda_{\max}(A)}.$$

2.10.4 Метод релаксации.

Этот метод основан на минимизации функции ошибки

$$z = x - x^*.$$

Для того, чтобы работать с этой функцией, определим норму, порожденную матрицей A :

$$\|z\|_A = \sqrt{(Az, z)}.$$

Будем говорить, что итерационный процесс сходится в норме матрицы A , если

$$\lim_{k \rightarrow \infty} \|z^k\|_A = \lim_{k \rightarrow \infty} \|x^k - x^*\| = 0.$$

Предположим, что у нас имеется s -ое приближение к x^*

$$x^s = (x_1^s, \dots, x_n^s).$$

Под следующим приближением мы будем понимать такое значение вектора x , у которого меняется лишь i -ая компонента

$$x^{s+1} = (x_1^s, \dots, x_i^{s+1}, \dots, x_n^s).$$

Формально можно записать итерационный процесс от s итерации к $s+1$ как

$$x^{s+1} = x^s - \alpha e_i, \quad \alpha = \text{const}, \quad e_i = (0, \dots, \underset{i}{1}, \dots, 0).$$

Вычислим

$$\|z^{s+1}\|_A^2 = (Az^{s+1}, z^{s+1}) = (A(z^s - \alpha e_i), z^s - \alpha e_i) = \dots = \|z^s\|_A^2 + \alpha^2 a_{ii} - 2\alpha r_i^s.$$

$$\|z^{s+1}\|_A^2 = \|z^s\|_A^2 + \frac{1}{a_{ii}}(\alpha a_{ii} - r_i^s)^2 - \frac{(r_i^s)^2}{a_{ii}}. \quad (5)$$

Как следует из формулы (5) минимум величины z^{s+1} достигается при $\alpha = \frac{r_i^s}{a_{ii}}$. При этом α значение $r_i^{s+1} = 0$. Действительно, можно показать, что

$$r_i^{s+1} = (Az^{s+1}, e_i) = (A(z^s - \alpha e_i), e_i) = (Az^s, e_i) - \alpha(Ae_i, e_i) = r_i^s - \alpha a_{ii} = 0$$

при выбранном $\alpha = \frac{r_i^s}{a_{ii}}$.

Однако при построении очередного приближения необязательно каждый раз минимизировать норму

$$\|z^{s+1}\|_A^2.$$

Нам достаточно потребовать, чтобы выполнялось

$$\|z^{s+1}\|_A^2 < \|z^s\|_A^2.$$

Оказывается, что последнее условие позволяет построить более оптимальные итерационные процессы. Если мы рассмотрим выполнение данного неравенства, то условие будет следующее

$$|\alpha a_{ii} - r_i^s| < |r_i^s|.$$

Теперь, если мы выберем

$$\alpha = \omega \frac{r_i^s}{a_{ii}}, \quad 0 < \omega < 2,$$

то получим следующее выражение

$$r_i^{s+1} = (Az^{s+1}, e_i) = (1 - \omega)r_i^s.$$

Это соотношение позволяет записать общий метод — **метод релаксации**, формулы которого имеют следующий вид: если предположить

$$x^s = (x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^k, x_{i+1}^k, \dots, x_n^k)$$

то

$$x^{s+1} = (x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^{k+1}, x_{i+1}^k, \dots, x_n^k).$$

Тогда запишем

$$x_i^{k+1} = (1 - \omega)x_i^k - \omega \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \omega \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^k + \omega \frac{b_i}{a_{ii}}, \quad i = \overline{1, n}, \quad k = 0, 1, \dots, \quad 0 < \omega < 2. \quad (6)$$

Итерационный процесс (6) называется **методом релаксации**. Если $\omega \in (0, 1)$, то получим **метод нижней релаксации**. Если $\omega \in (1, 2)$, то это **метод верхней релаксации**. Если $\omega = 1$, то получим **метод полной релаксации**. Также при $\omega = 1$ мы получим метод Гаусса-Зейделя.

Оптимальным значением ω , которое дает максимальную скорость сходимости, является значение $\omega \in (1, 2)$.

Теорема. Пусть $A = A^T > 0$, $0 < \omega < 2$. Тогда метод релаксации сходится.

Запишем итерационный процесс (6) в матричной форме: если матрицу системы представить в виде суммы

$$A = L + D + U$$

тогда итерационный процесс (6) можно записать в матричном виде:

$$(D + \omega L) \frac{x^{k+1} - x^k}{\omega} + Ax^k = b.$$

Оптимальное значение ω часто определяется экспериментально.

2.10.5 Метод минимальных невязок.

Зададим начальное приближение x^0 . Очередную итерацию, как в методе градиентного спуска, мы будем искать в виде

$$x^1 = x^0 - \tau_1 r^0.$$

В некотором смысле функция невязки и есть величина градиента. По определению

$$r^0 = Ax^0 - b.$$

Причем τ_1 мы выберем так, чтобы минимизировать функционал невязки

$$(r^0, r^0) = \|r^0\|^2 \rightarrow \min.$$

Докажем, что в качестве τ нам следует выбирать величину

$$x^{k+1} = x^k - \tau_{k+1} r^k, \quad \tau_{k+1} = \frac{(Ar^k, r^k)}{(Ar^k, Ar^k)} \quad (7)$$

Формула (7) определяет **метод минимальных невязок**.

Действительно так как

$$r^{k+1} = r^k - \tau_{k+1} Ar^k,$$

то посчитаем норму невязки на $k + 1$:

$$\begin{aligned} \|r^{k+1}\|^2 &= \|r^k - \tau_{k+1} Ar^k\|^2 = \underbrace{(r^k, r^k)}_{=\|r^k\|^2} - 2\tau_{k+1}(Ar^k, r^k) + \tau^2(Ar^k, Ar^k). \end{aligned}$$

Отсюда видно, что

$$\|r^{k+1}\|^2 \rightarrow \min$$

при

$$\tau_{k+1} = \frac{(Ar^k, r^k)}{(Ar^k, Ar^k)}.$$

Теорема. Если $A = A^T > 0$, то метод (7) сходится и для погрешности метода справедлива оценка

$$\|A(x^k - x^*)\| \leq \rho_0^k \|A(x^0 - x^*)\|, \quad k = 0, 1, \dots \quad \rho_0 = \frac{1 - \lambda_{\min}(A)}{1 + \lambda_{\max}(A)}.$$

Глава 3

Методы решения проблемы собственных значений (ПСЗ).

3.1 Общие положения ПСЗ.

Сформулируем задачу, изучаемую в этой главе. Пусть матрица A невырожденная, размерности $n \times n$ с элементами a_{ij} , $i, j = \overline{1, n}$.

- Число λ называется **собственным значением** матрицы A , если

$$\exists x \neq 0 : Ax = \lambda x. \quad (1)$$

Вектор x называется **собственным вектором** матрицы A , соответствующим данному собственному значению матрицы A .

Для существования задачи (1) необходимо, чтобы

$$|A - \lambda E| = \begin{vmatrix} a_{11} - \lambda & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} - \lambda \end{vmatrix} = 0 \quad (2)$$

- Левая часть равенства (2) называется **характеристическим многочленом**. А само уравнение — **характеристическим (вековым) уравнением**.

Очевидно, что $|A - \lambda E|$ есть алгебраический многочлен степени n от λ со старшим коэффициентом $(-1)^n$

$$|A - \lambda E| = (-1)^n (\lambda^n - p_1 \lambda^{n-1} - \dots - p_{n-1} \lambda - p_n) = (-1)^n P_n(\lambda) \quad (3)$$

- Многочлен $P_n(\lambda)$ в формуле (3) называется **собственным многочленом** матрицы A .
- Совокупность всех корней характеристического многочлена матрицы называется **спектром матрицы**.

Таким образом, задачу нахождения собственных значений и собственных векторов можно разбить на 3 этапа:

- построение собственного многочлена $P_n(\lambda)$ матрицы;
- решение уравнения $P_n(\lambda) = 0$ и нахождение его корней $\lambda_i(A)$;

- отыскание нетривиальных решений систем вида

$$Ax_i = \lambda_i x_i, \quad i = \overline{1, n}.$$

Если необходимо знать все собственные значения и соответствующие векторы, то такая задача называется **полной ПСЗ**. Если необходимо знать одно или несколько собственных значений и соответствующих векторов, то такая задача называется **частичной ПСЗ**. На основе этого различают и методы решений задачи ПСЗ. Для решения полной ПСЗ используются прямые методы, а для частичной — прямые и итерационные методы.

Рассмотрим вопрос об устойчивости ПСЗ. Исходными выходными данными в нашей задаче являются элементы матрицы A . Если мы имеем элементы матрицы A с погрешностью, то вместо точного решения x мы получим возмущенное решение, при этом этому x будет соответствовать и возмущенное λ :

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

Устойчивость задачи будет зависеть от величины, называемой **коэффициентом перекоса матрицы**:

$$\omega = \frac{\|x\| \cdot \|y\|}{|(x, y)|},$$

причем мы выбираем спектральную (третью) норму векторов. A x и y являются собственными векторами относительно матриц A и A^* соответственно:

$$Ax = \lambda x, \quad A^* y = \bar{\lambda} y.$$

Если случай действительный, то $\lambda = \bar{\lambda}$, $A^* = A^T$. Очевидно, что $\omega \geq 1$. По определению фактически

$$\omega = \frac{1}{|\cos \alpha|},$$

где α — угол между векторами x и y . Приведем основные оценки погрешности входных данных, которые обеспечивают устойчивость. В случае попарно различных собственных значений справедливы оценки

$$\begin{aligned} |\delta \lambda_i| &\leq \omega_i \|\delta A\|_{\text{III}} \\ \|\delta x_i\| &\leq \|x_i\| \left(\sum_{j=1}^n \frac{\omega_j}{|\lambda_i - \lambda_j|} \right) \|A\|_{\text{III}}, \quad i \neq j \end{aligned} \quad (4)$$

Как следует из оценок (4), можно сделать следующие выводы:

1. собственное значение устойчиво относительно вариации матричных элементов, если соответствующий ему коэффициент перекоса мал; если коэффициент велик, то устойчивость будет плохой;
2. собственный вектор устойчив относительно вариации, если все коэффициенты перекоса по всем собственным значениям достаточно малы;
3. если матрица эрмитова, т.е. $A = A^*$, то все $\omega_i = 1$, а значит ПСЗ устойчива по всем λ_i , а проблема определения собственных векторов устойчива, если все λ_i простые (отсутствуют кратные корни).

3.2 Метод Данилевского.

Идея метода опирается на известный факт, что подобное преобразование матрицы A не изменяет ее характеристического многочлена:

$$\exists S, |S| \neq 0 : B = S^{-1}AS.$$

Действительно

$$|B - \lambda E| = |S^{-1}AS - \lambda S^{-1}ES| = |S^{-1}| \cdot |A - \lambda E| \cdot |S| = |A - \lambda E|.$$

Для того, чтобы этот метод записать, рассмотрим каноническую форму Фробениуса:

$$\Phi = \begin{pmatrix} p_1 & p_2 & \dots & p_{n-1} & p_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}.$$

Непосредственная проверка показывает, что числа, стоящие в первой строке матрицы Фробениуса, являются коэффициентами собственного многочлена $P_n(\lambda)$. Действительно

$$\begin{aligned} |\Phi - \lambda E| &= \begin{vmatrix} p_1 - \lambda & p_2 & \dots & p_{n-1} & p_n \\ 1 & -\lambda & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 - \lambda & 0 \end{vmatrix} = (p_1 - \lambda)(-\lambda)^{n-1} - \begin{vmatrix} p_2 & p_3 & \dots & p_{n-1} & p_n \\ 1 & -\lambda & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 - \lambda & 0 \end{vmatrix} = \\ &= (p_1 - \lambda)(-\lambda)^{n-1} - p_2(-\lambda)^{n-2} - \dots + (-1)^{n+1}p_n = (-1)^n P_n(\lambda). \end{aligned}$$

3.2.1 Построение собственного многочлена.

Согласно методу Данилевского нам нужно найти такую матрицу S , что

$$\Phi = S^{-1}AS.$$

Алгоритм метода Данилевского состоит в том, чтобы последовательными преобразованиями матрицы A сделать ее вида Фробениуса. В зависимости от коэффициентов a_{ij} различают два алгоритма

1. Регулярный случай.

Нам нужно последнюю строку матрицы A превратить в последнюю строку матрицы Фробениуса

$$(a_{n1}, \dots, a_{nn-1}, a_{nn}) \rightarrow (0, \dots, 1, 0).$$

Предположим, что $a_{nn-1} \neq 0$. Разделим все элементы $n - 1$ -го столбца матрицы A на элемент a_{nn-1} , вновь полученный $n - 1$ -ый столбец умножим на a_{ni} и вычтем из столбцов номера i . Проделаем это для всех $i = 1, 2, \dots, n - 2, n$, мы приведем последнюю строку матрицы A к виду Фробениуса. Такое преобразование равносильно умножению матрицы A справа на матрицу простой структуры

$$M_{n-1} = \begin{pmatrix} 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ -\frac{a_{n1}}{a_{nn-1}} & \dots & \frac{1}{a_{nn-1}} & -\frac{a_{nn}}{a_{nn-1}} \\ 0 & \dots & 0 & 1 \end{pmatrix}.$$

После умножения получили

$$B = AM_{n-1}.$$

Домножим слева на M_{n-1}^{-1} (она существует, так как $|M_{n-1}| = \frac{1}{a_{n-1}} \neq 0$)

$$M_{nn-1}^{-1} = \begin{pmatrix} 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn-1} & a_{nn} \\ 0 & \dots & 0 & 1 \end{pmatrix}.$$

В итоге за счет двух умножений мы получили новую матрицу

$$A_1 = M_{n-1}^{-1}AM_{n-1} = \begin{pmatrix} a_{11}^1 & \dots & a_{1n-1}^1 & a_{1n}^1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n-11}^1 & \dots & a_{n-1n-1}^1 & a_{n-1n}^1 \\ 0 & \dots & 1 & 0 \end{pmatrix}.$$

Запишем, как пересчитываются элементы при умножении справа

$$b_{ij}a_{ij} + a_{in-1}m_{n-1j}, \quad j \neq n-1.$$

$$b_{in-1} = a_{in-1}m_{n-1n-1}, \quad i = \overline{1, n}$$

Поменяется только $n-1$ строка матрицы, поэтому

$$a_{ij}^1 = b_{ij}, \quad i = \overline{1, n-2}.$$

$$a_{n-1j}^1 = \sum_{k=1}^n a_{nk}b_{kj}, \quad j = \overline{1, n}.$$

Далее нам нужно привести вторую строку в матрице A_1 к виду Фробениуса. То есть, нам нужно сделать единицу на месте $a_{n-1n-2}^1 \neq 0$. Предположим, что он отличен от нуля. Теперь будем умножать на матрицу M_{n-2} и M_{n-2}^{-1} . Эти матрицы формируются аналогично M_{n-1} , за исключением того, что отличной от единичной будет $n-2$ -ая строка. После второго шага получим матрицу

$$A_2 = M_{n-2}^{-1}A_1M_{n-2}.$$

И так далее. Будем предполагать, что все остальные элементы находящиеся на диагонали над главной диагональю отличны от нуля

$$a_{nn-1} \neq 0, a_{n-1n-2}^1 \neq 0, \dots, a_{21}^{n-1} \neq 0.$$

после выполнения $n-1$ -го шага преобразования мы получим матрицу

$$\begin{aligned} A_{n-1} &= \underbrace{M_1^{-1}M_2^{-1} \dots M_{n-1}^{-1}}_{S^{-1}} A \underbrace{M_{n-1} \dots M_2 M_1}_S = \begin{pmatrix} a_{11}^{n-1} & a_{12}^{n-1} & \dots & a_{1n-1}^{n-1} & a_{1n}^{n-1} \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & \vdots & 1 & 0 \end{pmatrix} = \\ &= \begin{bmatrix} p_1 & p_2 & \dots & p_{n-1} & p_n \\ 1 & 0 & \dots & 0 & 0 \\ \dots & \ddots & \dots & \dots & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} = \Phi. \end{aligned}$$

По первой строке полученной матрицы составляется собственный многочлен матрицы A .

2. Нерегулярный случай.

Пусть процесс сведения матрицы A к форме Фробениуса доведен до строки k и при этом элемент

$$a_{kk-1}^{n-k} = 0.$$

Дальнейшее преобразование может быть выполнено по двум вариантам:

- (a) Мы должны проанализировать все элементы, стоящие левее элемента $a_{ki}^{n-k} \neq 0$, $i < k-1$. Столбец с номером ki мы можем поменять местами с этим столбцом. Также необходимо поменять и строки с этими номерами. Далее по алгоритму регулярного случая;
- (b) Все $a_{ki}^{n-k} = 0$, $i = \overline{1, k-1}$. В этом случае матрица

$$A_{n-k} = \left(\begin{array}{ccc|ccc} a_{11}^{n-k} & \dots & a_{1k-1}^{n-k} & a_{1k}^{n-k} & \dots & a_{1n}^{n-k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{k-11}^{n-k} & \dots & a_{k-1k-1}^{n-k} & a_{k-1k}^{n-k} & \dots & a_{k-1n}^{n-k} \\ \hline 0 & \dots & 0 & a_{kk}^{n-k} & \dots & a_{kn}^{n-k} \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{array} \right) = \left(\begin{array}{c|c} C_{n-k} & D_{n-k} \\ \hline 0 & \Phi_{n-k} \end{array} \right)$$

Рассмотрим определитель, который является на данном этапе искомым

$$|A_{n-k} - \lambda E| = \underbrace{|C_{n-k} - \lambda E_1|}_{P_{k-1}(\lambda)} \cdot \underbrace{|\Phi_{n-k} - \lambda E_2|}_{P_{n-k+1}(\lambda)} = 0.$$

В этой формуле E_1 и E_2 — единичные матрицы порядка $(k-1)$ и $(n-k+1)$ соответственно.

Поскольку второй сомножитель является определителем от матрицы Фробениуса, то для этого определителя мы можем выписать собственный многочлен по первой строке этой матрицы. Коэффициенты первого многочлена из этого произведения определяются по алгоритму регулярного случая.

Замечания.

1. Число операций в регулярном случае равно $O(n^3)$.
2. Чтобы повысить надежность вычислений, можно как в метода Гаусса с выбором элемента на каждом шаге преобразования матриц выбирать в качестве элемента, на который производится деление, можно ставить максимальный по модулю элемент матрицы, находящийся левее и выше него.
3. Собственные значения найти по этому алгоритму мы не сможем, потому что мы лишь строим полином $P_n(\lambda) = 0$.

3.2.2 Вычисление собственных векторов матриц.

Собственный вектор является решением системы

$$Ax = \lambda_i x, \quad i = \overline{1, n}.$$

Метод Данилевского позволяет построить систему собственных векторов матрицы A , используя вычисления, произведенные в процессе построения собственного многочлена.

- Аннулирующий многочлен наименьшей степени называется **минимальным** **многочленом** матрицы A .

Свойства:

1. $\exists!$ минимальный аннулирующий многочлен матрицы A ;
2. $\exists!$ минимальный аннулирующий $c \neq 0$ многочлен матрицы A и корни этого многочлена являются собственными значениями.

Теорема (Гамильтона-Кели). Если $P_n(\lambda)$ — собственный многочлен матрицы A , то

$$P_n(A) = A^n - p_1 A^{n-1} - \dots - p_{n-1} A - p_n E = 0.$$

Следовательно, $P_n(\lambda)$ — аннулирующий многочлен матрицы A .

3.3.1 Построение собственного многочлена.

Рассмотрим произвольный ненулевой вектор c^0 (например, $(1, \dots, 0)^T$), размерность которого согласована с матрицей A , и по этому вектору составим рекуррентную последовательность векторов:

$$c^0, \quad c^1 = Ac^0, \quad c^2 = Ac^1 = A^2c^0, \dots$$

Вычисление будем продолжать до тех пор, пока не получим вектор, являющийся линейной комбинацией предшествующих векторов. Мы дошли до какого-то шага, где

$$c^m = q_1 c^{m-1} + q_2 c^{m-2} + \dots + q_m c^0,$$

при этом очевидно, что c^0, \dots, c^{m-1} линейно независимы, то есть

$$\sum_{i=1}^m q_i^2 > 0.$$

Очевидно, что этот момент наступит по крайней мере в момент $m \leq n$. Для определения коэффициентов q_i запишем предельно возможную комбинацию, то есть в случае, когда $m = n$:

$$q_1c^{n-1} + q_2c^{n-2} + \dots + q_nc^0 = c^n.$$

Распишем данную систему по координатам:

[illegible]

(1) является неоднородной СЛАУ, определитель которой отличен от нуля лишь в случае линейной независимости векторов c^0, \dots, c^{n-1} . Применим метод Гаусса для решения системы (1). При этом возможны два варианта:

1. Все n шагов прямого хода метода Гаусса выполнимы \Rightarrow система приводится к верхнему треугольному виду

[illegible]

Тогда определитель системы (1) отличен от нуля и векторы c^0, \dots, c^{n-1} линейно независимые. Поэтому q_1, \dots, q_n найдутся единственным образом из системы (2) обратным ходом метода Гаусса. Нам осталось показать, что полученные коэффициенты q_1, \dots, q_n совпадают с соответствующими коэффициентами собственного многочлена

$$P_n(\lambda) = \lambda^n - p_1\lambda^{n-1} - \dots - p_n.$$

Для того, чтобы это доказать воспользуемся теоремой Гамильтона-Кели. Поскольку равенство

$$P_n(A) = A^n - p_1 A^{n-1} - \dots - p_{n-1} A - p_n E = 0$$

справедливо для матрицы A , то умножим это равенство на c^0 справа и воспользуемся тем фактом, что по построению

$$A^i c^0 = c^i, \quad i = \overline{1, n}.$$

В итоге получим

$$p_1 c^{n-1} + p_2 c^{n-2} + \dots + p_n c^0 = c^n. \quad (3)$$

С другой стороны,

$$q_1 c^{n-1} + q_2 c^{n-2} + \dots + q_n c^0 = c^n. \quad (4)$$

Вычитая из уравнения (3) уравнение (4), мы получим

$$(p_1 - q_1)c^{n-1} + (p_2 - q_2)c^{n-2} + \dots + (p_n - q_n)c^n = 0.$$

Ввиду линейной независимости векторов c^0, \dots, c^{n-1} такое равенство возможно тогда, когда все коэффициенты в этом разложении обращаются в 0. Отсюда следует, что

$$p_i = q_i, \quad i = \overline{1, n}.$$

Таким образом, мы построили собственный многочлен.

2. Прямой ход метода Гаусса возможен только на $m < n$ шагов преобразования. После их выполнения система будет иметь следующий вид:

[illegible]

Следовательно, ранг системы (1) равен m и среди всех векторов c^0, \dots, c^{n-1} есть только m линейно независимых. Обозначим их

$$c^0, \dots, c^{m-1}.$$

Поскольку эти векторы являются линейно независимыми и $m < n$, то мы можем записать соответствующую линейную комбинацию

$$q_1 c^{m-1} + q_2 c^{m-2} + \dots + c_m c^0 = c^m \quad (6)$$

Далее, записав эту систему покомпонентно, и выбрав m линейно независимых уравнений, мы найдем q_1, \dots, q_m . Если домножить уравнение (6) на c^0 и воспользоваться тем, что

$$A^i c^0 = c^i, \quad i = \overline{1, n},$$

мы получим следующее уравнение

$$\underbrace{(A^m - q_1 A^{m-1} - \dots - q_m E)}_{\varphi(A)} c^0 = 0.$$

Мы таким образом получили, что

$$\varphi(A) c^0 = 0,$$

то есть $\varphi(A)$ — аннулирующий многочлен для вектора c^0 . Тогда

$$\varphi(\lambda) = \lambda^m - q_1 \lambda^{m-1} - \dots - q_m$$

и он является делителем любого другого многочлена матрицы A аннулирующего вектор c^0 . В частности, он является делителем и собственного многочлена $P(\lambda)$. Решая уравнение

$$\varphi(\lambda) = 0,$$

мы найдем λ_i , $i = \overline{1, m}$, $m < n$ — лишь часть спектра матрицы A . Следовательно, мы можем взять другое начальное приближение. И прийти либо к случаю 1, либо прийти к случаю 2, но получить другие коэффициенты.

Замечание. Если изменением вектора c^0 не удастся найти больше m собственных чисел, то многочлен $\varphi(\lambda)$ будет являться минимальным многочленом матрицы A . Тогда получаются кратные значения.

Для определения коэффициентов по методу Крылова требуется количество операций порядка $Q = O(n^3)$. Число операций умножения и деления $\frac{3}{2}n^2(n+1)$.

3.3.2 Вычисление собственных векторов матрицы.

Рассмотрим для случая $m < n$. Пусть λ_i — корень минимального многочлена $\varphi(\lambda)$ матрицы A , аннулирующего вектор c^0 . Собственный вектор x_i , соответствующий данному значению λ_i будем находить в форме линейной комбинации построенных по методу Крылова собственных значений:

$$x_i = \beta_{i1} c^{m-1} + \beta_{i2} c^{m-2} + \dots + \beta_{im} c^0. \quad (7)$$

Подберем коэффициенты β_{ij} таким образом, чтобы выполнялось равенство

$$Ax_i = \lambda_i x_i \quad (8).$$

Подставим сейчас уравнение (7) в уравнение (8) и воспользуемся соотношением $Ac^j = c^{j+1}$, $j = \overline{0, n-1}$. Тогда получим вместо (8) следующее равенство

$$\beta_{i1} c^m + \beta_{i2} c^{m-1} + \dots + \beta_{im} c^1 = \lambda_i \cdot (\beta_{i1} c^{m-1} + \beta_{i2} c^{m-2} + \dots + \beta_{im} c^0).$$

Поскольку c^m выражается через c^0, \dots, c^{m-1} :

$$c^m = q_1 c^{m-1} + \dots + q_m c^0,$$

мы можем исключить его из полученного равенства. В итоге мы получим следующее уравнение

$$(q_1 \beta_{i1} + \beta_{i2} - \lambda_i \beta_{i1}) c^{m-1} + (q_2 \beta_{i1} + \beta_{i3} - \lambda_i \beta_{i2}) c^{m-2} + \dots + (q_m \beta_{i1} - \lambda_i \beta_{im}) c^0 = 0$$

Так как векторы c^0, \dots, c^{m-1} по построению линейно независимы, то нам нужно, чтобы все коэффициенты в этой сумме одновременно обращались в ноль.

[illegible]

Легко видеть, что эту систему можно разрешить и последовательно находить β_{ik} через β_{i1} :

$$\left\{ \begin{array}{l} \beta_{i2} = (\lambda_i - q_1)\beta_{i1}, \\ \beta_{i3} = (\lambda_i^2 - q_1\lambda_i - q_2)\beta_{i1}, \\ \dots\dots\dots \\ \beta_{im} = (\lambda_i^{m-1} - q_1\lambda_i^{m-2} - \dots - q_{m-1})\beta_{i1}, \\ (\lambda_i^m - q_1\lambda_i^{m-1} - \dots - q_m)\beta_{i1} = 0 \end{array} \right.$$

Последнее равенство выполнено, потому что λ_i — собственные значения. Следовательно, мы можем построить вектор x . Обычно полагают

[illegible]

И по формуле (7) строим собственный вектор. Описанный способ нахождения собственных вектор справедлив независимо от случая $m = n$ или $m < n$. Если данному собственному значению принадлежат несколько собственных векторов, то для их отыскания необходимо выбрать другие начальные векторы s^0 и повторить указанную процедуру.

3.4 Итерационный метод вращений решения ППСЗ.

Рассмотрим для простоты случай, когда матрица A действительная и симметричная, т.е. $A = A^T$, $a_{ij} \in \mathbb{R}$.

3.4.1 Предварительные сведения.

Известно, что всякая симметричная действительная матрица может быть приведена к диагональному виду подобным преобразованием. Иными словами, имеет место разложение вида

$$A = U\Lambda U^{-1}, \quad (1)$$

где U — это ортогональная матрица, Λ — диагональная матрица, причем диагональными элементами в этой матрице являются собственные значения λ_i , $i = \overline{1, n}$ матрицы A . При этом столбцы матрицы U являются столбцами координат собственных векторов, соответствующих этим собственным значениям.

- Это разложение называется **каноническим**, или **спектральным** разложением матрицы A .

Поскольку $U^T = U^{-1}$, мы можем записать наряду с (1)

$$U^T A U = \Lambda \quad (2).$$

Таким образом, можно построить последовательных ортогональных (подобных) преобразований, позволяющих неограниченно уменьшать модули недиагональных элементов матрицы A . Мету близости A к диагональному виду определим следующим образом. Введем функцию

$$\sigma_i(A) = \sum_{j=1, i \neq j}^n |a_{ij}|^2, \quad i = \overline{1, n} \quad (3)$$

Теперь за мету близости матрицы A к диагональной мы примем число

$$t(A) = \sigma_1 + \dots + \sigma_n = \sum_{i,j, i \neq j} |a_{ij}|^2. \quad (4)$$

Эта функция и будет являться критерием близости матрицы A к диагональной. Понятно, что $t(A)$ должно стремиться к нулю.

Пусть с помощью преобразования подобия с ортогональными матрицами построена последовательность

$$A = A^0, A^1, \dots, A^k.$$

- Процесс называется **монотонным**, если $t(A^k) < t(A^{k-1})$.

3.4.2 Алгоритм метода.

По заданной матрице A будем строить последовательность матриц A^k так, что каждая следующая матрица A^{k+1} получается из матрицы A^k при помощи преобразования подобия с матрицей вращения T_{ij} (п. 2.7.3). Предположим, что преобразования доведенные до шага k и построена матрица $A^k = \{a_{ij}^k\}$. Найдем в ней наибольший по модулю недиагональный элемент $a_{ij}^k, i \neq j$. Поскольку матрица A симметрическая, то искать элемент можно в верхнем треугольнике $i < j$ (если таких элементов несколько, то можно взять любой из них в качестве максимального по модулю элемента). По индексам i, j строим матрицу $T_{ij}^k = T_{ij}(\varphi^k)$. Значение угла φ мы определим ниже. Образум матрицу

$$A^{k+1} = (T_{ij}^k)^T A^k T_{ij}^k. \quad (5)$$

Такое преобразование является подобным. У этих матриц будет совпадать характеристический многочлен. Для вычисления введем матрицу

$$B^k = A^k T_{ij}^k.$$

У матрицы B от A^k будут отличаться только столбцы под номерами i и j , а именно

$$\begin{cases} b_{li}^k = a_{li}^k \cos \varphi^k + a_{lj}^k \sin \varphi^k, \\ b_{lj}^k = -a_{li}^k \sin \varphi^k + a_{lj}^k \cos \varphi^k. \end{cases}, \quad l = \overline{1, n} \quad (6)$$

Если мы будем умножать матрицу B^k слева на $(T_{ij}^k)^k$, то в матрице

$$A^{k+1} = (T_{ij}^k)^k B$$

будут отличаться две строки по сравнению с матрицей B^k , а именно i -ая и j -ая строки

$$\begin{cases} a_{il}^{k+1} = b_{il}^k \cos \varphi^k + b_{jl}^k \sin \varphi^k, \\ a_{jl}^{k+1} = -b_{il}^k \sin \varphi^k + b_{jl}^k \cos \varphi^k. \end{cases}, \quad l = \overline{1, n} \quad (7)$$

На основании формул (6) и (7) мы сразу можем записать выражение элементов матрицы A^{k+1} через элементы матрицы A^k :

$$a_{ij}^{k+1} = b_{ij}^k \cos \varphi^k + b_{jj}^k \sin \varphi^k = (-a_{ii}^k \sin \varphi^k + a_{ij}^k \cos \varphi^k) \cos \varphi^k + (-a_{ji}^k \sin \varphi^k + a_{jj}^k \cos \varphi^k) \sin \varphi^k$$

Учитывая симметрию задачи, мы упростим и получим $a_{ij}^k = a_{ji}^k$. В итоге

$$a_{ij}^{k+1} = a_{ij}^k \cos 2\varphi^k + \frac{1}{2}(a_{jj}^k - a_{ii}^k) \sin 2\varphi^k. \quad (8)$$

Выберем теперь угол φ^k так, чтобы элемент a_{ij}^{k+1} обратился в нуль. Требование того, что $a_{ij}^{k+1} = 0$ приводит к формулам

$$\operatorname{tg} 2\varphi^k = \frac{2a_{ij}^k}{a_{ii}^k - a_{jj}^k}, \quad |\varphi^k| \leq \frac{\pi}{4} \quad (9)$$

$$\cos 2\varphi^k = \frac{1}{\sqrt{1 + \operatorname{tg}^2 2\varphi^k}}.$$

Пользуясь последней формулой, вычислим

$$\cos \varphi^k = \sqrt{\frac{1 + \cos 2\varphi^k}{2}} \quad (10)$$

Аналогично

$$\sin \varphi^k = \pm \sqrt{\frac{1 - \cos 2\varphi^k}{2}} \quad (10)$$

Чтобы выполнялось условие $|\varphi| \leq \pi/4$, знак в формуле для \sin знак выбирается тот же, что и у выражения

$$a_{ij}^k(a_{ii}^k - a_{jj}^k) \quad (11)$$

Сформулируем алгоритм рассматриваемого метода. Для определения матриц Λ и U в разложении (1) по формуле (5) строим последовательность матриц

$$A^0 = A, \quad A^1, \quad A^2, \quad \dots, \quad A^k, \dots \longrightarrow \Lambda$$

При этом для вычисления матриц на каждом шаге метода используются формулы (9), (10), (11), (6) и (7). Одновременно с построением последовательности матриц A^k строим последовательность матриц U :

$$U^0 = E, \quad U^1, \quad U^2, \quad \dots, \quad U^k, \dots \longrightarrow U$$

где каждое

$$U^{k+1} = U^k \cdot T_{ij}^k, \quad k = 0, 1, \dots$$

Вычисления продолжаем до тех пор, пока не выполнится условие

$$|t(A^{k+1})| \leq \varepsilon,$$

где $t(A)$ — мера близости, определяемая по формуле (4). В итоге в рамках заданной точности ε диагональные элементы последней матрицы A^{k+1} принимаются за искомые собственные значения

$$\lambda_i = a_{ii}^{k+1}, \quad i = \overline{1, n}$$

Столбцы матрицы U^{k+1} принимаются за координаты собственных векторов, соответствующих этим собственным значениям.

3.4.3 Сходимость итерационного процесса.

Если мы докажем, что $t(A^k) \xrightarrow[k \rightarrow \infty]{} 0$, то мы получим сходящийся итерационный процесс. Рассмотрим величину

$$t(A^{k+1}) = \sum_{l=1}^n \sum_{m=1, m \neq l}^n (a_{lm}^{k+1})^2$$

Как следует из формул (6) и (7) при подобном преобразовании вращения сумма недиагональных элементов всех строк матрицы A^{k+1} кроме i -ой и j -ой остается неизменной. То есть изменение $t(A^{k+1})$ от $t(A^k)$ зависит только от i -ой и j -ой строки. Докажем это. С одной стороны,

$$(b_{li}^k)^2 + (b_{lj}^k)^2 = (a_{li}^k)^2 + (a_{lj}^k)^2.$$

Теперь мы возьмем i -ую и j -ую строки, и для них выполняется следующее соотношение:

$$(a_{ij}^{k+1})^2 + (a_{jl}^{k+1})^2 = (b_{il}^k)^2 + (b_{jl}^k)^2.$$

Для всех $l \neq i, l \neq j$

$$b_{il} = a_{il}, \quad b_{jl} = a_{jl}$$

по построению. Следовательно,

$$(a_{ij}^{k+1})^2 + (a_{jl}^{k+1})^2 = (a_{il}^k)^2 + (a_{ij}^k)^2, \quad l \neq i, l \neq j.$$

При $l = i$ и $l = j$ у нас будут два равенства

$$(a_{ii}^{k+1})^2 + (a_{ji}^{k+1})^2 = (a_{ii}^k)^2 + (a_{ji}^k)^2,$$

$$(a_{ij}^{k+1})^2 + (a_{jj}^{k+1})^2 = (a_{ij}^k)^2 + (a_{jj}^k)^2.$$

Теперь мы учтем тот факт, что при построении A^{k+1} мы выбирали угол φ таким образом, что $a_{ij}^{k+1} = a_{ji}^{k+1}$. В итоге из этих двух равенств у нас получится

$$(a_{ii}^{k+1})^2 + (a_{jj}^{k+1})^2 = (a_{ii}^k)^2 + (a_{jj}^k)^2 + 2(a_{ij}^k)^2.$$

Из этого равенства следует, что изменение от итерации к итерации определяется величиной $2(a_{ij}^k)^2$. Тогда мы можем записать следующее соотношение

$$t(A^{k+1}) = \sum_{l=1}^n \sum_{m=1, m \neq l}^n (a_{lm}^{k+1})^2 = \sum_{l=1}^n \sum_{m=1, m \neq l}^n (a_{lm}^k)^2 - 2(a_{ij}^k)^2.$$

Это равенстве есть ничто иное как

$$t(A^{k+1}) = t(A^k) - 2(a_{ij}^k)^2 \tag{12}$$

Так как a_{ij}^k есть наибольший по модулю недиагональный элемент, он отличен от нуля. Тогда

$$t(A^{k+1}) < t(A^k).$$

Таким образом, мы доказали сходимость итерационного процесса. Теперь покажем скорость стремления к нулю:

$$t(A^k) \leq n(n-1)(a_{ij}^k)^2.$$

Отсюда

$$(a_{ij}^k)^2 \geq \frac{1}{n(n-1)} t(A^k).$$

Используем формулу (12)

$$t(A^{k+1}) \leq t(A^k) - \frac{2}{n(n-1)}t(A^k) = qt(A^k), \quad q = 1 - \frac{2}{n(n-1)}.$$

В случае, если $n \geq 2$, то $0 \leq q \leq 1$. Используя последнее неравенство, можно записать

$$t(A^k) \leq qt(A^{k-1}) \leq \dots \leq q^k t(A^0).$$

Учитывая то, что $A^0 = A$, окончательно получим оценку

$$t(A^k) \leq qt(A) \tag{13}$$

Из (13) следует, что $t(A^k) \xrightarrow[k \rightarrow \infty]{} 0$, то есть процесс этот сходящийся и скорость сходимости равна скорости сходимости геометрической прогрессии со знаменателем $q < 1$.

Замечания.

1. Необходимо выполнить $\frac{n(n-1)}{2} \sim O(n^2)$ сравнений. Можно уменьшить объем вычислений, если взять наибольшую из сумм (3) $\sigma_i(A^k)$. В качестве a_{ij}^k можно брать максимальный элемент в этой строке, относящейся к σ_i . В этом случае нам понадобится $O(n)$ сравнений. При этом на следующей итерации меняются только величины

$$\sigma_i(A^{k+1}) = \sigma_i(A^k) + (a_{ii}^{k+1})^2 - (a_{ii}^k)^2 - (a_{ij}^k)^2$$

$$\text{и } \sigma_j(A^{k+1}) = \sigma_j(A^k) + \sigma_i(A^k) - \sigma_i(A^{k+1}).$$

2. Из формулы (13) можно получить априорную оценку количества итераций. Для того, чтобы получить точность ε нужно совершить k итераций, где

$$k \geq \frac{\lg \varepsilon - \lg t(A)}{\lg q}$$

3. Предельное количество арифметических операций требуемых для выполнения одного шага алгоритма $Q(n) = 2n^2 + O(n)$, $n \rightarrow \infty$.

3.5 Степенной метод.

Данный метод применяется для решения частной ПСЗ.

3.5.1 Нахождение наибольшего по модулю собственного значения и соответствующего собственного вектора.

Пусть матрица A вещественная и обладает полной системой n линейно независимых собственных векторов. Это обеспечивается в тех случаях, когда собственные значения различны между собой или матрица A симметрическая.

Обозначим собственные значения и собственные векторы $\lambda_1, \dots, \lambda_n$ и соответствующие им собственные векторы x^1, \dots, x^n . Пусть λ_i перенумерованы в порядке невозрастания модулей, то есть

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|.$$

Перед нами стоит задача нахождения наибольшего по модулю собственного значения и соответствующего ему собственного вектора.

Возьмем произвольный вектор $y^0 = (y_1^0, \dots, y_n^0)^T$ и построим рекуррентную последовательность:

$$y^0, \quad y^1 = Ay^0, \quad y^2 = Ay^1 = A^2y^0, \quad \dots, \quad y^k = Ay^{k-1} = \dots = A^ky^0, \dots$$

Разложим y^0 по системе полных векторов:

$$y^0 = \alpha_1 x^1 + \dots + \alpha_n x^n. \quad (1)$$

Выразим y^k по координатам, учитывая, что $A^k x^i = \lambda_i^k x^i$:

$$y_s^k = \beta_{1s} \lambda_1^k + \beta_{2s} \lambda_2^k + \dots + \beta_{ns} \lambda_n^k, \quad s = \overline{1, n}, \beta_{is} = \alpha_i x_s^i, i = \overline{1, n}. \quad (2)$$

Рассмотрим отношение компонент итерационной последовательности на k -ом и $k+1$ -ом шаге. То есть запишем отношение

$$\frac{y_s^{k+1}}{y_s^k} = \frac{\beta_{1s} \lambda_1^{k+1} + \beta_{2s} \lambda_2^{k+1} + \dots + \beta_{ns} \lambda_n^{k+1}}{\beta_{1s} \lambda_1^k + \beta_{2s} \lambda_2^k + \dots + \beta_{ns} \lambda_n^k} = \lambda_1 \cdot \frac{1 + \gamma_{2s} \mu_2^{k+1} + \dots + \gamma_{ns} \mu_n^{k+1}}{1 + \gamma_{2s} \mu_2^k + \dots + \gamma_{ns} \mu_n^k};$$

$$\gamma_{is} = \frac{\beta_{is}}{\beta_{1s}}, \mu_i = \frac{\lambda_i}{\lambda_1}, i = \overline{2, n}. \quad (4)$$

Очевидно, что $|\mu_i| < 1$. Тогда при $k \rightarrow \infty$ получится, что

$$\frac{y_s^{k+1}}{y_s^k} = \lambda_1 + O(|\mu_2|^k).$$

Отсюда следует, что в качестве λ_1 при достаточно больших k в пределах принятой точности можно взять величину

$$\lambda_1 \approx \frac{y_s^{k+1}}{y_s^k}. \quad (5)$$

Правая часть в формуле (5) зависит от s . Если будет одинаковое значение при всех s в пределах принятой точности, то это является некоторой гарантией достигнутой точности при достаточно больших k .

Теперь найдем соответствующий собственный вектор. Для этого используем формулу (2), учитывая тот факт, что при достаточно больших k все слагаемые начиная со второго можно отбросить. В итоге при $k \rightarrow \infty$ сохранится лишь первое слагаемое. А отсюда мы можем записать, что

$$x^1 \approx \frac{1}{\alpha_1 \lambda_1^k} y^k.$$

Понятно, что α_1 — это константа, которую мы заранее не знаем. Но учитывая тот факт, что собственный вектор находится с точностью до константы, то мы можем это заменить как

$$x^1 \approx \frac{1}{\alpha_1 \lambda_1^k} y^k = c_k^y y^k, \quad (6)$$

где c_k — произвольная константа. Обычно берут $c_k = 1$.

Видно, что скорость убывания величины $O(|\mu_2|^k)$ равна скорости убывания геометрической прогрессии с показателем меньше 1. Можно попытаться увеличить скорость сходимости, а значит уменьшить количество итераций для достижения необходимой точности.

В случае, если матрица $A = A^T$ можно увеличить скорость сходимости итерационного процесса. Поскольку для симметрической матрицы систему собственных векторов всегда можно считать ортонормированной. Это означает

$$(x_i, x_j) = \delta_{ij}, \quad i, j = \overline{1, n},$$

δ_{ij} — символ Кронекера. Используя формулу (3), можно вычислить скалярное произведение

$$(y^k, y^k) = \alpha_1^2 \lambda_1^{2k} + \alpha_2^2 \lambda_2^{2k} + \dots + \alpha_n^2 \lambda_n^{2k}.$$

Наряду с этим скалярным произведением рассмотрим

$$(y^{k+1}, y^k) = \alpha_1^2 \lambda_1^{2k+1} + \alpha_2^2 \lambda_2^{2k+1} + \dots + \alpha_n^2 \lambda_n^{2k+1}.$$

Легко видеть, что если мы составим отношение этих скалярных произведений, то есть вместо отношения компонент векторов рассмотрим отношение

$$\frac{(y^{k+1}, y^k)}{(y^k, y^k)} = \lambda_1 + O(|\mu_2|^{2k})$$

при выделении главной части. То есть на порядок увеличилась скорость. Но мы платим операциями вычисления скалярного произведения. Отсюда в качестве λ_1 можно взять

$$\lambda_1 \approx \frac{(y^{k+1}, y^k)}{(y^k, y^k)} \quad (7).$$

В данном случае в качестве собственного вектора будет значение y_k .

Замечания.

1. Понятно, что если мы будем каждый раз считать y^k , то нам надо умножать на матрицу A , то есть компоненты этого вектора будут либо возрастать, если $|\lambda_1| > 1$, либо неограниченно убывать, когда $|\lambda_1| < 1$. За счет этого получаются векторы с очень большими по модулю компонентами. Чтобы избежать чрезмерного роста в абсолютной величине координат векторов y^k (либо чрезмерного убывания), целесообразно эти векторы нормировать. При этом вместо последовательности

$$y^0, \quad y^1 = Ay^0, \quad \dots, \quad y^k = A^k y^0$$

мы будем строить последовательность

$$z^0 = y^0, \quad z^1 = \alpha_1 A y^0, \quad z^2 = \alpha_2 A^2 y^0, \quad \dots, \quad z^k = \alpha_k A^k y^0.$$

Здесь α — это нормирующий множитель. Например, можно взять

$$\alpha_k = \frac{1}{\|A^k y^0\|}.$$

Если выбрать первую норму, то у нас получится одна компонента единица, а остальные меньше единицы. Для получения приближения к собственному значению λ_1^k , $k = 1, 2, \dots$ в случае нормировки надо брать отношение координат Az^k и z^k .

2. В качестве критерия останова итерационного процесса используется условие

$$|\lambda_1^{k+1} - \lambda_1^k| \leq \varepsilon.$$

3.5.2 Нахождение наименьшего по модулю собственного значения (метод обратных итераций).

Будем предполагать, что по-прежнему

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots > |\lambda_n|.$$

Обозначим через $\nu_i, i = \overline{1, n}$ собственные значения матрицы A^{-1} . Далее, зная соотношения между матрицей и обратной к ней, получаем

$$\nu_i = \frac{1}{\lambda_i}, \quad \nu_{\max} = \nu_n = \frac{1}{\lambda_{\min}} = \frac{1}{\lambda_n}.$$

Поэтому для того, чтобы найти минимальное по модулю собственное число матрицы A , достаточно найти максимальное по модулю собственное число матрицы A^{-1} . После этого нужно применить алгоритм, описанный в пункте 5.1.

Можно организовать вычисления так, чтобы не нужно было обращать матрицу A . Для этого надо каждое y^k приближение находить из системы

$$Ay^k = y^{k-1}, \quad k = 1, 2, \dots$$

Легко видеть, что построенная таким образом итерационная последовательность будет сходиться со скоростью геометрической прогрессии со знаменателем

$$\left| \frac{\nu_{n-1}}{\nu_n} \right| = \left| \frac{\lambda_n}{\lambda_{n-1}} \right|$$

Аналогичным способом вычисляется собственный вектор, т.е. он будет равен y^k .

Если на первой итерации построить LU разложение матрицы A , то все последовательные операции могут быть реализованы за $O(n^2)$ арифметических операций. Аналогично, если $A = A^T$, то можно повысить скорость сходимости также как и в пункте 5.1. То есть вместо отношения $\frac{y^{k+1}}{y^k}$ будем рассматривать $\frac{(y^{k+1}, y^k)}{(y^k, y^k)}$, при этом скорость сходимости будет на порядок больше: $O\left(\left|\frac{\nu_{n-1}}{\nu_n}\right|^{2k}\right)$.

3.5.3 Случай кратного доминирующего собственного значения.

Известно, что есть кратный корень. Причем он является максимальным по модулю по отношению к остальным собственным значениям, то есть

$$\lambda_1 = \lambda_2 = \dots = \lambda_r$$

и

$$|\lambda_1| > |\lambda_{r+1}| \geq \dots \geq |\lambda_n|.$$

В данном случае разложение (2) примет следующий вид

$$y^k = \lambda_1^k (\alpha_1 x^1 + \dots + \alpha_r x^r) + \alpha_{r+1} \lambda_{r+1}^k x^{r+1} + \dots + \alpha_n \lambda_n^k x^n.$$

Легко видеть, что

$$y_s^k = \beta_{1s} \lambda_1^k + \beta_{r+1s} \lambda_{r+1}^k + \dots + \beta_{ns} \lambda_n^k, \quad \beta_{1s} = \alpha_1 x_s^1 + \dots + \alpha_r x_s^r, \quad \beta_{is} = \alpha_s x_s^i, \quad i = \overline{r+1, n}.$$

Так же как и в случае (5.1) мы получим

$$\frac{y_s^{k+1}}{y_s^k} = \lambda_1 + O(|\mu_{r+1}|^k), \quad \mu_{r+1} = \frac{\lambda_{r+1}}{\lambda_1}.$$

Эта формула практически приводит к формуле (5):

$$\lambda_1 \approx \frac{y_s^{k+1}}{y_s^k}.$$

Но для найденного значения λ мы получим собственный вектор, совпадающий с y^k . Но в данном случае

$$y^k = \lambda_1^k(\alpha_1 x^1 + \dots + \alpha_r x^r).$$

Для того, чтобы найти полную систему векторов, нам нужно изменить y^0 и получить другую итерационную последовательность, которая в принципе даст нам то же самое λ , но значения будут другие. Они и обеспечат нам получения новых собственных векторов. И так далее $r - 2$ раз.

3.5.4 Случай двух наибольших собственных значений, отличающихся знаками.

Пусть $\lambda_2 = -\lambda_1$. Тогда мы будем рассматривать последовательность

$$|\lambda_1| > |\lambda_3| \geq \dots \geq |\lambda_n|.$$

Тогда здесь

$$y^k = \lambda_1^k(\alpha_1 x^1 + (-1)^k \alpha_2 x^2) + \alpha_3 x^3 + \dots + \alpha_n x^n.$$

Учитывая, что знак в скобке будет все время меняться в зависимости от k , то рассмотрим четные и нечетные итерации:

$$\begin{aligned} y^{2k} &= \lambda_1^{2k}(\alpha_1 x^1 + \alpha_2 x^2) + \alpha_3 x^3 + \dots + \alpha_n x^n, \\ y^{2k+1} &= \lambda_1^{2k+1}(\alpha_1 x^1 - \alpha_2 x^2) + \alpha_3 x^3 + \dots + \alpha_n x^n. \end{aligned}$$

Если коэффициенты $\alpha_1 \neq \alpha_2$, то тогда можем рассматривать отношение четных и нечетных итераций:

$$\begin{aligned} \frac{y_s^{2k+2}}{y_s^{2k}} &= \lambda_1^2 + O(|\mu_3|^{2k}), \\ \frac{y_s^{2k+1}}{y_s^{2k-1}} &= \lambda_1^2 + O(|\mu_3|^{2k-1}). \end{aligned}$$

В данном случае при $k \rightarrow \infty$ получим формулу

$$\lambda_1 \approx \pm \sqrt{\frac{y_s^{2k+2}}{y_s^{2k}}}, \quad \lambda_2 \approx \pm \sqrt{\frac{y_s^{2k+1}}{y_s^{2k-1}}}. \quad (8)$$

Вычислим собственные векторы

$$\begin{aligned} y^{2k+1} &\approx \lambda_1^{2k+1}(\alpha_1 x^1 - \alpha_2 x^2), \\ y^{2k+2} &\approx \lambda_1^{2k+2}(\alpha_1 x^1 + \alpha_2 x^2). \end{aligned}$$

Теперь составим комбинации

$$y^{2k+2} + \lambda_1 y^{2k+1}, \quad y^{2k+2} - \lambda_1 y^{2k+1}$$

и из первой суммы мы найдем x^1 , а из второй — x^2 . Тогда формулы здесь будут

$$x^1 \approx \frac{1}{2\alpha_1 \lambda_1^{2k+2}} (y^{2k+2} + \lambda_1 y^{2k+1}), \quad (9)$$

$$x^2 \approx \frac{1}{2\alpha_2 \lambda_1^{2k+2}} (y^{2k+2} - \lambda_1 y^{2k+1}). \quad (9)$$

3.5.5 Нахождение собственного значения второго по величине

Пусть

$$|\lambda_1| > |\lambda_2| > |\lambda_3| \geq |\lambda_4| \geq \dots \geq |\lambda_n|,$$

где λ_1 известно. Рассмотрим составляющие векторы y_s^{k+1} , y_s^k , используя формулу (3). Добьемся исключения слагаемого, в котором есть λ_1 :

$$y_s^{k+1} - \lambda_1 y_s^k = \beta_{2s} \lambda_2^k (\lambda_2 - \lambda_1) + \beta_{3s} \lambda_3^k (\lambda_3 - \lambda_1) + \dots,$$

$$y_s^k - \lambda_1 y_s^{k-1} = \beta_{2s} \lambda_2^{k-1} (\lambda_2 - \lambda_1) + \beta_{3s} \lambda_3^{k-1} (\lambda_3 - \lambda_1) + \dots$$

И составим отношение этих выражений:

$$\frac{y_s^{k+1} - \lambda_1 y_s^k}{y_s^k - \lambda_1 y_s^{k-1}} = \lambda_2 \frac{1 + \gamma_{3s} \mu_3^k + \gamma_{4s} \mu_4^k + \dots}{1 + \gamma_{3s} \mu_3^{k-1} + \gamma_{4s} \mu_4^{k-1} + \dots} = \lambda_2 \cdot (1 + O(|\mu^{k-1}|)),$$

где

$$\gamma_{is} = \frac{\beta_{is}(\lambda_i - \lambda_1)}{\beta_{2s}(\lambda_2 - \lambda_1)}, \quad \mu_i = \frac{\lambda_i}{\lambda_2}, \quad i = 3, 4, \dots$$

Так как $|\mu_i| < 1$, то

$$\lambda_2 \approx \frac{y_s^{k+1} - \lambda_1 y_s^k}{y_s^k - \lambda_1 y_s^{k-1}}. \quad (10)$$

Замечания.

1. Теоретически степенным методом можно найти любое собственное значение. Однако, чем с большей точностью найдено λ_1 , тем с меньшей точностью будет найдено λ_2 .

$$\left(\left| \frac{\lambda_3}{\lambda_2} \right| \right)^k \leq \varepsilon \left(\left| \frac{\lambda_1}{\lambda_2} \right| \right)^k.$$

2. Метод λ -разности — это пункт 5.5.

3.6 Методы построения собственного многочлена основанные на использовании формулы Ньютона.

3.6.1 Метод Леверье.

Рассмотрим выражение

$$kp_k = S_k - p_1 S_{k-1} - p_2 S_{k-2} - \dots - p_{k-1} S_1, \quad k = \overline{1, n}, \quad (1)$$

[illegible]
$$S_k = \mathrm{Sp} A^k = \sum_{i=1}^n a_{ii}^k.$$
$$S_k = \sum_{i=1}^n a_{ii}^k, \quad k = \overline{1, n}.$$

- Сложность алгоритма

- Количество операций умножения равно $\frac{1}{2}(n-1)(2n^3 - 2n^2 + n + 2)$.
- Можно останавливать перемножение матриц, когда найдены все диагональные элементы. Вычислим A^2, A^3, \dots, A^m , где $m = \left\lfloor \frac{n+1}{2} \right\rfloor$, а S^{m+1}, \dots, A^n можно находить, минуя вычисление недиагональных элементов.

$$a_{ii}^p = \sum_{k=1}^n a_{ik}^s a_{ks}^{p-s}, \quad p < m, \quad s \leq m, \quad p - s \leq m.$$

[illegible]

61

Теорема. Для формулы (3) выполняется следующее:

1. $q_i = p_i$, где p_i — это коэффициенты собственного многочлена;
2. $B_n = 0$;
3. $A^{-1} = \frac{1}{p_n} B_{n-1}$.

◆

1. Докажем первое утверждение по индукции. При $i = 1$ имеем $p_1 = \text{Sp } A = \text{Sp } A_1 = q_1$.

Пусть $q_i = p_i$, $i = \overline{1, k-1}$. Покажем, что $p_k = q_k$. Согласно алгоритму построения по формуле (3), имеем

$$\begin{aligned} A_k &= A^k - q_1 A^{k-1} - \dots - q_{k-1} A, \\ A_k &= A^k - p_1 A^{k-1} - \dots - q_{k-1} A, \end{aligned} \quad (4)$$

где p — коэффициент собственного многочлена. Тогда

$$\begin{aligned} kq_k &= \text{Sp } A_k = \text{Sp } A^k - p_1 \text{Sp } A^{k-1} - \dots - p_{k-1} \text{Sp } A = [\text{из метода Леверье}] = \\ &= S_k - p_1 S_{k-1} - \dots - p_{k-1} S_1 = [\text{по формуле (1)}] = kp_k. \end{aligned}$$

То есть

$$q_k = p_k.$$

2. Докажем второе утверждение. Используя формулы (3) и (4), можно вычислить

$$B_n = A_n - q_n E = \underbrace{A^n - p_1 A^{n-1} - \dots - p_{n-1} A}_{P_n(A)} = 0.$$

3. Докажем третье утверждение. С учетом доказанного из первого и второго утверждения, по построению имеем

$$A_n = AB_{n-1} \Rightarrow p_n E = AB_{n-1} \Rightarrow E = \frac{1}{p_n} AB_{n-1} \Rightarrow A^{-1} = \frac{1}{p_n} B_{n-1}.$$

□

Обозначим

$$Q(\lambda) = \lambda^{n-1} E + \lambda^{n-2} B_1 + \lambda^{n-3} B_2 + \dots + \lambda B_{n-2} + B_{n-1}.$$

Если все $\lambda_i(A)$ различны, то $Q(\lambda_i) \neq 0$, $i = \overline{1, n}$ и любой ненулевой столбец может быть выбран в качестве собственного вектора. Докажем это утверждение:

$$\begin{aligned} (\lambda_i E - A)Q(\lambda_i) &= \lambda^{n-1} E + \lambda^{n-2} B_1 + \lambda^{n-3} B_2 + \dots + B_{n-1} = [\text{нужно показать, что } (\lambda_i E)Q(\lambda_i) = 0] = \\ &= (\lambda_i E - A)(\lambda^{n-1} E + \lambda^{n-2} B_1 + \lambda^{n-3} B_2 + \dots + B_{n-1}) = \\ &= \lambda_i^n E + \lambda_i^{n-1} (B_1 - A) + \lambda_i^{n-2} (B_2 - AB_1) + \dots + \lambda_i (B_{n-1} - AB_{n-2}) - AB_{n-1} = \\ &= [B_k - AB_{k-1} = q_k E = -p_k E, \quad k = \overline{1, n}] = (\lambda_i^n - p_1 \lambda_i^{n-1} - \dots - p_n) E = 0. \end{aligned}$$

Тогда $(\lambda_i E - A)X = 0$, где X — любой столбец матрицы $Q(\lambda)$.

Замечания.

1. Число операций

$$Q(n) \approx (n-1)n^3.$$

2. Нет необходимости строить всю матрицу $Q(\lambda)$, достаточно одного столбца.
3. В случае кратных собственных значений нужно рассматривать матрицы, полученные из Q дифференцированием по переменной λ .

отражений для СЛАУ, мы начнем преобразовывать матрицу, начиная с первого столбца. Все отличие будет являться в том, что мы должны подобрать матрицу V так, чтобы занулить не все элементы ниже главной диагонали, а только элементы ниже поддиагонали, а также умножать будем и с другой стороны для подобия преобразования.

Предположим, что процесс Хессенберга сделан до $k - 1$ столбца. Опишем очередной шаг алгоритма. Разобьем матрицу на клетки следующим образом

$$A = \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array} \right)$$

где A_1 — это матрица Хессенберга размерности $k \times k$, A_3 — прямоугольная матрица, у которой только последний столбец отличен от нуля, а A_2 и A_4 — матриц соответствующих размерностей, которые изменились в процессе $k - 1$ -го шага процесса Хессенберга. Для того, чтобы применить преобразование этой матрицы, подберем матрицу отражений так, чтобы занулить в k -ом столбце необходимое нам количество элементов. Для построения матрицы $V_k = E - 2\omega^k(\omega^k)^T$ нам нужно найти вектор ω^k . Для этого мы можем разбить матрицу V_k на клетки того же размера, что и матрицу A :

$$V = \left(\begin{array}{c|c} E & 0 \\ \hline 0 & W \end{array} \right).$$

Таким образом, искомое преобразование подобия имеет вид

$$B = V_k^{-1} A V_k = \left(\begin{array}{c|c} E & 0 \\ \hline 0 & W \end{array} \right) \cdot \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array} \right) \cdot \left(\begin{array}{c|c} E & 0 \\ \hline 0 & W \end{array} \right) = \left(\begin{array}{c|c} A_1 & A_2 W \\ \hline W A_3 & W A_4 W \end{array} \right). \quad (4)$$

Чтобы занулить элементы последнего столбца матрицы $W A_3$ необходимо взять векторы

$$s = (0, \dots, 0, a_{k+1k}, \dots, a_{nk})^T,$$

$$e = (1, 0, \dots, 0)^T.$$

И применить формулы (3), (4) и параграфа 2.7. Тогда для определения матрицы отражения V^k , можно получить формулы для определения вектора ω :

$$\left\{ \begin{array}{l} \alpha = \alpha_k = b_{k+1k} = \left(\sum_{i=k+1}^n a_{ik}^2 \right)^{\frac{1}{2}}, \\ (s - \alpha e)_i = \begin{cases} a_{k+1k} - \alpha_k, & i = k + 1, \\ a_{ik}, & i = \overline{k + 2, n} \end{cases}, \\ (s, s - \alpha e) = b_{k+1k}(b_{k+1k} - a_{k+1k}), \\ \kappa = \frac{1}{\sqrt{2(s, s - \alpha e)}}, \\ \omega_i = \begin{cases} \kappa(a_{k+1k} - \alpha_k), & i = k + 1, \\ \kappa a_{ik}, & i = \overline{k + 2, n}. \end{cases} \end{array} \right. \quad (5)$$

Последовательно полагаем $k = 1, 2, \dots, n - 2$, определяем ω^k и производя умножения приводим матрицу A к верхней форме Хессенберга.

В заключение этого пункта отметим, что для экономичной организации процесса следует иметь ввиду, что клетка A_1 в формуле (4) не изменяется, в клетке $W A_3$ имеется только один ненулевой элемент — b_{k+1k} . Умножения в остальных двух клетках можно заменить умножение матрицы на матрицу умножением матрицы на вектор:

$$A_2 W = A_2 (E - 2\omega\omega^T) = A_2 - 2(A_2\omega)\omega^T.$$

3.7.2 Преобразование матрицы по методу вращений.

Одно подобное преобразование вращения состоит из двух умножений. Сначала мы рассматриваем матрицу $B = AT_{ij}$, а потом домножим слева на T_{ij}^T . Столбцы матрицы B совпадают со столбцами матрицы A за исключением двух столбцов с номерами i и j . Формально запишем это как

$$\begin{cases} B_i = \cos \varphi A_i + \sin \varphi A_j, \\ B_j = -\sin \varphi A_i + \cos \varphi A_j \end{cases} \quad (6)$$

где A_i — это элементы i -го столбца матрицы A , т.е. A_i, A_j, B_i, B_j — столбцы матриц A и B соответственно. Рассмотрим матрицу $C = T_{ij}^T B$. При умножении слева, в матрицу B поменяются i -ая и j -ая строка. Выпишем формулу

$$\begin{cases} C^i = \cos \varphi B^i + \sin \varphi B^j, \\ C^j = -\sin \varphi B^i + \cos \varphi B^j. \end{cases}$$

где C^i, C^j, B^i, B^j — строки i и j соответствующих матриц. Угол поворота φ выбирается так, чтобы обнулить один элемент c_{ji-1} . Для этого достаточно выбрать

$$\begin{cases} \cos \varphi = \frac{a_{ii-1}}{\sqrt{a_{ii-1}^2 + a_{ji-1}^2}}, \\ \sin \varphi = \frac{a_{ji-1}}{\sqrt{a_{ii-1}^2 + a_{ji-1}^2}}. \end{cases}$$

Таким образом, алгоритм метода вращений может быть сформулирован следующим образом.

Первый этап. На первом этапе зануляем элементы первого столбца, начиная с третьего элемента, то есть обнуляем элементы $a_{31}, a_{41}, \dots, a_{n1}$. Для этого используем матрицы $T_{23}, T_{24}, \dots, T_{2n}$. При этом ранее уничтоженные элементы первого столбца с позиции $(i, 1)$ при дальнейшем преобразовании T_{2j} ($j > i$) меняться не будут.

Второй этап. С помощью подобных преобразований $T_{34}, T_{35}, \dots, T_{3n}$ уничтожаются элементы второго столбца, начиная с четвертого. При этом использование этих матриц не изменяет нули первого столбца получившейся матрицы, хотя первые два элемента могут поменяться, то есть $(i, 1), i = \overline{3, n}$.

И так далее. Необходимо сделать $n - 2$ таких этапа. Прделав $\frac{(n-1)(n-2)}{2}$ таких преобразования, мы приведем матрицу к форме Хессенберга.

3.8 QR-метод (алгоритм).

Любую матрицу можно представить в виде произведения ортогональной и верхнетреугольной матрицы

$$A = QR. \quad (1)$$

Построим последовательность матриц

$$A^0, A^1, \dots, A^k, \dots$$

по следующему правилу:

$$A^{k+1} = R^{k+1}Q^{k+1}, \quad \text{где } Q^{k+1}R^{k+1} = A^k, \quad k = 0, 1, \dots, \quad A^0 = A. \quad (2)$$

Таким образом берем матрицу A^k , представляем ее в виде QR -разложения, а затем получаем новую матрицу, производя перемножение, поменяв местами матрицы разложения. С каждой итерацией мы фактически изменяем матрицу A . Мы должны убедиться в том, что эти преобразования не изменяют собственный многочлен матрицы, то есть являются подобными. Это легко показать:

$$A^{k+1} = R^{k+1}Q^{k+1} = (Q^{k+1})^{-1}Q^{k+1}R^{k+1}Q^{k+1} = (Q^{k+1})^{-1}A^kQ^{k+1},$$

то есть преобразование подобное. Таким образом, все матрицы A^k построенные по правилу (2) подобны между собой и подобны исходной матрице A .

• Будем называть последовательность матриц A^1, \dots, A^k, \dots **последовательностью QR -алгоритма**.

Теорема (о QR -алгоритме). Пусть невырожденная матрица A имеет различные вещественные собственные значения

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|, \quad \lambda_i \in \mathbb{R}, \quad i = \overline{1, n}$$

и пусть также существует LU -разложение матрицы подобия

$$Q^{-1} = LU,$$

причем

$$A = Q\Lambda Q^{-1}, \quad \Lambda = \text{diag}\{\lambda_1, \dots, \lambda_n\}.$$

Тогда последовательность A^k QR -алгоритма сходится к верхнетреугольной матрице, диагональные элементы которой являются ее собственными значениями.

Одним из критериев остановки метода является следующий: если у нас есть точность ε , то мы проверяем условие, когда

$$|a_{ij}^k| < \varepsilon, \quad i > j.$$

Вторым критерием может быть случай, когда все λ на главной диагонали сойдутся.

Изложенный алгоритм носит название **QR -алгоритм в ортодоксальной форме**.

Усовершенствование алгоритма возможно по двум критериям:

1. уменьшение трудоемкости (количества операций);
2. улучшение скорости сходимости итерационного процесса.

Замечания.

1. Сформулированные в теореме условия сходимости могут быть ослаблены. Простейшее из таких ослаблений состоит в том, что в условии (3) достаточно только одного строгого неравенства.

2. Уменьшение объема работы алгоритма на одну итерацию возможно путем преобразования матрицы A к форме Хессенберга. При этом для получения QR -разложения достаточно порядка $O(n^2)$ арифметических операций взамен $O(n^3)$, если не приводить матрицу к форме Хессенберга.
3. Уменьшение количества итераций QR -алгоритма возможно путем ускорения сходимости метода за счет организации, так называемых, сдвигов. Отсюда будет название **QR -алгоритм со сдвигом**.

Вместо формулы (2) итерационный алгоритм для построения последовательностей задается следующим образом:

$$A^0 = A, \quad A^k - \mu E = Q^{k+1} R^{k+1},$$

где μ — специальным образом подобранный скалярный параметр. Тогда

$$A^{k+1} = R^{k+1} Q^{k+1} + \mu E, \quad k = 0, 1, \dots \quad (4)$$

Увеличение скорости сходимости итераций достигается за счет подбора параметра сдвига μ . Легко показать, что формула (4) позволяет построить последовательность матриц A^k , которая сохраняет спектр.

Вариант QR -алгоритма со сдвигом.

1. Сначала мы делаем некое количество m_0 итераций ортодоксального QR -алгоритма.
2. Затем берут $\mu = a_{nn}^{m_0}$.
3. Следующие m_1 этапов организуем со сдвигом по формулам (4). В результате мы получим матрицу $A^{m_0+m_1}$, у этой матрицы будет достаточно хорошее приближение к $\lambda_n = a_{nn}^{m_0+m_1} = \lambda_{\min}$. То есть мы нашли одно собственное значение.
4. Дальше мы продолжаем ортодоксальный QR -алгоритм к матрице размера $(n-1) \times (n-1)$, которая получена из матрицы $A^{m_0+m_1}$ вычеркиванием последней строки и последнего столбца.

В заключение отметим, что сходимость итерационного процесса определяется отношением

$$\frac{|\lambda_2|}{|\lambda_1|} = q.$$

Можно построить видоизменение QR -алгоритма, позволяющее добиться кубической скорости сходимости. Он называется **QR -алгоритмом с мультисдвигом Рэлея**.