

Саратовский государственный университет
Механико-математический факультет
Кафедра математической физики
и вычислительной математики
Учебная лаборатория вычислительных методов

СН. Кабанов, Д.С. Лукомский,
С.И. Поликарпов, А.Ю. Трынин, В.А.Юрко

Численные методы

Линейная алгебра и нелинейные уравнения

Учебное пособие для студентов
механико-математического и физического факультетов

2013

УДК 519.61(075.8)

С.Н. Кабанов, Д.С. Лукомский,
С.И. Поликарпов, А.Ю. Трынин, В.А.Юрко

Численные методы. Линейная алгебра и нелинейные уравнения: Учебное пособие для студентов механико-математического и физического факультетов . - Саратов:

В пособии в доступной для первоначального ознакомления с предметом форме излагаются сложившиеся в настоящее время методы численного решения важнейших математических задач, содержатся методы решения систем линейных алгебраических уравнений, нелинейных уравнений, методы поиска экстремума и вычислений собственных значений и собственных векторов матрицы.

Для студентов механико-математического и физического факультетов в качестве руководства при изучении курса «Численные методы».

Рекомендуют к печати:

кафедра математической физики и вычислительной математики
Саратовского госуниверситета,
кандидат физико-математических наук В.П. Складов

(С) С.Н. Кабанов, Д.С. Лукомский,
С.И. Поликарпов, А.Ю. Трынин, В.А.Юрко

2013

Далее процесс повторяется. На s -том шаге ($s=1, \dots, m$) в предположении $a_{ss}^{(s-1)} \neq 0$ вычисляем

$$a_{sj}^{(s)} = a_{sj}^{(s-1)} / a_{ss}^{(s-1)}, \quad j = \overline{s, m+1}; \quad (1.2)$$

$$a_{ij}^{(s)} = a_{ij}^{(s-1)} - a_{sj}^{(s-1)} a_{is}^{(s-1)} / a_{ss}^{(s-1)}, \quad i = \overline{s+1, m}; \quad j = \overline{s+1, m+1}.$$

В итоге после выполнения m шагов получаем систему с треугольной матрицей

$$\left. \begin{aligned} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \dots + a_{1, m-1}^{(1)} x_{m-1} + a_{1m}^{(1)} x_m &= a_{1, m+1}^{(1)} \\ x_2 + a_{23}^{(2)} x_3 + \dots + a_{2, m-1}^{(2)} x_{m-1} + a_{2m}^{(2)} x_m &= a_{2, m+1}^{(2)} \\ \dots & \\ x_{m-1} + a_{m-1, m}^{(m-1)} x_m &= a_{m-1, m+1}^{(m-1)} \\ x_m &= a_{m, m+1}^{(m)} \end{aligned} \right\} \quad (1.3)$$

или в матричной форме

$$\tilde{A}x = \tilde{b},$$

где

$$\tilde{A} = \begin{bmatrix} 1 & \tilde{a}_{12} & \tilde{a}_{13} & \dots & \tilde{a}_{1, m-1} & \tilde{a}_{1m} \\ & 1 & \tilde{a}_{23} & \dots & \tilde{a}_{2, m-1} & \tilde{a}_{2m} \\ & & \dots & & & \\ & & & & 1 & \tilde{a}_{m-1, m} \\ & & & & & 1 \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} \tilde{a}_{1, m+1} \\ \tilde{a}_{2, m+1} \\ \dots \\ \tilde{a}_{m-1, m+1} \\ \tilde{a}_{m, m+1} \end{bmatrix}, \quad \tilde{a}_{ij} = a_{ij}^{(i)}. \quad (1.4)$$

Мы выполнили так называемый “прямой ход” метода Гаусса. Для его реализации нужно, чтобы $a_{11}^{(0)} a_{22}^{(1)} \dots a_{mm}^{(m-1)} \neq 0$. Попутно в “прямом ходе” мы вычислили определитель матрицы A :

$$\det A = a_{11}^{(0)} a_{22}^{(1)} \dots a_{mm}^{(m-1)}. \quad (1.5)$$

Используя соотношение (1.4), последовательно находим x_i , $i=m, m-1, \dots, 1$ по рекуррентной формуле:

$$x_i = \tilde{a}_{i, m+1} - \sum_{j=i+1}^m \tilde{a}_{ij} x_j, \quad (1.6)$$

Это так называемый “обратный ход” метода Гаусса.

Этот метод позволяет вычислить обратную матрицу A^{-1} . Для этого нужно параллельно решить m СЛАУ вида (1.1) с одной и той же матрицей A , но с различными правыми частями

$$a_{i, m+1} = \begin{cases} 1 & \text{при } i = k \\ 0 & \text{при } i \neq k \end{cases} \quad k=1, 2, \dots, m.$$

Факторизация. “Прямой ход” метода Гаусса фактически заключается в представлении матрицы A и вектора b в виде

$$A = \tilde{C}\tilde{A}, \quad b = \tilde{C}\tilde{b}, \quad (1.7)$$

где \tilde{A}, \tilde{b} имеют вид (1.4), а

$$\tilde{C} = \begin{bmatrix} \tilde{c}_{11} & & & & \\ \tilde{c}_{21} & \tilde{c}_{22} & & & \\ \tilde{c}_{31} & \tilde{c}_{32} & \tilde{c}_{33} & & \\ \dots & \dots & \dots & \dots & \\ \tilde{c}_{m1} & \tilde{c}_{m2} & \tilde{c}_{m3} & \dots & \tilde{c}_{mm} \end{bmatrix}.$$

Ясно, что $\tilde{c}_{ss} = a_{ss}^{(s-1)}$. Записывая (1.7) в координатах, получим:

$$\left. \begin{aligned} \tilde{c}_{ik} &= a_{ik} - \sum_{j=1}^{k-1} \tilde{c}_{ij} \tilde{a}_{jk}, & k \leq i \\ \tilde{a}_{ik} &= \frac{1}{\tilde{c}_{ii}} (a_{ik} - \sum_{j=1}^{i-1} \tilde{c}_{ij} \tilde{a}_{jk}), & i < k \end{aligned} \right\} \quad (1.8)$$

Отсюда последовательно при $i=1, 2, \dots, m$ вычисляем $\tilde{c}_{ik}, \tilde{a}_{ik}$.

Метод Гаусса с выбором главного элемента. Недостатком вышеуказанного алгоритма является его неприменимость в случае, если $a_{ss}^{(s-1)} = 0$ при некотором s , или потеря точности, если $a_{ss}^{(s-1)}$ “мало”. Чтобы избежать этого недостатка, немного видоизменим метод Гаусса.

Прежде чем выполнить первый шаг метода Гаусса, выберем максимальный по модулю элемент $a_{i_0 j_0}^{(0)}$, т.е. $|a_{i_0 j_0}^{(0)}| = \max_{1 \leq i, j \leq m} |a_{ij}^{(0)}|$ и

поставим его на место $a_{11}^{(0)}$ путем перестановки строк с номерами i, i_0 и столбцов с номерами j, j_0 . Аналогично на s -м шаге выбираем элемент $a_{i_{s-1} j_{s-1}}^{(s-1)}$

такой, что $|a_{i_{s-1} j_{s-1}}^{(s-1)}| = \max_{s \leq i, j \leq m} |a_{ij}^{(s-1)}|$ и ставим его на место $a_{ss}^{(s-1)}$. Если

$\det A \neq 0$, то в таком алгоритме всегда $a_{ss}^{(s-1)} \neq 0, s = \overline{1, m}$. Часто выбор “главного” элемента упрощают, выбирая максимальный элемент только по столбцу, что приводит лишь к перестановке уравнений системы.

1.2 Метод квадратного корня

Рассмотрим СЛАУ (1.1) в частном случае, когда A – вещественная симметричная матрица, т.е. $A = A^T$ (T - знак транспонирования). В этом случае для решения СЛАУ можно применять более экономичный, чем метод Гаусса, метод квадратного корня. Он особенно эффективен, если $A > 0$ - положительно определенная матрица.

Симметричную вещественную матрицу A можно представить в виде

$$A = S^T S, \quad (1.9)$$

где

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ & s_{22} & \dots & s_{2m} \\ & & \dots & \\ & & & s_{mm} \end{bmatrix}.$$

Записывая равенство (1.9) в координатах, получим:

$$a_{ij} = \sum_{k=1}^i s_{ki} s_{kj}, \quad i \leq j,$$

и, следовательно,

$$s_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} s_{ki}^2}, \quad s_{ij} = \frac{1}{s_{ii}} (a_{ij} - \sum_{k=1}^{i-1} s_{ki} s_{kj}), \quad j > i. \quad (1.10)$$

Отсюда последовательно при $i=1, 2, \dots$ вычисляем s_{ij} . Отметим, что хотя a_{ij} вещественны, числа s_{ij} могут быть комплексными. Но если $A > 0$, то s_{ij} вещественны, $s_{ii} > 0$.

В силу равенства (1.9) СЛАУ (1.1) принимает вид $S^T S x = b$. Таким образом, если матрица S вычислена, то для нахождения вектора x достаточно решить две СЛАУ с треугольными матрицами

$$S^T y = b, \quad S x = y.$$

1.3. Метод прогонки

Рассмотрим СЛАУ следующего вида

$$\beta_i x_{i-1} + \alpha_i x_i + \gamma_i x_{i+1} = b_i, \quad i = \overline{1, m}; \quad x_0 = x_{m+1} = 0, \quad (1.11)$$

где $|\alpha_i| > |\gamma_i|$, $|\alpha_i| \geq |\beta_i| + |\gamma_i|$. Такие системы часто встречаются в приложениях. СЛАУ (1.11) - частный случай системы (1.1), когда матрица A является трехдиагональной: $a_{ij} = 0$ при $|i - j| > 1$.

При этом

$$\alpha_i = a_{ii}, \quad \gamma_i = a_{i,i+1}, \quad \beta_i = a_{i,i-1}, \quad b_i = a_{i,m+1}.$$

Решение (1.11) будем искать в виде

$$x_i = \tilde{b}_i - \tilde{a}_i x_{i+1}, \quad x_{m+1} = 0. \quad (1.12)$$

Подставляя выражение (1.12) в (1.11), получаем:

$$\beta_i (\tilde{b}_{i-1} - \tilde{a}_{i-1} x_i) + \alpha_i x_i + \gamma_i x_{i+1} = b_i$$

или

$$x_i = \frac{b_i - \beta_i \tilde{b}_{i-1}}{\alpha_i - \beta_i \tilde{a}_{i-1}} - \frac{\gamma_i}{\alpha_i - \beta_i \tilde{a}_{i-1}} x_{i+1}.$$

Сравнивая с выражением (1.12), получаем рекуррентные формулы:

$$\tilde{a}_i = \frac{\gamma_i}{\alpha_i - \beta_i \tilde{a}_{i-1}}, \quad \tilde{b}_i = \frac{b_i - \beta_i \tilde{b}_{i-1}}{\alpha_i - \beta_i \tilde{a}_{i-1}}, \quad i = \overline{1, m}; \quad \tilde{a}_0 = \tilde{b}_0 = 0, \quad (1.13)$$

причем $|\tilde{a}_i| < 1$, $\alpha_i - \beta_i \tilde{a}_{i-1} \neq 0$. Таким образом, имеем следующий алгоритм решения СЛАУ (1.11):

1. Прямая прогонка. Вычисляем $\tilde{a}_i, \tilde{b}_i, i = \overline{1, m}$ по рекуррентным формулам (1.13).

2. Обратная прогонка. Вычисляем $x_i, i = m, m-1, \dots, 1$ по рекуррентным формулам (1.12).

Реализация этого алгоритма называется методом прогонки. Отметим, что метод прогонки может быть получен и непосредственно из метода Гаусса. Так, формулы (1.13), (1.12) являются следствием формул (1.8), (1.6), в которых $\tilde{c}_{ij} = 0$ при $i > j+1$; $\tilde{a}_{ij} = 0$ при $j > i+1$; $\tilde{a}_i = \tilde{a}_{i+1}, \tilde{b}_i = \tilde{a}_{i+1}, \tilde{c}_{ii} = \alpha_i - \beta_i \tilde{a}_{i-1}$.

1.4. Метод простой итерации

Перейдем теперь к итерационным методам решения СЛАУ. Пусть $\|x\|$ обозначает какую-нибудь норму вектора $x = [x_i]_{i=\overline{1, m}} \in \mathbb{R}^m$, а $\|A\| = \sup_{\|x\|=1} \|Ax\|$ - соответствующая норма матрицы A . Наиболее употребительными являются следующие нормы :

$$\begin{aligned} \|x\|_\infty &= \max_{1 \leq j \leq m} |x_j|, & \|A\|_\infty &= \max_{1 \leq i \leq m} \left(\sum_{j=1}^m |a_{ij}| \right), \\ \|x\|_1 &= \sum_{j=1}^m |x_j|, & \|A\|_1 &= \max_{1 \leq j \leq m} \left(\sum_{i=1}^m |a_{ij}| \right), \\ \|x\|_2 &= \sqrt{\sum_{j=1}^m |x_j|^2}, & \|A\|_2 &\leq \sqrt{\sum_{i,j=1}^m |a_{ij}|^2}. \end{aligned}$$

Для применения метода простой итерации следует сначала преобразовать СЛАУ $Ax=b$ к виду

$$x = Bx + d, \quad B = [b_{ij}]_{i,j=\overline{1, m}}, \quad d = [d_i]_{i=\overline{1, m}}, \quad (1.14)$$

где $\|B\| < 1$. Приведение (1.1) к виду (1.14) - самостоятельная задача, решаемая каждый раз с учетом специфики системы. Например, если удастся представить матрицу A в виде $A = P + Q$, где Q - "мала", а P^{-1} легко вычисляется, то система $Ax=b$ равносильна системе (1.14), где $B = -P^{-1}Q, d = P^{-1}b$.

Пример. Пусть

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^m |a_{ij}|,$$

т.е. диагональные элементы преобладают. Разделим i -е уравнение (1.1) на a_{ii} и получим СЛАУ вида (1.14)

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^m \left(-\frac{a_{ij}}{a_{ii}}\right)x_j + \frac{a_{im+1}}{a_{ii}}.$$

Это соответствует случаю $P = \text{diag} [a_{ii}]_{i=1, \dots, m}$.

Метод простой итерации заключается в построении последовательности векторов $x^k = [x_i^k]_{i=1, \dots, m}$, $k=0, 1, 2, \dots$ по рекуррентной формуле

$$x^{k+1} = Bx^k + d \quad (1.15)$$

или в координатах

$$x_i^{k+1} = \sum_{j=1}^m b_{ij} x_j^k + d_i. \quad (1.16)$$

В качестве начального приближения можно брать любой вектор x^0 , но, как правило, берут $x^0 = b$.

Если $\|B\| < 1$, то СЛАУ (1.14) имеет единственное решение x^* , причем

$$\|x^k - x^*\| \leq \frac{\|B\|^k}{1 - \|B\|} \|x^1 - x^0\| \rightarrow 0 \text{ при } k \rightarrow \infty.$$

Таким образом, последовательность x^k сходится при $k \rightarrow \infty$ к решению x^* со скоростью геометрической прогрессии. По достижении заданной точности вычисления прекращаются.

Модификацией метода простой итерации является метод Зейделя, в котором вычисления ведутся не по (1.16), а по формуле

$$x_j^{k+1} = \sum_{j=1}^{i-1} b_{ij} x_j^{k+1} + \sum_{j=1}^m b_{ij} x_j^k + d_i,$$

т.е. в методе Зейделя при вычислении очередной координаты вектора x^{k+1} используются только что полученные координаты $x_1^{k+1}, \dots, x_{i-1}^{k+1}$.

1.5 Метод наискорейшего спуска

Пусть $A = A^T > 0$ - вещественная, симметричная, положительно определенная матрица. Пусть x^* - решение СЛАУ (1.1). Рассмотрим квадратичную функцию

$$F(x) = \frac{1}{2} \langle Ax, x \rangle - \langle b, x \rangle, \quad (1.17)$$

где

$$\langle x, y \rangle = \sum_{i=1}^m x_i y_i$$

- скалярное произведение векторов $x, y \in \mathbb{R}^m$. Так как градиент $F'(x) = [\partial F / \partial x_i]_{i=1, \dots, m}$ имеет вид $F'(x) = Ax - b$, то решение (1.1) совпадает с точкой минимума функции $F(x)$. Таким образом, решение СЛАУ (1.1) можно заменить равносильной задачей поиска минимума квадратичной функции (1.17).

Для нахождения минимума функции $F(x)$ построим, отправляясь от некоторого начального приближения x^0 , последовательность векторов $x^k = [x_i^k]_{i=1, \dots, m}$ по рекуррентной формуле

$$x^{k+1} = x^k + \alpha_k r^k, \quad \alpha_k > 0, \quad k=0, 1, 2, \dots, \quad (1.18)$$

где $r^k = -F'(x^k)$ - направление спуска, а шаг α_k выбран из условия минимизации функции $F(x)$ вдоль направления спуска :

$$F(x^k + \alpha_k r^k) = \min_{\alpha > 0} F(x^k + \alpha r^k),$$

т.е.

$$\alpha_k = \frac{\langle r^k, r^k \rangle}{\langle r^k, Ar^k \rangle}.$$

Таким образом, вычисления ведутся по формуле

$$x^{k+1} = x^k - \frac{\langle r^k, r^k \rangle}{\langle r^k, Ar^k \rangle} F'(x^k), \quad F'(x^k) = Ax^k - b, \quad k=0, 1, 2, \dots \quad (1.19)$$

Можно доказать, что $\lim_{k \rightarrow \infty} x^k = x^*$. Вычисления прекращаются при выполнении условия $\|F'(x^k)\| < \varepsilon$. Так как в (1.19) направление спуска $r^k = -F'(x^k)$ совпадает с антиградиентом, т.е. с направлением наибольшего убывания функции в окрестности точки x^k , то метод (1.19) называется методом наискорейшего спуска. Скорость сходимости метода зависит от вытянутости линий уровня функции $F(x)$.

Приведем алгоритм, реализующий метод наискорейшего спуска.

Алгоритм А. Заданы $x^0, \varepsilon > 0$.

1. $k := 0$.
2. Сравниваем $\|F'(x^k)\| < \varepsilon$ (?). Если “да”, то переходим на п. 5.
3. Вычисляем x^{k+1} по формуле (1.19).
4. $k := k+1$ и переходим на п.2.
5. $x^* := x^k$ - решение СЛАУ (1.1).

1.6 Метод сопряженных градиентов

Метод наискорейшего спуска, несмотря на свое название, сходится довольно медленно. В этом пункте рассмотрим более эффективный метод

сопряженных градиентов, позволяющий найти минимум функции (1.17) за m шагов.

Пусть матрица $A > 0$, т.е. $\langle Ax, x \rangle > 0$ для $x \neq 0$.

Будем называть вектора x, y A - сопряженными (или A -ортогональными), если $\langle x, Ay \rangle = 0$.

Пусть s^0, s^1, \dots, s^{m-1} - система A -ортогональных ненулевых векторов. Построим векторы x^k по рекуррентной формуле

$$x^{k+1} = x^k - \frac{\langle F'(x^k), s^k \rangle}{\langle s^k, As^k \rangle} s^k, \quad k = \overline{0, m-1}. \quad (1.20)$$

Здесь $F'(x^k) = Ax^k - b$, а шаг

$$\alpha_k = \frac{\langle F'(x^k), s^k \rangle}{\langle s^k, As^k \rangle}$$

выбран из условия минимизации функции $F(x)$ вдоль направления спуска s^k . Можно доказать, что $x^m = x^*$, т.е. алгоритм (1.20) при любом начальном приближении x^0 приводит нас за m шагов в точку минимума.

Выбор A -ортогональных векторов s^0, s^1, \dots, s^{m-1} - самостоятельная задача. В зависимости от этого выбора будем получать различные алгоритмы спуска. Например, одним из наиболее популярных алгоритмов является метод Флетчера-Ривса, в котором направление спуска s^k есть линейная комбинация антиградиента и предыдущего направления спуска s^{k-1} :

$$s^0 = -F'(x^0), \quad s^k = -F'(x^k) + \beta_{k-1} s^{k-1}, \quad k \geq 1, \quad (1.21)$$

причем коэффициенты β_{k-1} выбираются так, чтобы векторы s^k, s^{k-1} были A -ортогональными:

$$\beta_{k-1} = \frac{\langle F'(x^k), F'(x^k) \rangle}{\langle F'(x^{k-1}), F'(x^{k-1}) \rangle}. \quad (1.22)$$

Таким образом, получаем следующий алгоритм решения задачи.

Алгоритм В. Задано начальное приближение x^0 .

1. $k := 0$.
2. Вычисляем вектор x^{k+1} по формулам (1.20)-(1.22).
4. $k := k+1$. Если $k < m$, то переходим на п.2 данного алгоритма.
5. $x^* := x^m$ - решение СЛАУ (1.1).

2. Нелинейные уравнения и поиск экстремума

В этом параграфе рассматриваются две тесно связанные между собой задачи.

Задача 1. Решить нелинейное уравнение в R^m

$$f(x)=0 \Leftrightarrow \begin{cases} f_1(x_1, \dots, x_m) = 0, \\ \dots\dots\dots \\ f_m(x_1, \dots, x_m) = 0, \end{cases} \quad x = \begin{bmatrix} x_1 \\ \dots \\ x_m \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ \dots \\ f_m \end{bmatrix}. \quad (2.1)$$

Задача 2. Найти минимум функции $F(x)$, $x \in R^m$.

Данные задачи могут быть сведены одна к другой. Так, задачу 1 можно заменить на поиск минимума некоторой функции, например, $F(x) = \langle f(x), f(x) \rangle$, а задачу 2 - на решение системы уравнений $F'(x) = 0$, где $F'(x) = [\partial F / \partial x_i]_{i=1, \dots, m}$ - градиент функции F . Однако решение каждой задачи имеет свою специфику и не всегда удобно заменять одну задачу другой. Поэтому будем рассматривать эти задачи по отдельности.

2.1 Метод простой итерации

Преобразуем уравнение (2.1) к виду

$$x = \varphi(x) \Leftrightarrow \begin{cases} x_1 = \varphi_1(x_1, \dots, x_m), \\ \dots\dots\dots \\ x_m = \varphi_m(x_1, \dots, x_m), \end{cases} \quad (2.2)$$

и предположим, что $\varphi(x)$ определена в замкнутой области $V \subset R^m$, причем $\varphi(x) \in V$ для всех $x \in V$. Предположим также, что при всех $x, \tilde{x} \in V$

$$\|\varphi(x) - \varphi(\tilde{x})\| \leq q \|x - \tilde{x}\|, \quad q < 1. \quad (2.3)$$

Отметим, что условие (2.3) автоматически выполняется, если $\varphi(x)$ непрерывно дифференцируема и $\|\varphi'(x)\| \leq q < 1$, где $\varphi'(x) = [\partial \varphi_i / \partial x_j]_{i,j=1, \dots, m}$.

Пусть $x^0 \in V$. Построим последовательность векторов $x^k \in V$ по рекуррентной формуле

$$x^{k+1} = \varphi(x^k) \Leftrightarrow \begin{cases} x_1^{k+1} = \varphi_1(x_1^k, \dots, x_m^k), \\ \dots\dots\dots \\ x_m^{k+1} = \varphi_m(x_1^k, \dots, x_m^k), \end{cases} \quad k=0, 1, 2, \dots \quad (2.4)$$

При выше указанных условиях уравнение (2.2) имеет единственное решение $x^* \in V$, причем $\|x^k - x^*\| < q^k \|x^0 - x^*\|$, т.е. при $k \rightarrow \infty$ последовательность x^k сходится к x^* со скоростью геометрической прогрессии.

2.2. Метод Ньютона

Пусть x^* - корень уравнения (2.1), и пусть в окрестности x^* задано начальное приближение x^0 . Пользуясь формулой Тейлора, запишем уравнение (2.1) в виде:

$$f(x) = f(x^0) + f'(x^0)(x - x^0) + o(\|x - x^0\|) = 0,$$

где $f'(x) = [\partial f_i / \partial x_j]_{i,j=1,\overline{m}}$. Отбрасывая остаточный член, получаем уравнение:

$$y=0,$$

где

$$y = f(x^0) + f'(x^0)(x - x^0). \quad (2.5)$$

Корень этого уравнения $x^1 = x^0 - (f'(x^0))^{-1}f(x^0)$ естественно принять за новое уточненное приближение к корню x^* уравнения (2.1). Таким образом, в методе Ньютона, отправляясь от начального приближения x^0 , строится последовательность векторов x^k по рекуррентной формуле:

$$x^{k+1} = x^k - (f'(x^k))^{-1}f(x^k), \quad k=0,1,2,\dots \quad (2.6)$$

Можно доказать, что если в окрестности корня x^* матрица $f'(x)$ является неособой, то при достаточно хорошем начальном приближении $\lim_{k \rightarrow \infty} x^k = x^*$, т.е. последовательность x^k сходится к корню уравнения (2.1). Вычисления прекращаются при выполнении условия $\|f(x^k)\| < \varepsilon$, где $\varepsilon > 0$ - заданное число, характеризующее точность вычислений. Приведем алгоритм, реализующий метод Ньютона.

Алгоритм С. Заданы $x^0, \varepsilon > 0$.

1. $k:=0$.
2. Сравниваем $\|f(x^k)\| < \varepsilon$ (?). Если “да”, то переходим на п. 5.
3. Вычисляем вектор x^{k+1} по формуле (2.6).
4. $k:=k+1$ и переходим на п.2.
5. $x^* := x^k$ - решение уравнения (2.1).

Недостатком метода Ньютона является необходимость вычисления матрицы $(f')^{-1}$ и чувствительность к выбору начального приближения x^0 .

Так как уравнение (2.5) - уравнение касательной к графику функции $f(x)$, то метод Ньютона в одномерном случае часто называют методом касательных.

Пример. Требуется вычислить $x = \sqrt{\alpha}$, $\alpha > 0$, т.е. найти положительный корень уравнения $x^2 - \alpha = 0$. Здесь $f(x) = x^2 - \alpha$, $f'(x) = 2x$ и формула (2.6) принимает вид

$$x^{k+1} = \frac{1}{2}(x^k + \frac{\alpha}{x^k}), \quad k=0,1,2,\dots$$

2.3 Метод дихотомии

Рассмотрим уравнение (2.1) в однородном случае ($m=1$). Пусть $f(x) \in C[a,b]$, $f(a)f(b) < 0$, и пусть на отрезке $[a,b]$ уравнение $f(x)=0$ имеет только один корень x^* . Построим последовательность сужающихся отрезков $[a_k, b_k]$, содержащих точку x^* , по следующему алгоритму.

Алгоритм D. Заданы отрезок $[a, b]$ и точность вычислений $\varepsilon > 0$.

1. $a_0 := a; b_0 := b; k := 0$. Вычисляем $f(a_0), f(b_0)$.
2. Вычисляем $c_k = (a_k + b_k)/2, f(c_k)$.
3. Если $f(c_k) = 0$ или $b_k - a_k < \varepsilon$, то $x^* := c_k$ и прекращаем счет.
4. Если $\text{sign } f(c_k) = \text{sign } f(a_k)$, то $a_{k+1} := c_k, b_{k+1} := b_k$.
Если $\text{sign } f(c_k) = \text{sign } f(b_k)$, то $a_{k+1} := a_k, b_{k+1} := c_k$.
5. $k := k + 1$ и переходим на п.2.

Таким образом, в методе дихотомии отрезок на каждом шаге делится пополам и выбирается та половина, в которой находится корень x^* . При этом

$b_{k+1} - a_{k+1} = (b_k - a_k)/2; f(a_k)f(b_k) < 0$ при всех k . Вычисления прекращаются, когда длина отрезка $[a_k, b_k]$ становится меньше заданной точности $\varepsilon > 0$.

2.4. Решение алгебраических уравнений

В этом пункте мы рассмотрим важный частный случай скалярного уравнения $f(x) = 0$, когда $f(x)$ является алгебраическим многочленом степени n с вещественными коэффициентами:

$$f(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n, \quad a_0 \neq 0. \quad (2.7)$$

Задача заключается в приближенном нахождении всех корней x_1, \dots, x_n многочлена (2.7). Отметим, что среди этих корней могут быть как вещественные, так и комплексно сопряженные. Следующие теоремы дают информацию о расположении корней.

Теорема 1. Обозначим $A = \max_{1 \leq i \leq n} |a_i|, B = \max_{0 \leq i \leq n-1} |a_i|$. Тогда

$$\frac{|a_n|}{B + |a_n|} < |x_k| < 1 + \frac{A}{|a_0|}, \quad k = 1, \dots, n. \quad (2.8)$$

Теорема 2 (Лагранжа). Пусть $a_0 > 0, a_1 \geq 0, \dots, a_{m-1} \geq 0, a_m < 0$ (т.е. a_m - первый отрицательный коэффициент). Тогда для положительных корней многочлена справедлива оценка

$$x_j < 1 + \sqrt[m]{b/a_0}, \quad b = \max_{a_i < 0} |a_i|. \quad (2.9)$$

Если $a_i \geq 0$ при всех $i = \overline{0, n}$, то положительных корней нет.

Теорема 3 (Декарта). Число положительных корней (с учетом кратности) равно числу перемен знаков в последовательности a_0, a_1, \dots, a_n (коэффициенты, равные 0, не учитываются) или меньше этого числа на четное число.

Замечание. Полагая $f^*(x) = f(-x)$ и применяя к $f^*(x)$ теоремы Лагранжа и Декарта, можно получить информацию об отрицательных корнях многочлена $f(x)$.

Теорема 4 (Гюа). Если хотя бы для одного номера $k = \overline{1, n-1}$ выполнено неравенство $a_k^2 \leq a_{k-1} a_{k+1}$, то многочлен (2.7) имеет по крайней мере одну пару комплексно сопряженных корней.

Пример. Пусть $f(x)=x^3+5x^2+4x-6$. Тогда $-f(-x) = x^3-5x^2+4x-6$. Из теоремы 1 следует, что $6/11 < |x_k| < 7$. Из теорем 2,3 получаем:

- 1) многочлен $f(x)$ имеет один положительный корень $x_1 < 1+6$;
- 2) многочлен $f(x)$ либо не имеет отрицательных корней, либо имеет два отрицательных корня, причем $-x_j < 6$. На самом деле корни многочлена суть $x_1 = 1, x_2 = -2, x_3 = -3$.

Общая схема вычисления корней многочлена

1. Используя теоремы 1 - 4, получаем информацию о расположении корней многочлена. Информацию о корнях многочлена можно получить также, используя его специфику. Например, известно, что характеристический многочлен вещественной симметричной матрицы имеет только вещественные корни (см. [29]).

2. Отделяем вещественные корни многочлена, т.е. находим интервалы, которые содержат ровно по одному вещественному корню многочлена.

3. Вычисляем приближенно вещественные корни многочлена одним из выше рассмотренных методов, например, методом дихотомии.

4. Для поиска комплексно сопряженных корней (если они есть) удобно записать многочлен в виде $f(x) = f_1(x_1, x_2) + i f_2(x_1, x_2)$, где $x_1 = \text{Re}x$, $x_2 = \text{Im}x$, $f_1 = \text{Re}f$, $f_2 = \text{Im}f$. Тогда задача поиска корней многочлена равносильна задаче поиска минимума функции $F(x_1, x_2) = f_1^2(x_1, x_2) + f_2^2(x_1, x_2)$ или решению системы $f_i(x_1, x_2) = 0, i = 1, 2$. Методы поиска экстремума, как правило, предпочтительнее. Эти методы изложены в пункте 2.5.

Если есть подозрения, что многочлен имеет кратные корни, то полезно исследовать $f'(x), f''(x), \dots$

Пример. Вычислим корни многочлена $f(x) = 6x^4 + x^3 + 4x^2 + x - 2$.

1. Используя теоремы 1-4, получаем:

а) $1/4 < |x_k| < 5/3$;

б) у многочлена есть пара сопряженных корней,

в) у многочлена есть один отрицательный корень $x_1 \in (-4/3, 0)$ и один положительный корень $x_2 \in (0, 4/5)$.

2. Методом дихотомии вычисляем $x_1 = -2/3, x_2 = 1/2$.

3. Вычисляем $f_1(x_1, x_2) = 6x_1^4 - 36x_1^2x_2^2 + 6x_2^4 - 3x_1x_2^2 + 4x_1^2 - 4x_2^2 + x_1 - 2$, $f_2(x_1, x_2) = 24x_1^3x_2 - 24x_1x_2^3 + 3x_1^2x_2 - x_2^3 - 8x_1x_2 + x_2$. Применяя методы спуска, приведенные в п.2.5, к функции $F(x_1, x_2) = f_1^2(x_1, x_2) + f_2^2(x_1, x_2)$, находим комплексно сопряженные корни многочлена $x_{3,4} = \pm i$.

2.5. Методы минимизации

Метод спуска. Рассмотрим задачу

$$F(x) \rightarrow \min, x = [x_i]_{i=1, \dots, m} \in \mathbb{R}^m$$

поиска минимума функции $F(x)=F(x_1, \dots, x_m)$. В этом пункте приведено несколько наиболее употребительных численных методов решения данной задачи. Все эти методы являются итерационными. В них, отправляясь от некоторой начальной точки x^0 , строится последовательность x^k , сходящаяся к точке минимума x^* . Последовательность x^k строится по рекуррентной формуле

$$x^{k+1} = x^k + \alpha_k r^k, \quad (2.10)$$

где $r^k = [r_i^k]_{i=1, \dots, m}$ - вектор, характеризующий направление спуска, α_k - число, характеризующее величину шага. Методы отличаются друг от друга способом выбора направления спуска. Вычисления прекращаются при выполнении условия $\|x^{k+1} - x^k\| < \varepsilon$ или $\|F'(x^k)\| < \varepsilon$, где $\varepsilon > 0$ - заданное число, характеризующее точность вычислений.

Выбор шага. Предположим, что точка x^k и направление спуска r^k заданы. Как выбрать шаг α_k ? С теоретической точки зрения наиболее выгоден выбор α_k из условия минимизации функции $F(x)$ вдоль направления спуска:

$$F(x^k + \alpha_k r^k) = \min_{\alpha} F(x^k + \alpha r^k).$$

Таким образом, выбор шага сводится к задаче поиска минимума функции одной переменной. Однако такой способ выбора шага достаточно трудоёмкий и применяется сравнительно редко. Гораздо чаще используются более простые, но и более грубые методы выбора шага. Укажем два наиболее употребительных алгоритма.

Алгоритм Е. Исходные данные: x^0 , ε , α - начальное значение шага.

1. $k:=0$; $\alpha_0=\alpha$.
2. Вычисляем $F(x^0)$.
3. Вычисляем направление спуска r^k (своё в каждом методе).
4. Вычисляем $x^{k+1} = x^k + \alpha_k r^k$, $F(x^{k+1})$, $\delta = \|x^{k+1} - x^k\|$ (или $\delta = \|F'(x^k)\|$).
5. Сравниваем $\delta < \varepsilon$ (?). Если “да”, то переходим на п.8.
6. Сравниваем $F(x^{k+1}) < F(x^k)$ (?). Если “нет”, то $\alpha_k := \alpha_k / 2$ и переходим на п.4.
7. $k:=k+1$ и переходим на п.3.
8. Конец. $x^* := x^{k+1}$ - точка минимума.

Таким образом, алгоритм Е предусматривает дробление шага при нарушении условия монотонности спуска $F(x^{k+1}) < F(x^k)$. Более эффективен следующий алгоритм F, в котором предусмотрен не только механизм дробления шага, но и механизм его увеличения, так как слишком малый шаг может значительно уменьшить скорость сходимости.

Алгоритм F. Исходные данные: x^0 , ε , α .

1. $k:=0$; $\alpha_0=\alpha$.

2. Вычисляем $F(x^0)$.
3. Вычисляем направление спуска r^k (свое в каждом методе).
4. $l_1 := 0; l_2 := 0$.
5. Вычисляем $x^{k+1} = x^k + \alpha_k r^k$, $F(x^{k+1})$, $\delta = \|x^{k+1} - x^k\|$ (или $\delta = \|F'(x^k)\|$).
6. Сравниваем $\delta < \varepsilon$ (?). Если “да”, то переходим на п.14.
7. Сравниваем $F(x^{k+1}) < F(x^k)$ (?). Если “да”, то переходим на п.8. Если “нет”, то переходим на п.11.
8. $l_1 := 1$.
9. Сравниваем $l_1 + l_2 < 2$ (?). Если “нет”, то $k := k+1$ и переходим на п.3.
10. $\alpha_k := 2\alpha_k$ и переходим на п.5.
11. $l_2 := 1$, $\alpha_k := \alpha_k / 2$.
12. Сравниваем $l_1 + l_2 < 2$ (?). Если “да”, то переходим на п.5.
13. Вычисляем $x^{k+1} = x^k + \alpha_k r^k$, $F(x^{k+1})$.
14. Конец. $x^* := x^{k+1}$ - точка минимума.

Теперь приведём основные методы выбора направления спуска.

Градиентный метод. Здесь $r^k = -F'(x^k)$, $F'(x) = [\partial F / \partial x_i]_{i=1, \dots, m}$, т.е. направление спуска совпадает с антиградиентом и является направлением наибольшего убывания функции в окрестности точки x^k . Таким образом, вычисления вектора ведутся по формуле

$$x^{k+1} = x^k - \alpha_k F'(x^k), \quad k=0,1,2,\dots \quad (2.11)$$

Достоинства градиентного метода - в его простоте и неприхотливости, недостаток - в медленной сходимости.

Метод сопряженных градиентов. Вычисления можно вести по формулам

$$x^{k+1} = x^k + \alpha_k s^k, \quad k=0,1,2,\dots \quad (2.12)$$

$$s^0 = -F'(x^0), \quad s^k = -F'(x^k) + \beta_{k-1} s^{k-1}, \quad k \geq 1, \quad (2.13)$$

$$\beta_{k-1} = \frac{\langle F'(x^k), F'(x^k) \rangle}{\langle F'(x^{k-1}), F'(x^{k-1}) \rangle}. \quad (2.14)$$

В п.1.6 показано, что для квадратичной функции метод (2.12)-(2.14) даёт минимум за конечное число шагов. Для произвольной функции $F(x)$ процесс будет уже бесконечным. Метод сопряженных градиентов обладает более высокой скоростью сходимости, чем градиентный метод, но требует более точного начального приближения.

Метод Ньютона. Вычисления ведутся по формуле

$$x^{k+1} = x^k - (F''(x^k))^{-1} F'(x^k), \quad k=0,1,2,\dots \quad (2.15)$$

где $F''(x) = [\partial^2 F / \partial x_i \partial x_j]_{i,j=1, \dots, m}$.

Таким образом, здесь $r^k = - (F''(x^k))^{-1} F'(x^k)$, $\alpha_k = 1$. Формула (2.15) получается из формулы (2.6), если положить $f(x) = F'(x)$, и поиск минимума

$F(x)$ заменить решением уравнения $f(x)=0$. Метод Ньютона обладает высокой скоростью сходимости, но очень чувствителен к выбору начального приближения. Другой недостаток - необходимость вычисления матрицы вторых производных F'' и её обращение.

Метод покоординатного спуска. В этом методе направление спуска совпадает с направлением одной из координатных осей, а именно:

$$r^{pm+s} = 1; k=pm+s; s=\overline{1,m}; p=0,1,2,\dots$$

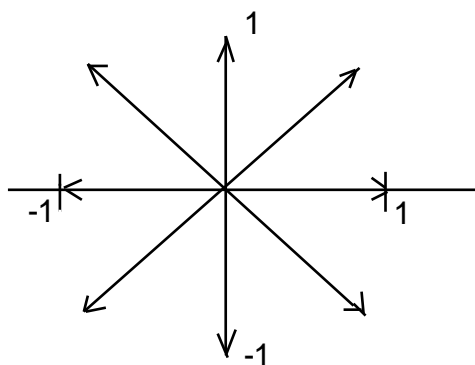
$$l_s = [l_{is}]_{i=\overline{1,m}}, \quad l_{is} = \begin{cases} \pm 1, & i = s, \\ 0, & i \neq s. \end{cases}$$

Знак “+” или “-” выбирается так, чтобы выбранное направление было направлением убывания функции $F(x)$.

Метод прямого спуска. Задаём конечный набор направлений Ω . Направление спуска r^k в (2.10) выбирается из заданного набора Ω по условию

$$F(x^k + hr^k) = \min_{\omega \in \Omega} F(x^k + h\omega),$$

где $h>0$ - заданное число. Другими словами, мы делаем пробные шаги из точки x^k с некоторым шагом h по всем направлениям $\omega \in \Omega$ и выбираем то из направлений, вдоль которого функция имеет наибольшее убывание. Обычно в качестве Ω берут множество векторов с координатами $0, \pm 1$ (см. рис.1 для случая $m=2$).



$$\Omega = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\}$$

Рис.1

Дроблением h , если необходимо, желательно добиться выполнения условия $\min_{\omega \in \Omega} F(x^k + h\omega) < F(x^k)$, т.е. чтобы хотя бы одно из направлений $\omega \in \Omega$ было направлением убывания функции $F(x)$.

Отметим, что фактически метод покоординатного спуска является частным случаем метода прямого спуска, когда на каждом шаге множество $\Omega = \Omega_k$ состоит из двух элементов (положительное и отрицательное направления соответствующей координатной оси).

Метод золотого сечения. Метод предназначен для поиска минимума функции одной переменной, т.е. для случая $m=1$. Пусть $F(x) \in C[a,b]$ является строго выпуклой вниз (рис.2)

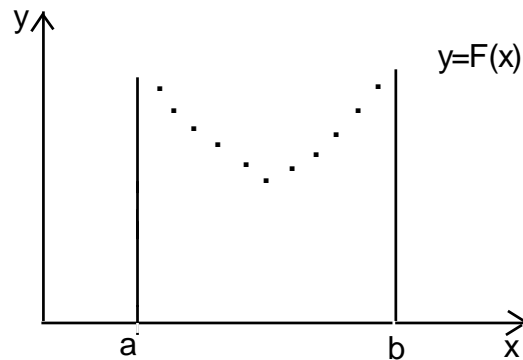


Рис.2

Тогда $F(x)$ имеет на $[a,b]$ точно один локальный минимум X^* . Построим последовательность сужающихся отрезков $[a_k, b_k]$, содержащих точку x^* , по следующему алгоритму.

Алгоритм G. Заданы отрезок $[a,b]$ и точность вычислений $\varepsilon > 0$.

1. $a_0 := a; b_0 := b; k := 0$. Вычисляем $F(a_0), F(b_0)$.

2. Если $b_k - a_k < \varepsilon$, то счет прекращается.

3. Выберем две точки $c_k, d_k \in [a_k, b_k]$ по формулам

$$c_k = a_k + \xi(b_k - a_k), d_k = b_k - \xi(b_k - a_k), \xi = 0.38 \quad (2.16)$$

и вычисляем $F(c_k), F(d_k)$.

4. Сравниваем $F(c_k) < F(d_k)$ (?). Если “да” (рис.3), то $a_{k+1} := a_k, b_{k+1} := d_k$.

Если “нет” (рис.4), то $a_{k+1} := c_k, b_{k+1} := b_k$.

5. $k := k+1$ и переходим на п.2.

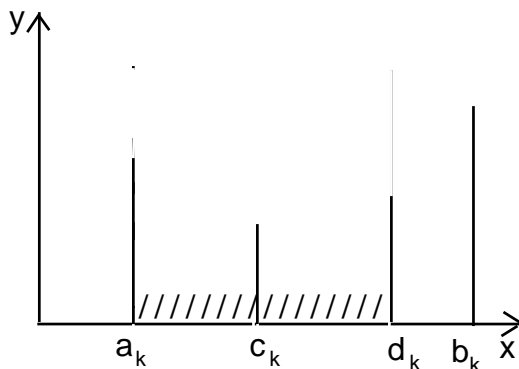


Рис.3

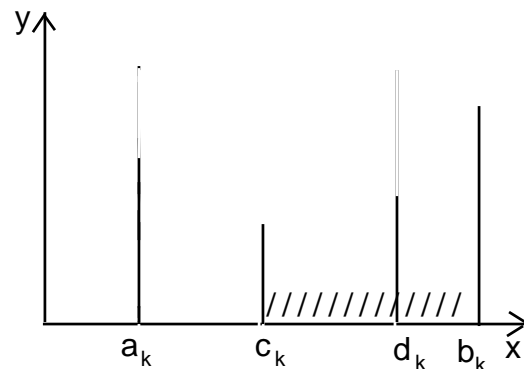


Рис.4

Замечание. Точки $c_k, d_k \in [a_k, b_k]$ можно брать любыми. Однако наиболее выгодно брать их так, чтобы следующий отрезок был поделен подобно предыдущему

$$\frac{c_k - a_k}{b_k - a_k} = \frac{d_k - c_k}{d_k - a_k}.$$

Отсюда и из (2.16) получаем уравнение $\xi^2 - 3\xi + 1 = 0$ и, следовательно, $\xi = (3 - \sqrt{5}) / 2 \approx 0.38$.

3. Вычисление собственных значений и собственных векторов матрицы

Рассмотрим уравнение

$$Ax = \lambda x, \quad (3.1)$$

где $A = [a_{ij}]_{i,j=1,n}$ - квадратная матрица с вещественными элементами,

$x = [x_i]_{i=1,n} \in \mathbb{R}^n$ - вектор-столбец, λ - параметр.

Те значения λ , при которых уравнение (3.1) имеет ненулевые решения, называются собственными значениями (СЗ). Ненулевые решения уравнения (3.1) называются собственными векторами (СВ). СЗ являются корнями характеристического многочлена (ХМ)

$$\Delta(\lambda) = \det[\lambda E - A] = \begin{vmatrix} \lambda - a_{11} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & \lambda - a_{22} & \dots & -a_{2n} \\ \dots & \dots & \dots & \dots \\ -a_{n1} & -a_{n2} & \dots & \lambda - a_{nn} \end{vmatrix} = \lambda^n - p_1 \lambda^{n-1} - \dots - p_n. \quad (3.2)$$

СЗ вещественной матрицы A могут быть как вещественными, так и комплексно сопряженными. Если же A симметрична ($A=A^T$), то её СЗ вещественны, а соответствующие СВ ортогональны. Методы вычисления СЗ и СВ можно разбить на две группы. В первой группе методов сначала строится ХМ $\Delta(\lambda)$, т.е. вычисляются коэффициенты p_1, \dots, p_n , а затем ищутся корни многочлена (см. п.2.4.) и вычисляются СВ. Во второй группе методов СЗ и СВ вычисляются непосредственно без построения ХМ $\Delta(\lambda)$. В п.3.1 и 3.2 изложены методы построения ХМ, а в п.3.3 и 3.4 приведены методы, относящиеся ко второй группе.

3.1. Метод Данилевского

Матрицы A и B называются подобными, если существует неособая матрица S такая, что $B = S^{-1}AS$. Очевидно, что подобные матрицы имеют одинаковые ХМ. В методе Данилевского преобразование подобия S подбирается так, чтобы матрица B имела форму Фробениуса

$$B = \begin{bmatrix} p_1 & p_2 & \dots & p_{n-1} & p_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \quad (3.3)$$

При этом элементы первой строки матрицы B и являются коэффициентами ХМ $\Delta(\lambda)$. Переход от A к B осуществляется последовательным преобразованием строк матрицы A , начиная с последней, в соответствующие строки матрицы B .

Шаг 1. Пусть $a_{n,n-1} \neq 0$. Рассмотрим матрицу

$$S_{n-1} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{a_{n1}}{a_{n,n-1}} & \frac{a_{n2}}{a_{n,n-1}} & \dots & \frac{a_{n,n-2}}{a_{n,n-1}} & \frac{1}{a_{n,n-1}} & \frac{a_{nn}}{a_{n,n-1}} \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

Ясно, что

$$S_{n-1}^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 0 \\ a_{n1} & a_{n2} & \dots & a_{n,n-2} & a_{n,n-1} & a_{nn} \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

Тогда матрица $A^{(1)} = S_{n-1}^{-1} A S_{n-1}$ имеет вид

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1,n-2}^{(1)} & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n-1,1}^{(1)} & a_{n-1,2}^{(1)} & \dots & a_{n-1,n-2}^{(1)} & a_{n-1,n-1}^{(1)} & a_{n-1,n}^{(1)} \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}.$$

Шаг 2. Пусть $a_{n-1,n-2}^{(1)} \neq 0$. Рассмотрим матрицу

$$S_{n-2} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{a_{n-1,1}^{(1)}}{a_{n-1,n-2}^{(1)}} & \frac{a_{n-1,2}^{(1)}}{a_{n-1,n-2}^{(1)}} & \dots & \frac{1}{a_{n-1,n-2}^{(1)}} & \frac{a_{n-1,n-1}^{(1)}}{a_{n-1,n-2}^{(1)}} & \frac{a_{n-1,n}^{(1)}}{a_{n-1,n-2}^{(1)}} \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

Ясно, что

$$S_{n-2}^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n-1,1}^{(1)} & a_{n-1,2}^{(1)} & \dots & a_{n-1,n-1}^{(1)} & a_{n-1,n}^{(1)} \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

Тогда матрица $A^{(2)} = S_{n-2}^{-1} A^{(1)} S_{n-2}$ имеет вид

$$A^{(2)} = \begin{bmatrix} a_{11}^{(2)} & \dots & a_{1,n-2}^{(2)} & a_{1,n-1}^{(2)} & a_{1n}^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n-2,1}^{(2)} & \dots & a_{n-2,n-2}^{(2)} & a_{n-2,n-1}^{(2)} & a_{n-2,n}^{(2)} \\ 0 & \dots & 1 & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}.$$

Далее процесс продолжается и аналогично строятся матрицы $A^{(3)}, A^{(4)}, \dots, A^{(n-2)}, A^{(n-1)}=B$. Таким образом, если

$$a_{n,n-1} \neq 0, a_{n-1,n-2}^{(1)} \neq 0, \dots, a_{2,1}^{(n-2)} \neq 0, \quad (3.4)$$

то после $n-1$ шагов получим

$$B = S^{-1}AS,$$

где B имеет вид (3.3), а $S = S_{n-1}S_{n-2} \dots S_1$.

Вычисление СВ. Пусть λ_* - СЗ, т.е. корень $\Delta(\lambda)$. Для вычисления соответствующего СВ γ матрицы A , вообще говоря, нужно решить уравнение $A\gamma = \lambda_*\gamma$. Но в данном случае проще воспользоваться приведенными выше вычислениями и следующими очевидными фактами:

1. Вектор $\xi = [\lambda_*^{n-1}, \lambda_*^{n-2}, \dots, \lambda_*, 1]^T$ является СВ матрицы B при $\lambda = \lambda_*$.
2. Вектор $\gamma = S\xi$ является СВ матрицы A при $\lambda = \lambda_*$.

Исключительные случаи в методе Данилевского. Метод работает при выполнении условия (3.4). Предположим теперь, что при некотором k в матрице

$$A^{(n-k)} = \begin{bmatrix} a_{11}^{(n-k)} & \dots & a_{1,k-1}^{(n-k)} & a_{1k}^{(n-k)} & \dots & a_{1,n-1}^{(n-k)} & a_{1n}^{(n-k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{k-1,1}^{(n-k)} & \dots & a_{k-1,k-1}^{(n-k)} & a_{k-1,k}^{(n-k)} & \dots & a_{k-1,n-1}^{(n-k)} & a_{k-1,n}^{(n-k)} \\ a_{k1}^{(n-k)} & \dots & a_{k,k-1}^{(n-k)} & a_{kk}^{(n-k)} & \dots & a_{k,n-1}^{(n-k)} & a_{kn}^{(n-k)} \\ 0 & \dots & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 \end{bmatrix}$$

имеем $a_{k,k-1}^{(n-k)} = 0$. Здесь нужно различать два случая.

1. Предположим, что слева от элемента $a_{k,k-1}^{(n-k)}$ существует ненулевой элемент $a_{k1}^{(n-k)} \neq 0, 1 < k-1$. Тогда следует поменять местами столбцы и строки с номерами 1 и $k-1$. После этого счет можно продолжить.

2. Предположим, что $a_{k,k-1}^{(n-k)} = \dots = a_{k1}^{(n-k)} = 0$. Тогда матрица $A^{(n-k)}$ распадается на блоки

$$A^{(n-k)} = \begin{bmatrix} D_1 & M & D_3 \\ L & L & L \\ 0 & M & D_2 \end{bmatrix},$$

причем $\det[\lambda E - A^{(n-k)}] = \det[\lambda E - D_1] \det[\lambda E - D_2]$. Матрица D_2 уже имеет форму Фробениуса, а для вычисления ХМ матрицы D_1 можно снова применить метод Данилевского.

3.2. Другие методы вычисления характеристического многочлена

Развертывание определителя. Раскрывая определитель (3.2) и приравнявая коэффициенты при степенях λ^k , получим:

$$p_1 = \sum_{i=1}^n a_{ii}, \quad -p_2 = \sum_{i < j} \begin{vmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{vmatrix}, \dots, \quad (-1)^{k-1} p_k = \sum_{i_1 < \dots < i_k} \begin{vmatrix} a_{i_1 i_1} & \dots & a_{i_1 i_k} \\ \dots & \dots & \dots \\ a_{i_k i_1} & \dots & a_{i_k i_k} \end{vmatrix}, \dots, \quad (-1)^{n-1} p_n = \det A.$$

Метод прост, но применяется лишь при небольших n , так как с ростом n объем вычислений катастрофически нарастает.

Метод Левеерье. Обозначим $\text{sp}(A) = a_{11} + a_{22} + \dots + a_{nn}$ - след матрицы A . Коэффициенты p_1, \dots, p_n ХМ $\Delta(\lambda)$ вычисляются по следующим формулам:

$$s_k = \text{sp}(A^k), \quad k = \overline{1, n}, \quad (3.5)$$

$$p_k = \frac{1}{k} (s_k + p_1 s_{k-1} + \dots + p_{k-1} s_1), \quad k = \overline{1, n}. \quad (3.6)$$

Отметим, что формулы (3.5), (3.6) являются следствием теоремы Виета и формулы $\text{sp}(A^k) = \lambda_1^k + \dots + \lambda_n^k$, где $\lambda_1, \dots, \lambda_n$ - корни ХМ $\Delta(\lambda)$.

Интерполяционный метод. Вычислим $f_i = \det[\mu_i E - A]$, $i = \overline{1, n}$ при некоторых фиксированных значениях μ_1, \dots, μ_n . Тогда коэффициенты ХМ p_1, \dots, p_n будут решениями СЛАУ

$$\mu_i^n - p_1 \mu_i^{n-1} - \dots - p_n = f_i, \quad i = \overline{1, n} \quad (3.7)$$

с отличным от нуля определителем, который является определителем Вандермонда.

Метод Крылова. По теореме Гамильтона-Кэли матрица A является корнем своего ХМ, т.е.

$$A^n - p_1 A^{n-1} - \dots - p_n A - p_n E = 0. \quad (3.8)$$

Отправляясь от некоторого начального вектора $x^{(0)} = [x_i^{(0)}]_{i=\overline{1, n}}$, построим последовательность векторов $x^{(k)} = [x_i^{(k)}]_{i=\overline{1, n}}$ по формуле $x^{(k)} = A^k x^{(0)}$, $k = \overline{0, n}$. Умножая равенство (3.8) на $x^{(0)}$, получим:

$$x^{(n)} - p_1 x^{(n-1)} - \dots - p_n x^{(0)} = 0$$

или в координатах

$$p_1 x_i^{(n-1)} + \dots + p_n x_i^{(0)} = x_i^{(n)}, \quad i = \overline{1, n}. \quad (3.9)$$

Если определитель в системе (3.9) отличен от нуля, т.е. если векторы $x^{(0)}, \dots, x^{(n-1)}$ линейно независимы, то решая (3.9), находим коэффициенты p_1, \dots, p_n ХМ $\Delta(\lambda)$. Если же при некотором m векторы $x^{(0)}, \dots, x^{(m-1)}$ линейно независимы, а $x^{(m)} = q_1 x^{(m-1)} + \dots + q_m x^{(0)}$, то многочлен $\varphi(\lambda) = \lambda^m - q_1 \lambda^{m-1} - \dots - q_m$ является делителем $\Delta(\lambda)$. Меняя, если необходимо, начальный вектор $x^{(0)}$, можно найти все делители $\Delta(\lambda)$, т.е. вычислить все корни ХМ.

3.3 Степенной метод

Степенной метод предназначен для вычисления наибольшего по модулю СЗ и соответствующего СВ. Пусть λ_i , $i = \overline{1, n}$ - СЗ матрицы A , а l_j - соответствующие линейно независимые СВ (мы предполагаем для

простоты изложения, что матрица A имеет n линейно независимых СВ). Пусть среди СВ есть наибольший по модулю

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

Возьмем фиксированный вектор x^0 . Разложим его по базису

$$x^0 = \sum_{i=1}^m c_i l_i$$

и предположим, что $c_1 \neq 0$. Тогда

$$A^k x^0 = \sum_{i=1}^m c_i \lambda_i^k l_i = c_1 \lambda_1^k l_1 + o(|\lambda_2|^k), \quad k = 0, 1, 2, \dots$$

Отсюда вычисляем

$$\lambda_1 = \lambda_1^{(k)} + o\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \quad (3.10)$$

где

$$\lambda_1^{(k)} = \frac{\langle A^{k+1} x^0, A^k x^0 \rangle}{\langle A^k x^0, A^k x^0 \rangle}.$$

Так как $|\lambda_2|/|\lambda_1| < 1$, то $\lim_{k \rightarrow \infty} \lambda_1^{(k)} = \lambda_1$ и при достаточно больших k $\lambda_1 \approx \lambda_1^{(k)}$ в пределах заданной точности.

Замечания. 1. При $|\lambda_1| > 1$ $\lim_{k \rightarrow \infty} \|Ax^k\| = \infty$, и возникает опасность переполнения. При $|\lambda_1| < 1$ $\lim_{k \rightarrow \infty} \|Ax^k\| = 0$, и возможна потеря значимости.

Поэтому вычисления удобно вести по формулам

$$l_1^{(k-1)} = \frac{x^{k-1}}{\|x^{k-1}\|}, \quad x^k = A l_1^{(k-1)}, \quad \lambda_1^{(k)} = \langle x^k, l_1^{(k-1)} \rangle, \quad k=1, 2, \dots \quad (3.11)$$

2. Если вектор x^0 выбран так, что $c_1 = 0$, то в процессе вычисления из-за ошибки округления всё равно появится составляющая с l_1 . Впрочем, чтобы избежать этого случая, полезно менять начальный вектор x^0 .

3. Для СВ l_1 аналогично получаем

$$l_1 = \lim_{k \rightarrow \infty} l_1^{(k)} \frac{|c_1 \lambda_1^k|}{c_1 \lambda_1^k}.$$

Так как СВ определяется с точностью до постоянного множителя, то в качестве СВ можно брать вектор

$$\gamma_1 = \lim_{k \rightarrow \infty} l_1^{(k)} \frac{|\lambda_1^k|}{\lambda_1^k}.$$

3.4. Метод вращений

Пусть $A = A^T$, т.е. матрица A является вещественной симметричной, тогда её СВ $\lambda_1, \dots, \lambda_n$ вещественны, а соответствующие СВ ортогональны.

2. Находим наибольший по модулю элемент $a_{ij}^{(k)}$ ($i < j$) матрицы A_k и номера i, j . (Если таких элементов несколько, то берем любой из них).

3. Вычисляем φ_k по формуле (3.13).

4. Строим матрицу $B_k = [b_{vs}^{(k)}]_{v,s=1,\dots,n}$ по формуле $B_k = A_k V_k$ или в координатах

$$\begin{aligned} b_{vi}^{(k)} &= a_{vi}^{(k)} \cos \varphi_k + a_{vj}^{(k)} \sin \varphi_k, & b_{vs}^{(k)} &= a_{vs}^{(k)}, \quad s \neq i, j. \\ b_{vj}^{(k)} &= -a_{vi}^{(k)} \sin \varphi_k + a_{vj}^{(k)} \cos \varphi_k, \end{aligned}$$

5. Строим матрицу A_{k+1} по формуле $A_{k+1} = V_k^T B_k$ или в координатах

$$\begin{aligned} a_{is}^{(k+1)} &= b_{is}^{(k)} \cos \varphi_k + b_{js}^{(k)} \sin \varphi_k, & a_{vs}^{(k+1)} &= b_{vs}^{(k)}, \quad v \neq i, j. \\ a_{js}^{(k+1)} &= -b_{is}^{(k)} \sin \varphi_k + b_{js}^{(k)} \cos \varphi_k, \end{aligned}$$

6. Вычисляем $t(A_{k+1})$.

7. Сравниваем $t(A_{k+1}) < \varepsilon$ (?). Если “нет”, то $k := k+1$ и переходим на п.2. Если “да”, то переходим на п.8.

8. $\lambda_i := a_{ii}^{(k+1)}$, $V := V_0 \dots V_k$ - СЗ и СВ матрицы A .

Библиографический список

1. Березин И.С., Жидков Н.П. Методы вычислений. М., 1962.
2. Демидович Б.П., Марон И.А. Основы вычислительной математики. М., 1966.
3. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. М., 1987.
4. Воеводин В.В. Вычислительные основы линейной алгебры. М., 1977.
5. Бобков В.В., Крылов В.И., Монастырский П.И. Начала теории вычислительных методов. Линейная алгебра и нелинейные уравнения. Минск, 1983.
6. Васильев Ф.П. Численные методы решения экстремальных задач. М., 1980.

Оглавление

1. Системы линейных алгебраических уравнений	3
1.1. Метод Гаусса	3
1.2. Метод квадратного корня	5
1.3. Метод прогонки	6
1.4. Метод простой итерации	7
1.5. Метод наискорейшего спуска	8
1.6. Метод сопряженных градиентов	9
2. Нелинейные уравнения и поиск экстремума	11
2.1. Метод простой итерации	11
2.2. Метод Ньютона	11
2.3. Метод дихотомии	12
2.4. Решение алгебраических уравнений	13
2.5. Методы минимизации	14
3. Вычисление собственных значений и собственных векторов матрицы	19
3.1. Метод Данилевского	19
3.2. Другие методы вычисления характеристического многочлена	22
3.3. Степенной метод	23
3.4. Метод вращений	24
Библиографический список	26